



# Proceedings of the 15th Conference of the International Society for Music Information Retrieval (ISMIR 2014)

October 27 - 31, 2014

Taipei, Taiwan

EDITED BY

Hsin-Min Wang  
Yi-Hsuan Yang  
Jin Ha Lee





# **Proceedings of the 15th Conference of the International Society for Music Information Retrieval (ISMIR)**

EDITED BY

Hsin-Min Wang

Yi-Hsuan Yang

Jin Ha Lee

October 27 - 31, 2014

Taipei, Taiwan

<http://www.terasoft.com.tw/conf/ismir2014/>

ISMIR2014 is organized by the International Society for Music Information Retrieval.

Website: <http://www.terasoft.com.tw/conf/ismir2014/>

EDITED BY

Hsin-Min Wang (Academia Sinica, Taipei, Taiwan)

Yi-Hsuan Yang (Academia Sinica, Taipei, Taiwan)

Jin Ha Lee (University of Washington, Seattle, WA, USA)

## Sponsors

### Gold Sponsors



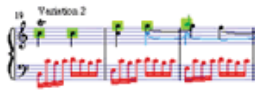
### Silver Sponsors



### Bronze Sponsors



## Co-Hosts



中華民國電腦音樂學會  
*Taiwan Computer Music Association*



經濟部  
國際貿易局 經貿資訊網  
Bureau of Foreign Trade



觀光傳播局  
*Department of Information and Tourism*

## Corporate Partner



## **Organizing Committee**

### **General Chairs**

Jyh-Shing Roger Jang (National Taiwan University)

Masataka Goto (National Institute of Advanced Industrial Science and Technology (AIST))

Xiao Hu (University of Hong Kong)

### **Program Chairs**

Hsin-Min Wang (Academia Sinica)

Yi-Hsuan Yang (Academia Sinica)

Jin Ha Lee (University of Washington)

### **Steering Committee**

Shigeki Sagayama (University of Tokyo)

Malcolm Slaney (Microsoft)

Frank Soong (Microsoft Research Asia)

Shyh-Kang Jeng (National Taiwan University)

Arbee Chen (National Chengchi University)

Homer Chen (National Taiwan University)

Alvin Su (National Cheng Kung University)

### **Tutorial Chair**

Li Su (Academia Sinica)

### **Music Chairs**

Jeff (Chih-Fang) Huang (Kainan University, Chair of Taiwan Computer Music Association)

Yi-Wen Liu (National Tsing Hua University)

Yu-Chung Tseng (National Chiao Tung University)

### **Late-breaking and Demo Chair**

Ju-Chiang Wang (University of California, San Diego)

### **Unconference Chair**

Eric Humphrey (New York University)

### **Finance Chair**

Ming-Feng Tsai (National Chengchi University)

### **Local Arrangement Chairs**

Jia-Lien Hsu (Fu Jen Catholic University)

Wei-Ho Tsai (National Taipei University of Technology)

**Web/Publication Chair**

Ching-Hua Chuan (University of North Florida)

Gang Ren (University of Rochester)

**Publicity Chair**

Ye Wang (National University of Singapore)

**Registration Chair**

Jia-Ching Wang (National Central University)

## Technical Program Committee

Jean-Julien Aucouturier (Institut de Recherche et Coordination Acoustique/Musique, France)  
Juan Pablo Bello (New York University, USA)  
Elaine Chew (Queen Mary University of London, UK)  
Darrell Conklin (Universidad del Pais Vasco, Spain)  
Sally Jo Cunningham (University of Waikato, New Zealand)  
Matthew Davies (Institute for Systems and Computer Engineering of Porto, Portugal)  
J. Stephen Downie (University of Illinois at Urbana-Champaign, USA)  
Zhiyao Duan (Rochester University, USA)  
Andreas Ehmann (Pandora, USA)  
Dan Ellis (Columbia University, USA)  
Slim Essid (ParisTech, France)  
Arthur Flexer (Austrian Research Institute for Artificial Intelligence, Austria)  
Rebecca Fiebrink (Princeton University, USA)  
Ichiro Fujinaga (McGill University, Canada)  
Emilia Gomez (Universitat PompeuFabra, Spain)  
Perfecto Herrera (Universitat PompeuFabra, Spain)  
Andre Holzapfel (University of Crete, Greece)  
Hirokazu Kameoka (Tokyo University, Japan)  
Peter Knees (Johannes Kepler University, Austria)  
Audrey Laplante (University of Montreal, Canada)  
Luis Gustavo Martins (Catholic University of Porto, Portugal)  
Brian Mcfee (Columbia University, USA)  
Meinard Mueller (University Erlangen-Nuremberg, Germany)  
Geoffroy Peeters (Institut de Recherche et Coordination Acoustique/Musique, France)  
Markus Schedl (Johannes Kepler University, Austria)  
Erik Schmidt (Pandora Internet Radio, USA)  
Joan Serra (Artificial Intelligence Research Institute, Spain)  
Mohamed Sordo (Universitat PompeuFabra, Spain)  
Bob Sturm (Aalborg University Copenhagen, Denmark)  
Li Su (Academia Sinica, Taiwan)  
Douglas Turnbull (Ithaca College, USA)  
George Tzanetakis (University of Victoria, Canada)  
Emmanuel Vincent (Institut National de Recherche en Informatique et en Automatique, France)  
Anja Volk (Utrecht University, The Netherlands)  
Ju-Chiang Wang (University of California, San Diego, USA)  
Frans Wiering (Utrecht University, The Netherlands)  
Kazuyoshi Yoshii (Kyoto University, Japan)



## Reviewers

Samer Abdallah	Matthew Davies	Enric Guaus
Jakob Abesser	Bas de Haas	Sankalp Gulati
Teppo Ahonen	Alceu de Souza Britto Jr.	Michael Gurevich
Joshua Albrecht	Norberto Degara	Emilia Gomez
Anna Aljanaki	Francois Deliege	Philippe Hamel
Vinoo Alluri	Arnaud Desein	Jinyu Han
Joakim Anden	Johanna Devaney	Andrew Hankinson
Tom Arjannikov	Sander Dieleman	Pierre Hanna
Andreas Arzt	Christian Dittmar	Mikael Henaff
Jean-Julien Aucouturier	J. Stephen Downie	Martin Hermant
Roland Badeau	Jonathan Driedger	Perfecto Herrera
Isabel Barbancho	Zhiyao Duan	Jason Hockman
Ana Maria Barbancho	Georgi Dzhabazov	Matt Hoffman
Mathieu Barthet	Tuomas Eerola	Andre Holzapfel
Eric Battenberg	Andreas Ehmann	Yajie Hu
Dogac Basaran	Katherine Ellis	Po-Sen Huang
Juan Pablo Bello	Dan Ellis	Anna Huang
Emmanouil Benetos	Valentin Emiya	Eric Humphrey
Ciril Bohak	Paulo Esquef	Fernando Iazzetta
Antonio Bonafonte	Slim Essid	Vaiva Imbrasaite
Juanjo Bosch	Sebastian Ewert	Vignesh Ishwar
Baris Bozkurt	Angel Faraldo	Akinori Ito
Ashley Burgoyne	George Fazekas	Katsutoshi Itoyama
Sebastian Bock	Rebecca Fiebrink	Ozgur Izmirli
Marcelo Caetano	Thomas Fillon	Jordi Janer
Emilios Cambouropoulos	Derry Fitzgerald	Tristan Jehan
Estefania Cano	Nicole Flaig	Kristoffer Jensen
Mark Cartwright	Arthur Flexer	Cyril Joder
Tak-Shing Chan	Nuno Fonseca	Sergi Jorda
Chih-Ming Chen	Frederic Font	Hirokazu Kameoka
Alex Chen	Jose Fornari	Kunio Kashino
Elaine Chew	Ichiro Fujinaga	Haruhiro Katayose
Ching-Hua Chuan	Satoru Fukayama	Damian Keller
Andrea Cogliati	Daniel Gaertner	Johannes Kepper
Gina Collecchia	Martin Gasser	Corey Kereliuk
Tom Collins	Ali Cenk Gedik	Tetsuro Kitahara
Darrell Conklin	Mathieu Giraud	Anssi Klapuri
Arshia Cont	Aggelos Gkiokas	Peter Knees
Courtenay Cotton	Fabien Gouyon	Ian Knopke
Emanuele Coviello	Maarten Grachten	Gopala Krishna Koduri
Sally Jo Cunningham	Garth Griffin	Noam Koenigstein
Michael Scott Cuthbert	Thomas Grill	Alessandro Koerich
Roger Dannenberg	David Grunberg	Filip Korzeniowski

Florian Krebs	Yasunori Ohishi	Ian Simon
Nadine Kroher	Nobutaka Ono	George Sioros
Lun-Wei Ku	Carthach ONuanain	Paris Smaragdis
Frank Kurth	Nicola Orio	Jordan Smith
Mathieu Lagrange	Alexei Ozerov	Lloyd Smith
Paul Lamere	Francois Pachet	Yading Song
Thibault Langlois	Rui Pedro Paiva	Reinhard Sonnleitner
Audrey Laplante	Helene Papadopoulos	Mohamed Sordo
Olivier Lartillot	Tae Hong Park	Ajay Srinivasamurthy
Kyogu Lee	Jouni Paulus	Adam Stark
Andreas Lehmann	Johan Pauwels	Sebastian Stober
Bernhard Lehner	Steffen Pauws	Dan Stowell
Kjell Lemstrom	Geoffroy Peeters	Bob Sturm
David Lewis	Graham Percival	Bob Sturm
Dawen Liang	Antonio Pertusa	Li Su
Cynthia Liem	Pedro Pestana	Alvin Wen Yu Su
Daryl Lim	Aggelos Pikrakis	Atau Tanaka
Jen-Yu Liu	Thomas Praetzelich	damien tardieu
Yi-Wen Liu	Matthew Prockup	David Temperley
Antoine Liutkus	Laurent Pugin	Steve Tjoa
Jorn Loviscach	Marcelo Queiroz	Marko Tkalcic
Hanna Lukashevich	Stanislaw Raczynski	Petri Toiviainen
Esteban Maestre	Colin Raffel	Godfried Toussaint
Adolfo Maia Jr.	Zafar Rafii	Wei-Ho Tsai
Michael Mandel	Mathieu Ramona	Douglas Turnbull
Matija Marolt	Andreas Rauber	George Tzanetakis
Luis Gustavo Martins	Ana Rebelo	Julian Urbano
Agustin Martorell	Gang Ren	Yonatan Vaizman
Matthias Mauch	Christophe Rhodes	Jan Van Balen
Rudolf Mayer	Gael Richard	Remco Veltkamp
Brian McFee	David Rizo	Emmanuel Vincent
Cory McKay	Matthias Robine	Richard Vogl
Matt McVicar	Martin Rocamora	Anja Volk
Nicola Montecchio	Marcelo Rodriguez Lopez	Ge Wang
Josh Moore	Perry Roland	Xing Wang
Marcela Morvidone	Gerard Roma	Ju-Chiang Wang
Manuel Moussallam	Justin Salamon	Ron Weiss
Meinard Mueller	Andy Sarroff	Felix Weninger
Daniel Mullensiefen	Markus Schedl	Tillman Weyde
Hidehisa Nagano	Jan Schlueter	Ian Whalley
Masahiro Nakano	Erik Schmidt	Frans Wiering
Tomoyasu Nakano	Bjoern Schuller	Geraint Wiggins
Juhan Nam	Jeff Scott	Ben Wu
Eric Nichols	Sertan Senturk	Fu-Hai Frank Wu
Oriol Nieto	Xavier Serra	Guangyu Xia
Mitsunori Ogihara	Joan Serra	Kazuyoshi Yoshii

# Contents

<b>Preface .....</b>	<b>xvii</b>
<b>Keynotes .....</b>	<b>xxi</b>
Keynote 1: Automatic Music Transcription: From Music Signals to Music Scores .....	xxii
<i>Axel Roebel</i>	
Keynote 2: Sound and Music Computing for Exercise and (Re-)Habilitation .....	xxiv
<i>Ye Wang</i>	
<b>Tutorials .....</b>	<b>xxv</b>
Tutorial 1: Why is Greek Music Interesting? Towards an Ethics of MIR .....	xxvi
<i>Andre Holzapfel and George Tzanetakis</i>	
Tutorial 2: Musical Structure Analysis .....	xxvii
<i>Meinard Mueller and Jordan Smith</i>	
Tutorial 3: Jingju Music: Concepts and Computational Tools for Analysis .....	xxviii
<i>Rafael Caro Repetto, Ajay Srinivasamurthy, Sankalp Gulati, and Xavier Serra</i>	
Tutorial 4: MiningSuite, a Comprehensive Framework for Music Analysis.....	xxix
Articulating Audio (MIRtoolbox 2.0) and Symbolic Approaches	
<i>Olivier Lartillot</i>	
<b>Oral Session 1: Classification .....</b>	<b>1</b>
On Cultural, Textual and Experiential Aspects of Music Mood .....	3
<i>Abhishek Singhi and Daniel Brown</i>	
Sparse Cepstral and Phase Codes for Guitar Playing Technique Classification .....	9
<i>Li Su, Li Fan Yu, Yi-Hsuan Yang</i>	
Automated Detection of Single- and Multi-Note Ornaments in Irish Traditional Flute Playing .....	15
<i>Münevver Köküer, Peter Jančovič, Islah Ali-MacLachlan, Cham Athwal</i>	
The Kiki-Bouba Challenge: Algorithmic Composition for Content-Based MIR .....	21
Research & Development	
<i>Bob Sturm and Nick Collins</i>	
<b>Poster Session 1 .....</b>	<b>27</b>
Transfer Learning by Supervised Pre-Training for Audio-Based Music Classification .....	29
<i>Aäron van den Oord, Sander Dieleman, Benjamin Schrauwen</i>	

Estimating Musical Time Information from Performed MIDI Files .....	35
<i>Harald Grohganz, Michael Clausen, Meinard Müller</i>	
Estimation of the Direction of Strokes and Arpeggios .....	41
<i>Isabel Barbancho, George Tzanetakis, Lorenzo J. Tardón, Peter F. Driessen, Ana M. Barbancho</i>	
Predicting Expressive Dynamics in Piano Performances Using Neural Networks .....	47
<i>Sam van Herwaarden, Maarten Grachten, W. Bas de Haas</i>	
An RNN-based Music Language Model for Improving Automatic Music Transcription .....	53
<i>Siddharth Sigtia, Emmanouil Benetos, Srikanth Cherla, Tillman Weyde, Artur S. d'Avila Garcez, Simon Dixon</i>	
Towards Modeling Texture in Symbolic Data .....	59
<i>Mathieu Giraud, Florence Levé, Florent Mercier, Marc Rigaudière, Donatien Thorez</i>	
Computational Models for Perceived Melodic Similarity in A Cappella Flamenc .....	65
<i>Nadine Kroher, Emilia Gómez, Catherine Guastavino, Francisco Gómez-Martín, Jordi Bonada</i>	
The VIS Framework: Analyzing Counterpoint in Large Datasets .....	71
<i>Christopher Antila and Julie Cumming</i>	
Hierarchical Approach to Detect Common Mistakes of Beginner Flute Players .....	77
<i>Yoonchang Han and Kyogu Lee</i>	
Robust Joint Alignment of Multiple Versions of a Piece of Music .....	83
<i>Siyang Wang, Sebastian Ewert, Simon Dixon</i>	
Formalizing the Problem of Music Description .....	89
<i>Bob Sturm, Rolf Bardeli, Thibault Langlois, Valentin Emiya</i>	
An Association-based Approach to Genre Classification in Music .....	95
<i>Tom Arjannikov and John Z. Zhang</i>	
Multiple Viewpoint Melodic Prediction with Fixed-Context Neural Networks .....	101
<i>Srikanth Cherla, Tillman Weyde, Artur d'Avila Garcez</i>	
Verovio: A library for Engraving MEI Music Notation into SVG .....	107
<i>Laurent Pugin, Rodolfo Zitellini, Perry Roland</i>	
Music Classification by Transductive Learning Using Bipartite Heterogeneous Networks .....	113
<i>Diego Furtado Silva, Rafael Geraldeli Rossi, Solange Oliveira Rezende, Gustavo Enrique de Almeida Prado Alves Batista</i>	
Automatic Melody Transcription based on Chord Transcription .....	119
<i>Antti Laaksonen</i>	

Audio-to-Score Alignment at the Note Level for Orchestral Recordings .....	125
<i>Marius Miron, Julio José Carabias-Orti, Jordi Janer</i>	
A Compositional Hierarchical Model for Music Information Retrieval .....	131
<i>Matevž Pesek, Aleš Leonardis, Matija Marolt</i>	
An Analysis and Evaluation of Audio Features for Multitrack Music Mixtures .....	137
<i>Brecht De Man, Brett Leonard, Richard King, Joshua D. Reiss</i>	
Detecting Drops in Electronic Dance Music: Content Based Approaches .....	143
to a Socially Significant Music Event	
<i>Karthik Yadati, Martha Larson, Cynthia C. S. Liem, Alan Hanjalic</i>	
Towards Automatic Content-Based Separation of DJ Mixes into Single Tracks .....	149
<i>Nikolay Glazyrin</i>	
MedleyDB: A Multitrack Dataset for Annotation-Intensive MIR Research .....	155
<i>Rachel M. Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, Juan P. Bello</i>	
Melody Extraction from Polyphonic Audio of Western Opera .....	161
A Method based on Detection of the Singer's Formant	
<i>Zheng Tang and Dawn A. A. Black</i>	
Codebook-Based Scalable Music Tagging with Poisson Matrix Factorization .....	167
<i>Dawen Liang, John Paisley, Daniel P. W. Ellis</i>	
<b>Oral Session 2: Transcription .....</b>	<b>173</b>
Template Adaptation for Improving Automatic Music Transcription .....	175
<i>Emmanouil Benetos, Roland Badeau, Tillman Weyde, Gaël Richard</i>	
Note-Level Music Transcription by Maximum Likelihood Sampling .....	181
<i>Zhiyao Duan and David Temperley</i>	
Drum Transcription via Classification of Bar-Level Rhythmic Patterns .....	187
<i>Lucas Thompson, Simon Dixon, Matthias Mauch</i>	
<b>Oral Session 3: Symbolic .....</b>	<b>193</b>
Developing Tonal Perception Through Unsupervised Learning .....	195
<i>Carlos Eduardo Cancino Chacón, Stefan Lattner, Maarten Grachten</i>	
Exploiting Instrument-Wise Playing/Non-Playing Labels .....	201
for Score Synchronization of Symphonic Music	
<i>Alessio Bazzica, Cynthia C. S. Liem, Alan Hanjalic</i>	

Multi-Strategy Segmentation of Melodies .....	207
<i>Marcelo Rodríguez-López, Anja Volk, Dimitrios Bountouridis</i>	
A Data Set for Computational Studies of Schenkerian Analysis .....	213
<i>Phillip Kirlin</i>	
Systematic Multi-Scale Set-Class Analysis .....	219
<i>Agustín Martorell and Emilia Gómez</i>	
<b>Oral Session 4: Retrieval .....</b>	<b>225</b>
Spotting a Query Phrase from Polyphonic Music Audio Signals .....	227
Based on Semi-Supervised Nonnegative Matrix Factorization	
<i>Taro Masuda, Kazuyoshi Yoshii, Masataka Goto, Shigeo Morishima</i>	
Bayesian Audio Alignment based on a Unified Model of Music Composition and Performance ....	233
<i>Akira Maezawa, Katsutoshi Itoyama, Kazuyoshi Yoshii, Hiroshi Okuno</i>	
Automatic Set List Identification and Song Segmentation for Full-Length Concert Videos .....	239
<i>Ju-Chiang Wang, Ming-Chi Yen, Yi-Hsuan Yang, Hsin-Min Wang</i>	
On Inter-Rater Agreement in Audio Music Similarity .....	245
<i>Arthur Flexer</i>	
<b>Poster Session 2 .....</b>	<b>251</b>
Emotional Predisposition of Musical Instrument Timbres with Static Spectra .....	253
<i>Bin Wu, Andrew Horner, Chung Lee</i>	
Panako - A Scalable Acoustic Fingerprinting System .....	259
Handling Time-Scale and Pitch Modification	
<i>Joren Six and Marc Leman</i>	
Perceptual Analysis of the F-Measure to Evaluate Section Boundaries in Music .....	265
<i>Oriol Nieto, Morwaread Farbood, Tristan Jehan, Juan Pablo Bello</i>	
Keyword Spotting in A-Capella Singing .....	271
<i>Anna Kruspe</i>	
The Importance of F0 Tracking in Query-by-Singing-Humming .....	277
<i>Emilio Molina, Lorenzo J. Tardón, Isabel Barbancho, Ana M. Barbancho</i>	
Vocal Separation using Singer-Vowel Priors Obtained from Polyphonic Audio .....	283
<i>Shrikant Venkataramani, Nagesh Nayak, Preeti Rao, Rajbabu Velmurugan</i>	
Improving Query by Tapping via Tempo Alignment .....	289
<i>Chun-Ta Chen, Jyh-Shing Roger Jang, Chun-Hung Lu</i>	

Automatic Instrument Classification of Ethnomusicological Audio Recordings .....	295
<i>Dominique Fourer, Jean-Luc Rouas, Pierre Hanna, Matthias Robine</i>	
Music Analysis as a Smallest Grammar Problem .....	301
<i>Kirill Sidorov, Andrew Jones, David Marshall</i>	
Frame-Level Audio Segmentation for Abridged Musical Works .....	307
<i>Thomas Prätzlich and Meinard Müller</i>	
Creating a Corpus of Jingju (Beijing Opera) Music and Possibilities for Melodic Analysis .....	313
<i>Rafael Caro Repetto and Xavier Serra</i>	
Modeling Temporal Structure in Music for Emotion Prediction using Pairwise Comparisons .....	319
<i>Jens Madsen, Bjørn Sand Jensen, Jan Larsen</i>	
Musical Structural Analysis Database Based on GTTM .....	325
<i>Masatoshi Hamanaka, Keiji Hirata, Satoshi Tojo</i>	
Theoretical Framework of A Computational Model of Auditory Memory .....	331
for Music Emotion Recognition	
<i>Marcelo Caetano and Frans Wiering</i>	
Improving Music Structure Segmentation using lag-priors .....	337
<i>Geoffroy Peeters and Victor Bisot</i>	
Study of the Similarity between Linguistic Tones and Melodic Pitch Contours .....	343
in Beijing Opera Singing	
<i>Shuo Zhang, Rafael Caro Repetto, Xavier Serra</i>	
A Proximity Grid Optimization Method to Improve Audio Search for Sound Design .....	349
<i>Christian Frisson, Stéphane Dupont, Willy Yvart, Nicolas Riche, Xavier Siebert, Thierry Dutoit</i>	
Introducing a Dataset of Emotional and Color Responses to Music .....	355
<i>Matevž Pesek, Primož Godec, Mojca Poredoš, Gregor Strle, Jože Guna, Emilija Stojmenova, Matevž Pogačnik, Matija Marolt</i>	
In-Depth Motivic Analysis based on Multiparametric Closed Pattern .....	361
and Cyclic Sequence Mining	
<i>Olivier Lartillot</i>	
MIR_EVAL: A Transparent Implementation of Common MIR Metrics .....	367
<i>Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel P. W. Ellis</i>	
Computational Modeling of Induced Emotion Using GEMS .....	373
<i>Anna Aljanaki, Frans Wiering, Remco C. Veltkamp</i>	

Cognition-Inspired Descriptors for Scalable Cover Song Retrieval .....	379
<i>Jan Van Balen, Dimitrios Bountouridis, Frans Wiering, Remco Veltkamp</i>	
A Cross-Cultural Study on the Mood of K-POP Songs .....	385
<i>Xiao Hu1, Jin Ha Lee, Kahyun Choi, J. Stephen Downie</i>	
Cadence Detection in Western Traditional Stanzaic Songs using Melodic and Textual Features ....	391
<i>Peter Van Kranenburg and Folgert Karsdorp</i>	
Discovering Typical Motifs of a Raga from One-Liners of Songs in Carnatic Music .....	397
<i>Shrey Dutta and Hema A Murthy</i>	
<b>Oral Session 5: Structure .....</b>	<b>403</b>
Analyzing Song Structure with Spectral Clustering .....	405
<i>Brian McFee and Daniel P.W. Ellis</i>	
Identifying Polyphonic Musical Patterns From Audio Recordings .....	411
Using Music Segmentation Techniques <i>Oriol Nieto and Morwaread Farbood</i>	
Boundary Detection in Music Structure Analysis using Convolutional Neural Networks .....	417
<i>Karen Ullrich, Jan Schlüter, Thomas Grill</i>	
<b>Oral Session 6: Cultures .....</b>	<b>423</b>
Tracking the "Odd": Meter Inference in a Culturally Diverse Music Corpus .....	425
<i>Andre Holzapfel, Florian Krebs, Ajay Srinivasamurthy</i>	
Transcription and Recognition of Syllable based Percussion Patterns .....	431
The Case of Beijing Opera <i>Ajay Srinivasamurthy, Rafael Caro Repetto, Harshavardhan Sundar, Xavier Serra</i>	
<b>Oral Session 7: Recommendation &amp; Listeners .....</b>	<b>437</b>
Taste Space Versus the World: an Embedding Analysis of Listening Habits and Geography .....	439
<i>Joshua Moore, Thorsten Joachims, Douglas Turnbull</i>	
Enhancing Collaborative Filtering Music Recommendation .....	445
by Balancing Exploration and Exploitation <i>Zhe Xing, Xinxi Wang, Ye Wang</i>	
Improving Music Recommender Systems: What Can We Learn from Research on Music Tastes? .	451
<i>Audrey Laplante</i>	



Social Music in Cars .....	457
<i>Sally Jo Cunningham, David M. Nichols, David Bainbridge, Hassan Ali</i>	
<b>Poster Session 3 .....</b>	<b>463</b>
A Combined Thematic and Acoustic Approach for a Music Recommendation Service .....	465
in TV Commercials	
<i>Mohamed Morchid, Richard Dufour, Georges Linarès</i>	
Are Poetry and Lyrics All That Different? .....	471
<i>Abhishek Singhi and Daniel G. Brown</i>	
Singing-Voice Separation from Monaural Recordings Using Deep Recurrent Neural Networks ....	477
<i>Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, Paris Smaragdis</i>	
Impact of Listening Behavior on Music Recommendation .....	483
<i>Katayoun Farrahi, Markus Schedl, Andreu Vall, David Hauger, Marko Tkalčič</i>	
Towards Seamless Network Music Performance: Predicting an Ensemble's .....	489
Expressive Decisions for Distributed Performance	
<i>Bogdan Vera and Elaine Chew</i>	
Detection of Motor Changes in Violin Playing by EMG Signals .....	495
<i>Ling-Chi Hsu, Yu-Lin Wang, Yi-Ju Lin, Cheryl D. Metcalf, Alvin W.Y. Su</i>	
Automatic Key Partition Based on Tonal Organization Information of Classical Music .....	501
<i>Wang Kong Lam and Tan Lee</i>	
Bayesian Singing-Voice Separation .....	507
<i>Po-Kai Yang, Chung-Chien Hsu, Jen-Tzung Chien</i>	
Probabilistic Extraction of Beat Positions from a Beat Activation Function .....	513
<i>Filip Korzeniowski, Sebastian Böck, Gerhard Widmer</i>	
Geographical Region Mapping Scheme Based on Musical Preferences .....	519
<i>Sanghoon Jun, Seungmin Rho, Eenjun Hwang</i>	
On Comparative Statistics for Labelling Tasks: What Can We Learn from MIREX ACE 2013? ....	525
<i>John Ashley Burgoyne, W. Bas de Haas, Johan Pauwels</i>	
Merged-Output HMM for Piano Fingering of Both Hands .....	531
<i>Eita Nakamura, Nobutaka Ono, Shigeki Sagayama</i>	
Modeling Rhythm Similarity for Electronic Dance Music .....	537
<i>Maria Panteli, Niels Bogaards, Aline Honingh</i>	
MuSe: A Music Recommendation Management System .....	543
<i>Martin Przyjaciel-Zablocki, Thomas Hornung, Alexander Schätzle, Sven Gauß, Io Taxidou, Georg Lausen</i>	

Tempo- and Transposition-Invariant Identification of Piece and Score Position .....	549
<i>Andreas Arzt, Gerhard Widmer, Reinhard Sonnleitner</i>	
Gender Identification and Age Estimation of Users Based on Music Metadata .....	555
<i>Ming-Ju Wu, Jyh-Shing Roger Jang, Chun-Hung Lu</i>	
Information-Theoretic Measures of Music Listening Behaviour .....	561
<i>Daniel Boland and Roderick Murray-Smith</i>	
Evaluation Framework for Automatic Singing Transcription .....	567
<i>Emilio Molina, Ana M. Barbancho, Lorenzo J. Tardón, Isabel Barbancho</i>	
What is the Effect of Audio Quality on the Robustness of MFCCs and Chroma Features? .....	573
<i>Julián Urbano, Dmitry Bogdanov, Perfecto Herrera, Emilia Gómez, Xavier Serra</i>	
Music Information Behaviors and System Preferences of University Students in Hong Kong .....	579
<i>Xiao Hu, Jin Ha Lee, Leanne Ka Yan Wong</i>	
LyricsRadar: A Lyrics Retrieval System Based on Latent Topics of Lyrics .....	585
<i>Shoto Sasaki, Kazuyoshi Yoshii, Tomoyasu Nakano, Masataka Goto, Shigeo Morisihima</i>	
JAMS: A JSON Annotated Music Specification for Reproducible MIR Research .....	591
<i>Eric J. Humphrey, Justin Salamon, Oriol Nieto, Jon Forsyth, Rachel M. Bittner, Juan P. Bello</i>	
On The Changing Regulations of Privacy and Personal Information in MIR .....	597
<i>Pierre Saurel, Francis Rousseaux, Marc Danger</i>	
A Multi-Model Approach to Beat Tracking Considering Heterogeneous Music Styles .....	603
<i>Sebastian Böck, Florian Krebs, Gerhard Widmer</i>	
<b>Oral Session 8: Source Separation .....</b>	<b>609</b>
Extending Harmonic-Percussive Separation of Audio Signals .....	611
<i>Jonathan Driedger, Meinard Müller, Sascha Disch</i>	
Singing Voice Separation Using Spectro-Temporal Modulation Features .....	617
<i>Frederick Yen, Yin-Jyun Luo, Tai-Shih Chi</i>	
Harmonic-Temporal Factor Decomposition Incorporating Music Prior Information .....	623
for Informed Monaural Source Separation	
<i>Tomohiko Nakamura, Kotaro Shikata, Norihiro Takamune, Hirokazu Kameoka</i>	
<b>Oral Session 9: Rhythm &amp; Beat .....</b>	<b>629</b>
Design And Evaluation of Onset Detectors using Different Fusion Policies .....	631
<i>Mi Tian, György Fazekas, Dawn A. A. Black, Mark Sandler</i>	

Evaluating the Evaluation Measures for Beat Tracking .....	637
<i>Matthew Davies and Sebastian Böck</i>	
Improving Rhythmic Transcriptions via Probability Models Applied Post-OMR .....	643
<i>Maura Church and Michael Scott Cuthbert</i>	
Classifying EEG Recordings of Rhythm Perception .....	649
<i>Sebastian Stober, Daniel J. Cameron, Jessica A. Grahn</i>	
<b>MIEX Oral Session .....</b>	<b>655</b>
Ten Years of MIREX (Music Information Retrieval Evaluation eXchange) .....	657
Reflections, Challenges and Opportunities	
<i>J. Stephen Downie, Xiao Hu, Jin Ha Lee, Kahyun Choi, Sally Jo Cunningham, Yun Hao</i>	
<b>Author Index .....</b>	<b>663</b>

## Preface

Welcome to the 15th ISMIR (International Society for Music Information Retrieval) Conference in Taipei, the capital city of Taiwan where you'll enjoy a mild sub-tropical climate, a multitude of exotic fruits, gourmet cuisine, easy and informal hospitality and a thriving and fascinating cultural scene.

The present volume contains all the peer-reviewed papers presented at ISMIR 2014.

- 252 papers were received, of which 222 were complete and well-formatted. These 222 papers were subjected to a double-blind review process in which both the authors and reviewers remained anonymous.
- 106 papers were accepted based on reviews and meta-reviews provided by 267 reviewers and 37 PC members. The overall acceptance rate is 47.7% (=106/222).
- 33 papers were selected for oral presentations based on both research quality and topic; the remainder 73 were selected for poster presentations.

The following table summarizes the publication statistics over the past ISMIRs:

Location	Oral	Poster	Total Papers	Total Pages	Total Authors	Unique Authors	Pages/ Paper	Authors/ Paper	U. Authors/ Paper
Plymouth	19	16	35	155	68	63	4.4	1.9	1.8
Indiana	25	16	41	222	100	86	5.4	2.4	2.1
Paris	35	22	57	300	129	117	5.3	2.3	2.1
Baltimore	26	24	50	209	132	111	4.2	2.6	2.2
Barcelona	61	44	105	582	252	214	5.5	2.4	2
London	57	57	114	697	316	233	6.1	2.8	2
Victoria	59	36	95	397	246	198	4.2	2.6	2.1
Vienna	62	65	127	486	361	267	3.8	2.8	2.1
Philadelphia	24	105	105	630	296	253	6	2.8	2.4
Kobe	38	85	123	729	375	292	5.9	3	2.4
Utrecht	24	86	110	656	314	263	6	2.	2.4
Miami	36	97	133	792	395	322	6	3	2.4
Porto	36	65	101	606	324	264	6	3.2	2.6
Curitiba	31	67	98	587	395	236	5.9	3	2.4
<b>Taipei</b>	<b>33</b>	<b>73</b>	<b>106</b>	<b>635</b>	<b>343</b>	<b>271</b>	<b>6</b>	<b>3.2</b>	<b>2.6</b>

As in past ISMIR conferences, the selected papers will be presented over a period of 3.5 days, preceded by a day of tutorials and followed by a half-day late-breaking/demo & unconference sessions. Moreover, we have two satellite events, including a music hack day of Hacking Audio and Music Research (HAMR) on Oct. 25 and 26, and Workshop on Computer Music and Audio Technology (WOCMAT) on Oct. 28 and 29. Highlights of the conference include:

## **Tutorials**

Four tutorials will take place on Monday, with two on ethnic music and two on music analysis, providing a good balance between culture and technology.

Morning sessions:

- “Why is Greek music interesting? Towards an ethics of MIR” by Andre Holzapfel and George Tzanetakis
- “Musical structure analysis” by Meinard Müller and Jordan Smith

Afternoon sessions:

- “Jingju music: Concepts and computational tools for its analysis” by Rafael Caro Repetto, Ajay Srinivasamurthy, Sankalp Gulati and Xavier Serra
- “MiningSuite, a comprehensive framework for music analysis, articulating audio (MIRtoolbox 2.0) and symbolic approaches” by Olivier Lartillot

## **Keynote Speakers**

We are honored to have two distinguished keynote speakers, Dr. Axel Roebel from IRCAM and Prof. Ye Wang from the National University of Singapore. Dr. Roebel will talk about audio music transcription, a challenging task and one of the ultimate goals of MIR, while Prof. Wang will describe innovative applications using music for exercise and rehabilitation.

- Axel Roebel: Automatic Music Transcription: From Music Signals to Music Scores
- Ye Wang: Sound and Music Computing for Exercise and (Re-)habilitation

## **MIREX**

Music Information Retrieval Evaluation eXchange (MIREX) is a collective effort to evaluate cutting-edge methods for various MIR tasks, which is an integral part of ISMIR. This year we

are celebrating the 10th anniversary of MIREX with the following events on Friday morning:

- Prof. J. Stephen Downie’s talk on “Ten Years of MIREX: Reflections, Challenges and Opportunities”
- A poster session on 20 MIREX tasks, including the Grand Challenge 2014 for user experience, and several other new tasks related to audio downbeat estimation, audio fingerprinting, and singing voice separation.

## **Late-breaking/Demo & Unconference**

Friday afternoon is dedicated to late-breaking papers and MIR system demonstrations. Abstracts for these presentations will be available online. Moreover, after the late-breaking/demo session, we have a special “unconference” session (following the late-breaking sessions in ISMIR 2012 and 2013) in which participants can break into groups to discuss MIR issues of particular interest. This will be an informal and informative opportunity to get to know your peers and colleagues from around the world.

## **Music Program**

On Wednesday night, a 2.5-hour concert will be held in the main conference hall. The goal of this year’s music program is not only to encourage the use of MIR techniques in creating new music, but also to promote the composition of music that reflects Asian philosophy. The concert will feature 10 pieces selected from participant submissions, and 6 pieces specially-commissioned for the conference.

## **Social Events**

As in past ISMIRs, a reception will be held on Monday night and a banquet on the following Thursday night. Moreover, we have “Women in MIR meeting” (for connecting female researchers in MIR) on Wednesday early morning, and “Mixer” (for people to get to know one another) on Wednesday late afternoon.

## **Get to know Taipei**

The Bureau of Foreign Trade has kindly provided all foreign participants vouchers for English-language half- and whole-day local tours, including an all-around tour of Taipei (including visits to National Palace Museum, Shilin Night Market, Taipei 101, etc), a sky lantern tour in the mountains outside Taipei, a tea culture tour (including Maokong Gondola), a luxurious foot massage (with dinner at world-famous Michelin-starred Din Tai Fung), and a spa in Taipei’s famous hot springs. More tour options can be found at ISMIR-2014 website. Be sure to take the chance to explore Taipei City and enjoy your stay at Taiwan!

The proceedings of ISMIR 2014 were made possible by the hard work of the organizing team, the PC members, the reviewers, and the authors, to whom we would like to express our deep gratitude. Special thanks also go to this year’s sponsors and supporters, including

MediaTek, KKBox, Pandora, Shazam, FormosaSoft, Merry, Realtek, Gracenote, iKala, Dolby, Samsung, Deansoft, Google, Doreso, Terasoft, III, ITRI, iNDIEVOX, and ACLCLP.

We hope you all have a wonderful and unforgettable stay at Taipei!

## **General Chairs**

Jyh-Shing Roger Jang  
*National Taiwan University, Taiwan*

Masataka Goto  
*National Institute of Advanced Industrial Science and Technology (AIST), Japan*

Xiao Hu  
*University of Hong Kong, Hong Kong S.A.R., China*

## **Program Chairs**

Hsin-Min Wang  
*Academic Sinica, Taiwan*

Yi-Hsuan Yang  
*Academic Sinica, Taiwan*

Jin Ha Lee  
*University of Washington, USA*



## Keynotes



## **Keynote 1: Automatic Music Transcription: From Music Signals to Music Scores**

**Axel Roebel**

*Analysis/Synthesis Team, IRCAM*

*Paris, France*

### **Abstract**

Deriving the symbolic annotation of a piece of music from the audio signal is one of the important long term objectives of research in music information retrieval. The related signal processing task is denoted in short as: Automatic Music Transcription. It consists of deriving a complete score including the timing and frequency information of the notes (instruments and drums) present, and the instruments that have produced each note. A solution of this task would have an important impact on the research on MIR because it would open the door to use a symbolic music representation for the analysis of arbitrary audio signals. On the other hand one may note that the solution of the AMT task may benefit from results of many individual MIR tasks: e.g. tonality, chords, tempo, structure (notably repetitions), instrumentation.

The present talk aims to situate today's research related to the AMT problem. It will start with an introduction into the problem and the main obstacles to be resolved. Then a brief summary of the history of research related to Automatic Music Transcription will be presented leading to a description of the state of the art. An overview of the algorithms that are currently employed will be given together with a few examples using existing software implementations. Finally, potential directions for improving the state of the art AMT algorithms will be discussed covering instrument models (ANR project SOR2), multi channel audio analysis (EU FP7 project 3DTVS), as well as music theoretic constraints.

### **Biography**

Axel Roebel is the head of the research team analysis/synthesis of sound at IRCAM. He received the Diploma in electrical engineering from Hannover University in 1990 and the Ph.D. degree (summa cum laude) in computer science from the Technical University of Berlin in 1993. In 1994 he joined the German National Research Centre for Information Technology (GMD-First) in Berlin where he continued his research on adaptive modeling of time series of nonlinear dynamical systems. In 1996 he became assistant professor for digital signal processing in the communication science department of the Technical University of Berlin. Since 2000 he is working at IRCAM doing research on spectral domain algorithms for sound analysis, synthesis and transformation. In summer 2006 he was Edgar-Varese guest professor for computer music at the Electronic studio of the Technical University of Berlin and in 2011 he became the head of the analysis/synthesis team.

His research centers around problems in audio signal analysis, synthesis and transformation covering music and speech. His recent research projects are related to spectral modeling of musical instruments (ANR project Sample Orchestrator II), audio to midi transcription (industrially funded project Audio2Note), detection and classification of sound events in multi channel audio (EU FP7 project 3DTVS), modeling and transformation of sound textures (ANR project PHYSIS), synthesis of singing voice (ANR project CHANTER). He is the main author of IRCAM's SuperVP software library for sound analysis and transformation.

## **Keynote 2: Sound and Music Computing for Exercise and (Re-)Habilitation**

**Ye Wang**

*Sound and Music Computing Lab, National University of Singapore  
Singapore*

### **Abstract**

The use of music as an aid in healing body and mind has received enormous attention over the last 20 years from a wide range of disciplines, including neuroscience, physical therapy, exercise science, and psychological medicine. We have attempted to transform insights gained from the scientific study of music and medicine into real-life applications that can be delivered widely, effectively, and accurately. We have been trying to use music in evidence-based and/or preventative medicine. In this talk, I will describe three clinically-focused tools to facilitate the delivery of established music-enhanced therapies, harnessing the synergy of sound and music computing (SMC), mobile computing, and cloud computing technologies to promote healthy lifestyles and to facilitate disease prevention, diagnosis, and treatment in both developed countries and resource-poor developing countries. I will present some of our past and ongoing research projects that combine wearable sensors, smartphone apps, and a cloud-based therapy delivery system to facilitate music-enhanced physical and speech therapy, as well as the joys and pains working in such a multidisciplinary environment.

### **Biography**

Ye Wang is an Associate Professor in the Computer Science Department at the National University of Singapore (NUS) and NUS Graduate School for Integrative Sciences and Engineering (NGS). He established and directed the sound and music computing (SMC) Lab ([www.smcnus.org](http://www.smcnus.org)). Before joining NUS he was a member of the technical staff at Nokia Research Center in Tampere, Finland for 9 years. His research interests include sound analysis and music information retrieval (MIR), mobile computing, and cloud computing, and their applications in music edutainment and e-Health, as well as determining their effectiveness via subjective and objective evaluations. His most recent projects involve the design and evaluation of systems to support 1) therapeutic gait training using Rhythmic Auditory Stimulation (RAS), and 2) Melodic Intonation Therapy (MIT). He is also affiliated with the School of Computer Science of Fudan University and Harvard Medical School.



## **Tutorials**

## **Tutorial 1: Why is Greek Music Interesting? Towards an Ethics of MIR**

**Andre Holzapfel**

*Department of Computer Science*

*University of Crete*

*Crete, Greece*

**George Tzanetakis**

*Department of Computer Science*

*University of Victoria*

*Victoria, BC, Canada*

The initial goal of this tutorial is to provide an overview of musical styles in Greek culture, and to indicate various features of these musics that make them challenging and interesting for research in Music Information Retrieval (MIR). This tutorial is addressed to everybody interested in extending the diversity of her/his evaluation data, this way targeting generality of MIR approaches. On the other hand, the tutorial is aimed to provide a lively overview over a range of styles, that we hope will be informative and inspiring for any music listener. The tutorial will initially provide an overview of various styles of rural and urban music styles in the various areas of Greece. Then, we will focus on some styles we are particularly familiar with, and point out a variety of research tasks that is apparently quite challenging for those musics, such as beat tracking, mood estimation, transcription and chord estimation. In conclusion, inspired by the diversity of Greek music and the problems such diversity poses for our research, we reflect on the possibility of universal approaches to music processing, and discuss ethical implications for our work on recommendation systems for the musics of the world.

## **Tutorial 2: Musical Structure Analysis**

**Meinard Mueller**

*International Audio Laboratories Erlangen  
Erlangen, Germany*

**Jordan Smith**

*Centre for Digital Music  
Queen Mary University of London  
London, UK*

One of the attributes distinguishing music from other sound sources is the hierarchical structure in which music is organized. On the lowest level, one may have sound events such as individual notes, which are characterized by the way they sound, their timbre, pitch and duration. Such sound events combine to form larger structures such as motives, phrases, and chords, and these elements again form larger constructs that determine the overall layout of the composition. This higher structural level is specified in terms of musical parts and their mutual relations. For example, in popular music such parts can be the intro, chorus, and verse sections of the song. Or, in classical music, it can be the exposition, development, and recapitulation of a sonata movement. The goal of music structure analysis is to divide a given music representation into temporal segments that correspond to musical parts and to group these segments into musically meaningful categories.

In this tutorial, we review the most important segmentation and structure analysis principles and then discuss state-of-the-art techniques—many published in just the last few years—that exploit specific characteristics of music. The goals of this tutorial are: first, to explicitly discuss the simplifying model assumptions that each computational procedure is based on; second, to present recent research directions within music structure analysis and to show how the various principles can be applied and combined; and third, to discuss problems involving the evaluation of automated procedures and the use of so-called "ground-truth" reference annotations.

## **Tutorial 3: Jingju Music: Concepts and Computational Tools for Analysis**

**Rafael Caro Repetto, Ajay Srinivasamurthy, Sankalp Gulati, and Xavier Serra**

*Music Technology Group*

*University of Pompeu Fabra*

*Barcelona, Spain*

Jingju (also known as Peking or Beijing opera) is one of the most representative genres of Chinese traditional music. From an MIR perspective, jingju music offers interesting research topics that challenge current MIR tools. The singing/acting characters in jingju are classified into predefined role-type categories with characteristic singing styles. Their singing is accompanied by a small instrumental ensemble, within which a high pitched fiddle, the jinghu, is the most prominent instrument within the characteristic heterophonic texture. The melodic conventions that form jingju modal systems, known as shengqiang, and the percussion patterns that signal important structural points in the performance offer interesting research questions. Also the overall rhythmic organization into pre-defined metrical patterns known as banshi makes tempo tracking and rhythmic analysis a challenging problem. Being Chinese a tonal language, the intelligibility of the text would require the expression of tonal categories in the melody, what offers an appealing scenario for the research of lyrics-melody relationship. The role of the performer as a core agent of the music creativity gives jingju music a notable space for improvisation. The lyrics and scores cannot be taken as authoritative sources, but as transcriptions of particular performances.

In this tutorial we will give an overview of Jingju music, of the relevant problems that can be studied from an MIR perspective and of the use of specific computational tools for its analysis. The tutorial will be organized in three parts. The first will be an introduction to Jingju from a musicological perspective, the second will cover diverse audio analysis tools of relevance to the study of Jingju (using <http://essentia.upf.edu>), and finally in the last part we will present and discuss specific examples of analyzing Jingju arias using those tools (work done in the context of <http://compmusic.upf.edu>).

## **Tutorial 4: MiningSuite, a Comprehensive Framework for Music Analysis, Articulating Audio (MIRtoolbox 2.0) and Symbolic Approaches**

**Olivier Lartillot**

*Department of Architecture, Design and Media Technology*

*Aalborg University*

*Aalborg, Denmark*

This tutorial presents an in-depth introduction to MiningSuite, a continuation of MIRtoolbox, an innovative environment featuring a large range of audio and music analysis tools. Thanks to an adaptive syntactic layer on top of Matlab, complex design of audio or music analysis operations can be written in a very concise way through a simple assemblage of operators featuring a large set of options. The integration of expertise developed in separate areas of study into common modules encourages further reuse of these individual methods and their intermingling into a common framework. The MiningSuite features an innovative and integrative set of symbolic-based musicological tools related to, among others, segmentation in the form of hierarchical grouping, melodic reduction and modal analysis. An innovative method for exhaustive pattern mining allows detailed motivic and metrical analyses. Audio and symbolic representations (in MIDI and score-like formats) and processes are tightly interconnected: Operators dedicated to high-level musical features extraction (tonal, metrical, structural analyses) integrate signal processing, statistical and symbolic-based methods, and accept both symbolic and audio input.

The tutorial, suitable for both novices and experts, will give an overview of these different audio and symbolic approaches available in the framework, and will explain how to take benefit of the capabilities of the environment via the user-friendly syntax. At the last part of the tutorial, we will dwell a little into the description of the architecture of the MiningSuite (significantly different from the previous MIRtoolbox project) and of the core classes that govern the general capabilities of the framework. Will be described for instance the rich format of the output results, or a syntactic layer within the operators' Matlab code that simplifies and clarifies the code while taking care of the matrix optimisations in the background. We will explain how you can write new modules, and will present the open-source collaborative platform hosting the MiningSuite project, with versioning control, integrated source code browsing and code review, issue tracker and user's manual available in a wiki environment.





Oral Session 1  
**Classification**

This Page Intentionally Left Blank

# ON CULTURAL, TEXTUAL AND EXPERIENTIAL ASPECTS OF MUSIC MOOD

Abhishek Singhi

Daniel G. Brown

University of Waterloo  
Cheriton School of Computer Science  
{asinghi,dan.brown}@uwaterloo.ca

## ABSTRACT

We study the impact of the presence of lyrics on music mood perception for both Canadian and Chinese listeners by conducting a user study of Canadians not of Chinese origin, Chinese-Canadians, and Chinese people who have lived in Canada for fewer than three years. While our original hypotheses were largely connected to cultural components of mood perception, we also analyzed how stable mood assignments were when listeners could read the lyrics of recent popular English songs they were hearing versus when they only heard the songs. We also showed the lyrics of some songs to participants without playing the recorded music. We conclude that people assign different moods to the same song in these three scenarios. People tend to assign a song to the mood cluster that includes “melancholy” more often when they read the lyrics without listening to it, and having access to the lyrics does not help reduce the difference in music mood perception between Canadian and Chinese listeners significantly. Our results cause us to question the idea that songs have “inherent mood”. Rather, we suggest that the mood depends on both cultural and experiential context.

## 1. INTRODUCTION

Music mood detection has been identified as an important Music Information Retrieval (MIR) task. For example, there is a MIREX audio mood classification task [12]. Though most automatic mood classification systems are solely based on the audio content of the song, some systems have used lyrics or have combined audio and lyrics features (*e.g.*, [3-5] and [6-7]). Previous studies have shown that combining these features improves classification accuracy (*e.g.*, [6-7] and [9]) but as mentioned by Downie et al. in [3], there is no consensus on whether audio or lyrical features are more useful.

Implicit in “mood identification” is the belief that songs have “inherent mood,” but in practice this assignment is unstable. Recent work has focused on associating songs with more than one mood label, where similar

mood tags are generally grouped together into the same label (*e.g.*, [2]), but this still tends to be in a stable listening environment.

Our focus is instead on the cultural and experiential context in which people interact with a work of music. People's cultural origin may affect their response to a work of art, as may their previous exposure to a song, their perception of its genre, or the role that a song or similar songs has had in their life experiences.

We focus on people's cultural origin, and on how they interact with songs (for example, seeing the lyrics sheet or not). Listening to songs while reading lyrics is a common activity: for example, there are “lyrics videos” (which only show lyrics text) on YouTube with hundreds of millions of views (*e.g.* “Boulevard of Broken Dreams”), and CD liner notes often include the text of lyrics. Our core hypothesis is that there is enough plasticity in assigning moods to songs, based on context, to argue that many songs have no inherent “mood”.

Past studies have shown that there exist differences in music mood perception among Chinese and American listeners (*e.g.*, [8]). We surmised that some of this difference in mood perception is due to weak English language skills of Chinese listeners: perhaps such listeners are unable to grasp the wording in the audio. We expected that they might more consistently match the assignments of native English-speaking Canadians when shown the lyrics to songs they are hearing than in their absence. We addressed the cultural hypothesis by exploring Canadians of Chinese origin, most of whom speak English natively but have been raised in households that are at least somewhat culturally Chinese. If such Chinese-Canadians match Canadians not of Chinese origin in their assignments of moods to songs, this might at least somewhat argue against the supposition that being Chinese in culture had an effect on mood assignment, and would support our belief that linguistic skills account for at least some of the differences. Our campus has many Chinese and Chinese-Canadians, which also facilitated our decision to focus on these communities.

In this study we use the same five mood clusters as are used in the MIREX audio mood classification task and ask the survey participants to assign a song to only one mood cluster. A multimodal mood classification could be a possible extension to our work here. Earlier works in MIR [11] had used Russell's valence-arousal model where the mood is determined by the valence and arousal scores of the song; we stick to the simpler classification here.



© Abhishek Singhi, Daniel G. Brown.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Abhishek Singhi, Daniel G. Brown. “On Cultural, Textual And Experiential Aspects Of Music Mood”, 15th International Society for Music Information Retrieval Conference, 2014.

In practice, our hypotheses about language expertise were not upheld by our experimental data. Rather, our data support the claim that both cultural background and experiential context have significant impact on the mood assigned by listeners to songs, and this effect makes us question the meaningfulness of “mood” as a category in MIR.

## 2. RELATED WORK

Mood classification is a classic task in MIR, and is one of the MIREX challenges. Several projects have used lyrics as part of the mood prediction task. Lu et al. [1] and Trohidis et al. [2] come up with an automatic mood classification system solely based on audio. Several projects like Downie et al. [3], Xiong et al. [4] and Chen et al. [5], have used lyrics as part of the mood prediction task. Downie et al. [3] show that features derived from lyrics outperform audio features in 7 out of the 8 categories. Downie et al. [6], Laurier et al. [7] and Yang et al. [9] show that systems which combine audio and lyrics features outperform systems using only audio or only lyrics features. Downie et al. [6] show that using a combination of lyrics and audio features reduces the need of training data required to achieve the same or better accuracy levels than only-audio or only-lyrics systems.

Lee et al. [8] study the difference in music mood perception between Chinese and American listeners on a set of 30 songs and conclude that mood judgment differs between Chinese and American participants and that people belonging to the same culture tend to agree more on music mood judgment. That study primarily used the common Beatles data set, which may have been unfamiliar to all audiences, given its age. Their study collected mood judgments solely based on the audio; we also ask participants to assign mood to a song based on its lyrics or by presenting both audio and lyrics together. To our knowledge no work has been done on the mood of a song when both audio and lyrics of the song is made available to the participants, which as we have noted is a common experience. Kosta et al. [11] study if Greeks and non-Greeks agree on arousal and valence rating for Greek music. They conclude that there is a greater degree of agreement among Greeks compared to non-Greeks possibly because of acculturation to the songs.

Downie et al. [3], Laurier et al. [7] and Lee et al. [8] use 18 mood tags derived from social tags and use multimodal mood classification system. Trohidis et al. [2] use multi modal mood classification into six mood clusters. Kosta et al. [11] use Russell’s valence-arousal model which has 28 emotion denoting adjectives in a two dimensional space. Downie et al. [10] use the All Music Guide datasets to come up with 29 mood tags and cluster it into five groups. These five mood clusters are used in the MIREX audio music mood classification task. We

use these clusters where each song is assigned a single mood cluster.

Mood Clusters	Mood Tags
Cluster 1	passionate, rousing, confident, boisterous, rowdy
Cluster 2	rollicking, cheerful, fun, sweet, amiable/good natured
Cluster 3	literate, poignant, wistful, bittersweet, autumnal, brooding
Cluster 4	humorous, silly, campy, quirky, whimsical, witty, wry
Cluster 5	aggressive, fiery, tense/anxious, intense, volatile, visceral

**Table 1.** The mood clusters used in the study.

## 3. METHOD

### 3.1 Data Set

We selected fifty very popular English-language songs of the 2000’s, with songs from all popular genres, and with an equal number of male and female singers. We verified that the selected songs were international hits by going to the songs’ Wikipedia pages and analyzing the peak position reached in various geographies.

We focus on English-language popular music in our study, because it is the closest to “universally” popular music currently extant, due to the strength of the music industry in English-speaking countries. Our data set includes music from the US, Canada and the UK and Ireland.

### 3.2 Participants

The presence of a large Chinese and Canadian population at our university, along with obvious cultural differences between the two communities, convinced us to use them for the study. We also include Canadians of Chinese origin; we are unaware of any previous MIR work that has considered such a group. We note that the Chinese-Canadian group is diverse: while some speak Chinese languages, others have comparatively little exposure to Chinese language or culture.

We recruited 100 participants, mostly university students, from three groups:

- 33 Chinese, living in Canada for less than 3 years.
- 33 Canadians, not of Chinese origin, born and brought up in Canada, with English as their mother tongue.
- 34 Canadians, of Chinese origin, born and brought up in Canada.

### 3.3 Survey

Each participant was asked to assign a mood cluster to each song in a set of 10 songs. For the first three songs

they saw only the lyrics; for the next three songs they only heard the first 90 seconds of the audio; and for the last four songs they had access to both the lyrics and the first 90 seconds of the audio simultaneously. They assigned each song to one of the five mood clusters shown in Table 1. We collected 1000 music mood responses for 50 songs, 300 each based solely either on audio or lyrics and 400 based on both audio and lyrics together. We note that due to their high popularity, some songs shown only via lyrics may have been known to some participants. We did not ask participants if this was the case.

#### 4. RESULTS

We hypothesized that the difference in music mood perception between American and Chinese listeners demonstrated by Hu and Lee [8] is because of the weak spoken English language skills of Chinese students, and that this might give them some difficulty in understanding the wording of songs; this is why we allowed our participants to see the lyrics for seven out of ten songs.

We had the following set of hypotheses before our study:

- People often assign different mood to the same song depending on whether they read the lyrics, or listen the audio or both simultaneously.
- Chinese-born Chinese listeners will have less consistency in the assignment of moods to songs than do Canadian-born non-Chinese when given only the recording of a song.
- Chinese-born Chinese will more consistently match Canadians when they are shown the lyrics to songs.
- Just reading the lyrics will be more helpful in matching Canadians than just hearing the music for Chinese-born Canadians.
- Canadian-born Chinese participants will be indistinguishable from Canadian-born non-Chinese participants.
- A song does not have an inherent mood: its "mood" depends on the way it is perceived by the listener, which is often listener-dependent.

##### 4.1 Lyrics and music mood perception between cultures

We started this study with the hypothesis that difference in music mood perception between Chinese and Canadian cultures is partly caused by English language skills, and that if participants are asked to assign mood to a song based on its lyrics, we will see much more similarity in judgment between two different groups.

We used the Kullback-Leibler distance between the distribution of responses from one group and the distribution of responses from that group and another group to

identify how similar the two groups' assignments of moods to songs were, and we used a permutation test to identify how significantly similar or different the two groups were. In Table 2, we show the number of songs for which different population groups are surprisingly similar. What we find is that the three groups actually agree quite a bit in uncertainty of assigning mood to songs when they are presented only with the recording: if one song has uncertain mood assignment for Canadian listeners, our Chinese listeners also typically did not consistently assign a single mood to the same song.

Our original hypothesis was that adding presented lyrics to the experience would make Chinese listeners agree more with the Canadian listeners, due to reduced uncertainty in what they were hearing. In actuality, this did not happen at all: in fact, presence of both audio and lyrics resulted in both communities having both more uncertainty and disagreeing about the possible moods to assign to a song.

This confusion in assigning a mood might be because a lot of hit songs ("Boulevard of Broken Dreams", "Viva La Vida", "You're Beautiful", *etc.*) use depressing words with very upbeat tunes. It could also be that by presenting both lyrics and audio changes the way a song is perceived by the participants and leads to a completely new experience. (We note parenthetically that this argues against using lyrics only features in computer prediction of song mood.)

The number of songs with substantial agreement between Chinese and Canadian, not of Chinese origin, participants remains almost the same with lyrics only and audio only, but falls drastically when both are presented together. (Note again: in this experiment, we are seeing how much the distribution of assignments differs for the two communities.) This contradicts our hypothesis that the difference in music mood perception between Chinese and Canadians is because of their difference in English abilities. It could of course be the case that many Chinese participants did not understand the meaning of some of the lyrics.

We had hypothesized that Canadians, of Chinese and non-Chinese origin would have very similar mood judgments because of similar English language skills but they do tend to disagree a lot on music mood. The mood judgment agreement between Chinese and Canadian, of Chinese and non-Chinese origin seem to be similar and we conclude that we can make no useful claims about the Chinese-Canadian participants in our sample.

On the whole we conclude that the presence of lyrics does not significantly increase the music mood agreement between Chinese and Canadian participants: in fact, being able to read lyrics while listening to a recording seems to significantly decrease the music mood agreement between the groups.

		lyrics	audio	audio+lyrics
Chinese	Canadians	25	22	14
Chinese	Canadian-Chinese	36	31	27
Chinese	non-Chinese Canadians	31	32	23
non-Chinese Canadians	Canadian-Chinese	36	29	31

**Table 2.** The number of statistically significantly similar responses between the different cultures for the three different ways they interact with the songs. “Canadians” refer to Canadians of both Chinese and non-Chinese origin.

#### 4.2 Stability across the three kinds of experiences

We analyze the response from participants when they are made to listen to the lyrics, hear the audio or both simultaneously across all the three groups. We calculate Shannon entropy of this mood assignment for each of the 50 songs for the three ways we presented a song to the participants: some songs have much more uncertainty in how the participants assign mood cluster to them. We then see if this entropy is correlated across the three kinds of experience, using Spearman’s rank correlation coefficient of this entropy value between the groups. A rank correlation of 1.0 would mean that the song with the most entropy in its mood assignment in one experience category is also the most entropic in the other, and so on.

	Spearman’s rank correlation coefficient
only lyrics & only audio	0.0504
only lyrics & audio+lyrics	0.1093
only audio & audio+lyrics	0.0771

**Table 3.** Spearman’s rank correlation coefficient between the groups. The groups “only lyrics” and “only audio” identify when participants had access to only lyrics and audio respectively while “audio+lyrics” refers to when they had access to both simultaneously.

The low value of the correlation analysis suggests that there is almost no relationship between “certainty” in music mood across the three different kinds of experiences: for songs like “Wake Up” by Hillary Duff and “Maria Maria” by Santana, listeners who only heard the song were consistent in their opinion that the song was from the second cluster, “cheerful”, while listeners who heard the song and read the lyrics were far more uncertain as to which class to assign the song to.

#### 4.3 “Melancholy” lyrics

For each song, we identify the mood cluster to which it was most often assigned, and show these in Table 4.

Mood Clusters	only lyrics	only audio	audio+lyrics
Cluster 1	8	9	13
Cluster 2	5	15	11
Cluster 3	28	14	18
Cluster 4	4	6	3
Cluster 5	5	6	5

**Table 4.** The most commonly assigned mood clusters for each experimental context. Most songs are assigned to the third mood cluster when participants are shown only the lyrics.

Songs experienced only with the lyrics are most often assigned to the third mood cluster, which includes the mood tags similar to “melancholy”. In the presence of audio or both audio and lyrics there is a sharp decline in the number of songs assigned to that cluster; this may be a consequence of “melancholy” lyrics being attached to surprisingly cheery tunes that cause listeners to assign them to the first two clusters. The number of songs assigned to the fourth and fifth cluster remains more similar across all experiential contexts. Even between the two contexts where the listener does hear the recording of the song, there is a good deal of inconsistency in assignment of mood to songs: for 27 songs, the most commonly identified mood is different between the “only audio” and “audio+lyrics” data.

#### 4.4 Rock songs

We explored different genres in our test set, to see if our different cultural groups might respond in predictable ways when assigning moods to songs.

Things that might be considered loud to Chinese listeners could be perceived as normal to Canadian listeners. Thus, we examined how responses differed across these two groups for rock songs, of which we had twelve in our data set. We calculate the Shannon entropy of the response of the participants and present the result in table 5.

We see that for many rock songs, there is high divergence in the mood assigned to the song by our listeners from these diverse cultures. For seven of the twelve rock songs, the most diversity of opinion is found when listeners both read lyrics and hear the audio, while for three songs, all participants who only read the lyrics agreed exactly on the song mood (zero entropy).

We see that for 3 of 12 cases all the participants tend to agree on the mood for the song when they are given access to the lyrics. The data for lyrics only have lower entropy than audio for 5 of 12 cases and all five of these songs are “rebellious” in style. For the five cases where the audio-only set has lower entropy than lyrics-only, the song has a more optimistic feel to it. This is consistent with our finding in the last section about melancholy song lyrics.

For example, the lyrics of “Boulevard of Broken Dreams”, an extremely popular Green Day song, evoke isolation and sadness, consistent with the third mood cluster. On the other hand the song’s music is upbeat which

may give the increased confusion when the participant has access to both the audio and lyrics for the song.

Song	only lyrics	only audio	audio+lyrics
“Complicated”	1.0	<b>0.918</b>	1.148
“American Idiot”	1.792	<b>1.459</b>	1.792
“Apologize”	<b>1.0</b>	1.25	1.0
“Boulevard of Broken Dreams”	<b>0.0</b>	1.792	2.155
“Bad Day”	1.792	1.459	<b>1.061</b>
“In the End”	<b>0.65</b>	1.459	1.061
“Viva La Vida”	<b>0.0</b>	1.5849	1.75
“It’s My life”	<b>0.0</b>	0.65	1.298
“Yellow”	1.792	<b>0.65</b>	1.351
“Feel”	0.918	<b>0.650</b>	1.148
“Beautiful Day”	1.584	<b>1.459</b>	1.836
“Numb”	1.25	1.918	<b>0.591</b>

**Table 5.** Entropy values for rock songs for the three different categories.

#### 4.5 Hip-Hop/ Rap

Lee et al. [8] show that mood agreement among Chinese and American listeners is least for dance songs. Our test set included five rap songs, and since this genre is often used at dance parties, we analyzed user response for this genre. Again, we show the entropy of mood assignment for the three different experiential contexts in Table 6.

What is again striking is that seeing the lyrics (which in the case of rap music is the primary creative element of the song) creates more uncertainty among listeners as to the mood of the song, while just hearing the audio recording tends to yield more consistency. Perhaps this is because the catchy tunes of most rap music pushes listeners to make a spot judgment as to mood, while being reminded of lyrics pushes them to evaluate more complexity.

In general we see that there is high entropy in mood assignment for these songs, and so we confirm the previous claim that mood is less consistent for “danceable” songs.

#### 5. DOES MUSIC MOOD EXIST?

For music mood classification to be a well-defined task, the implicit belief is that songs have “inherent mood(s),” that are detectable by audio features. Our hypothesis is that many songs have no inherent mood, but that the perceived mood of a song depends on cultural and experiential factors. The data from our study supports our hypothesis.

We have earlier shown that the mood judgment of a song depends on whether it is heard to or its lyrics is read or both together, and that all three contexts produce mood assignments that are strikingly independent.

We have shown that participants are more likely to assign a song to the “melancholic” mood cluster when only reading its lyrics, and we have shown genre-specific cultural and experiential contexts that affect how mood appears to be perceived. Together, these findings suggest that the concept of music mood is fraught with uncertainty.

The result of the MIREX audio mood classification task has had a maximum classification accuracy of less than 70% [12], with no significant recent improvements. Perhaps, this suggests that the field is stuck at a plateau, and we need to redefine “music mood” and change our approach to the music mood classification problem. Music mood is highly affected by external factors like the way a listener interacts with the song, the genre of the song, the mood and personality of the listener, and future systems should take these factors into account.

Song	only lyrics	only audio	audio+lyrics
“London Bridge”	1.459	<b>0.918</b>	1.405
“Don’t Phunk With My Heart”	1.459	<b>1.251</b>	1.905
“I Wanna Love You”	<b>0.918</b>	1.459	1.905
“Smack That”	1.918	<b>1.792</b>	1.905
“When I’m Gone”	1.251	<b>0.918</b>	1.448

**Table 6.** Entropy values for hip-hop/ rap songs for the three different categories.

#### 6. CONCLUSION

Our experiment shows that the presence of lyrics has a significant effect on how people perceive songs. To our surprise, reading lyrics alongside listening to a song does not significantly reduce the differences in music mood perception between Canadian and Chinese listeners. Also, while we included two different sets of Canadian listeners (Canadian-Chinese, and Canadians not of Chinese origin), we can make no useful conclusions about the Chinese-Canadian group.

We do consistently see that presence of both audio and lyrics reduces the consistency of music mood judgment between Chinese and Canadian listeners. This phenomenon may be because of irony caused by negative words presented in proximity to upbeat beats, or it could be that presenting both audio and lyrics together might be a completely different experience for the listener. This is an obvious setting for further work.

We have shown that the mood of a song depends on its experiential context. Interestingly, songs where listeners agree strongly about the mood of the song when only listening to the recording are often quite uncertain in their mood assignments when the lyrics are shown alongside the recording. Indeed, there is little correlation between the entropy in mood assignment between the different ways we presented songs to participants.

We also show that many “melancholy” lyrics are found in songs assigned to a more cheerful mood by listeners, again suggesting that for such songs, the extent to which listeners focus on the lyrics may influence how sad they view a song to be. We analyzed the mood assignments of participants on rock and hip-hop/rap songs. We see that people tend to agree much more to the mood of a hip-hop/rap song when they are made to listen to the song. We found that for rebellious/negative rock songs lyrics leads to more agreement in music mood but audio is better for positive songs. In both the genres we found that hearing audio while reading lyrics lead to less agreement on music mood of songs.

Our results suggest that music mood is so dependent on cultural and experiential context to make it difficult to claim it as a true concept. With the classification accuracy of mood classification systems reaching a plateau with no significant improvements we suggest that we need to re-define the term “music mood” and change our approach toward music mood classification problem.

A possible extension to our work could be running a similar study using a larger set of songs and more participants, possibly from more diverse cultures than the ones we studied. Future studies could focus on multi-modal music mood classification where a song could belong to more than one mood, to see if even in this more robust domain there is a stable way to assign songs to clusters of moods when they are experienced in different contexts. We also wonder if other contextual experiments can show other effects about mood: for example, if hearing music while in a car or on public transit, or in stores, makes the “mood” of a song more uncertain.

We fundamentally also wonder if “mood” as an MIR concept needs to be reconsidered. If listeners disagree more or less about the mood of a song when it is presented alongside its lyrics, that suggests a general uncertainty in the concept of “mood”. We leave more evidence gathering about this concept to future work as well.

## 7. ACKNOWLEDGEMENTS

Our research is supported by a grant from the Natural Sciences and Engineering Research Council of Canada to DGB.

## 8. REFERENCES

- [1] L. Lu, D. Liu, and H. Zhang, “Automatic mood detection and tracking of music audio signals”, in *IEEE Transactions on Audio, Speech, and Language Processing*, volume 14, number 1, pages 5-18, 2006.
- [2] K. Trohidis, G. Tsoumakas, G. Kalliris and I. Vlahvas, “Multi-label classification of music into emotions”, in *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR '08)*, pages 325-330, 2008.
- [3] X. Hu and J.S. Downie, “When lyrics outperform audio for music mood classification: a feature analysis”, in *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR '10)*, pages 1–6, 2010.
- [4] H. He, J. Jin, Y. Xiong, B. Chen, W. Sun, and L. Zhao, “Language feature mining for music emotion classification via supervised learning from lyrics”, in *Proceedings of Advances in the 3rd International Symposium on Computation and Intelligence (ISICA '08)*, pages 426-435, 2008.
- [5] Y. Hu, X. Chen and D. Yang, “Lyric-based song emotion detection with affective lexicon and fuzzy clustering method”, in *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR '09)*, pages 123-128, 2009.
- [6] X. Hu and J. S. Downie, “Improving mood classification in music digital libraries by combining lyrics and audio”, in *Proceedings of Joint Conference on Digital Libraries (JCDL)*, pages 159–168, 2010.
- [7] C. Laurier, J. Grivolla, and P. Herrera, “Multimodal music mood classification using audio and lyrics”, in *Proceedings of the International Conference on Machine Learning and Applications (ICMLA '08)*, pages 688-693, 2008.
- [8] X. Hu and J.H. Lee, “A cross-cultural study of music mood perception between American and Chinese listeners,” in *Proceedings of International Society for Music Information Retrieval (ISMIR '12)*, pages 535-540, 2012.
- [9] Y.H. Yang, Y.C. Lin, H.T. Cheng, I.B. Liao, Y.C. Ho, and H.H. Chen, “Toward multi-modal music emotion classification”, in *Proceedings of Pacific Rim Conference on Multimedia (PCM '08)*, pages 70-79, 2008.
- [10] X. Hu. and J.S. Downie, “Exploring mood metadata: relationships with genre, artist and usage metadata”, in *Proceedings of the 8th International Conference on Music Information Retrieval, (ISMIR '07)*, pages 67-72, 2007.
- [11] K. Kosta, Y. Song, G. Fazekas and M. Sandler, “A study of cultural dependence of perceived mood in Greek music”, in *Proceedings of the 14th International Society for Music Information Retrieval Conference, (ISMIR '13)*, pages 317-322, 2013.
- [12] X. Hu, J. S. Downie, C. Laurier, M. Bay, and A. F. Ehmann: “The 2007 MIREX audio mood classification task: Lessons learned,” in *Proceedings of the 9th International Society for Music Information Retrieval Conference, (ISMIR '08)* , pages 462–467, 2008.



# SPARSE CEPSTRAL AND PHASE CODES FOR GUITAR PLAYING TECHNIQUE CLASSIFICATION

Li Su, Li-Fan Yu and Yi-Hsuan Yang

Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan

lisu@citi.sinica.edu.tw, a999frank@gmail.com, yang@citi.sinica.edu.tw

## ABSTRACT

Automatic recognition of guitar playing techniques is challenging as it is concerned with subtle nuances of guitar timbres. In this paper, we investigate this research problem by a comparative study on the performance of features extracted from the magnitude spectrum, cepstrum and phase derivatives such as group-delay function (GDF) and instantaneous frequency deviation (IFD) for classifying the playing techniques of electric guitar recordings. We consider up to 7 distinct playing techniques of electric guitar and create a new individual-note dataset comprising of 7 types of guitar tones for each playing technique. The dataset contains 6,580 clips and 11,928 notes. Our evaluation shows that sparse coding is an effective means of mining useful patterns from the primitive time-frequency representations and that combining the sparse representations of logarithm cepstrum, GDF and IFD leads to the highest average F-score of 71.7%. Moreover, from analyzing the confusion matrices we find that cepstral and phase features are particularly important in discriminating highly similar techniques such as pull-off, hammer-on and bending. We also report a preliminary study that demonstrates the potential of the proposed methods in automatic transcription of real-world electric guitar solos.

## 1. INTRODUCTION

The use of various instrumental techniques is essential in music. A practical, interpretable automatic transcription system should provide information about playing techniques in addition to information about pitch or onset. For example, various fingering styles of the guitar, such as pull-off, hammer-on or bending, are all important elements of a guitar performance. A novice guitar player might be eager to learn the playing techniques employed in a musical excerpt of interest. Similar to some popular online automatic chord recognizer (e.g. Chordify<sup>1</sup>), a tool transcribing the note-by-note playing techniques of a guitar recording enhances the interactivity of music learning

<sup>1</sup><http://chordify.net/>

or listening experiences, and thereby offers important educational, recreational and even cultural values.

While extracting the pitch, onset, chord and instrument information from a musical excerpt has received great attention in the music information retrieval (MIR) community [3, 5, 16–18, 24], relatively little effort has been invested in transcribing the playing technique of instruments [23]. In addition, due to the use of various guitar tones (i.e. audio effects such as distortion, reverb, delay, and chorus effect) in everyday guitar performances, conventional timbre descriptors extracted from the spectrum might not be enough in modeling the electric guitar playing techniques. For instance, as the chorus effect is usually implemented by temporal delay [6], information about the phase spectrum might be important. On the other hand, for distortions that involve a filtering effect, cepstral features might be useful to characterize the respective source and filter components [8].

Motivated by the above observations, we present in this paper a comparative study evaluating the accuracy of playing technique classification of electric guitar using a variety of spectral, cepstral and phase features. The contribution of the paper is three-fold. First, to investigate more subtle variation of musical timbre, we compile an open dataset of 7 playing techniques of electric guitar, covering a variety of pitches and 7 tones (cf. Section 4). We have made the full dataset and its detailed information available online.<sup>2</sup> Second, as feature learning techniques such as dictionary learning and deep learning have garnered increasing attention in audio signal processing [12, 18, 22, 25], we evaluate the performance of sparse representations of audio signals using a dictionary adapted to the signals of interest (Section 5). Our evaluation shows that, to better model the playing techniques, it is useful to combine the sparse representation of different types of features, such as logarithm cepstrum and phase derivatives (Section 6). Finally, a preliminary study using a guitar solo demonstrates the potential of the proposed methods in automatic guitar transcription (Section 7).

## 2. RELATED WORK

Designing useful musical timbre descriptors has been a long-studied topic, and has achieved high performance in some fundamental problems such as instrument classifica-

<sup>2</sup><http://mac.citi.sinica.edu.tw/GuitarTranscription>



© Li Su, Li-Fan Yu and Yi-Hsuan Yang.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Li Su, Li-Fan Yu and Yi-Hsuan Yang. “Sparse cepstral and phase codes for guitar playing technique classification”, 15th International Society for Music Information Retrieval Conference, 2014.

tion of monophonic signals [13]. Nowadays, researchers turn to more challenging problems like multiple instrument recognition, which deals with a highly complicated timbre space [10]. Besides the complexity of multiple instruments, another challenge in timbre classification is to identify all the styles of timbre that one instrument can produce, such as to identify the playing techniques of an instrument. For example, Abeßer *et al.* and Reboursière *et al.* [1, 21] pioneered the problem of automatic guitar playing technique classification, and used timbre descriptors such as spectral flux, weighted phase divergence, spectral crest factors, brightness, and irregularity, amongst others. Most of these features are physically related to the characteristics of a plucked, vibrating string. However, these studies were not evaluated using a dataset comprising of various playing techniques and guitar tones.

In addition to larger and more realistic datasets, novel feature learning techniques might be helpful for modeling subtle timbre variations. Recently, sparse coding (SC) as a feature learning technique has been shown effective for MIR. This approach uses a predefined dictionary (codebook) to encode the prominent information of a given low-level feature representation of an input signal. One can encode any sensible audio representation by SC to capture different signal characteristics. For instance, Nam *et al.* [17] applied SC on short-time mel-spectra for music auto-tagging; Yu *et al.* [25] applied SC on logarithm cepstra and power-scale cepstra for predominant instrument recognition. Our work goes one step further and exploits phase information for SC.

### 3. ELECTRIC GUITAR PLAYING TECHNIQUE

Table 1 lists the 7 playing techniques we consider in this work. Most guitar solos are constructed with these techniques. For example, muting is widely used alternatively in place of normal in guitar *riffs* for rhythmic and punched phrases in rock and metal music, and bending is commonly considered to be the most important technique for expressing emotion.

To gain more insights into the signal-level properties of the playing techniques, in Fig. 1 we show the spectrograms (the first row) and the short-time cepstra (the second row) of the individual-note examples played with the 7 playing techniques. The first three columns are individual notes F4 of normal, vibrato and mute, the fourth column the consecutive notes F4–E4 of pull-off, and the last three columns the consecutive notes F4–#F4 of hammer-on, sliding and bending. The length of all samples is 0.6s. The window size is 46ms and the hop size is 10ms. From the spectrograms and the short-time cepstra, we see that muting has a ‘noisier’ attack and a faster decay comparing to normal. Moreover, hammer-on, sliding and bending have quite different transition behaviors, although they have the same note progression. The transition is sharp for hammer-on; smooth for bending; and there is a two-stage transition for sliding. Therefore, it seems that both the spectrogram and the cepstra contain useful information that can be exploited for automatic classification.

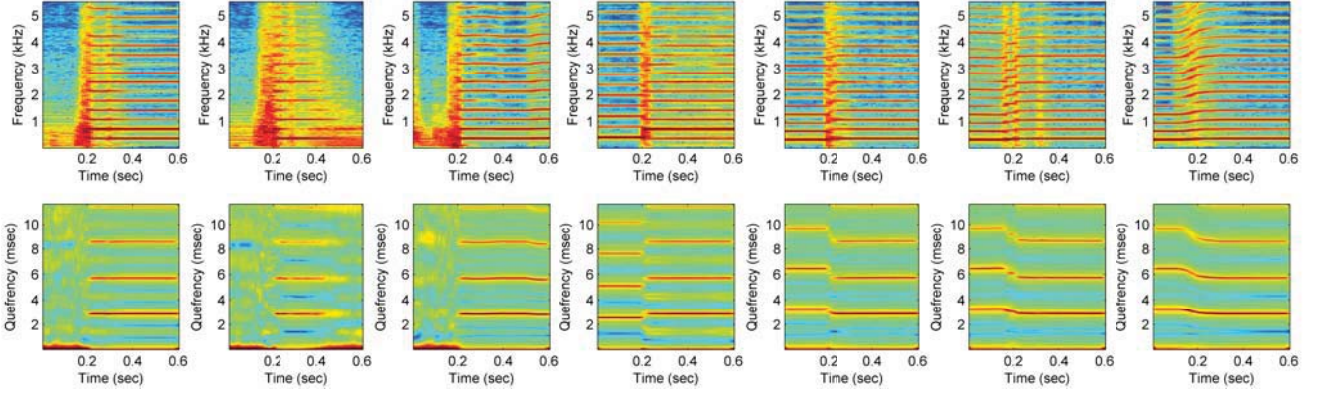
Technique	Description	# clips
Normal	Normal sound	2,009
Muting	Sounds muted (by right hand) to create great attenuation	385
Vibrato	Trilled sound produced by twisting left hand finger on the string	637
Pull-off	Sound similar to normal but with the smoother attack created by pulling off the string by left hand finger	525
Hammer-on	Sound similar to normal but with the smoother attack created by hammering on the string by left hand finger	581
Sliding	Discrete change to the target note with a smooth attack by left hand finger sliding through the string	1,162
Bending	Continuous change to the target note without an apparent attack by bending the string by left hand fingers	1,281

**Table 1.** Description of the playing techniques considered.

## 4. DATASET

While there is no publicly available dataset for guitar playing technique classification across different tones, we establish our own one with the aforementioned 7 playing techniques. The dataset is recorded by a professional guitarist using a recording interface, PreSonus’ AudioBox USB, with bit depth of 24 bits and frequency response from 14 Hz to 70 kHz. We directly line-in the guitar to recording interface to catch every nuance of sound and exclude environmental noise. The guitar for recording is ESP’s MII with Seymour Duncan’s pickup and Ebony finger board, which is a high-quality guitar especially for metal and rock music. To make the quality of the sound recordings akin to that of real-world performance, we augment the single clean tone source to different guitar tones, which is done in the post-production stage using music production software Cubase. In addition, we assign each audio clips to 7 different guitar tones, which involve different levels of *distortion*, *reverb*, *delay* and *chorus*. Such tones may represent different genres such as rock, metal, funk, and country music solos. Moreover, the tones are carefully tuned to meet the quality for listening.

Because of the different characteristics of the techniques, the clips are recorded in slightly different ways. All the clips of sliding and bending have 2 notes for each clip with both whole step (2 semitones) and half step (1 semitone); all the clips of hammer-on and pull-off have 2 notes with only half step; and the clips of vibrato and muting have only one note for each clip. As for normal, we record whole steps, one steps, and single notes to cover all possible cases which might occur in the other 6 techniques. For sliding and bending, we record the clips only with the first three strings of the guitar since these techniques are less frequently applied on the last 3 strings. Similarly, we record muting clips with only the last 3 strings because it is commonly used in rhythm guitar with low pitch. Other



**Figure 1.** Spectrograms (the first row) and short-time cepstra (the second row) of the seven playing techniques considered in this study. From left to right: normal, muting, vibrato, pull-off, hammer-on, sliding, bending.

playing techniques are recorded with all the 6 strings. As a result, we can see from Table 1 that the numbers of clips of the 7 techniques are different, where normal has the largest number of 2,009 notes and muting has the smallest number of 385 notes. In total there are 6,580 clips.

## 5. METHODS

### 5.1 Feature representation

Our feature processing procedures have two steps: low-level feature extraction and sparse coding. In low-level feature extraction, we select spectrogram (SG), group-delay function (GDF), instantaneous frequency deviation (IFD), logarithm cepstrum (CL) and power cepstrum (CP), all of which are derived quantities from the short-time Fourier transformation (STFT):

$$S^h(t, \omega) = \int x(\tau) h(\tau - t) e^{-j\omega\tau} d\tau = M^h(t, \omega) e^{j\Phi^h(t, \omega)}, \quad (1)$$

where  $x(t) \in \mathbb{R}$  is the input signal,  $S^h(t, \omega) \in \mathbb{C}$  stands for the two-dimensional STFT representation on time-frequency plane, and  $h(t)$  refers to the window function. SG is the magnitude part of the STFT representation:  $SG^h(t, \omega) = |S^h(t, \omega)|$ . Phase spectrum is the imaginary part of the logarithm spectrum:  $\Phi^h(t, \omega) = \text{Im}(\log S^h(t, \omega))$ . IFD and GDF are the derivative of phase  $\Phi$  over time and frequency, respectively:

$$\text{IFD}^h(t, \omega) = \frac{\partial \Phi^h}{\partial t} = \text{Im} \left( \frac{S^{\mathcal{D}h}(t, \omega)}{S^h(t, \omega)} \right), \quad (2)$$

$$\text{GDF}^h(t, \omega) = -\frac{\partial \Phi^h}{\partial \omega} - t = \text{Re} \left( -\frac{S^{\mathcal{T}h}(t, \omega)}{S^h(t, \omega)} \right), \quad (3)$$

where  $\mathcal{D}$  and  $\mathcal{T}$  represent operators on window functions:  $\mathcal{D}h(t) = h'(t)$  and  $\mathcal{T}h(t) = t \cdot h(t)$ . Detailed derivation procedures of GDF and IFD can be found in [2]. On the other hand, CL and CP are calculated as

$$\text{CL}^h(t, q) = (S^h)^{-1}(\log |S^h(t, \omega)|), \quad (4)$$

$$\text{CP}^h(t, q) = (S^h)^{-1}(|S^h(t, \omega)|^{1/3}), \quad (5)$$

where  $(S^h)^{-1}(\cdot)$  denotes the inverse STFT and  $q$  denotes quefrency [19]. Features derived from CL, such as the

Mel-frequency cepstral coefficients (MFCCs), are often employed in audio signal processing [8, 16].

### 5.2 Sparse coding and dictionary learning

For any one of the aforementioned low-level features, denoted as  $\mathbf{y} \in \mathbb{R}^m$ , we further convert it to a sparse representation  $\alpha \in \mathbb{R}^k$  by SC. Specifically, SC involves the following  $l_1$ -regularized LASSO problem [7] to encode  $\mathbf{y}$  over a given dictionary  $\mathbf{D} \in \mathbb{R}^{m \times k}$ .

$$\hat{\alpha} = f_{\text{SC}}(\mathbf{D}, \mathbf{y}) = \arg \min_{\alpha} \|\mathbf{y} - \mathbf{D}\alpha\|_2^2 + \lambda \|\alpha\|_1. \quad (6)$$

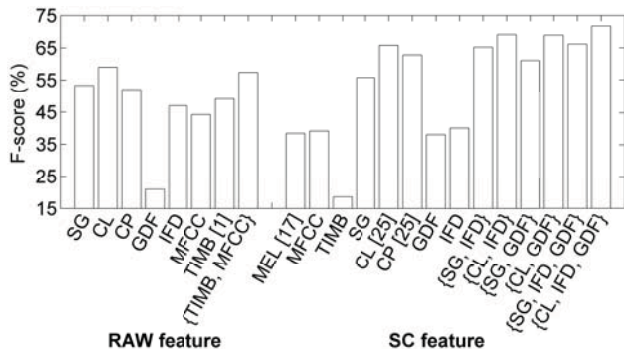
The LASSO problem can be efficiently solved by for example the least angle regression (LARS) algorithm [7]. Moreover, the dictionary  $\mathbf{D}$  is learned by the online dictionary learning (ODL) [15] implemented by the open-source package SPAMS (<http://spams-devel.gforge.inria.fr/>). The SC result when the input  $\mathbf{y}$  is CL has been referred to as the sparse cepstral code [25].

## 6. EXPERIMENT

### 6.1 Experimental setup of individual notes

As Fig. 1 illustrates, the playing techniques can be better identified around the onsets for most cases. Therefore, our system starts from detecting the onset of each clip and then extracts features from each segment starting from the time before the onset by  $t_a$  second to the time after the onset by  $t_b$  second. We use the well-known spectral flux method [11] for onset detection, and empirically set  $t_a = 0.1$  and  $t_b = 0.2$  for all the clips. For STFT, we use Hanning window of window size 46 ms (1,024 samples) and hop size of 10 ms (441 samples). Under the sampling rate of 44.1 kHz, the dimension of all the low level features is 512 (i.e. considering only positive frequency).

We adopt a five-fold jack-knife cross-validation (CV) scheme for the evaluation. For all the fold partitions, the distribution of clips over the playing techniques is balanced. We learn both the classifier and the ODL dictionary from the training folds only, without using the test fold. The number of atoms  $k$  of each dictionary is set to



**Figure 2.** Average accuracies (in F-scores) of playing technique classification using various feature combination. Left part: RAW features; right part: SC features.

512.<sup>3</sup> After obtaining the frame-level sparse codeword  $\alpha$ , a clip-level feature representation is constructed by mean pooling. Finally, the features, either with or without sparse coding, are fed into linear support vector machine (SVM) [9], with the parameter  $C$  optimized through an inside CV on the training data from the range  $\{2^{-10}, 2^{-9}, \dots, 2^{10}\}$ . The evaluation results on the test set are reported in terms of F-score, which is the harmonic mean of precision and recall. All the evaluation is done at the clip-level.

We consider a number of baseline approaches for comparison. First, we use the MIRtoolbox (version 1.3.4) [14] to compute a total number of 41 features covering the temporal, spectral, cepstral and harmonic aspects of music signals (denoted as ‘TIMB’ in Fig. 2) as an implementation of a prior art on guitar playing technique classification [1]. Second, the conventional MFCC,  $\Delta$ MFCC and  $\Delta\Delta$ MFCC are also used for their popularity (denoted as ‘MFCC’). Third, we try the early fusion of MFCC and TIMB (i.e. by concatenating the corresponding clip-level representations to form a longer feature vector). Finally, for the features learned by SC, we note that the sparse representation of the mel-spectra (denoted as ‘MEL’) was used in [17], and the sparse representations of CL and CP were used in [25]. However, please note that the focus here is to compare the performance of using different features for the task, so our implementation does not faithfully follow the ones described in the prior arts. For example, Nam *et al.* uses automatic gain control as a pre-processing and uses multiple frame representation instead of frame-level features as input to feature encoding [17]. For simplicity the feature extraction and classification pipelines have been kept simple in this study.

We apply SC to all the five low-level features described in Section 5.1 and consider a number of early fusion of them. No normalization is performed for SC features. However, for non-SC features (referred to as ‘RAW’), it is useful to apply a z-score normalization so that each feature dimension has zero mean and unit variance.

<sup>3</sup> Using an over-complete dictionary (i.e.  $k \gg m$ ) usually improves the performance of SC features [25], but we leave this as a future work.

(a) SC+SG

	predicted class							F-score	
	nor	mut	vib	pul	ham	sli	ben		
actual class	nor	<b>92.6</b>	3.26	1.07	1.01	0.85	0.90	0.38	55.2
	mut	44.0	<b>43.7</b>	6.94	1.13	0.32	0.97	2.90	56.1
	vib	31.0	4.93	<b>63.8</b>	0.27	0.00	0.00	0.00	74.1
	pul	21.0	1.75	0.00	<b>21.8</b>	16.9	34.2	4.47	29.7
	ham	31.4	0.36	0.18	12.6	<b>25.8</b>	25.6	4.14	33.1
	sli	11.9	0.94	0.00	7.92	10.9	<b>52.7</b>	15.6	46.1
	ben	3.56	0.92	0.11	2.18	1.26	14.5	<b>77.5</b>	75.6

(b) SC+CL

	predicted class							F-score	
	nor	mut	vib	pul	ham	sli	ben		
actual class	nor	<b>95.6</b>	1.01	0.41	0.82	0.63	1.20	0.30	58.6
	mut	38.9	<b>54.4</b>	4.35	0.16	0.00	0.65	1.45	66.3
	vib	14.3	6.03	<b>79.7</b>	0.00	0.00	0.00	0.00	86.3
	pul	27.2	0.58	0.19	<b>28.2</b>	14.6	25.6	3.69	38.9
	ham	31.4	0.00	0.00	9.55	<b>38.2</b>	18.2	2.70	47.2
	sli	14.9	1.42	0.00	4.43	7.26	<b>61.8</b>	10.2	56.7
	ben	3.79	0.69	0.00	1.84	1.03	10.5	<b>82.2</b>	81.9

(c) SC+{CL,GDF,IFD}

	predicted class							F-score	
	nor	mut	vib	pul	ham	sli	ben		
actual class	nor	<b>95.6</b>	1.59	0.33	0.55	0.79	0.79	0.36	64.1
	mut	35.0	<b>57.9</b>	4.52	0.32	0.16	0.32	1.77	68.7
	vib	12.3	6.85	<b>80.8</b>	0.00	0.00	0.00	0.00	86.9
	pul	19.6	0.58	0.19	<b>41.2</b>	11.7	22.5	4.27	52.0
	ham	24.3	0.18	0.00	10.5	<b>45.8</b>	17.5	1.80	55.2
	sli	10.2	1.13	0.19	5.66	6.60	<b>70.4</b>	5.85	65.0
	ben	1.38	0.23	0.00	0.23	0.80	5.17	<b>92.2</b>	89.4

**Table 2.** Confusion matrix (in %) of playing technique classification of electric guitar individual notes using different feature combinations.

## 6.2 Experiment results

From the left hand side of Figure 2, we find that both RAW+TIMB [1] and RAW+MFCC perform worse than RAW+SG, RAW+CL and RAW+CP, possibly because the feature dimension of the latter three is larger. However, after fusing TIMB and MFCC, the F-score is improved to 57.4%, which is not significantly worse than the result of RAW+CL (i.e. 59.0%) under the two-tailed t-test. It turns out that using sophisticated features such as those computed by the MIRtoolbox does not offer gain for this task. Note that the F-score of random guess would be  $1/7=14.3\%$ , because each fold is balanced across the 7 techniques. The performance of most RAW features is greatly better than the chance level.

In contrast, from the right hand side of Figure 2, we find that SC features usually performs much better than the non-SC (i.e. RAW) counterparts. For example, SC+SG, SC+CL and SC+CP are better than RAW+SG, RAW+CL and RAW+CP, respectively. These improvements are all significant under the two-tailed t-test ( $p < 0.01$ ,  $d.f. = 8$ ). Similar observations have been made in existing works that

apply SC features to MIR tasks (e.g. [17, 25]). We also find that using SC+CL already leads to significantly better F-score than RAW+{TIMB,MFCC} ( $p < 0.0001$ ,  $d.f.=8$ ). Moreover, from the data of SC features we see that fusing GDF and IFD generally improves the accuracy, and that the best F-score 71.7% is obtained by fusing sparse-coded cepstral and phase features (i.e. SC+{CL,GDF,IFD}). The F-score of SC+{SG,GDF,IFD} is worse (66.1%) than SC+{CL,GDF,IFD}, but is still significantly better than SC+SG. We also note that SC does not improve the performance for MEL, MFCC, TIMB and IFD. This implies that sophisticated features like TIMB are not suitable for SC. Although SC+IFD is worse than IFD, its fusion with other SC features still results in better performance. In a nutshell, this evaluation shows that it is promising to use SC for playing technique classification, especially when we fuse multiple features derived from STFT.

Table 2 displays the confusion matrices for three different feature combinations with sparse coding. Table 2(a) shows the result of SC+SG, from which we see that normal and bending have relatively high F-scores of 74.1% and 75.6% (see the rightmost column), yet the other five techniques have F-scores lower. We see that many playing techniques can be easily misclassified as normal. We also see ambiguities between for example pull-off versus sliding and hammer-on versus sliding, showing that such techniques are difficult to be discriminated from one another in the logarithm-scale spectrogram.

In contrast, we see from Table 2(b) that SC+CL leads to consistent improvement in F-score for all the playing techniques, comparing to SC+SG. The largest performance gain (+14.1%) is obtained for hammer-on. We also see that the ambiguity between normal and vibrato is mitigated.

Finally, comparing Tables 2 (b) and (c) we see that SC+{CL,GDF,IFD} consistently improves the F-score for all the playing techniques. More interestingly, it seems that adding phase derivatives effectively alleviate the aforementioned confusions without compromising the discriminability of other classes. The F-scores of all the playing techniques are now above 50.0%.

## 7. REAL-WORLD MUSIC

The automatic transcription flow contains frame-level pitch detection, onset detection, and playing technique classification, one after another. We adopt the method proposed by Peeters [20] and use spectral and cepstral features for pitch detection. For onset detection, we use again the spectral flux method [4, 11]. Finally, we apply the playing technique classifier trained from the individual note dataset to classify the playing techniques of the guitar solo.

We present a qualitative evaluation of a real-world electric guitar solo excerpt performed by same professional guitarist. It is an interpretation of Sonata Artica's Tallulah released in 2001, for the fragment 3:59–4:08. We show in the first two subfigures of Fig. 3 its scoresheet and spectrogram. In the third subfigure we show the pitch and onset, using black horizontal bars, gray horizontal bars, and vertical dashed lines to denote the estimated frame-

level pitches, ground truth pitches, and estimated onsets, respectively. We see that the estimated pitches and onsets match the ground truth quite well, except for some cases such as the mismatch between the onset at 7.70s and the change of pitch at 7.84s, which probably results from the ambiguity of the onset of bending.

The last subfigure of Fig. 3 compares the result of SC+SG and SC+{CL,GDF,IFD} for playing technique classification. Since our classification is performed with respect to the detected onsets, the errors in the stage of onset detection will fully propagate into the stage of playing technique classification. Therefore, the techniques which are not characterized by onset (e.g., a long-sustaining vibrato) cannot be transcribed. A true positive of onset is defined as an onset position which is detected within 100ms of the ground truth onset time. A true positive of playing technique is accordingly defined as a correct prediction of playing technique at a true positive of onset. We can see that the performance of playing classification degrades a lot in comparison to the case of individual notes. Specifically, we have 7 true positives (4 normal and 3 bending) for SC+{CL,GDF,IFD} and 5 true positives (2 sliding, 2 bending and 1 normal) for SC+SG, while there are in total 17 targets in the ground truth. The 2 muting at 2.38s and 4.60s and the hammer-on at 9.24 second are not recalled by both methods. Although SC+{CL,GDF,IFD} fails to recall sliding, SC+SG recalls 2 sliding. While SC+{CL,GDF,IFD} has many false positives of vibrato, SC+SG has many false positives of sliding. In general, SC+{CL,GDF,IFD} performs better.

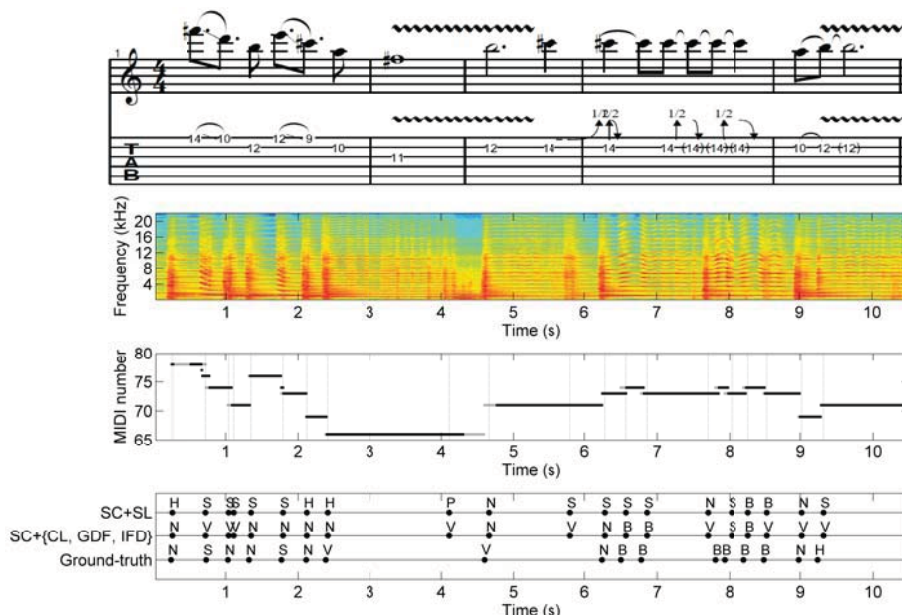
The two estimated events at 4.11s and 5.80s are interesting. Although the two events do not present in the ground truth, the prediction of SC+{CL,GDF,IFD} is musically correct as the two false alarms of onset indeed occur in a long-sustaining vibrato. In contrast, SC+SG misclassifies the two events as pull-off and sliding, respectively.

## 8. CONCLUSION

In this study, we have reported a comparative study on the performance of a number of timbre modeling methods for the relatively unexplored task of guitar playing technique classification. The evaluation is performed on a large-scale individual-note dataset comprising of 6,580 clips and a real-world guitar solo recording. Our evaluation shows that sparse coding works well in learning features that are useful for the task, and that using features extracted from the cepstra and phase derivatives helps resolve the confusion among similar playing techniques. We also report a qualitative evaluation on guitar solo transcription. We are currently collecting more individual notes and solos to deeply understand the signal-level characteristics for these playing techniques. Although the present study might be at best preliminary, we hope it can call for more attention towards playing technique modeling.

## 9. ACKNOWLEDGMENTS

This work was supported by the Academia Sinica Career Development Award 102-CDA-M09.



**Figure 3.** Result of transcribing a real-world guitar solo excerpt. From top to bottom: scoresheet, guitar tab, spectrogram, pitch and onset (gray bar: ground truth; black bar: estimated pitch; vertical dashed line: estimated onset), and result of playing technique classification by using SC+SG and SC+{CL,GDF,IFD}. Abbreviation: N=normal, V=vibrato, M=muting, P=pull-off, H=hammer-on, S=sliding, B=bending.

## 10. REFERENCES

- [1] J. Abeßer et al. Feature-based extraction of plucking and expression styles of the electric bass guitar. In *ICASSP*, pages 2290–2293, 2010.
- [2] F. Auger and P. Flandrin. Improving the readability of time-frequency and time-scale representations by the method of reassignment. *IEEE Trans. Sig. Proc.*, 43(5):1068–1089, 1995.
- [3] A. M. Barbancho et al. Automatic transcription of guitar chords and fingering from audio. *IEEE Trans. Audio, Speech, and Language Processing*, 20(3):915–921, 2012.
- [4] J. P. Bello et al. A tutorial on onset detection in music signals. *IEEE Speech Audio Process.*, 13(5-2):1035–1047, 2005.
- [5] E. Benetos et al. Automatic music transcription: challenges and future directions. *J. Intelligent Information Systems*, 41(3):407–434, 2013.
- [6] J. Dattorro. Effect design, part 2: Delay line modulation and chorus. *J. Audio engineering Society*, 45(10):764–788, 1997.
- [7] B. Efron et al. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.
- [8] A. Eronen and A. Klapuri. Musical instrument recognition using cepstral coefficients and temporal features. In *ICASSP*, pages 753–756, 2000.
- [9] R.-E. Fan et al. LIBLINEAR: A library for large linear classification. *J. Machine Learning Research*, 2008.
- [10] P. Hamel et al. Automatic identification of instrument classes in polyphonic and pply-instrument audio. In *ISMIR*, 2009.
- [11] A. Holzapfel et al. Three dimensions of pitched instrument onset detection. *IEEE Trans. Audio, Speech, Language Process.*, 18(6):1517–1527, 2010.
- [12] E. J. Humphrey et al. Feature learning and deep architectures: new directions for music informatics. *J. Intelligent Information Systems*, 41(3):461–481, 2013.
- [13] A. Klapuri and M. Davy, editors. *Signal Processing Methods for Music Transcription*, chapter 6. Springer, 2006.
- [14] O. Lartillot and P. Toivainen. A Matlab toolbox for musical feature extraction from audio. In *DAFx*, 2007.
- [15] J. Mairal et al. Online dictionary learning for sparse coding. In *Int. Conf. Machine Learning*, pages 689–696, 2009.
- [16] M. Müller et al. Signal processing for music analysis. *IEEE J. Sel. Topics Signal Processing*, 5(6):1088–1110, 2011.
- [17] J. Nam et al. Learning sparse feature representations for music annotation and retrieval. In *ISMIR*, pages 565–560, 2012.
- [18] K. O’Hanlon and M. D Plumbley. Automatic music transcription using row weighted decompositions. In *ICASSP*, 2013.
- [19] A. V. Oppenheim and R. W. Schaffer. *Discrete-Time Signal Processing*. Prentice Hall, 2010.
- [20] G. Peeters. Music pitch representation by periodicity measures based on combined temporal and spectral representations. In *ICASSP*, 2006.
- [21] L. Reboursière et al. Left and right-hand guitar playing techniques detection. In *NIME*, 2012.
- [22] L. Su and Y.-H. Yang. Sparse modeling for artist identification: Exploiting phase information and vocal separation. In *ISMIR*, pages 565–560, 2013.
- [23] L. Su and Y.-H. Yang. Sparse modeling of subtle timbre: a case study on violin playing technique. In *WOCMAT*, 2013.
- [24] K. Yazawa et al. Automatic transcription of guitar tablature from audio signals in accordance with player’s proficiency. In *ICASSP*, 2014.
- [25] L.-F. Yu et al. Sparse cepstral codes and power scale for instrument identification. In *ICASSP*, 2014.

# AUTOMATED DETECTION OF SINGLE- AND MULTI-NOTE ORNAMENTS IN IRISH TRADITIONAL FLUTE PLAYING

Münevver Köküer<sup>1,2</sup>, Peter Jančovič<sup>2</sup>, Islah Ali-MacLachlan<sup>1</sup>, Cham Athwal<sup>1</sup>

<sup>1</sup>DMT Lab, Birmingham City University, UK

<sup>2</sup>School of Electronic, Electrical & Systems Engineering, University of Birmingham, UK  
 {munevver.kokuer, islah.ali-maclachlan, cham.athwal}@bcu.ac.uk  
 p.jancovic@bham.ac.uk

## ABSTRACT

This paper presents an automatic system for the detection of single- and multi-note ornaments in Irish traditional flute playing. This is a challenging problem because ornaments are notes of a very short duration. The presented ornament detection system is based on first detecting onsets and then exploiting the knowledge of musical ornamentation. We employed onset detection methods based on signal envelope and fundamental frequency and customised their parameters to the detection of soft onsets of possibly short duration. Single-note ornaments are detected based on the duration and pitch of segments, determined by adjacent onsets. Multi-note ornaments are detected based on analysing the sequence of segments. Experimental evaluations are performed on monophonic flute recordings from Grey Larsen's CD, which was manually annotated by an experienced flute player. The onset and single- and multi-note ornament detection performance is presented in terms of the precision, recall and  $F$ -measure.

## 1. INTRODUCTION

Within Irish traditional music, ornaments are used extensively by all melody instruments. They are central to the style of the music, adding to its liveliness and expression. Amongst traditional players, the melody is merely a framework [3, 4] – dynamics, ornamentation and context will be added in real time. This is often different from classical music where a standard notation for each piece of music usually includes ornaments as written by the composer.

Ornaments are notes of a very short duration. They can be categorised into single-note and multi-note ornaments. Single-note ornaments are amongst the most common in Irish traditional music. Multi-note ornaments consist of a specific sequence of note and single-note ornaments.

Methods for ornament detection are typically based on detection of note onsets. Note onsets may be categorised as hard or soft. A hard onset, typical in percussive instruments, is characterised by a sudden change in energy. A soft onset shows a more gradual change in energy and it occurs in wind instruments, like flute. A variety of methods have been proposed for the detection of note onsets in music recordings, e.g., [1, 8, 11, 13, 17]. The methods typically exploit the change in the energy of the signal, which may be estimated in temporal or spectral domain. The use of phase has also been investigated, e.g., [1, 11], and combined with the fundamental frequency in [11]. It has been reported that reliable note onset detection for non-percussive instruments is more difficult to obtain due to the soft nature of the onsets [11].

An automated detection of ornaments is a challenging problem. This is because ornaments are of very short durations, which may cause them being easily omitted or falsely detected. Unlike note onset detection, this research area has received relatively little attention. An automatic location of ornaments for flute recordings based on MPEG-7 features was investigated in [5]. Transcription of baroque ornaments in two piano recordings by analysing rhythmic groupings and expressive timing was studied in [2]. This work used onset values from manually edited time-tagged audio. Several works employed spectral-domain energy-based onset detection, e.g., [9, 10, 16]. The work in [16] analysed ornamentation from Bassoon recordings. The work of a group from Dublin Institute of Technology, summarised in [9], is the only study on the detection of ornaments in Irish traditional flute music. This provided only some initial results and on a considerably smaller dataset.

In this paper, we extend our recent work presented in [14] and investigate automatic detection of single- and multi-note ornaments in flute playing. The presented ornament detection system is based on first detecting onsets and then exploiting knowledge of musical ornamentation. We explore the use of several different methods for onset detection and customisation of their parameters to detection of soft onsets of notes which may be also of very short duration. The detected onsets provide segmentation of the signal, where a segment is defined by the adjacent detected onsets. This segmentation, together with the musical knowledge of ornamentation is then used for the detection of single- and multi-note ornaments. Experimental evalua-



© Münevver Köküer<sup>1,2</sup>, Peter Jančovič<sup>2</sup>, Islah Ali-MacLachlan<sup>1</sup>, Cham Athwal<sup>1</sup>.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Münevver Köküer<sup>1,2</sup>, Peter Jančovič<sup>2</sup>, Islah Ali-MacLachlan<sup>1</sup>, Cham Athwal<sup>1</sup>. "AUTOMATED DETECTION OF SINGLE- AND MULTI-NOTE ORNAMENTS IN IRISH TRADITIONAL FLUTE PLAYING", 15th International Society for Music Information Retrieval Conference, 2014.

tions are performed using recordings of Irish traditional tunes played by flute from Grey Larsen’s CD [15]. Results of ornament detection are presented in terms of the precision, recall and  $F$ -measure. The average  $F$ -measure performance for single- and multi-note ornaments is over 76% and 67%, respectively.

## 2. SINGLE- AND MULTI-NOTE ORNAMENTS IN IRISH TRADITIONAL FLUTE PLAYING

Ornaments are used as embellishments in Irish traditional music [15]. They are notes of a very short duration, created through the use of special fingered articulations.

Single-note ornaments, namely ‘cut’ and ‘strike’, are pitch articulations. The ‘cut’ involves quickly lifting and replacing a finger from a tonehole, and corresponds to a higher note than the ornamented note. The ‘strike’ is performed by momentarily closing an open hole, and corresponds to a lower note than the ornamented note.

Multi-note ornaments are successive use of single-note ornaments. To simplify the description, we refer to the ornamented note as the base note throughout the rest of this paper. The ‘roll’ consists of the base note, a ‘cut’, base note, a ‘strike’ and then returning to the base note. A shorter version of the roll, referred to as short-roll, omits the starting base note. The ‘crann’ consists of the base note that is cut three times in rapid succession and then returning to the base note. The short-crann omits the starting base note. The ‘shake’ commences with a ‘cut’, followed by a base note and a second ‘cut’ and then returning to the base note.

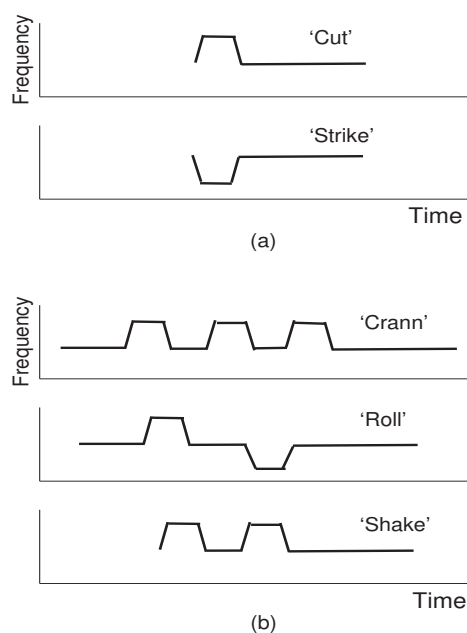
A schematic visualisation of the single- and multi-note ornaments is given in Figure 1. In the multi-note ornaments figure, the proportions of the length of the individual parts aim to approximately indicate the typical duration proportions. For instance, in theory, a roll would be split equally into three parts by the cut and the strike but in reality different players will time this differently according to the ‘swing’ of the tune, their muscle control and a host of other attributes that make up their personal style.

## 3. AUTOMATIC DETECTION OF ORNAMENTS

This section presents the developed automatic ornament detection system. We first give a brief description of the onset detection methods we employed and then describe how the detected onsets are used for the detection of single- and multi-note ornaments.

### 3.1 Methods for detection of onsets

Here we briefly describe three onset detection methods we employed. Two of the methods exploit the change of the signal amplitude over time, with processing performed in the temporal and spectral domain [1, 8]. The third method is based on the fundamental frequency [6, 11]. Each of the method requires several parameters to be set and their values are explored during experimental evaluations and presented later in Section 4.3. The implementation of the



**Figure 1.** A schematic representation of single-note (a) and multi-note (b) ornaments.

temporal domain energy-based method used in parts some functions from the MIRtoolbox.

#### 3.1.1 Signal energy: spectral domain

This method, also sometimes referred to as spectral-flux method, performs onset detection in the spectral domain. The signal is segmented into overlapping frames. Each signal frame is multiplied by Hamming window. The windowed frames are then zero padded and the Fourier transform is applied to provide the short-term Fourier spectrum. For each frequency bin, the differences between the short-term magnitude spectra of successive signal frames is computed. This is then half-wave rectified and the  $L_2$  norm is calculated to provide the value of the detection function at the current frame. The peaks of the detection function, whose amplitude is above a threshold are used as the detected onsets. We explored the use of a fixed threshold value as well as computing the value adaptively based on the median of the detection function values around the current frame. Finally, if two consecutive peaks are found within a given time distance, only the first peak is used.

#### 3.1.2 Signal energy: temporal domain

Another method we employed performs the detection in temporal domain. The signal is passed through a bank of fourteen band pass filters, each tuned to a specific note on the flute in the range from  $D_4$  to  $B_5$ . The filters have non-overlapping bands, with the lower and the upper frequency being half way between the adjacent note frequencies. These fourteen notes are readily playable on an unkeyed concert flute. The signal in each band is full-wave rectified and then smoothed, resulting in amplitude envelope. The time derivative of the amplitude envelope is calculated in each band and this is smoothed by convolving



it with a half-Hanning window. We explored several ways of making decision about detected onsets. The information from all bands can be combined by summing together their smoothed derivative signals. Alternatively, a single band can be chosen as a representative at each time based on assessment of amplitudes of peaks around that time across all bands. Onsets are obtained by comparing the values of peaks to a threshold, which may be fixed or adaptive over time.

### 3.1.3 Fundamental frequency

In addition to methods exploiting the signal envelope, we also explore the use of the fundamental frequency ( $F_0$ ). This has been reported to be beneficial for soft onset detection in [11]. Among a large variety of existing  $F_0$  estimation algorithms, we employed the YIN algorithm [7] in this work. The  $F_0$  estimation may result in so called doubling / halving errors. To help dealing with these errors, the  $F_0$  estimates are postprocessed using a median filter. The length of this filter needs to be set sensitively – a longer filter may be preferable to deal with the  $F_0$  estimation errors but this may also cause filtering out ornaments, which are characterised by their short duration.

The detection function at the frame time  $n$ , denoted as  $R_n$ , is based on calculating the change of  $F_0$  over time. This can be performed by taking the difference between the  $F_0$  estimate at the frame  $(n + \Theta)$  and  $(n - \Theta)$ . The onset is detected as the first frame for which  $abs(R_n) > \alpha_{F_0}$ , where the value of the threshold  $\alpha_{F_0}$  relates to the difference between frequencies of the closest possible notes.

## 3.2 Ornament detection

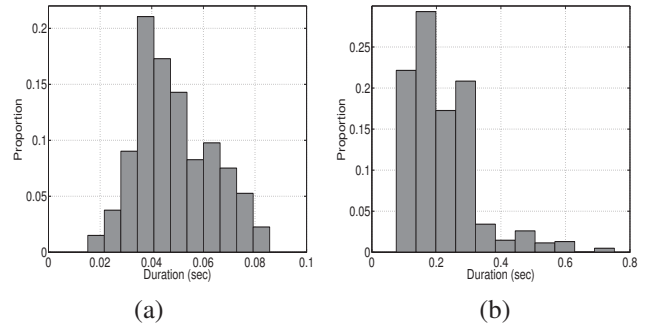
The detected onsets, as obtained using the methods described in Section 3.1, provide a segmentation of the signal, where each segment is formed based on the adjacent detected onsets.

We characterise each detected segment by some features, specifically, here we use the duration of the segment and its segmental fundamental frequency. For a given segment, its duration, denoted by  $D^{seg}$ , is obtained based on the detected onsets and its fundamental frequency, denoted by  $F_0^{seg}$ , is calculated as the median value of the  $F_0$ s corresponding to all signal frames assigned to that segment. Finally, these segment features are used to determine whether the detected segment corresponds to a note or a single-note ornament and whether the sequence of segments corresponds to a multi-note ornament, and if single- or multi-note ornament is detected, then to determine its type.

### 3.2.1 Single-note ornament detection

As single-note ornaments are expected to be of a shorter duration than notes, we examined whether the duration of the detected segments can be used to discriminate these ornaments from notes. We conducted statistical analysis of the duration of notes and single-note ornaments in our recordings. This was performed using the manual onset annotations. The obtained distributions of the durations are depicted in Figure 2 – these indicate that the duration

can indeed provide a good discrimination between notes and ornaments. Based on these results, we consider that a segment is classified as a single-note ornament when its duration is below 90 ms, otherwise it is classified as a note.



**Figure 2.** The distribution of the duration of single-note ornaments (a) and notes (b) obtained using the development set.

The decision whether the detected single-note ornament is a ‘cut’ or ‘strike’ can be made based on comparing the values of the  $F_0^{seg}$  of the current and the following segment. This reflects the musical knowledge of ornamentation. If  $F_0^{seg}$  of the segment detected as ornament is higher than  $F_0^{seg}$  of the following segment, the ornament is classified as ‘cut’ and as ‘strike’ otherwise.

### 3.2.2 Multi-note ornament detection

The detection of multi-note ornaments, namely ‘crann’, ‘roll’ and ‘shake’, is based on analysing the features of a sequence of detected consecutive segments. We used a set of rules to determine whether the sequence corresponds to one of the multi-note ornament types or not. These rules reflect the definition of the multi-note ornaments as presented in Section 2 and for each ornament type are described below. Let us consider that  $r$  denotes the index of the first segment in the sequence of detected segments we are currently analysing. Let us denote by  $\Delta F_0^{seg}(j, j+2)$  the difference between the  $F_0^{seg}$  for the segment  $(r+j)$  and  $F_0^{seg}$  for the segment  $(r+j+2)$ , where  $j$  is an index to be set.

‘Crann’ is detected if the following is fulfilled: i) the sequence of  $F_0^{seg}$  follows the pattern ‘BHBHBHB’, where ‘B’ stands for a base note and ‘H’ for a note higher than the base note; ii) the segmental  $F_0^{seg}$  is similar for segments corresponding to the base note, i.e., the  $\Delta F_0^{seg}(j, j+2)$  is within the given tolerance range  $\beta_{F_0}$  when  $j$  is individually set to 0, 2, and 4; and iii) the segment duration  $D^{seg}$  is below  $\beta_D$  for segments given by setting  $j$  from 1 to 5 and is above  $\beta_D$  for  $j$  set to 0 and 6. The ‘Short-Crann’ is using the same rules but taking into account that the starting base note is omitted.

‘Roll’ is detected if the following is fulfilled: i) the sequence of  $F_0^{seg}$  follows the pattern ‘BHBLB’, where ‘L’ stands for a note lower than the base note; ii) the value of  $\Delta F_0^{seg}(j, j+2)$  is within the tolerance range  $\beta_{F_0}$  for  $j$  set

to 0 and 2; and iii) the segment duration  $D^{seg}$  is above  $\beta_D$  for  $j$  being 0 and 2 and is below  $\beta_D$  when  $j$  is 1 and 3. Again, the ‘Short-Roll’ is using the same rules but taking into account that the starting base note is omitted.

‘Shake’ is detected if the following is fulfilled: i) the sequence of notes follows the pattern ‘HBHB’; ii) the value of  $\Delta F_0^{seg}(j, j+2)$  is within a given tolerance range  $\beta_{F_0}$  for  $j$  set to 1; and iii) the segment duration  $D^{seg}$  is below  $\beta_D$  when  $j$  is 1 and 2, and is above  $\beta_D$  when  $j$  is 3.

The parameters  $\beta_{F_0}$  and  $\beta_D$  were set to 20 Hz (except for ‘crann’ when 30 Hz was used) and 90 ms, respectively.

## 4. EXPERIMENTAL RESULTS

### 4.1 Data description

Evaluations are performed using recordings of Irish traditional tunes and training exercises played by flute from Grey Larsen’s CD which accompanied his book “Essential Guide to Irish Flute and Tin Whistle” [15]. The tunes are between 20 sec and 1 min 11 sec long. All recordings are monophonic and are sampled at 44.1 kHz sampling frequency. Manual annotation of the recordings to indicate the times of onsets and offsets and the identity of notes and ornaments was performed by the third author of this paper, who is a highly experienced musician with over 10 years of flute playing. The manual annotation is used as the ground truth in evaluations. The data was split into separate development and evaluation sets. The development set, consisting of 6 tunes (namely ‘Study5’, ‘Study6’, ‘Study17’, ‘Lady on the Island’, ‘The Lonesome Jig’, ‘The Drunken Landlady’), was used for finding the best parameter values of onset detection methods. The evaluation set, consisting of 13 tunes, was used to obtain the presented results. The list of the tunes from the evaluation set, with the number of notes and ornaments, is given in Table 1. In total, this set contains 3025 onsets, including notes and ornaments. Out of these there are 301 single-note ornaments, consisting of 257 cuts and 44 strikes, and 152 multi-note ornaments, consisting of 117 rolls (including short-rolls), 19 cranns (including short-cranns), and 16 shakes.

### 4.2 Evaluation measures

Performance of the onset and ornament detection is evaluated in terms of the precision ( $P$ ), recall ( $R$ ) and  $F$ -measure. The definition of these measures is the same as used in MIREX onset detection evaluations, specifically,

$$P = \frac{N_{tp}}{N_{tp} + N_{fp}}, R = \frac{N_{tp}}{N_{tp} + N_{fn}}, F = \frac{2PR}{P + R}$$

where  $N_{tp}$  is the number of correctly detected onsets / ornaments and  $N_{fp}$  and  $N_{fn}$  is the number of inserted and deleted onsets / ornaments, respectively. The onset detection is considered as correct when it is within  $\pm 50$  ms around the onset annotation.

The single-note and multi-note ornaments are considered to be detected correctly when the onsets, corresponding to the start and to the end of the ornament are within  $\pm 50$  ms and  $\pm 100$  ms range, respectively.

Tune Title	Number of		Time (sec.)
	Notes	Ornaments (C-S-Ro-Cr-Sh)	
Study 11	76	20-0-0-0-0	26
Study 22	127	0-28-0-0-0	47
Maids of Ardagh	98	23-0-5-0-0	32
Hardiman the ..	112	12-0-7-1-0	28
The Whinny Hills ..	117	15-1-5-2-4	30
The Frost is All ..	151	27-2-12-0-0	41
The Humours of ..	289	59-7-12-14-0	82
The Rose in the ..	152	22-2-11-0-0	39
Scotsman over ..	153	18-0-9-2-0	38
A Fig for a Kiss	105	17-3-6-0-2	28
Roaring Mary	176	15-1-21-0-3	44
The Mountain Road	105	8-0-6-0-3	25
The Shaskeen	181	21-0-23-0-4	42

**Table 1.** The list of tunes contained in the evaluation set, with the number of onsets and ornaments and duration of each tune. The notation ‘C’, ‘S’, ‘Ro’, ‘Cr’ and ‘Sh’ stands for ‘cut’, ‘strike’, ‘roll’, ‘crann’ and ‘shake’, respectively.

### 4.3 Results of onset detection

We have performed extensive evaluations on the development set with different parameter values for each of the onset detection method. The best values of parameters for each of the method are given in Table 2. The achieved performance on the evaluation set using these parameters for each method is presented in Table 3. Note that these results include the onsets corresponding to both notes and ornaments. Performance difference of less than 1% was observed when the parameters were tuned specifically for the evaluation set. It can be seen that all methods provide good onset detection performance, with the  $F_0$ -based method being slightly better than the energy-based methods. A method based on  $F_0$  was shown to perform best for wind instruments also in [11], where it was also shown that its combination with other methods provided only slight improvement at similar  $P$  and  $R$  values. As such, in the following, we use only the  $F_0$ -based method for evaluating the ornament detection performance. An example of a signal extract from one of the tune and the corresponding  $F_0$  estimate and the detection function, with indicated true label and detected onsets, are depicted in Figure 3.

### 4.4 Results of single-note ornament detection

The results of single-note ornament detection are presented in Table 4 separately for ‘cut’ and ‘strike’. The achieved detection performance is significantly higher than that presented in previous flute studies using similar data [9]. The performance for ‘cut’ is close to the overall onset detection performance as presented in Table 3. The performance for ‘strike’ is considerably lower than for ‘cut’. This has also been observed in previous research and may be due to the nature the ‘strike’ is created. There was 5 substitutions of

Onset detection method with best values of the parameters	
sig-energy (spectral):	
–	frame length of 1024 samples (23.2 ms)
–	frame shift of 896 samples (20.3 ms)
–	threshold set as fixed at 2% of the maximum of the normalised detection function
–	minimum distance between peaks set to 10 ms
sig-energy (temporal):	
–	half-Hanning window of 35 ms
–	threshold set as fixed at 15% of the maximum of the normalised detection function
–	minimum distance between peaks set to 20 ms
$F_0$ :	
–	frame length of 1024 samples (23.2 ms)
–	frame shift of 128 samples (2.9 ms)
–	median filter of length 9 frames
–	parameter $\Theta$ set to 6 frames (17 ms)
–	parameter $\alpha_{F_0}$ set to 10 Hz

**Table 2.** Parameters of each onset detection method and their best values obtained based on the development set.

Algorithm	Evaluation performance (%)		
	Precision	Recall	$F$ -measure
sig-energy (spectral)	94.9	85.0	89.7
sig-energy (temporal)	87.9	88.6	88.3
$F_0$	89.1	92.9	91.0

**Table 3.** Results of onset detection obtained by each of the employed method.

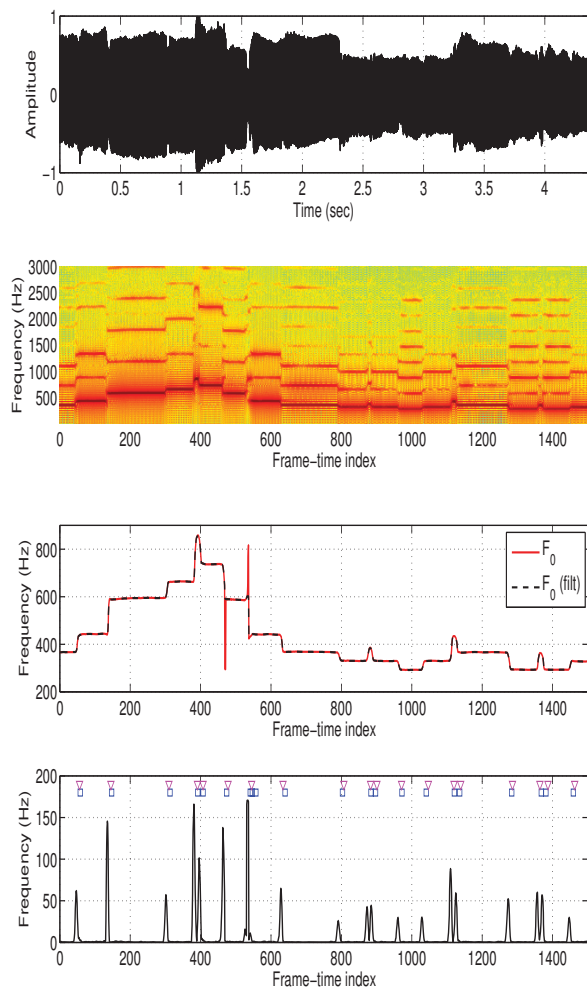
cut for strike and 1 substitution of strike for cut. These errors were contributed by slight inaccuracies in onset detection and  $F_0$  misestimation.

	Single-note Ornament Detection		
	Precision (%)	Recall (%)	$F$ -measure (%)
Cut	88.4	86.4	87.4
Strike	63.8	68.2	65.9

**Table 4.** Results of single-note ornament detection obtained by employing the  $F_0$ -based onset detection method.

#### 4.5 Results of multi-note ornament detection

Experiments for multi-note ornament detection were performed by analysing all the possible sequence patterns resulting from the detected segments – this consisted here of 3020 sequence pattern candidates. The results of multi-note ornament detection are presented in Table 5 separately for ‘roll’, ‘crann’ and ‘shake’. These results include also the short versions for ‘roll’ and ‘crann’. It can be seen that



**Figure 3.** An extract from the tune ‘The Lonesome Jig’, depicting (from top to bottom) the waveform, spectrogram,  $F_0$  estimation (unfiltered (red) and filtered (dashed black)) and the detection function with indicated detected onsets (blue  $\square$ ) and true label (magenta  $\nabla$ ).

the performance for ‘shake’ is considerably lower than that for ‘roll’ and ‘crann’. This is due to the short sequence pattern of ‘shake’, consisting of only 4 parts, which makes it more likely to be accidentally match with other note sequence. We have also analysed the performance separately for the short and normal versions of the ‘roll’ and ‘crann’ ornaments. This showed that the  $F$ -measure performance for ‘roll’ was approximately 17% better than for ‘short-roll’. This trend was not observed for ‘short-crann’, which may be due its longer note sequence.

## 5. CONCLUSION AND FUTURE WORK

In this paper, we presented work on detection of single- and multi-note ornaments in Irish traditional flute music. We employed three different methods for onset detection and customised their parameter values to detecting soft onsets of possibly very short notes. The method based on the fundamental frequency ( $F_0$ ) achieved around 91% onset detection performance in terms of the  $F$ -measure and

Multi-note Ornament Detection			
	Precision (%)	Recall (%)	$F$ -measure (%)
Roll	87.5	67.0	75.9
Crann	86.7	68.4	76.5
Shake	50.0	50.0	50.0

**Table 5.** Results of multi-note ornament detection obtained by employing the  $F_0$ -based onset detection method.

outperformed slightly the other two energy-based methods. The  $F_0$ -based method was then used for evaluating the ornament detection performance. The discrimination between notes and single-note ornaments was based on the duration of segments defined by the adjacent detected onsets. The  $F_0$  information of the current and the following segment was used to distinguish between ‘cut’ and ‘strike’ single-note ornaments. The achieved  $F$ -measure performance for ‘cut’ was over 87%, while for ‘strike’ over 65%. The multi-note ornament detection system was based on analysing the properties of a sequence of detected segments. This included the sequential pattern of segmental  $F_0$ ’s, the duration of each segment, and the relationship of the segmental  $F_0$ ’s among the segments. The average  $F$ -measure performance over all types of multi-note ornaments was over 67%.

There are several points we are currently considering to extend this work. First, we plan to analyse the errors made by each of the onset detection methods and accordingly explore whether their combination could lead to detection performance improvements. This would also include exploration of the use of other onset detection methods, including other  $F_0$  estimation algorithms and possible incorporation of the sinusoidal detection method we presented in [12]. Second, we will explore a compensation for variations in tempo across the recordings. Finally, we plan to employ probabilistic rules for detection of multi-note ornaments which should allow for better handling of the variations due to player’s style.

#### Acknowledgement

This work was supported by a project under the ‘Transforming Musicology’ programme funded by Arts and Humanities Research Council (UK).

#### 6. REFERENCES

- [1] J. P. Bello, L. Daudet, S. Abdallah, Ch. Duxbury, M. Davies, and M. B. Sandler. A tutorial on onset detection in music signals. *IEEE Trans. on Speech and Audio Processing*, pages 1–13, 2005.
- [2] G. Boenn. Automated quantisation and transcription of musical ornaments from audio recordings. In *Proc. of the Int. Computer Music Conf. (ICMC)*, pages 236–239, Copenhagen, Denmark, Aug. 2007.
- [3] B. Breathnach. *Folk music and dances of Ireland*. Osian, London, 1996.
- [4] C. Carson. *Last night’s fun: in and out of time with Irish music*. North Point Press, New York, 1997.
- [5] M. Casey and T. Crawford. Automatic location and measurement of ornaments in audio recording. In *Proc. of the 5th Int. Conf. on Music Information Retrieval (ISMIR)*, pages 311–317, Barcelona, Spain, 2004.
- [6] N. Collins. Using a pitch detector for onset detection. In *Proc. of the 5th Int. Conf. on Music Information Retrieval (ISMIR)*, pages 100–106, Spain, 2005.
- [7] A. de Cheveigne and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111(4):1917–1930, April 2002.
- [8] S. Dixon. Onset detection revisited. In *Proc. of the 9th Int. Conf. on Digital Audio Effects (DAFx)*, pages 133–137, Montreal, Canada, Sep. 2006.
- [9] M. Gainza and E. Coyle. Automating ornamentation transcription. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Honolulu, Hawaii, 2007.
- [10] M. Gainza, E. Coyle, and B. Lawlor. Single-note ornaments transcription for the Irish tin whistle based on onset detection. In *Proc. of the Digital Audio Effects (DAFX)*, Naples, Italy, 2004.
- [11] A. Holzapfel, Y. Stylianou, A. C. Gedik, and B. Bozkurt. Three dimensions of pitched instrument onset detection. *IEEE Trans. on Audio, Speech, and Language Processing*, 18(6):1517–1527, Aug. 2010.
- [12] P. Jančovič and M. Kökür. Detection of sinusoidal signals in noise by probabilistic modelling of the spectral magnitude shape and phase continuity. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 517–520, Prague, Czech Republic, May 2011.
- [13] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3089–3092, March 1999.
- [14] M. Kökür, I. Ali-MacLachlan, P. Jančovič, and C. Athwal. Automated detection of single-note ornaments in Irish traditional flute playing. In *Int. Workshop on Folk Music Analysis*, Istanbul, Turkey, June 2014.
- [15] G. Larsen. *The Essential Guide to Irish Flute and Tin Whistle*. Mel Bay Publications, Pacific, Missouri, USA, 2003.
- [16] M. Puiggros, E. Gómez, R. Ramirez, X. Serra, and R. Bresin. Automatic characterization of ornamentation from bassoon recordings for expressive synthesis. In *9th Int. Conf. on Music Perception and Cognition*, Bologna, Italy, 2006.
- [17] E. D. Scheirer. Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, 103(1):588–601, 1998.

# THE KIKI-BOUBA CHALLENGE: ALGORITHMIC COMPOSITION FOR CONTENT-BASED MIR RESEARCH & DEVELOPMENT

**Bob L. Sturm**

Audio Analysis Lab, Aalborg University, Denmark  
bst@create.aau.dk

**Nick Collins**

Dept. Music, Durham University, UK  
nick.collins@durham.ac.uk

## ABSTRACT

We propose the “Kiki-Bouba Challenge” (KBC) for the research and development of content-based music information retrieval (MIR) systems. This challenge is unencumbered by several problems typically encountered in MIR research: insufficient data, restrictive copyrights, imperfect ground truth, a lack of specific criteria for classes (e.g., genre), a lack of explicit problem definition, and irreproducibility. KBC provides a limitless amount of free data, a perfect ground truth, and well-specifiable and meaningful characteristics defining each class. These ideal conditions are made possible by open source algorithmic composition — a hitherto under-exploited resource for MIR.

## 1. INTRODUCTION

Before attempting to solve a complex problem, one should approach it by first demonstrably solving simpler, well-defined, and more restricted forms, and *only then* increase the complexity. However, there are key problems of research in content-based music information retrieval (MIR) [8] where this has yet to be done. For example, much of the enormous amount of research that attempts to address the problem of music genre recognition (MGR) [26] has started with genre in the “real world” [30]. The same is seen for research in music mood recognition [28, 29, 37], and music autotagging [6]. On top of this, the problem of describing music using genre, mood, or tags in general, has rarely, if ever, been explicitly defined [32].

In lieu of an explicit definition of the problem, the most common approach in much of this research is to implicitly define it via datasets of real music paired with “ground truth.” The problem then becomes reproducing as much of the “ground truth” as possible by pairing feature extraction and machine learning algorithms, and comparing the resulting numbers to those of other systems (including humans). Thousands of numerical results and publications have so far been produced, but it now appears as if most of it has tenuous relevance for *content-based* MIR [3, 27, 30, 31, 34]. The crux of the argument is that the lack of scientific validity in evaluation in much of this

work [3, 27, 30] has led to the development of many MIR systems that appear as if they are “listening” to the music when they are actually just exploiting confounded characteristics in a test dataset [31]. Thus, in order to develop MIR systems that address the goal of “making music, or information about music, easier to find” [8] in the real-world, there is a need to first demonstrably solve simple, well-defined and restricted problems.

Toward this end, this paper presents the “Kiki-Bouba Challenge” (KBC), which is essentially a simplification of the problem of MGR. On a higher level, we propose KBC to refocus the goals in content-based MIR. We devise KBC such that solving it is unencumbered by six significant problems facing content-based MIR research and development: 1) the lack of formal definition of retrieving information in recorded music; 2) the large amount of data necessary to ensure representativeness and generalization for machine learning; 3) the problem of obtaining “ground truth”; 4) the stifling affect of intellectual property (e.g., music copyright) on collecting and sharing recorded music; 5) the lack of validity of standard evaluation approaches of systems; and 6) a lack of reproducible research. KBC employs algorithmic composition to generate a limitless amount of music from two categories, named *Kiki* and *Bouba*. Music from each category are thereby free from copyright, are based in well-defined programs, and have a perfect ground truth. Solving KBC represents a veritable contribution of content-based MIR research and development, and promises avenues for solving parallel problems in less restricted and real-world domains.

Instead of being merely the reproduction of a “ground truth” of some dataset, the MIR “flagship application” of MGR [4] — and that which KBC simplifies — has as its principal goals the *imitation of the human ability to organize, recognize, distinguish between, and imitate genres used by music* [28]. To “imitate the human ability” is not necessarily to replicate the physiological processes humans use to hear, process and describe a piece of music, but merely to describe as humans do a piece of music *according to its content*, e.g., using such musically meaningful attributes as rhythm, instrumentation, harmonic progression, or formal structure. Solving the problem of MGR means creating an artificial system that can work with music like humans, but unencumbered by human limitations.

The concept of genre [12, 13, 16] is notoriously difficult to define such that it can be addressed by algorithms [23]. Researchers building MGR systems have by and large posed



© Bob L. Sturm, Nick Collins.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Bob L. Sturm, Nick Collins. “The Kiki-Bouba Challenge: Algorithmic composition for content-based MIR Research & Development”, 15th International Society for Music Information Retrieval Conference, 2014.

the problem, implicitly or explicitly, from an Aristotelean viewpoint, i.e., “genre” is a categorization of music just as “species” is a categorization of living things, e.g., [5,33].<sup>1</sup> The problem then is to automatically learn the characteristics that place a piece of music on one branch of a taxonomy, distinguish it from a piece of music on a different branch, and avoid contradiction in the process [7]. Researchers have combined signal processing and machine learning with datasets of real music recordings in hopes that the resulting system can discover Aristotelean criteria by which music can be categorized according to genre. The majority of the resulting work, however, documents how much “ground truth” an algorithm replicates in benchmark datasets [26], but rarely illuminates the criteria a system has learned and is using to categorize music [27]. The former quantity is meaningless when the latter is senseless.

In the next section, we discuss the use of algorithmic music composition for data generation. Then we present KBC in its most general form. We follow this with a concrete and specific realisation of KBC, available at the relevant webpage: <http://composerprogrammer.com/kikibouba.html>. We present an unacceptable solution to KBC, and discuss aspects of an acceptable solution. We conclude this paper with a discussion of KBC, and how it relates to content-based MIR in the “real world.”

## 2. ALGORITHMIC MUSIC COMPOSITION FOR GENERATING DATA

Algorithmic composition [1,9,19,21,22,25,36] has a long history back to mainframe computer experiments in the mid 1950s, predating by a decade MIR’s first explicit paper [17]. Ames and Domino [2] differentiate *empirical style modeling* (of historic musical styles) and *active style synthesis* (of novel musical style). In the practical work of this article we concentrate more on the latter, but there is a rich set of techniques for basing generation of music on models trained on existing musical data. Many musical models deployed to capture regularities in data sets are generative, in that a model trained from a corpus can generalise to production of new examples in that style [11].

Though anticipated by some authors, it is surprising how few studies in computer music have utilised algorithmic composition to create the ground truth. Although [24] present a four category taxonomy of algorithmic composition, they do not explicitly discuss the option of using algorithmic composition to produce data sets. The closest category is where “theories of a musical style are implemented as computer programs” [24], essentially empirical style modeling as above.

Sample CD data, especially meta-data on splices, have also rarely been used. But the advantage of algorithmic composition techniques are the sheer volume of data which can potentially be generated, and appropriately handled should be free of the copyright issues that plague databases of music recordings and hinder research access.

We believe that algorithmic generation of datasets within

<sup>1</sup> This of course belies the profound issues that biologists face in recognizing “speciation” events [10].

a framework of open source software has the following potential benefits to MIR and computer music analysis:

- Limitless data set generation, with perfect ground truth (the originating program is fully accessible, and can be devised to log all necessary elements of the ground truth during generation. Random seeds can be used to recover program runs exactly as necessary)
- A fully controlled musical working space, where all assumptions and representational decisions are clear
- Copyright free as long as license free samples or pure synthesis methods are utilised, under appropriate software licensing
- Established data sets can be distributed free of the originating software once accepted by the community, though their origins remain open to investigation by any interested researcher

The greatest issue with dependence on algorithmic generation of music is the ecological validity of the music being generated. A skeptic may question the provenance of the music, especially with respect to the established cultural and economically proven quality of existing human driven recorded music production. Nonetheless, humans are intimately involved in devising algorithmic composition programs. We believe that there is place for expert judgement here, where experts in algorithmic composition can become involved in the process of MIR evaluation. The present paper serves as one humble example; but ultimately, a saving grace of any such position is that the generation code is fully available, and thus accessible to reproduction and evaluation by others.

## 3. THE KIKI-BOUBA CHALLENGE

We now present KBC in its most general form: *develop a system that can organize, recognize, distinguish between, and imitate Aristotelean categories of “music.”* We define these in the subsections below, after we specify the domain.

### 3.1 Domain

The music universe of KBC is populated by “music” belonging to either one of two categories, *Kiki* and *Bouba*.<sup>2</sup> In KBC, music from either category is algorithmically composed such that there is available a limitless number of recordings of music from both categories, and which are entirely unencumbered by copyrights. A music recording from this universe therefore embeds music from *Kiki* and not from *Bouba*, or vice versa, for several reasons that are neither ambiguous nor disputable, and which can be completely garnered from the music recording. The ground truth of a dataset of recordings of music from the music universe then is absolute. Note that a music recording need not be an audio recording, but can be a notated score, or other kind of representation. Now, given that this is ideal

<sup>2</sup> Shapes named “Kiki” and “Bouba” (the two are spiky and rounded, respectively) were originally introduced in gestalt psychology to investigate cross-cultural associations of visual form and language [18,20]. Our example realization of KBC involves two distinctive artificial musical “genres” meant to illustrate in sonic terms a similar opposition.

Attribute	<i>Kiki</i>	<i>Bouba</i>
<b>Form</b>	Alternating accelerando rises and crazy section (“freak out”)	Steady chorale
<b>Rhythm</b>	Accelerando and complex “free” rhythm, fast	Limited set of rhythmic durations, slow
<b>Pitch</b>	Modulo octave tuning system	Recursive subdivision tuning system
<b>Dynamics</b>	Fade ins and outs during accelerando and close of “freak out” sections	Single dynamic
<b>Voicing</b>	All voices in most of the time	Arch form envelope of voice density, starting and ending with single voice
<b>Timbre</b>	Percussive sounds alongside fast attack and decay bright pitched sounds. Second rise has an additional siren sound.	Slow attack and decay sounds with initial portamento and vibrato, with an accompanying dull thud
<b>Harmony</b>	Accidental coincidences only, no overall precepts	System of tonality, with a harmonic sequence built from relatively few possible chords
<b>Texture</b>	More homophonic in accelerando, heterogenous with independent voices in “freak out” sections	Homophonic, homogenous
<b>Expression</b>	Ensemble timing loose on accelerando, independent during “freak out” sections	Details of vibrato, portamento and “nervousness” (chance of sounding on a given chord) differ for each voice in the texture
<b>Space</b>	Little or no reverb	Very reverberant

**Table 1.** Musical attributes of our realization of *Kiki* and *Bouba*.

for toolboxes of algorithms in an Aristotelean world, we pose the following tasks.

### 3.2 The discrimination task (unsupervised learning)

Given an unlabelled collection of music recordings from the music universe, build a system that determines there exist two categories in this music universe, *and* high-level (content) criteria that discriminate them. In machine learning, this can be seen as unsupervised learning, but ensuring discrimination is caused by content and not criteria that are irrelevant to the task.

### 3.3 The identification task (supervised learning)

Given a labelled collection of music recordings from the music universe, build a system that can learn to identify, using high-level (content) criteria, recordings of music (either from this music universe or from others) as being from *Kiki*, *Bouba*, or from neither. In machine learning, this can be seen as supervised learning, but ensuring identification is caused by content and not criteria that are irrelevant to the task.

### 3.4 The recognition task (retrieval)

Given a labelled collection of music recordings from this music universe, build a system that can recognize content in *real world music recordings* as being similar to contents in music from *Kiki*, *Bouba*, both, or neither. In information retrieval, this can be seen as relevance ranking.

### 3.5 The composition task (generation)

Given a labelled collection of music recordings from this music universe, build a system that composes music having content similar to music from *Kiki*, and/or music from *Bouba*. The rules that the system uses to create the music must themselves be meaningful. For example, a music analyst would find the program that generates the music to provide a high-level breakdown of the characteristics of a category. In one sense, this challenge is a necessary precursor to those above, in that a human composer must design the ground truth of the music universe. The production of a dataset of music recordings with algorithmic

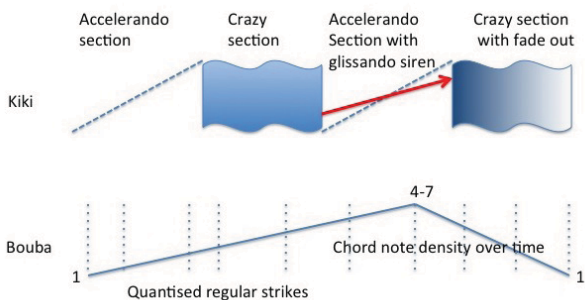
composition necessitates creation in real musical terms. The machine challenge here is to backwards engineer, or to learn in short, the compositional ability to work in the pre-established music universe. However, backwards engineering the compositional mechanisms of such a system, as an expert human musician can potentially do when encountering a musical style unfamiliar to them, is itself an important challenge of high-level musical understanding.

## 4. AN EXAMPLE REALIZATION OF KBC

We now present an example realization of KBC. We specify *Kiki* and *Bouba* via computer programs for algorithmic composition, which we use to create unlimited recordings of music from *Kiki* and *Bouba*, each varying subtly in the fine details (we discuss the practical range of this variation further below). Our computer program is written in the SuperCollider audio programming language [35], with SuperCollider used here in non-realtime mode for fast synthesis of music recordings (which in this case are monophonic digital audio files). We measure the speed of generation of music recordings to be around 60×real-time, so that one piece of around one minute can be created every second by our code. With this we easily created a multi-gigabyte dataset of ten hours, and could very easily create far more.

As *Kiki* and *Bouba* are designed here by humans, they are not independent of “real” music, even though they are fully specified via open source code.<sup>3</sup> Table 3 outlines properties of music from *Kiki* and *Bouba* with respect to some high and low level musical properties. This conveys a sense of why *Kiki* and *Bouba* are well-differentiated in musically meaningful ways. Figure 1 further attempts to illustrate the formal structure of the two styles, again as a demonstration of their distinctiveness. Although the musical description is not as simple as the visual manifestation of the original shapes of “kiki” and “bouba” [18,20], it was designed to avoid too much overlap of musical characteristics. Each output piece is around 40-60 seconds, since

<sup>3</sup> We make available this source code, as well as a few representative sound examples at the accompanying webpage: <http://composerprogrammer.com/kikibouba.html>.



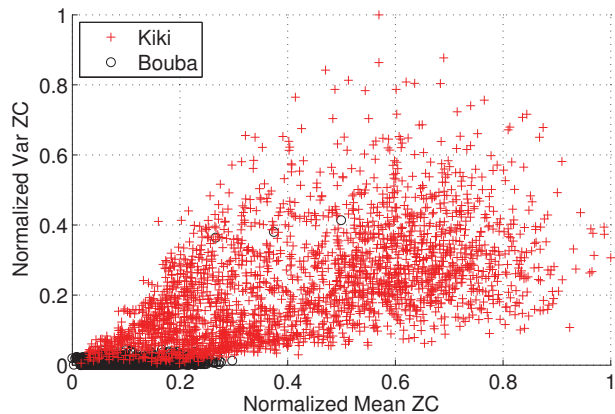
**Figure 1.** Comparative musical forms of our realization of music from *Kiki* and *Bouba* (labeled).

the actual length of sections is itself generative. It is beyond the scope of this article to discuss every detail of the code and the variability of output allowed, but this gives some idea. To anthropomorphise and allow a little literary conceit, our realization envisages music from *Kiki* to be ecstatic, chaotic and ritualistic, characterised by alternating build-ups (accelerando rises) and cathartic “freak-outs.” Our realization envisages music from *Bouba* as an abstract choral funeral march, steady and affected.

#### 4.1 An unacceptable solution

A typical approach to attempt to address an identification task is by computing a variety of low-level and short-time features from music recordings, modelling collections of these by probability distributions (bags of frames), and specifying criteria for classification, such as maximum likelihood. To this end, we use supervised learning to build a single nearest neighbor classifier trained with features computed from a dataset consisting of 250 recordings of music from *Kiki* and 250 from *Bouba*. As features, we first compute the number of zero crossings for 46.3 ms Hann-windowed audio frames, overlapped 50% across the entire recording. We then compute the mean and variance of the number of zero crossings from texture windows of 129 consecutive frames. Finally, we normalize the feature dimensions in the training dataset observations, and use the same normalization parameters to transform input observations. Figure 2 shows a scatter plot of these training dataset observations. To classify an input music recording as being of music from *Kiki* or *Bouba*, we use majority vote from the nearest neighbor classification of the first 10 consecutive texture windows.

We test the system using a stratified test dataset of 500 music recordings from *Kiki* or *Bouba*. For each input, we compare the system output to the ground truth. Our system produces a classification error of 0.00! It has thus successfully labeled all observations in the test dataset with the correct answer. However, this system is not a solution to the identification task of KBC, let alone the three other KBC tasks, *simply because it is not using high-level criteria (content)*. Of course, the statistics of low-level zero crossings across short-time frames has *something* to do with content [15], but this relationship is quite far removed and ambiguous. In other words, people listen to and describe music in terms related to key, tempo and timbre, but not zero crossings. Statistics of zero crossings are



**Figure 2.** Scatter plot of features extracted from recordings of music from *Kiki* and *Bouba*.

not meaningful musical information for solving any task of KBC. That this feature contributes to the perfect figure of merit of this system, it does not illuminate what makes music *Kiki*, and what makes music *Bouba*.

#### 4.2 An acceptable solution

As of the current time, we have yet to find any acceptable solution to our realization of KBC, or any of its tasks — which motivates this challenge. (Furthermore, as discussed below, the goal of KBC is not “a solution” but “solving.”) We can, however, describe aspects of solutions acceptable for our specific realization of KBC. An acceptable solution to the discrimination task determines that in a set of music recordings from the music universe, there exist two different kinds of music, which are discriminable by high-level content, some of which are listed in Table 3, and shown in Fig. 1. An acceptable solution to the identification task determines for any given music recording whether its high-level contents are or are not consistent with all the musical attributes of *Kiki* or *Bouba*. An acceptable solution to the recognition task might recognize as *Bouba* characteristics the slow plodding rhythm, wailing timbre, and homophonic texture of some jazz funeral music. It might recognize as *Kiki* characteristics the glissando siren of some rave music, or the complex, unpredictable and ametrical rhythm of some free improvisation. It would recognize as not characteristic of either *Kiki* or *Bouba* the form of 12-bar blues. Finally, an acceptable solution to the composition task generates music that mimics particular characteristics of music from *Kiki* and *Bouba*.

### 5. DISCUSSION

In essence, KBC is a general exercise, of which we have provided one realization. KBC simplifies MGR — and music description in general — to the degree that many problems typically encountered in MIR research are not an issue, i.e., lack of data, copyright restrictions, cost and inaccuracy of ground truth, poor problem definition, and evaluations that lack validity with respect to meaningful musical understanding by machine. While most research in MGR searches for an Aristotelean categorization of real music (or the reverse engineering of the categorization used to create benchmark music datasets like GTZAN [30, 33]),



it sustains most of the complexity inherent to the problem of MGR. KBC simplifies it to be Aristotelean and well-defined. Essentially, KBC defines categories of music as well-specified and open-source programs, which comports with an Aristotelean conception of music genre. This allows us to benefit from algorithmic composition since we can generate from these programs any quantity of data, free of copyright, and with a perfect ground truth and specified classification criteria.

It can be speculated that KBC is too much of a simplification of MGR, that defining music using programs has little “ecological validity,” and thus that a solution to KBC will be of little use for music in the “real world.” To the first claim, the tasks of KBC are much more complex than reproducing ground truth labels of datasets by any means — the implicit goal of the majority of work addressing MGR [27, 30] — *because solving the tasks requires machine listening*, i.e., “intelligent, automated processing of music” [8]. To the second claim, our realizations of music from *Kiki* and *Bouba* actually originate in higher-level musical processes defined by humans trained and practiced in music composition. Fundamentally, “algorithmic music” and “non-algorithmic music” is a false dichotomy; but this is not to say all algorithms create equally “valid” music. One non-sensical realization of KBC is defining music from *Kiki* and *Bouba* as 50 ms long compositions, each consisting of a single sine, but with frequencies separated by 1 Hz between the two categories. To the final claim, we emphasize an important distinction between “a solution to KBC” and “solving KBC.” We are not claiming that, e.g., a system that has learned to discriminate between music from *Kiki* and *Bouba* will be useful for discriminating between “real” music using any two “real” genres. The system (the actual finished product and black box [29]) will likely be useless. Rather, *solving KBC* is the goal because this requires developing a system that demonstrates a capacity to listen to acoustic signals in ways that consider high level (musical) characteristics.

If one desires more complexity than KBC offers, one can conceive of a music universe with more than two categories, and/or various mixings of “base” categories, e.g., giving rise to cross-genres *Bouki* and *Kiba* (the code at our link already has the capacity to generate these hybrid forms). However, we contend the best strategy is to first demonstrably solve the simplest problems before tackling ones of increased difficulty. If the components of a proposed MGR system result in a system that does not solve KBC, then why should they be expected to result in a system that can discriminate between, or identify, or recognize, or compose music using “real” genres of music from a limited amount of data having a ground truth output by a complex culturally negotiated system that cannot be as unambiguously specified as *Kiki* and *Bouba*?

## 6. CONCLUSION

Simply described, content-based MIR research and development aims to design and deploy artificial systems that are useful for retrieving, using or making music content. The enormous number of published works [6, 14, 26, 38],

not to mention the participation during the past ten years of MIREX,<sup>4</sup> show many researchers are striving to build machine listening systems that imitate the human ability to listen to, search for, and describe music. Examples of such research include music genre recognition, music mood recognition, music retrieval by similarity, cover song identification, and various aspects of music analysis, such as rhythmic and harmonic analysis, melody extraction, and segmentation. These pursuits, however, are hindered by several serious problems: a limited amount of data, the sharing of which is restricted by copyright; the problematic nature of obtaining “ground truth,” and explicitly defining its relationship to music content; and a lack of validity in the evaluation of content-based MIR systems with respect to the task they are supposedly addressing. We are thus left to ask: *Have the simplest problems been demonstrably solved yet?*

In this paper, we show how algorithmic music composition facilitates limitless amounts of data, with perfect ground truth and no restricting copyright, thus holding appreciable potential for MIR research and development. We propose the “Kiki-Bouba Challenge” (KBC) as a simplification of the problem of MGR, and produce an example realization of it facilitated by algorithmic composition. We do not present an acceptable solution to our realization of KBC, but discuss aspects of such a solution. We also illustrate an unacceptable solution, which fails to reveal anything relating to musical meaning even though it still perfectly labels a test dataset. We emphasize, *the goal of KBC is not the system itself, but in solving the challenge*. Solving KBC changes the incentive of research and development in content-based MIR from one of developing systems obtaining high figures of merit by any means, to one of developing systems obtaining high figures of merit by *relevant* means.

## 7. ACKNOWLEDGMENTS

We wish to acknowledge the anonymous reviewers who helped significantly in the revision of this paper. The work of BLS was supported in part by Independent Postdoc Grant 11-105218 from Det Frie Forskningsråd.

## 8. REFERENCES

- [1] C. Ames. Automated composition in retrospect: 1956-1986. *Leonardo*, 20(2):169–185, 1987.
- [2] C. Ames and M. Domino. Cybernetic Composer: an overview. In M. Balaban, K. Ebcioğlu, and O. Laske, editors, *Understanding Music with AI: Perspectives on Music Cognition*, pages 186–205. The AAAI Press/MIT Press, Menlo Park, CA, 1992.
- [3] J.-J. Aucouturier and E. Bigand. Seven problems that keep MIR from attracting the interest of cognition and neuroscience. *J. Intell. Info. Systems*, 41(3):483–497, 2013.

<sup>4</sup><http://www.music-ir.org/mirex>

- [4] J.-J. Aucouturier and E. Pampalk. Introduction – from genres to tags: A little epistemology of music information retrieval research. *J. New Music Research*, 37(2):87–92, 2008.
- [5] J. G. A. Barbedo and A. Lopes. Automatic genre classification of musical signals. *EURASIP J. Adv. Sig. Process.*, 2007:064960, 2007.
- [6] T. Bertin-Mahieux, D. Eck, and M. Mandel. Automatic tagging of audio: The state-of-the-art. In W. Wang, editor, *Machine Audition: Principles, Algorithms and Systems*. IGI Publishing, 2010.
- [7] G. C. Bowker and S. L. Star. *Sorting things out: Classification and its consequences*. The MIT Press, 1999.
- [8] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. *Proc. IEEE*, 96(4):668–696, Apr. 2008.
- [9] D. Cope, editor. *Virtual Music : Computer Synthesis of Musical Style*. MIT Press, Cambridge, MA, 2001.
- [10] D. C. Dennett. *Darwin’s Dangerous Idea: Evolution and the Meanings of Life*. Simon and Schuster, 1996.
- [11] S. Dubnov, G. Assayag, O. Lartillot, and G. Bejerano. Using machine learning methods for musical style modelling. *Computer*, pages 73–80, October 2003.
- [12] F. Fabbri. A theory of musical genres: Two applications. In *Proc. Int. Conf. Popular Music Studies*, 1980.
- [13] J. Frow. *Genre*. Routledge, New York, NY, USA, 2005.
- [14] Z. Fu, G. Lu, K. M. Ting, and D. Zhang. A survey of audio-based music classification and annotation. *IEEE Trans. Multimedia*, 13(2):303–319, Apr. 2011.
- [15] F. Gouyon, F. Pachet, and O. Delrue. On the use of zero-crossing rate for an application of classification of percussive sounds. In *Proc. DAFX*, 2000.
- [16] Scott Johnson. The counterpoint of species. In J. Zorn, editor, *Arcana: Musicians on Music*, pages 18–58. Granary Books, Inc., New York, NY, 2000.
- [17] M. Kassler. Towards musical information retrieval. *Perspectives of New Music*, 4(2):59–67, 1966.
- [18] W. Köhler. *Gestalt Psychology*. Liveright, New York, 1929.
- [19] J. McCormack and M. d’Inverno. *Computers and Creativity*. Springer-Verlag, Berlin Heidelberg, 2012.
- [20] E. Milan, O. Iborra, M. J. de Cordoba, V. Juarez-Ramos, M. A. Rodríguez Artacho, and J. L. Rubio. The kiki-bouba effect a case of personification and ideesthesia. *J. Consciousness Studies*, 20(1-2):1–2, 2013.
- [21] E. R. Miranda, editor. *Readings in Music and Artificial Intelligence*. Harwood Academic Publishers, Amsterdam, 2000.
- [22] G. Nierhaus. *Algorithmic Composition: Paradigms of Automated Music Generation*. Springer-Verlag/Wien, New York, NY, 2009.
- [23] F. Pachet and D. Cazaly. A taxonomy of musical genres. In *Proc. Content-based Multimedia Information Access Conference*, Paris, France, Apr. 2000.
- [24] M. Pearce, D. Meredith, and G. Wiggins. Motivations and methodologies for automation of the compositional process. *Musicae Scientiae*, 6(2):119–147, 2002.
- [25] C. Roads. Research in music and artificial intelligence. *Computing Surveys*, 17(2), June 1985.
- [26] B. L. Sturm. A survey of evaluation in music genre recognition. In *Proc. Adaptive Multimedia Retrieval*, Oct. 2012.
- [27] B. L. Sturm. Classification accuracy is not enough: On the evaluation of music genre recognition systems. *J. Intell. Info. Systems*, 41(3):371–406, 2013.
- [28] B. L. Sturm. Evaluating music emotion recognition: Lessons from music genre recognition? In *Proc. ICME*, 2013.
- [29] B. L. Sturm. Making explicit the formalism underlying evaluation in music information retrieval research: A look at the MIREX automatic mood classification task. In *Post-proc. Computer Music Modeling and Research*, 2014.
- [30] B. L. Sturm. The state of the art ten years after a state of the art: Future research in music information retrieval. *J. New Music Research*, 43(2):147–172, 2014.
- [31] B. L. Sturm. A simple method to determine if a music information retrieval system is a “horse”. *IEEE Trans. Multimedia*, 2014 (in press).
- [32] B. L. Sturm, R. Bardeli, T. Langlois, and V. Emiya. Formalizing the problem of music description. In *Proc. ISMIR*, 2014.
- [33] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Process.*, 10(5):293–302, July 2002.
- [34] J. Urbano, M. Schedl, and X. Serra. Evaluation in music information retrieval. *J. Intell. Info. Systems*, 41(3):345–369, Dec. 2013.
- [35] S. Wilson, D. Cottle, and N. Collins, editors. *The SuperCollider Book*. MIT Press, Cambridge, MA, 2011.
- [36] I. Xenakis. *Formalized Music*. Pendragon Press, Stuyvesant, NY, 1992.
- [37] Y.-H. Yang, D. Bogdanov, P. Herrera, and M. Sordo. Music retagging using label propagation and robust principal component analysis. In *Proc. Int. Conf. Companion on World Wide Web*, pages 869–876, 2012.
- [38] Y.-H. Yang and H. H. Chen. *Music Emotion Recognition*. CRC Press, 2011.



## Poster Session 1

This Page Intentionally Left Blank

# TRANSFER LEARNING BY SUPERVISED PRE-TRAINING FOR AUDIO-BASED MUSIC CLASSIFICATION

Aäron van den Oord, Sander Dieleman, Benjamin Schrauwen  
 Electronics and Information Systems department, Ghent University  
 {aaron.vandenoord, sander.dieleman, benjamin.schrauwen}@ugent.be

## ABSTRACT

Very few large-scale music research datasets are publicly available. There is an increasing need for such datasets, because the shift from physical to digital distribution in the music industry has given the listener access to a large body of music, which needs to be cataloged efficiently and be easily browsable. Additionally, deep learning and feature learning techniques are becoming increasingly popular for music information retrieval applications, and they typically require large amounts of training data to work well. In this paper, we propose to exploit an available large-scale music dataset, the Million Song Dataset (MSD), for classification tasks on other datasets, by reusing models trained on the MSD for feature extraction. This transfer learning approach, which we refer to as *supervised pre-training*, was previously shown to be very effective for computer vision problems. We show that features learned from MSD audio fragments in a supervised manner, using tag labels and user listening data, consistently outperform features learned in an unsupervised manner in this setting, provided that the learned feature extractor is of limited complexity. We evaluate our approach on the GTZAN, 1517-Artists, Unique and Magnatagatune datasets.

## 1. INTRODUCTION

With the exception of the Million Song Dataset (MSD) [3], public large-scale music datasets that are suitable for research are hard to come by. Among other reasons, this is because unwieldy file sizes and copyright regulations complicate the distribution of large collections of music data. This is unfortunate, because some recent developments have created an increased need for such datasets.

On the one hand, content-based music information retrieval (MIR) is finding more applications in the music industry, in a large part due to the shift from physical to digital distribution. Nowadays, online music stores and streaming services make a large body of music readily available to the listener, and content-based MIR can fa-

ilitate cataloging and browsing these music collections, for example by automatically tagging songs with relevant terms, or by creating personalized recommendations for the user. To develop and evaluate such applications, large music datasets are needed.

On the other hand, the recent rise in popularity of feature learning and deep learning techniques in the domains of computer vision, speech recognition and natural language processing has caught the attention of MIR researchers, who have adopted them as well [13]. Large amounts of training data are typically required for a feature learning approach to work well.

Although the initial draw of deep learning was the ability to incorporate large amounts of unlabeled data into the models using an unsupervised learning stage called *unsupervised pre-training* [1], modern industrial applications of deep learning typically rely on purely supervised learning instead. This means that large amounts of labeled data are required, and labels are usually quite costly to obtain.

Given the scarcity of large-scale music datasets, it makes sense to try and leverage whatever data is available, even if it is not immediately usable for the task we are trying to perform. We can use a *transfer learning* approach to achieve this: given a target task to be performed on a small dataset, we can train a model for a different, but related task on another dataset, and then use the learned knowledge to obtain a better model for the target task.

In image classification, impressive results have recently been attained on various datasets by reusing deep convolutional neural networks trained on a large-scale classification problem: ImageNet classification. The ImageNet dataset contains roughly 1.2 million images, divided into 1,000 categories [5]. The trained network can be used to extract features from a new dataset, by computing the activations of the topmost hidden layer and using them as features. Two recently released software packages, *OverFeat* and *DeCAF*, provide the parameters of a number of pre-trained networks, which can be used to extract the corresponding features [7,20]. This approach has been shown to be very competitive for various computer vision tasks, sometimes surpassing the state of the art [18,26].

Inspired by this approach, we propose to train feature extractors on the MSD for two large-scale audio-based song classification tasks, and leverage them to perform other classification tasks on different datasets. We show that this approach to transfer learning, which we will refer to as *supervised pre-training* following Girshick et al. [9],



© Aäron van den Oord, Sander Dieleman, Benjamin Schrauwen.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Aäron van den Oord, Sander Dieleman, Benjamin Schrauwen. "Transfer learning by supervised pre-training for audio-based music classification", 15th International Society for Music Information Retrieval Conference, 2014.

consistently improves results on the tasks we evaluated.

The rest of this paper is structured as follows: in Section 2, we give an overview of the datasets we used for training and evaluation. In Section 3 we describe our proposed approach and briefly discuss how it relates to transfer learning. Our experiments and results are described in Section 4. Finally, we draw conclusions and point out some directions for future work in Section 5.

## 2. DATASETS

The **Million Song Dataset** [3] is a collection of meta-data and audio features for one million songs. Although raw audio data is not provided, we were able to obtain 30 second preview clips for almost all songs from 7digital.com. A number of other datasets that are linked to the MSD are also available. These include the **Taste Profile Subset** [15], which contains listening data from 1 million users for a subset of about 380,000 songs in the form of play counts, and the **last.fm dataset**, which provides tags for about 500,000 songs. We will use the combination of these three datasets to define two *source tasks*: user listening preference prediction and tag prediction from audio.

We will evaluate four *target tasks* on different datasets:

- genre classification on the **GTZAN dataset** [22], which contains 1,000 audio clips, divided into 10 genres.
- genre classification on the **Unique** dataset [21], which contains 3,115 audio clips, divided into 14 genres.
- genre classification on the **1517-artists** dataset [21], which contains 3,180 full songs, divided into 19 genres.
- tag prediction on the **Magnatagatune** dataset [14], which contains 25,863 audio clips, annotated with 188 tags.

## 3. PROPOSED APPROACH

### 3.1 Overview

There are many ways to transfer learned knowledge between tasks. Pan and Yang [17] give a comprehensive overview of the transfer learning framework, and of the relevant literature. In their taxonomy, our proposed supervised pre-training approach is a form of *inductive transfer learning with feature representation transfer*: target labels are available for both the source and target tasks, and the feature representation learned on the source task is reused for the target task.

In the context of MIR, transfer learning has been explored by embedding audio features and labels from various datasets into a shared latent space with linear transformations [10]. The same shared embedding approach has previously been applied to MIR tasks in a multi-task learning setting [24]. We refer to these papers for a discussion of some other work in this area of research.

For supervised pre-training, it is essential to have a source task that requires a very rich feature representation, so as to ensure that the information content of this representation is likely to be useful for other tasks. For computer vision problems, ImageNet classification is one such

task, since it involves a wide range of categories. In this paper, we will evaluate two source tasks using the MSD: tag prediction and user listening preference prediction from audio. The goal of tag prediction is to automatically determine which of a large set of tags are associated with a given song. User listening preference prediction involves predicting whether users have listened to a given song or not.

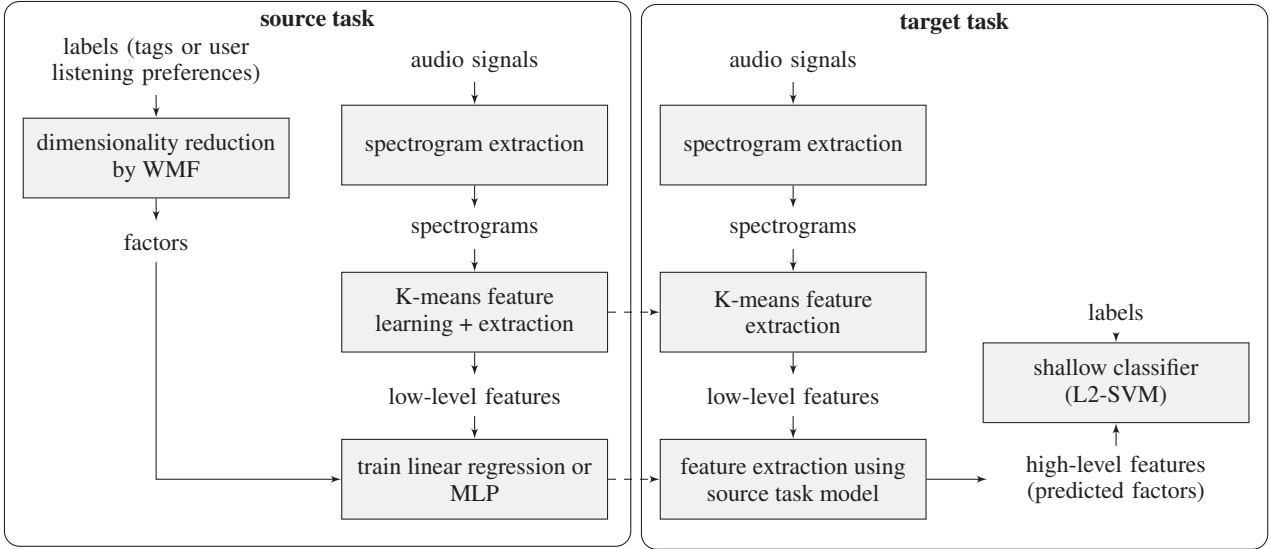
Both tasks differ from typical classification tasks in a number of ways:

- Tag prediction is a *multi-label classification* task: each song can be associated with multiple tags, so the classes are not disjoint. The same goes for user listening preference prediction, where we attempt to predict for each user whether they have listened to a song. The listening preferences of different users are not disjoint either, and one song is typically listened to by multiple users.
- There are large numbers of tags and users; orders of magnitude larger than the 1,000 categories of ImageNet.
- The data is weakly labeled: if a song is not associated with a particular tag, the tag may still be applicable to the song. In the same way, if a user has not listened to a song, they may still enjoy it (i.e. it would be a good recommendation). In other words, some positive labels are missing.
- The labels are redundant: a lot of tags are correlated, or have the same meaning. For example, songs tagged with *disco* are more likely to also be tagged with *80's*. The same goes for users: many of them have similar listening preferences.
- The labels are very sparse: most tags only apply to a small subset of songs, and most users have only listened to a small subset of songs.

We will tackle some of the problems created by these differences by first performing dimensionality reduction in the label space using *weighted matrix factorization* (WMF, see Section 3.2), and then training models to predict the reduced label representations instead.

We will first use the spherical K-means algorithm (see Section 3.3) to learn low-level features from audio spectrograms, and use them as input for the supervised models that we will train to perform the source tasks. Feature learning using K-means is very fast compared to other unsupervised feature learning methods, and yields competitive results. It has recently gained popularity for content-based MIR applications [6, 19, 25].

In summary, our workflow will be as follows: we will first learn low-level features from audio spectrograms, and apply dimensionality reduction to the target labels. We will train supervised models to predict the reduced label representations from the extracted low-level audio features. These models can then be used to perform the source tasks. Next, we will use the trained models to extract higher-level features from other datasets, and use those features to train shallow classifiers for different but related target tasks. We will compare the higher-level features obtained from different model architectures and different source tasks by



**Figure 1:** Schematic overview of the workflow we will use for our supervised pre-training approach. Dashed arrows indicate transfer of the learned feature extractors from the source task to the target task.

evaluating their performance on these target tasks. This workflow is visualized in Figure 1. The key learning steps are detailed in the following subsections.

### 3.2 Dimensionality reduction in the label space

To deal with large numbers of overlapping labels, we first consider the matrix of labels for all examples, and perform weighted matrix factorization (WMF) on it [12]. Given a binary  $m \times n$ -matrix  $A$  ( $m$  examples and  $n$  labels), WMF will find an  $m \times f$ -matrix  $U$  and an  $n \times f$ -matrix  $V$ , so that  $A \approx UV^T$ . The hyperparameter  $f$  controls the rank of the resulting approximation. This approximation is found by optimizing the following weighted objective function:

$$J(U, V) = C \circ (A - UV^T)^2 + \lambda(\|U\|_F^2 + \|V\|_F^2),$$

where  $C$  is a  $m \times n$  confidence matrix,  $\circ$  represents elementwise multiplication, the squaring is elementwise as well, and  $\lambda$  is a regularization parameter. If the confidence values in  $C$  are chosen to be 1 for all zeroes in  $A$ , an efficient alternating least squares (ALS) method exists to optimize  $J(U, V)$ , provided that  $A$  is sparse. For details, we refer to Hu et al. [12].

After optimization, each row of  $U$  can be interpreted as a reduced representation of the  $m$  labels associated with the corresponding example, which captures the latent factors that affect its classification. We can then train a model to predict these  $f$  factors instead, which is much easier than predicting  $m$  labels directly (typically  $f \ll m$ ). We have previously used a similar approach to do content-based music recommendation with a convolutional neural network [23]. In that paper, we showed that these factors capture a lot of relevant information and can also be used for tag prediction. We use the same settings and hyperparameter values for the WMF algorithm in this work.

Our choice for WMF over other dimensionality reduction methods, such as PCA, is motivated by the particular

structure of the label space described earlier. WMF allows for the sparsity and redundancy of the labels to be exploited, and we can take into account that the data is weakly labeled by choosing  $C$  so that positive signals are weighed more than negative signals.

The original label matrix for the tag prediction task has 173,203 columns, since we included all tags from the last.fm dataset that occur more than once. The matrix for the user listening preference prediction task has 1,129,318 columns, corresponding to all users in the Taste Profile Subset. By applying WMF, we obtain reduced representations with 400 factors for both tasks. These factors will be treated as ground truth target values in the supervised learning phase.

### 3.3 Unsupervised learning of low-level features

We learn a low-level feature representation from spectrograms in an unsupervised manner, to use as input for the supervised pre-training stage. First, we extract log-scaled mel-spectrograms from single channel audio signals, with a window size of 1024 samples and a hop size of 512. Conversion to the mel scale reduces the number of frequency components to 128. We then use the spherical K-means algorithm (as suggested by Coates et al. [4]) to learn 2048 bases from randomly sampled PCA-whitened windows of 4 consecutive spectrogram frames. This is similar to the feature learning approach proposed by Dieleman et al. [6].

To extract features, we divide the spectrograms into overlapping windows of 4 frames, and compute the dot product of each base with each PCA-whitened window. We then aggregate the feature values across time by computing the maximal value for each base across groups of consecutive windows corresponding to about 2 seconds of audio. Finally, we take the mean of these values across the entire audio clip to arrive at a 2048-dimensional feature representation for each example. This two-stage temporal pooling approach turns out to work well in practice.

### 3.4 Supervised learning of high-level features

For both source tasks, we train three different model architectures to predict the reduced label representations from the low-level audio features: a linear regression model, a multi-layer perceptron (MLP) with a hidden layer with 1000 rectified linear units (ReLUs) [16], and an MLP with two such hidden layers. The MLPs are trained using stochastic gradient descent (SGD) to minimize the mean squared error (MSE) of the predictions, and dropout regularization [11]. The training procedure was implemented using Theano [2].

We trained all these models on a subset of the MSD, consisting of 373,855 tracks for which we were able to obtain audio samples, and for which listening data is available in the Taste Profile Subset. We used 308,443 tracks for training, 18,684 for validation and 46,728 for testing. For the tag prediction task, the set of tracks was further reduced to 253,588 tracks, including only those for which tag data is available in the last.fm dataset. For this task, we used 209,218 tracks for training, 12,763 for validation and 31,607 for testing.

The trained models can be used to extract high-level features simply by computing predictions for the reduced label representations and using those as features, yielding feature vectors with 400 values. For the MLPs, we can alternatively compute the activations of the topmost hidden layer, yielding feature vectors with 1000 values instead. The latter approach is closer to the original interpretation of supervised pre-training as described in Section 1, but since the trained models attempt to predict latent factor representations, the former approach is viable as well. We will compare both.

To evaluate the models on the source tasks, we compute the predicted factors  $U'$  and obtain predictions for each class by computing  $A' = U'V^T$ . This matrix can then be used to compute performance metrics.

### 3.5 Evaluation of the features for target tasks

To evaluate the high-level features for the target tasks outlined in Section 2, we train linear L2-norm support vector machines (L2-SVMs) for all tasks with liblinear [8], using the features as input. Although using more powerful classifiers could probably improve our results, the use of a shallow, linear classifier helps to assess the quality of the input features.

## 4. EXPERIMENTS AND RESULTS

### 4.1 Source tasks

To assess whether the models trained for the source tasks are able to make sensible predictions, we evaluate them by computing the normalized mean squared error (NMSE)<sup>1</sup> of the latent factor predictions, as well as the area under the ROC curve (AUC) and the mean average precision (mAP)

<sup>1</sup> The NMSE is the MSE divided by the variance of the target values across the dataset.

User listening preference prediction			
Model	NMSE	AUC	mAP
Linear regression	0.986	0.750	0.0076
MLP (1 hidden layer)	0.971	0.760	0.0149
MLP (2 hidden layers)	0.961	0.746	0.0186
Tag prediction			
Model	NMSE	AUC	mAP
Linear regression	0.965	0.823	0.0099
MLP (1 hidden layer)	0.939	0.841	0.0179
MLP (2 hidden layers)	0.924	0.837	0.0179

**Table 1:** Results for the source tasks. For all three models, we report the normalized mean squared error (NMSE) on the validation set, and the area under the ROC curve (AUC) and the mean average precision (mAP) on a separate test set.

of the class predictions<sup>2</sup>. They are reported in Table 1. Note that the latter two metrics are computed on a separate test set, but the former is computed on the validation set that we also used to optimize the hyperparameters for the dimensionality reduction of the labels. This is because the ground truth latent factors, which are necessary to compute the NMSE, are not available for the test set.

It is clear that using a more complex model (i.e. an MLP) results in better predictions of the latent factors in the least-squares sense, as indicated by the lower NMSE values. However, when using the AUC metric, this does not always seem to translate into better performance for the task at hand: MLPs with only a single hidden layer perform best for both tasks in this respect. The mAP metric seems to follow the NMSE on the validation set more closely.

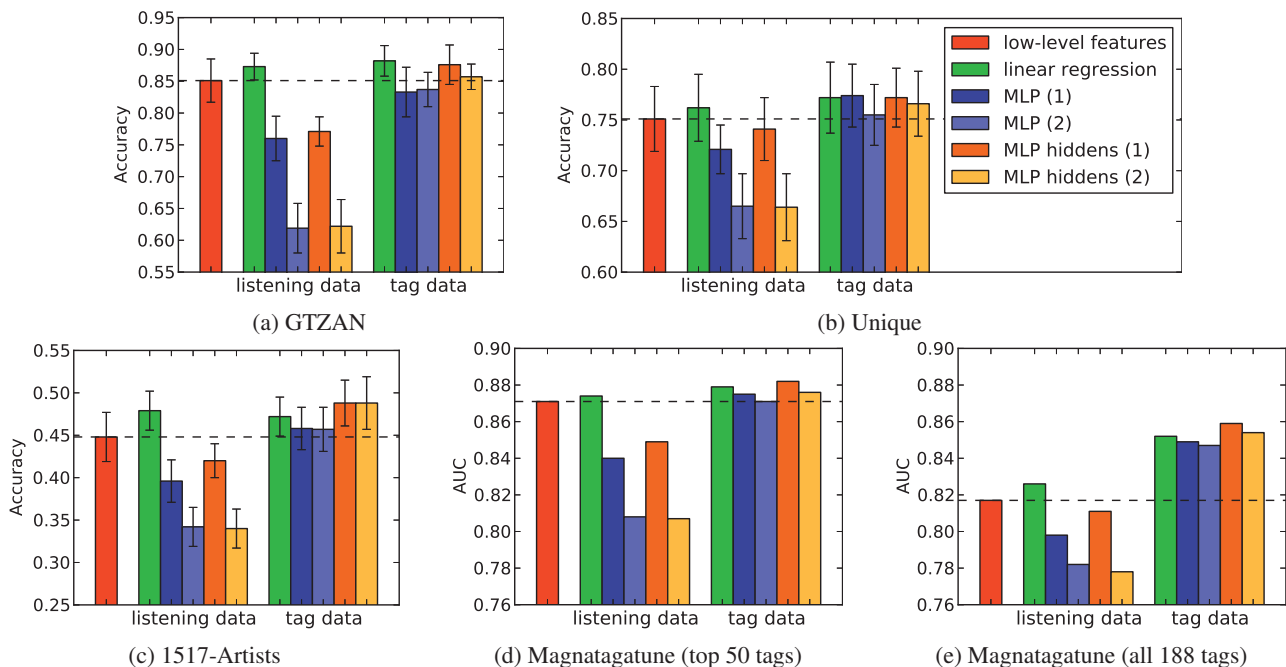
Although the NMSE values are relatively high, the class prediction metrics indicate that the predicted factors still yield acceptable results for the source tasks. In our preliminary experiments we also observed that using fewer factors tends to result in lower NMSE values. In other words, as we add more factors, they become less predictable. This implies that the most important latent factors extracted from the labels are also the most predictable from audio.

### 4.2 Target tasks

We report the L2-SVM classification performance of the different feature sets across all target tasks in Figure 2. For the GTZAN, Unique and 1517-Artists datasets, we report the average cross-validation classification accuracy across 10 folds. Error bars indicate the standard deviations across folds. We optimize the SVM regularization parameter using nested cross-validation with 5 folds. Magnatagatune comes divided into 16 parts; we use the first 11 for training and the next 2 for validation. After hyperparameter optimization, we retrain the SVMs on the first 13 parts, and the last 3 are used for testing. We report the AUC aver-

<sup>2</sup> The class predictions are obtained by multiplying the factor predictions with the matrix  $V^T$ , as explained in the previous section.





**Figure 2:** Target task performance of the different feature sets. The dashed line represents the performance of the low-level features. From left to right, the five bars in the bar groups represent high-level features extracted with linear regression, an MLP with 1 hidden layer, an MLP with 2 hidden layers, the hidden layer of a 1-layer MLP, and the topmost hidden layer of a 2-layer MLP respectively. Error bars for the first three classification tasks indicate the standard deviation across cross-validation folds. For Magnatagatune, no error bars are given because no cross-validation was performed.

aged across tags for the 50 most frequently occurring tags (Figure 2d), and for all 188 tags (Figure 2e).

The single bar on the left of each graph shows the performance achieved when training an L2-SVM directly on the low-level features learned using spherical K-means. The two groups of five bars show the performance of the high-level features trained in a supervised manner for the user listening preference prediction task and the tag prediction task respectively.

Across all tasks, using the high-level features results in improved performance over the low-level features. This effect is especially pronounced for Magnatagatune, when predicting all 188 tags from the high-level features learned on the tag prediction source task. This makes sense, as some of the Magnatagatune tags are quite rare, and features learned on this closely related source task must contain at least some relevant information for these tags.

Comparing the performance of different source task models for user listening preference prediction, model complexity seems to play a big role. Across all datasets, features learned with linear regression perform much better than MLPs, despite the fact that the MLPs perform better for the source task. Clearly the MLPs are able to achieve a better fit for the source task, but in the context of transfer learning, this is actually a form of overfitting, as the features generalize less well to the target tasks – they are too specialized for the source task. This effect is not observed when the source task is tag prediction, because this task is much more closely related to the target tasks. As a result, a better fit for the source task is more likely to result in better generalization across tasks.

For MLPs, there is a limited difference in performance between using the predictions or the topmost hidden layer activations as features. Sometimes the latter approach works a bit better, presumably because the feature vectors are larger (1000 values instead of 400) and sparser.

On GTZAN, we are able to achieve a classification accuracy of  $0.882 \pm 0.024$  using the high-level features obtained from a linear regression model for the tag prediction task, which is competitive with the state of the art. If we use the low-level features directly, we achieve an accuracy of  $0.851 \pm 0.034$ . This is particularly interesting because the L2-SVM classifier is linear, and the features obtained from the linear regression model are essentially linear combinations of the low-level features.

## 5. CONCLUSION AND FUTURE WORK

We have proposed a method to perform supervised feature learning on the Million Song Dataset (MSD), by training models for large-scale tag prediction and user listening preference prediction. We have shown that features learned in this fashion work well for other audio classification tasks on different datasets, consistently outperforming a purely unsupervised feature learning approach.

This transfer learning approach works particularly well when the source task is tag prediction, i.e. when the source task and the target task are closely related. Acceptable results are also obtained when the source task is user listening preference prediction, although it is important to restrict the complexity of the model in this case. Otherwise, the features become too specialized for the source task,

which hampers generalization to other tasks and datasets.

In future work, we would like to investigate whether we can achieve transfer from more complex models trained on the user listening preference prediction task, and other tasks that are less closely related to the target tasks. Since a lot of training data is available for this task, using more powerful models than linear regression to learn features is desirable, especially considering the complexity of models used for supervised pre-training in the computer vision domain. This will require a different regularization strategy that takes into account generalization to other tasks and datasets, and not just to new examples within the same task, as it seems that these two do not always correlate. We will also look into whether using different dimensionality reduction techniques instead of WMF can lead to representations that enable better transfer to new tasks.

## 6. REFERENCES

- [1] Yoshua Bengio. Learning deep architectures for AI. Technical report, Dept. IRO, Université de Montreal, 2007.
- [2] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010.
- [3] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, 2011.
- [4] Adam Coates and Andrew Y. Ng. Learning feature representations with k-means. *Neural Networks: Tricks of the Trade, Reloaded*, 2012.
- [5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.
- [6] Sander Dieleman and Benjamin Schrauwen. Multiscale approaches to music audio feature learning. In *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, 2013.
- [7] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *CoRR*, abs/1310.1531, 2013.
- [8] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.
- [9] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.
- [10] Philippe Hamel, Matthew EP Davies, Kazuyoshi Yoshii, and Masataka Goto. Transfer learning in MIR: sharing learned latent representations for music audio classification and similarity. In *ISMIR 2013*, 2013.
- [11] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. Technical report, University of Toronto, 2012.
- [12] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, 2008.
- [13] Eric J. Humphrey, Juan P. Bello, and Yann LeCun. Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*, 2012.
- [14] Edith Law and Luis von Ahn. Input-agreement: a new mechanism for collecting data using human computation games. In *Proceedings of the 27th international conference on Human factors in computing systems*, 2009.
- [15] Brian McFee, Thierry Bertin-Mahieux, Daniel P.W. Ellis, and Gert R.G. Lanckriet. The million song dataset challenge. In *Proceedings of the 21st international conference companion on World Wide Web*, 2012.
- [16] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010.
- [17] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.
- [18] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. *CoRR*, abs/1403.6382, 2014.
- [19] Jan Schlüter and Christian Osendorfer. Music Similarity Estimation with the Mean-Covariance Restricted Boltzmann Machine. In *Proceedings of the 10th International Conference on Machine Learning and Applications (ICMLA)*, 2011.
- [20] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.
- [21] Klaus Seyerlehner, Gerhard Widmer, and Tim Pohle. Fusing block-level features for music similarity estimation. In *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, pages 225–232, 2010.
- [22] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10:293–302, 2002.
- [23] Aäron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems 26*, 2013.
- [24] Jason Weston, Samy Bengio, and Philippe Hamel. Large-scale music annotation and retrieval: Learning to rank in joint semantic spaces. *Journal of New Music Research*, 2011.
- [25] J. Wülfing and M. Riedmiller. Unsupervised learning of local features for music classification. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, 2012.
- [26] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

# ESTIMATING MUSICAL TIME INFORMATION FROM PERFORMED MIDI FILES

**Harald Grohgan, Michael Clausen**  
 Bonn University  
 {grohgan, clausen}@cs.uni-bonn.de

**Meinard Müller**  
 International Audio Laboratories Erlangen  
 meinard.mueller@audiolabs-erlangen.de

## ABSTRACT

Even though originally developed for exchanging control commands between electronic instruments, MIDI has been used as quasi standard for encoding and storing score-related parameters. MIDI allows for representing musical time information as specified by sheet music as well as physical time information that reflects performance aspects. However, in many of the available MIDI files the musical beat and tempo information is set to a preset value with no relation to the actual music content. In this paper, we introduce a procedure to determine the musical beat grid from a given performed MIDI file. As one main contribution, we show how the global estimate of the time signature can be used to correct local errors in the pulse grid estimation. Different to MIDI quantization, where one tries to map MIDI note onsets onto a given musical pulse grid, our goal is to actually estimate such a grid. In this sense, our procedure can be used in combination with existing MIDI quantization procedures to convert performed MIDI files into semantically enriched score-like MIDI files.

## 1. INTRODUCTION

MIDI (Music Instrument Digital Interface) is used as a standard protocol for controlling and synchronizing electronic instruments and synthesizers [10]. Even though MIDI has not originally been developed to be used as a symbolic music format and imposes many limitations of what can be actually represented [11, 13], the importance of MIDI results from its widespread usage over the last three decades and the abundance of MIDI data freely available on the web. An important feature of the MIDI format is that it can handle musical as well as physical onset times and note durations. In particular, the header of a MIDI file specifies the number of basic time units (referred to as ticks) per quarter note. Physical timing is then given by means of additional tempo messages that determine the number of microseconds per quarter note. On the one hand, disregarding the tempo messages makes it possible



**Figure 1.** The first measure of the prelude BWV 888 by J. S. Bach. (a) Original score. (b) Score from P-MIDI of a performed version without musical pulse grid. (c) Score from S-MIDI based on an estimated musical pulse grid.

to generate a mechanical version of constant tempo, which closely relates to the musical time axis (given in beats) of a score. On the other hand, by including the tempo messages, one may generate a performed version with a physical time axis (given in seconds). However, many of the available MIDI files do not follow this convention. For example, MIDI files are often generated by freely performing a piece of music on a MIDI instrument without explicitly specifying the tempo. As a result, neither the ticks-per-quarter-note parameter nor the tempo messages are set in a musically meaningful way. Instead, these parameters are given by presets, which makes it possible to derive the physical but not the musical time information.

In the following, we distinguish between two types of MIDI files. When the musical beat and tempo messages are set correctly in a MIDI file, then a musical time axis as specified by a score can be derived. In this case, we speak of a *score-informed MIDI file* or simply S-MIDI. When the actual tempo and beat positions are not known (using some presets), we speak of a *performed MIDI file* or simply P-MIDI. This paper deals with the general problem of converting a P-MIDI into a reasonable approximation of an S-MIDI file. The main step is to estimate a musically informed beat or pulse grid from which one can derive the musical time axis. The general problem of estimating beat- and rhythm-related information from music representation (including MIDI and audio representations) is a difficult problem [1, 7]. Typically approaches are based on Hidden



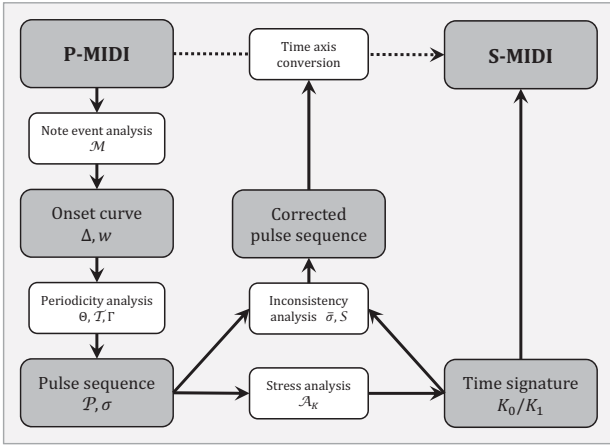


Figure 2. Overview of algorithmic pipeline.

Markov Models [12] and dynamic programming [6, 15]. Even when knowing the note onset positions explicitly (as is the case for MIDI files), finding beats and measure is by far not trivial—in particular when dealing with performed MIDIs having local tempo fluctuations. In [2], an approach based on salience profiles of MIDI notes is used for estimating the time signature and measure positions. Based on a trained dictionary of rhythmical patterns, a more general approach for detecting beat, measure, and rhythmic information is described in [14]. Note that the extraction of such musical time information from MIDI files is required before software for MIDI quantization and score generation can be applied in a meaningful way. This is demonstrated by Figure 1, which shows the original score, the score generated from a P-MIDI, and a score generated from an estimated S-MIDI.

In this paper, we introduce a procedure for estimating the musical beat grid as well as the time signature from a given P-MIDI file, which can then be converted into an approximation of an S-MIDI file.<sup>1</sup> The main idea is to adapt a beat tracking procedure originally developed for audio representations to estimate a first pulse grid. Despite of local errors, this information suffices to derive an estimate of a global time signature. This information, in turn, is then used to correct the local pulse irregularities. In Section 2, we describe the algorithmic details of our proposed method. Then, in Section 3, we evaluate our method and discuss a number of explicit examples to illustrate benefits and limitations. We conclude the paper with Section 4 with possible applications and an outlook on future work. Further related work is discussed in the respective sections.

## 2. ALGORITHMIC PIPELINE

In this section, we describe our procedure for converting P-MIDI files into (approximations of) S-MIDI files by mapping the physical time axis of the P-MIDI to an appropriate musical time axis. As shown in Figure 2, we extract an onset curve from the P-MIDI, and perform periodicity

<sup>1</sup>Our implementation in Java with GUI is available at <http://midi.sechsachtel.de>

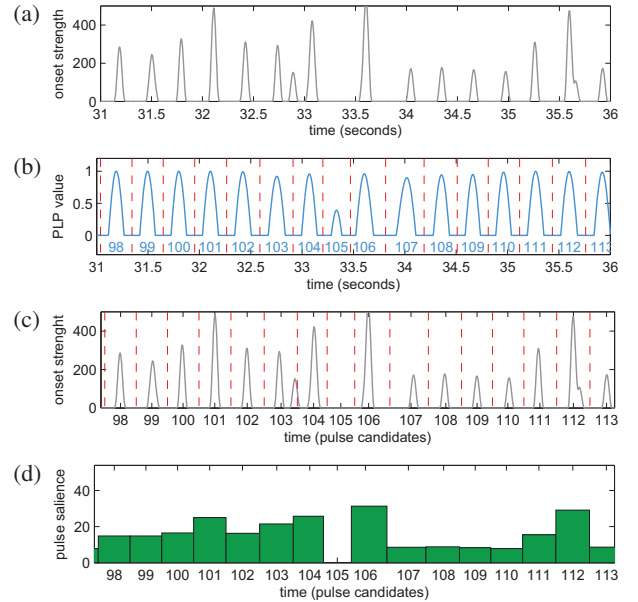


Figure 3. Computation of pulse salience for a 5-second excerpt of BWV 888: (a) MIDI onset curve  $\Delta$ , (b) PLP curve  $\Gamma$  with pulse region boundaries  $b$ , (c) Onset curve  $\Delta$  with boundaries  $b$ , (d) Pulse salience sequence  $\sigma$ .

analysis by adapting a pulse tracking method to obtain a sequence of pulse candidates (Section 2.1). In Section 2.2 we introduce a method to estimate the global time signature by analyzing the stress distribution of the pulses. This information is used for detecting and resolving inconsistencies in the pulse sequence.

In the following we use the notation  $[a : n : b] := ([a] + n\mathbb{N}_0) \cap [a, b]$ , where  $[a, b] := \{t \in \mathbb{R} \mid a \leq t \leq b\}$  for  $a, b \in \mathbb{R}$  and  $n \in \mathbb{N}$ . If  $n = 1$ , we use the notation  $[a : b] := [a : 1 : b]$ .

### 2.1 Pulse Detection

For pulse tracking, we build upon the method introduced by [9] which detects the local predominant periodicity in onset curves, and generates a pulse curve indicating the most likely positions for a pulse-grid. The peaks of this curve are then interpreted as pulse candidates. Although this method was originally developed for audio data like other beat tracking methods (see, e. g., [4,6]), it also works for onset curves derived from MIDI files.

We assume that the MIDI file is already converted to a physical time axis  $[0, T]$ , where  $T$  denotes the end of the last MIDI note, and we have a *MIDI note list* for a suitable finite index set  $I \subset \mathbb{N}$ :

$$\mathcal{M} := (t_i, d_i, p_i, v_i)_{i \in I},$$

where  $t_i \in [0, T)$  describes the start time of the  $i^{\text{th}}$  MIDI note,  $d_i$  its duration (also in seconds),  $p_i \in [0 : 127]$  its pitch, and  $v_i \in [0 : 127]$  its note onset velocity. Based on these notes, we define for a weighting parameter  $w = (w_1, w_2, w_3) \in \mathbb{R}^3$ , a MIDI onset curve

$$\Delta_w(t) := \sum_{i \in I} (w_1 + w_2 \cdot d_i + w_3 \cdot v_i) \cdot h(t - t_i),$$

for  $t \in [0, T]$ , with  $h$  describing a Hann window centered

at 0 of length 50 ms, cf. Figure 3a. Thus the components of the parameter  $w$  corresponds to weights of the presence of an onset, the duration, and the velocity, respectively. In our procedure, we fix  $w := (1, 20, \frac{50}{128})$  to balance the components of each MIDI note, so it will be omitted in the notation. Experiments have shown that the method is robust to slight changes of these values.

Using a short-time Fourier transform, we compute from  $\Delta$  a *tempogram*  $\mathcal{T} : [0, T] \times \Theta \rightarrow \mathbb{C}$  for a given set  $\Theta$  of considered BPM values as explained in [9], using parameters for smoothness (window length) and time granularity (step size). First, we compute a coarse tempogram  $\mathcal{T}^{\text{coarse}}$  using the tempo set  $\Theta = [40 : 4 : 240]$ , window length 8 sec, and step size 1 sec. The dominant global tempo  $T_0$  is derived by summing up the absolute values of  $\mathcal{T}^{\text{coarse}}$  row-wise and by detecting the maximum. Next, we compute a second tempogram  $\mathcal{T}^{\text{fine}}$  based on the new set  $\Theta = \left[ \frac{1}{\sqrt{2}} \cdot T_0, \sqrt{2} \cdot T_0 \right] \cap \mathbb{N}$ , which is the *tempo octave* around  $T_0$ . For this tempogram, the window length is set to  $5 \cdot \frac{60}{T_0}$  sec, and we use a finer step size of 0.2 sec. Choosing the BPM range in such a manner prevents unexpected jumps between multiples of the detected tempo; the window length corresponds to five expected pulses based on the assumption that a stable tempo remains almost constant for at least five beats.

Following [9], we estimate the predominant tempo for each time position from the tempogram  $\mathcal{T}^{\text{fine}}$ , and use this information to derive sinusoidal kernels which best describe local periodicity of the underlying onset curve  $\Delta$ . These kernels are combined to a *predominant local pulse* (PLP) curve  $\Gamma : [0, T] \rightarrow [0, 1]$ , which indicates positions of pulses on the physical time axis, see Figure 3b. The points in time corresponding to the local maxima of  $\Gamma$  form a *pulse candidate* sequence  $\mathcal{P} = (\mathcal{P}_1, \dots, \mathcal{P}_N)$ , which is suitable to estimate the beats in a first approximation. But this sequence may contain additional pulses (not describing a musical beat) or missing pulses. Thus we introduce a post-processing method in the next section which detects and corrects these errors.

## 2.2 Optimizing the pulse sequence

The main idea of the method described in this section relies on finding a global time signature and using it for resolving inconsistencies of the detected pulse sequence. Here, we assume that the measure type of the considered musical piece is not changed throughout the piece. The time signature can be estimated by periodicity analysis of pulse stress using short-time autocorrelation. In a second step, we compare the relative position of each pulse candidate to a measure grid induced by the time signature and detect deviations to correct *isolated* erroneous pulses. Finally, the pulses are interpreted as a new musical time axis, and the tick position of all MIDI events are mapped to this axis.

Now we describe our optimization procedure in more detail. First, we accumulate the onset strength for the  $n^{\text{th}}$  pulse candidate by defining its *pulse salience*

$$\sigma(n) := \int_{b(n-1)}^{b(n)} \Delta(t) dt \quad (n \in [1 : N]), \quad (1)$$

where the pulse region boundaries are given by  $b(n) = \frac{1}{2} \cdot (\mathcal{P}_n + \mathcal{P}_{n+1})$  for  $1 \leq n < N$ ,  $b(0) = 0$ , and  $b(N) = T$ . The boundaries  $b$  are illustrated in Figures 3b and 3c, and for the salience values  $\sigma$  see Figures 3d and 4a.

Our next goal is to compute an estimation of the time signature  $K_0/K_1$ . To this end we perform a salience analysis via autocorrelation. However, to ensure that errors in  $\mathcal{P}$  and  $\sigma$  have only a local influence, we use short-time autocorrelation. For a fixed window size  $K > 12$  ( $K = 32$  in our implementation), we consider the  $K \times N$  matrix

$$\mathcal{A}(k, n) := |I_k|^{-1} \sum_{i \in I_k} \sigma(n+i) \cdot \sigma(n+i+k),$$

where  $I_k := [0 : k : K - k - 1]$  and  $\sigma(n) := 0$  for  $n \in \mathbb{Z} \setminus [1 : N]$ . Thus  $\mathcal{A}(k, n)$  quantifies the plausibility of period length  $k$  around the  $n^{\text{th}}$  pulse candidate. Our predominant salience period  $K_0$ , the nominator of the estimated time signature, is obtained by row-wise summation and maximum picking of parts of  $\mathcal{A}$ :

$$K_0 := \arg \max_{k \in [3:12]} \sum_{n=1}^N \mathcal{A}(k, n).$$

For robustness and musical reasons we have excluded the cases  $k < 3$  and  $k > 12$ , respectively. (Excluding the case  $k = 2$  is not a serious problem as we can use, e. g.,  $4/8$  as surrogate for  $2/4$ .) The relevant rows of the matrix  $\mathcal{A}$  are illustrated in Figure 4b where  $K_0 = 6$ . The denominator  $K_1$  is not necessary for further computation. It is chosen accordingly to the main tempo  $T_0$  to ensure a value between 70 and 140 quarter notes per minute.

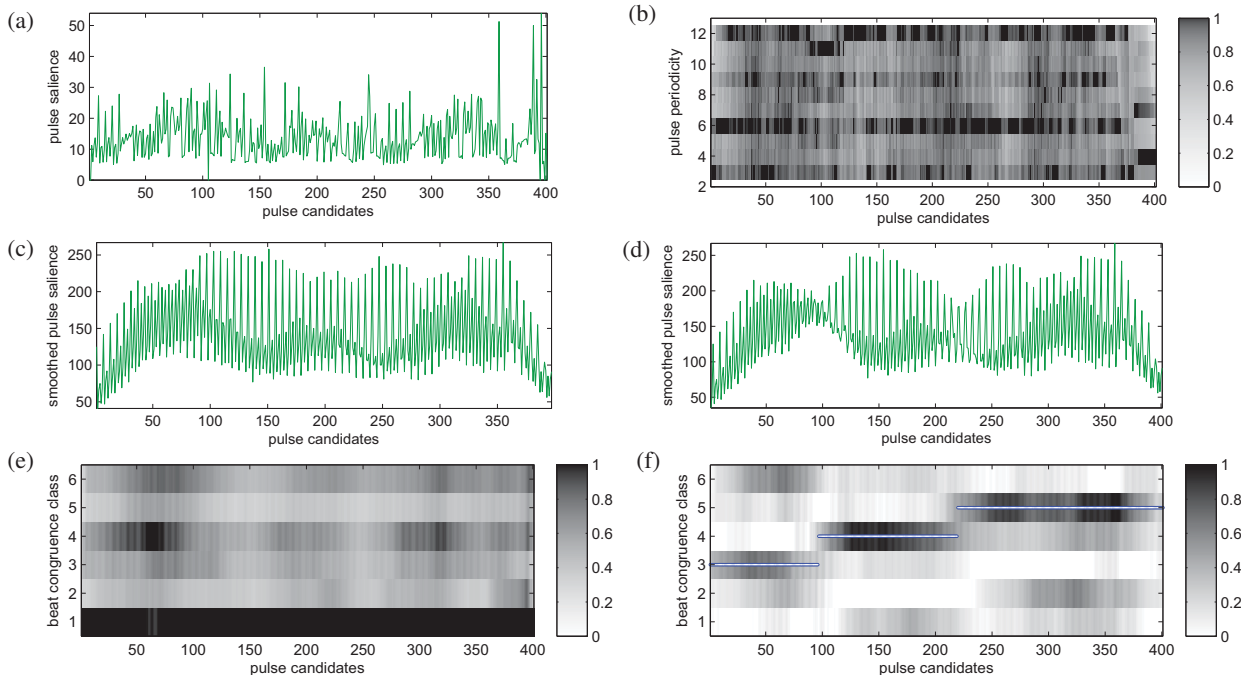
With the help of  $K_0$  we are now able to perform the inconsistency analysis. For now, we primarily consider the case where all detected pulse candidates are actually correct beats. In this idealized scenario, the restriction of  $\mathcal{P}$  to the  $n^{\text{th}}$   $K_0$ -congruence class  $[n : K_0 : N]$ ,  $n \in [1 : K_0]$ , describes the  $n^{\text{th}}$  position within the measures in a semantically meaningful way. In particular, the first class ( $n = 1$ ) corresponds to all downbeat positions if the considered piece does not start with an upbeat. An analogous decomposition applied to  $\sigma$  leads to salience patterns of each position in the measure. Due to rhythmic variations, we expect that the first class of  $\sigma$  *mostly* shows the highest salience value. To enhance robustness,  $\sigma$  is smoothed locally within the  $K_0$ -congruence classes

$$\bar{\sigma}(n) := \sigma(n) + \sum_{k=1}^{\lfloor K/K_0 \rfloor} \sigma(n \pm k \cdot K_0),$$

as illustrated in Figure 4c. Since the restriction to a congruence class is reminiscent of a comb, we call  $\bar{\sigma}$  the  *$K_0$ -combed* version of  $\sigma$ .

Erroneously detected pulse candidates disturb the assignment of all downbeats to a specific class. In such cases, the class containing highest salience values changes at some points of time. To make this visible, a  $K_0 \times N$  matrix  $\mathcal{S}$  is defined which shows the local salience distribution of the congruence classes. More precisely, we define

$$\mathcal{S}(k, n) := \bar{\sigma}(k) \cdot \delta(k \equiv_{K_0} n),$$



**Figure 4.** Step-by-step illustration how to detect inconsistencies in the stress sequence for BWV 888: (a) Pulse salience sequence  $\sigma$  as in Figure 3d. (b) Excerpt of short-time autocorrelation matrix  $\mathcal{A}$  of  $\sigma$  showing maximal energy in 6<sup>th</sup> row. (c) 6-combed salience sequence  $\bar{\sigma}$  if all pulse candidates are correctly detected. (d) 6-combed salience sequence  $\bar{\sigma}$  if two pulses are additionally inserted. (e) Stressgram showing maximal salience in 1<sup>st</sup> congruence class. (f) Stressgram showing two stress changes and path of highest salience.

where  $\delta(A) := 1$  if statement  $A$  holds and 0 else. Smoothing  $S$  along the temporal axis using a Hann window of length  $2 \cdot K_0$  yields a so-called *stressgram*  $\mathcal{S}$ . Such stressgrams are visualized for the ideal scenario (Figure 4e) as well as under presence of two additional pulses (Figure 4f).

Now we discuss this last case in more detail. First, the estimation of  $K_0$  is only locally disturbed which does not lead to a change of the estimated time signature. However, the decomposition into  $K_0$ -congruence classes does no longer coincide semantically with the position in the measures, since all pulses after the additional one are shifted by one beat position. In the stressgram  $\mathcal{S}$  this is indicated by changes of the rows showing high salience. To enhance robustness, we switch to a more global point of view by computing a path of highest energy through  $\mathcal{S}$  using dynamic programming. Each point in this path shows the congruence class with the highest coincidence of representing the downbeats at a specific time. More precisely, if the downbeats are in the class having index  $i$ , then a change to index  $i+1$  near the additional pulse can be noticed, see Figure 4f. The case of a missing pulse is similar, here the row index of the maximal salience changes to  $i-1$ .

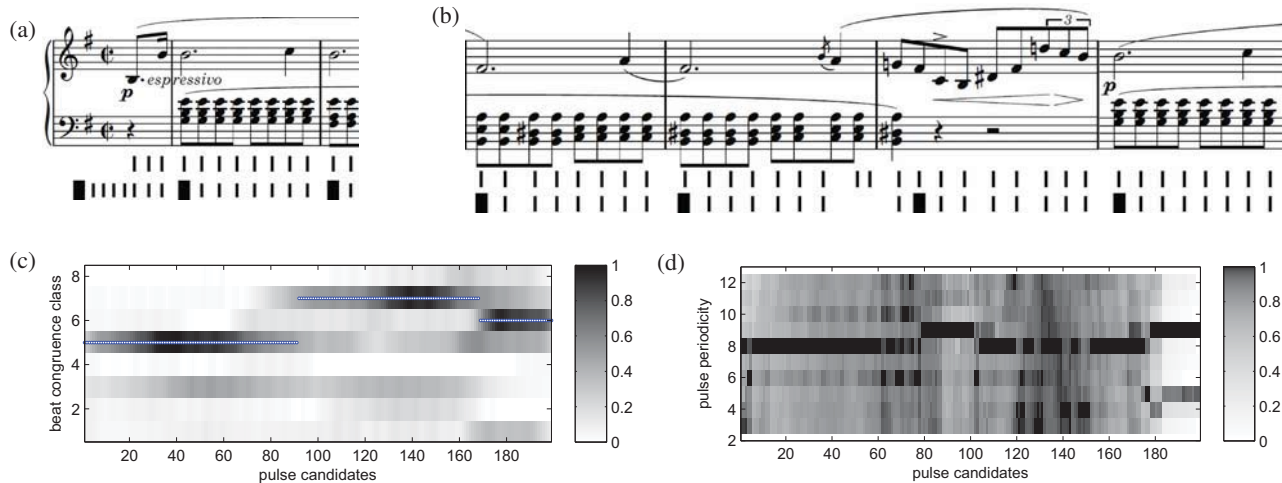
These detected irregularities can now be solved by choosing either falsely added pulse candidates or finding positions to insert an apparently missing pulse. For lack of space, we only sketch our correction procedure. To remove a candidate, one can delete the pulse having lowest salience  $\sigma$  or lowest PLP score (for this, replace  $\Delta$  by  $\Gamma$  in Equation 1). For adding an additional pulse, one may look for two adjacent relatively low values of the PLP curve.

Finally, this corrected pulse sequence defines a beat grid in the P-MIDI file, which allows to detect a sequence of tick positions corresponding to beats. By mapping them to equally distributed new tick positions, adding appropriate tempo change MIDI messages, and performing linear interpolation between the beat positions, the previous time axis of the P-MIDI is replaced by a musical time axis. In case of an upbeat, additional beats are added to the beginning of the piece such that the first  $K_0$ -congruence class corresponds to the estimated downbeats. Lastly, the time signature  $K_0/K_1$  is added to the new MIDI file.

### 3. EXPERIMENTS AND DISCUSSION

Evaluating the output of a beat tracking procedure is a non-trivial task due to the vague definition of beat times as described in [5]. Particularly determining the beat granularity, i. e., the decision between similar time signatures like 6/8 and 3/4, or multiples such as 4/4 or 8/4, appears as an ill-posed and negligible problem. Even for humans, beat and measure tracking can be challenging especially in the presence of rhythmic variations and expressive timing. Our evaluation is inspired by [14], where among others the visual impression of the computed score is considered, and by [5], where comparison to a ground truth annotation by a human and listening tests for a perceptual evaluation are suggested.

Because of its modeling, our procedure is not suitable for all kinds of P-MIDI files. The PLP approach described in Section 2.1 has some constraints like a stable rhythm or an almost stable tempo for a certain amount of time (in our



**Figure 5.** Prelude No. 4 from Chopin (Op. 28). The score excerpts show the detected pulses (upper row) and post-processed measure grid (lower row). Downbeats are indicated by bold lines. **(a)** Correctly detected upbeat. **(b)** Joint correction of two subsequent errors. **(c)** Stressgram with path of highest salience. **(d)** Short-time autocorrelation matrix.

implementation, this time window is roughly five seconds). In addition, tempo octave confusions are not considered here. For the optimization step described in Section 2.2, a global time signature is required. Furthermore, downbeats must be detectable by their length or stress. In the following, we discuss in detail two typical examples of P-MIDI files, and then perform an automatic analysis on a small test set of artificially distorted MIDI files.

### 3.1 Qualitative Evaluation

The performance of the proposed procedure for Bach's Prelude BWV 888 is already indicated in Figures 1, 3, and 4. Two insertions of additional pulses caused by ritardandi are corrected well, and also the erroneous rests at the beginning are eliminated. The estimated time signature 6/8 is perceptually similar to the notated time signature 12/8.

Our second example is the Prelude No. 4 from Chopin's romantic Piano Preludes. Figure 5 shows two score excerpts together with the detected pulse candidates and the estimated measure structure as well as the corresponding stressgram and the short-time autocorrelation matrix for the whole piece.<sup>2</sup> Prelude No. 4 contains some long notes at downbeat positions leading to a good measure tracking result. Because of their strong presence, an eighth note pulse grid was detected, see Figure 5d. This piece of music starts with an upbeat of a quarter note. Since the MIDI format does not support upbeat information directly, our method adds enough additional pulses such that the first pulse lies in the congruence class of the downbeats as shown in Figure 5a.

Noticeable tempo changes together with short appoggiatura and a triplet around pulse No. 90 causes the PLP procedure to detect two additional pulses in consecutive measures (Fig. 5b). In the stressgram this is indicated by

<sup>2</sup>The examples are recordings on a MIDI piano taken from Saarland Music Data (<http://www.mpi-inf.mpg.de/resources/SMD/>), and the scores are picked from Mutopia (<http://www.mutopiaproject.org/>).

a jump of the salience path across two classes (Fig. 5c). Note that this error has only little influence on the estimation of the time signature (Fig. 5d). As indicated by the stressgram, two pulses are removed in this region during our post-processing. Although the deletion of a correctly detected pulse in the 2<sup>nd</sup> measure of Figure 5b leads to a wrong downbeat position in the subsequent measure, the global measure grid is restored in the 4<sup>th</sup> measure. This shows how our method optimizes the measure grid without having to correct each single pulse error.

### 3.2 Automatic Evaluation

Furthermore, we evaluated our method on score-like MIDI files which have been automatically disturbed by adding additional tempo change events. A similar approach was used in [8] to show that smooth tempo changes are detected well by the PLP method.

In particular, the goal of our procedure is to recognize measure positions correctly, therefore we use standard precision (P), recall (R), and F-measure (F) on the set of the MIDI notes. A note is considered relevant if it starts at a downbeat position in the S-MIDI file, and it is "retrieved" if it was mapped by our method to a downbeat position of the distorted MIDI file. Since no quantization step is included, we allow a tolerance of  $\pm 5\%$  of each measure as its downbeat position.

By neglecting the musical time axis and using only the physical time position (in milliseconds) of all MIDI events, we simulate a performed MIDI of an S-MIDI file. The systematic distortion is done by adding a tempo change of  $\pm 20\%$  around the normal tempo each 10 seconds.

As test set, we consider the Fifteen Fugues by Beethoven from IMSLP<sup>3</sup>. Here, the note durations are sufficient for a good estimation of the PLP pulses. Adding note velocity information from real performed MIDI files suggests an further improvement of the results.

<sup>3</sup>Petrucci Music Library, [http://imslp.org/wiki/15\\_Fugues\\_\(Beethoven,\\_Ludwig\\_van\)](http://imslp.org/wiki/15_Fugues_(Beethoven,_Ludwig_van))

Piece	Full method			PLP only			# Corrections		
	F	P	R	F	P	R	add	delete	upbeat
No. 1	0.477	0.587	0.402	0.372	0.509	0.293	0	2	0
No. 2	0.978	1	0.956	0.397	0.56	0.308	1	0	1
No. 3	0.656	0.663	0.649	0.144	0.196	0.113	1	0	0
No. 4	0.945	0.984	0.909	0.738	0.804	0.682	2	0	0
No. 5	0.966	0.971	0.962	0	0	0	0	0	2
No. 6	0.996	1	0.993	0.996	1	0.993	0	0	0
No. 7	0.826	0.832	0.82	0.324	0.386	0.28	1	4	1
No. 8	0.953	0.985	0.923	0.821	0.945	0.725	0	1	0
No. 9	0.896	0.916	0.876	0.787	0.855	0.73	1	0	1
No. 10	0.581	0.579	0.582	0.008	0.013	0.005	2	2	1
No. 11	1	1	1	1	1	1	0	0	0
No. 12	0.994	1	0.988	0.792	0.842	0.748	3	1	0
No. 13	0.656	0.884	0.522	0.245	0.393	0.178	0	2	2
No. 14	0.975	0.995	0.957	0.432	0.75	0.303	1	3	0
No. 15	0.692	0.98	0.535	0.633	0.992	0.465	0	0	4
Mean	0.839	0.892	0.805	0.513	0.616	0.455	0.8	1	0.8

**Table 1.** Evaluation results for 15 Fugues from Beethoven for full method and PLP-based beat tracking only

The results for the automatic evaluation are shown in Table 1. We evaluated both the original pulse sequence derived from the PLP pulse tracking method introduced in Section 2.1, and the post-processed version. In both cases, the detected time signature was used to locate the downbeats. For all pieces except No. 11, the annotated 2/2 signature was mostly detected as 4/4, sometimes as 8/4. Compared to the results of the PLP pulse tracker, which is not designed for detecting downbeats, the results for some pieces were improved significantly by our method. For example, in Fugue No. 7 our post-processing method added one pulse and removed four other pulses. At the beginning, another single pulse was inserted to prevent upbeat shifts. These changes lead to an increase of the F-measure from 0.324 to 0.826, which has major consequences, e. g., on the amount of additional work for a human importing this MIDI file into a score notation software to optimize the score manually.

#### 4. CONCLUSION

We presented a bottom-up method to derive a musically meaningful time axis for performed MIDI files, and converting them into semantically enriched score-like MIDI files. Our proposed procedure optimized an estimated pulse sequence by insertion of missing pulses as well as removal of spurious pulses to derive an overall consistent measure grid.

Since the output of the presented method is another MIDI file, our procedure can be used in combination with any MIDI quantization software by using it for preprocessing performed MIDI files having no musically meaningful time information. Essentially, the physical time axis remains unchanged, so it can be used further in combination with rhythm transcription approaches. Deriving a musical time axis without quantization is also meaningful for real-time interaction with MIDI synthesizers, e. g., as a variation of [3]. Because of its generality, our procedure can be simply extended by including other rhythmic or harmonic aspects.

**Acknowledgments:** This work has been supported by the German Research Foundation (DFG CL 64/8-1,

DFG MU 2686/5-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer IIS.

#### 5. REFERENCES

- [1] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013.
- [2] E. Cambouropoulos. From MIDI to traditional musical notation. In *Proc. of the AAAI Workshop on Artificial Intelligence and Music*, vol. 30, 2000.
- [3] R. B. Dannenberg and C. Raphael. Music score alignment and computer accompaniment. *Communications of the ACM, Special Issue: Music information retrieval*, 49(8):38–43, 2006.
- [4] M. E. P. Davies and M. D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transact. on Audio, Speech and Language Processing*, 15(3):1009–1020, 2007.
- [5] S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30:39–58, 2001.
- [6] D. P. W. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.
- [7] F. Gouyon and S. Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 29:34–54, 2005.
- [8] P. Grosche and M. Müller. A mid-level representation for capturing dominant tempo and pulse information in music recordings. In *Proc. of the Intern. Conf. on Music Information Retrieval (ISMIR)*, pp. 189–194, 2009.
- [9] P. Grosche and M. Müller. Extracting predominant local pulse information from music recordings. *IEEE Transact. on Audio, Speech, and Language Processing*, 19(6):1688–1701, 2011.
- [10] D. M. Huber. *The MIDI manual*. Focal Press, 3rd edition, 2006.
- [11] F. R. Moore. The dysfunctions of MIDI. *Computer Music Journal*, 12(1):19–28, 1988.
- [12] C. Raphael. Automated rhythm transcription. In *Proc. of the Intern. Conf. on Music Information Retrieval (ISMIR)*, 2001.
- [13] E. Selfridge-Field, editor. *Beyond MIDI: the handbook of musical codes*. MIT Press, Cambridge, MA, USA, 1997.
- [14] H. Takeda, T. Nishimoto, and S. Sagayama. Rhythm and tempo analysis toward automatic music transcription. In *IEEE Intern. Conf. on Acoustics, Speech and Signal Processing*, vol. 4, pp. IV–1317, 2007.
- [15] A. C. Yang, E. Chew, and A. Volk. A dynamic programming approach to adaptive tatum assignment for rhythm transcription. In *IEEE Intern. Symposium on Multimedia*, 2005.



# ESTIMATION OF THE DIRECTION OF STROKES AND ARPEGGIOS

Isabel Barbancho<sup>1</sup>, George Tzanetakis<sup>2</sup>, Lorenzo J. Tardón<sup>1</sup>, Peter F. Driessen<sup>2</sup>, Ana M. Barbancho<sup>1</sup>

<sup>1</sup>Universidad de Málaga, ATIC Research Group, ETSI Telecomunicación,  
Dpt. Ingeniería de Comunicaciones, 29071 Málaga, Spain

<sup>2</sup> University of Victoria, Department of Computer Science, Victoria, Canada  
ibp@ic.uma.es, gtzan@cs.uvic.ca, lorenzo@ic.uma.es,  
peter@ece.uvic.ca, abp@ic.uma.es

## ABSTRACT

Whenever a chord is played in a musical instrument, the notes are not commonly played at the same time. Actually, in some instruments, it is impossible to trigger multiple notes simultaneously. In others, the player can consciously select the order of the sequence of notes to play to create a chord. In either case, the notes in the chord can be played very fast, and they can be played from the lowest to the highest pitch note (upstroke) or from the highest to the lowest pitch note (downstroke).

In this paper, we describe a system to automatically estimate the direction of strokes and arpeggios from audio recordings. The proposed system is based on the analysis of the spectrogram to identify meaningful changes. In addition to the estimation of the up or down stroke direction, the proposed method provides information about the number of notes that constitute the chord, as well as the chord playing speed. The system has been tested with four different instruments: guitar, piano, autoharp and organ.

## 1. INTRODUCTION

The design and development of music transcription systems has been an open research topic since the first attempts made by Moorer in 1977 [15]. Since then, many authors have worked in different aspects of the transcription problem [12], [17]. A common task in this context is automatic chord transcription [13], [1], [3], [7], [14], but also, other aspects beyond the mere detection of the notes played are nowadays considered, shifting the focus of the research to different pieces of information related to the way in which these notes are played, i.e. musical expressiveness [18], [4], [7], [11].

A chord can be defined as a specific set of notes that sound at the same time. Often, when a chord is played, not all the notes in the chord start at the same time. Because

of the mechanics of actuation of some instruments like the guitar, the mandolin, and the autoharp [20], it is hard to excite different strings at the same time. Instead the performer typically actuates them sequentially in a stroke. A stroke is a single motion across the strings of the instrument. The stroke can have two different directions: UP, when the hand moves from the lowest to the highest note, and DOWN, when the hand moves from the highest to the lowest note. A strum is made up of several strokes combined in a rhythmic pattern. In other instruments like the piano or the organ, all the notes that belong to a certain chord can be played at the same time. However, the musician can still choose to play the chord in arpeggio mode, i.e., one note after another. Again, the arpeggio direction can be up or down.

In this paper, we propose a new chord related analysis task focused on the identification of the stroke or arpeggio direction (up or down) in chords. Because the movement can be fast it is not feasible to look for onsets [6] to detect each note individually. Therefore, a different approach will be proposed. In addition to the detection of the stroke direction, our proposed method also detects the speed with which the chord has been played as well as the number of notes. The estimation of the number of notes played in a chord is a problem that has not been typically addressed, although some references can be found related to the estimation of the number of instruments in polyphonic music [16], which constitutes a related but different problem. Regarding the chord playing speed, to the best of our knowledge there are no published works to identify this parameter except when specific hardware is used for the task [19], [9]. The paper is organized as follows: in Section 2, the proposed system model is explained. Section 3 presents some experimental results and Section 4 draws some conclusions.

## 2. STROKE AND ARPEGGIO ANALYSIS

The main goal of this work is the analysis of audio excerpts to detect if a chord has been played from lower to higher notes (UP) or vice versa (DOWN). The movement to play a chord may be quite fast and all the information about the movement is contained at the very beginning of the chord waveform. After all the strings of the chord have been played, it is no longer possible to know whether the

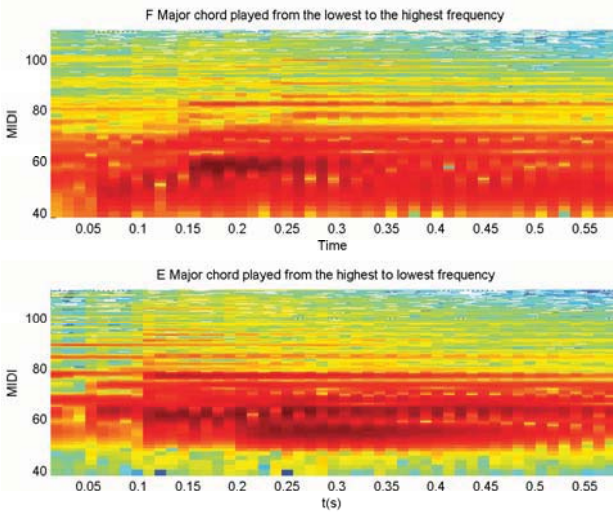


© Isabel Barbancho<sup>1</sup>, George Tzanetakis<sup>2</sup>, Lorenzo J. Tardón<sup>1</sup>, Peter F. Driessen<sup>2</sup>, Ana M. Barbancho<sup>1</sup>.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Isabel Barbancho<sup>1</sup>, George Tzanetakis<sup>2</sup>, Lorenzo J. Tardón<sup>1</sup>, Peter F. Driessen<sup>2</sup>, Ana M. Barbancho<sup>1</sup>. "ESTIMATION OF THE DIRECTION OF STROKES AND ARPEGGIOS", 15th International Society for Music Information Retrieval Conference, 2014.

movement was up or down because the resulting sound contains all the component pitches. This means that any feature that may provide information about how the spectrum varies when the chord is being played has to be calculated at the very beginning of the chord. We will consider that the time needed to complete a stroke varies from 1 s (relatively slow) to less than 0.2 s, when the chord is played fast.

Let  $x$  denote the samples of the played chord under study. In order to calculate a spectrogram, the samples  $x$  are divided into segments  $x_m = [x_m[1], \dots, x_m[M]]^T$ , where  $M$  is the selected window size for the spectrogram calculation. Let  $PSD_m$  denote the Power Spectral Density of each segment  $x_m$  and  $L_m$  the logarithm of the  $PSD_m$  i.e  $L_m = 10 \log_{10}(PSD_m)$ . In Fig. 1, the log spectrogram of an ‘F Major’ guitar chord played from the lowest to the highest string is shown (up stroke). The exact fret position employed to play this chord is  $frets = [2, 2, 3, 4, 4, 2]$  where the vector  $frets$  represents the frets pressed to play the chord from string 1 (highest string) to string 6 (lowest string). This chord has been generated synthetically to control the exact delay between each note in the chord (in this case the delay is  $\tau = 4ms$ ). The guitar samples have been extracted from the RWC database [10]. As it can be observed in Fig. 1, the information about the stroke direction is not directly visible in the spectrogram. Therefore, in order to detect the stroke direction, the spectrogram needs to be further analysed.



**Figure 1.** Spectrogram of an F Major chord UP stroke in a classic guitar (upper figure) and an E Major chord DOWN stroke. Audio file is sampled with  $f_s = 44100$  Hz. For the spectrogram, the selected parameters are window size  $M = 1024$ ,  $overlapp = 512$  with a Hamming window. The DFT size is  $K = 4096$ . For convenience, the MIDI numbers are shown in the y-axis instead of the frequency bins:  $MIDI = 69 + 12 \log_2(f/440)$ .

## 2.1 Detection of new spectral components

Whenever a new note is played, it is expected that new spectral components corresponding to the new note will be added to the existing components of the previous note (if any). In auditory scene analysis [8] this is termed the ‘old+new heuristic’. The main idea is to take advantage of this heuristic by detecting whether the current spectrum contains new components or, conversely, whether it simply retains the components from the previous spectrum. As we are frequently dealing with sounds that decay quickly our model of sustained notes will also contain a decay component. In order to detect ‘old+new’ changes we minimize the following equation:

$$\epsilon[m] = \min_{\alpha[m]} \left[ \sum_{k=1}^K |L_m[k] - \alpha[m]L_{m-1}[k]| \right] \quad (1)$$

The goal is to find a local  $\alpha[m]$  (decay factor) that minimizes  $\epsilon[m]$  for two consecutive windows  $m$  and  $m-1$ . The minimization is carried out by means of the unconstrained nonlinear minimization Nelder-Mead method [21]. The idea is to remove from window  $m$  all the spectral components that were also present in window  $m-1$  with a gain adjustment so that any new spectral component becomes more clearly visible. Thus, if there are no new played notes in window  $m$  with respect to window  $m-1$ ,  $\epsilon[m]$  will be small, otherwise  $\epsilon[m]$  will become larger because of the presence of the new note.

In Fig. 2 (a) and (b), the normalized evolutions of  $\alpha[m]$  and  $\epsilon[m]$  respectively are displayed for the F Major UP chord shown in Fig.1 (a). The vertical lines represent the instants when new notes appear in the chord. When a new note is played in the chord,  $\alpha[m]$  attains a minimum and  $\epsilon[m]$  a maximum. In order to automatically detect the instants when the new notes appear, the following variables are defined:

$$\epsilon'[m] = \begin{cases} \epsilon[m] - \epsilon[m-1] & \text{if } \epsilon[m] - \epsilon[m-1] > 0.5 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

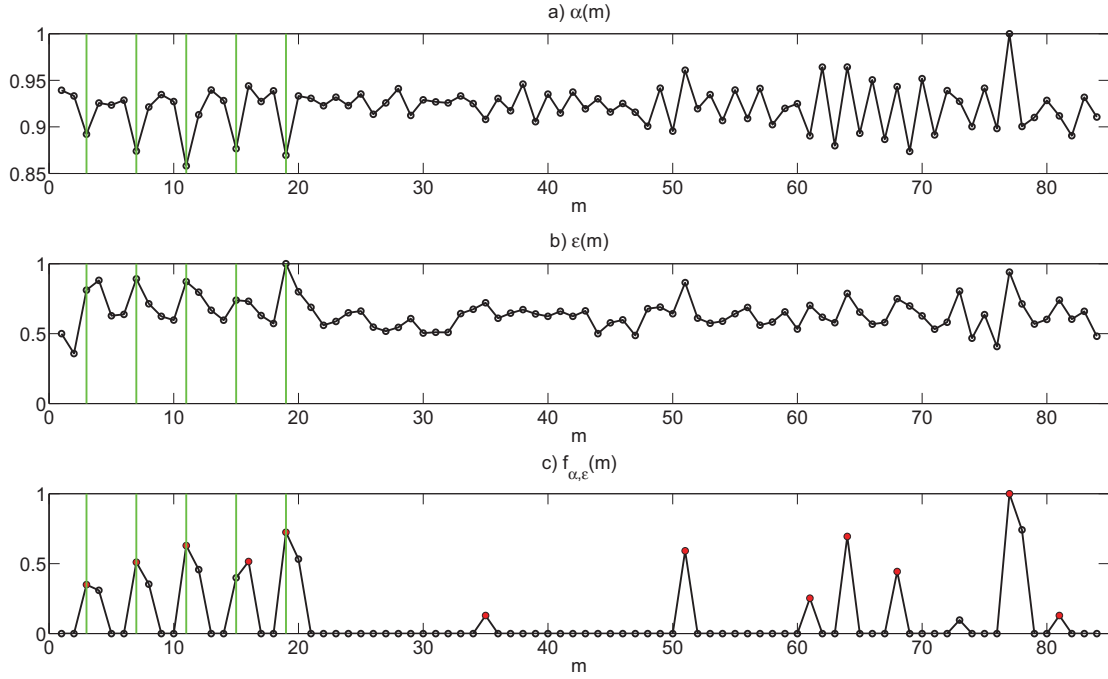
$$\alpha'[m] = \alpha[m] - \alpha[m-1] \quad (3)$$

$$f_{\alpha, \epsilon}[m] = \frac{\epsilon'[m]}{\max(\epsilon')} \cdot \left| \frac{\alpha'[m]}{\max(\alpha')} \right| \quad (4)$$

Fig. 2 (c) shows the behaviour of  $f_{\alpha, \epsilon}$ , where it becomes easy to identify the presence of new notes. In addition, it is also possible to estimate the number of notes played in the chord (in this case 6), as well as the stroke speed.

## 2.2 Estimation of number of notes and stroke speed

After a measure that highlights the presence of new notes has been defined, the next step is to find the peaks of  $f_{\alpha, \epsilon}$ . Each sample of the function  $f_{\alpha, \epsilon}(m)$  is compared against  $f_{\alpha, \epsilon}(m-1)$  and  $f_{\alpha, \epsilon}(m+1)$ . If  $f_{\alpha, \epsilon}(m)$  is larger than both neighbors (local maximum) and  $f_{\alpha, \epsilon}(m) > 0.1$ , then a candidate local peak is detected. Finally, if there are two



**Figure 2.** F Major chord UP stroke in classic guitar: (a) Evolution of  $\alpha[m]$  that minimizes equation (1), (b) Evolution of the error  $\epsilon[m]$  as defined in equation (1) and (c) Evolution of  $f_{\alpha,\epsilon}[m]$  (equation (4)) where the presence of new notes becomes apparent.

peaks less than two points apart, the smallest one is not considered. Once these selected peaks have been localized, the final step is to determine which ones belong to played notes so that the number of played notes can be estimated together with the speed of the stroke. The key observation is that the time difference between the note onsets that belong to the same stroke or arpeggio will be approximately constant. The reason is that, because of human physiology, in most cases the stroke is performed with fixed speed.

Let  $f_{locs}$  stand for a function that contains the positions where the selected peaks of  $f_{\alpha,\epsilon}$  are located. The objective is to detect sets of approximately equispaced peaks which will correspond to the played notes in a chord or arpeggio. Then, the number of played notes  $NPN_e$  will be estimated as follows:

$$NPN_e = n_{neq} + 2 \quad (5)$$

where  $n_{neq}$  represents the minimum value of  $n$  such that the distance between the positions of peaks contained in  $f_{locs}$  is no longer kept approximately constant.  $n_{neq}$  is defined as:

$$n_{neq} = \underset{n}{\operatorname{argmin}} \left( |f''_{locs}(n)| > 3 \right) \quad (6)$$

where  $f''_{locs}(n)$  stands for the second order difference of  $f_{locs}(n)$ .

Finally, the stroke speed estimate in notes per second is given by:

$$V = \frac{f_{locs}(NPN_e - 3) \cdot (\text{window size} - \text{overlap})}{f_s \cdot NPN} \quad (7)$$

Once the location of every new note is estimated using the method described, the feature to detect the stroke direction is computed.

### 2.3 Feature to detect stroke direction

In Fig. 3, the details of the windows in which the spectral changes occur are depicted for the two guitar chords that are being analysed. The stroke direction can be guessed from those figures, but we still need to derive a meaningful computational feature that can be used for automatic classification.

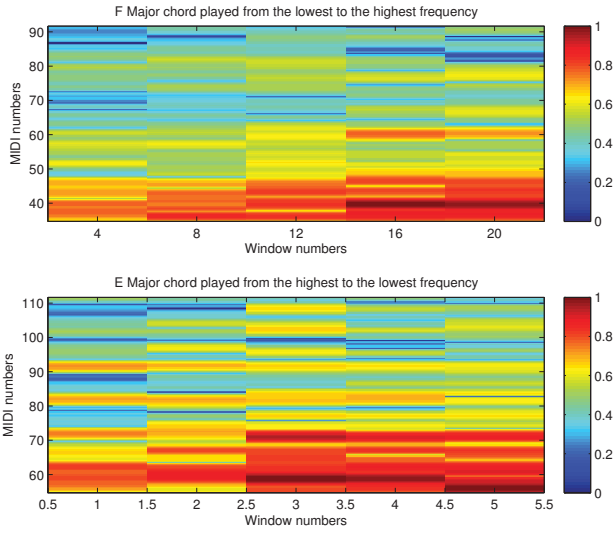
In order to reduce the amount of information to be processed by the classifier that will decide the stroke direction, a meaningful feature must be considered. Thus, the spectral centroid in each of the windows in which the changes have been detected is calculated.

The spectral centroid is the centre of gravity of the spectrum itself [22], [24] and, in our case, it is estimated in each of the windows  $x_m$  where the change has been detected. This feature will be denoted  $SPC_m$  (Spectral Centroid of window  $m$ ) and it is calculated as follows:

$$SPC_m = \left( \frac{\sum_{k=1}^K f_m(k) PSD_m(k)}{\sum_{k=1}^K PSD_m(k)} \right) \quad (8)$$

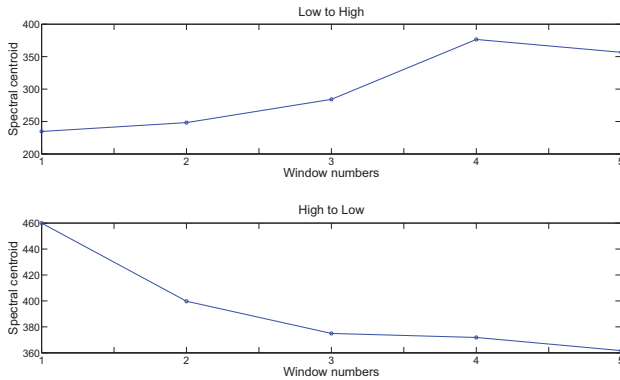
where  $PSD_m$  is the power spectral density of the window  $x_m$  and  $f_m$  is the corresponding frequency vector.

Note that we will use SPCs-KD when the SPCs are estimated with the delays known beforehand and SPCs-ED when the delays are estimated according to the procedure described in section 2.1.



**Figure 3.** Windows of the spectrogram of the UP F Major chord and the DOWN E Major chord in which new notes appear.

Fig. 4 illustrates the behaviour of the SPC in the selected windows in which a change of the spectral content is detected for the UP F Major chord and the DOWN E Major chord shown in the previous illustrations.



**Figure 4.** Spectral centroid evolution for the UP F Major chord and the DOWN E Major chord in the windows of the spectrogram in which the changes happen.

### 3. CLASSIFICATION RESULTS OF UP AND DOWN STROKES

The proposed scheme has been tested with four different instruments: guitar, piano, organ and autoharp. The guitar and organ samples have been extracted from the RWC database [10], the piano recordings have been extracted from [2] and the autoharp recordings have been specifically made by the research team.

A subset of the chords used in the experiment contains chords artificially assembled so that all the information regarding the number of notes played and the delay is available for assessing the proposed system performance. All

audio file are sampled with  $f_s = 44100$  Hz. The delay between consecutive notes in a chord ranges between 1000 samples (11 ms) and 5000 samples (55 ms).

With the guitar and the autoharp, the natural way of playing a chord is by adding the sound of one string after another. The guitar is a well known instrument [5], but the autoharp is not. The autoharp is an American instrument invented in 1881 by Charles F. Zimmerman. It was very popular in Canada and the USA for teaching music fundamentals because it is easy to play and introduces in a very intuitive way harmony concepts. Briefly, the instrument has 36 strings and the musician can select which ones can vibrate by pressing buttons corresponding to different chords. The buttons in the autoharp mute the strings corresponding to the notes that do not belong to the chord to be played. Then, the musician actuates the strings by strumming with the other hand. In the guitar, the decay of each string is exponential and very fast. In the case of the autoharp, due to the resonance box, the decay of the sound is slower. In the piano and in the organ, the musician can play the chords arpeggiated. In the piano the decay is also exponential but in the organ the sound of a note is sustained and decays slowly.

In Tables 1 and 1, the UP and DOWN classification results are summarized for the artificially assembled chords. In all the cases, 100 chords have been used for training (50 UP and 50 DOWN) and a total of 500 chords equally distributed among UP and DOWN have been used to evaluate the classification performance. The chord recordings used for training and testing purposes are separate and different.

The performance of the proposed feature is compared against a baseline that makes use of MFCCs (Mel Frequency Cepstral Coefficients) calculated as explained in [22]. More specifically, 15 coefficients are considered with the first one, corresponding to the DC component, removed.

A Fisher Linear Discriminant and a linear Support Vector Machine (SVM) classifier [23] have been evaluated.

Looking at Tables 1 and 2, we observe that the results of the proposed method and feature are satisfactory. In almost all the cases, the performance of the proposed scheme is better than the one achieved by the baseline based on MFCCs.

The error in the determination of the number of played notes is estimated as follows:

$$Error_{NPN} = A \left( \frac{|NPN_e - NPN_r|}{NPN_r} \right) \cdot 100 \quad (9)$$

where  $A()$  is the averaging operator,  $NPN_e$  stands for the estimated Number of Played Notes in (5) and  $NPN_r$  represents the the actual number of notes.

The error in the estimated delay between consecutive notes is evaluated as follows:

$$Error_W = A \left( \frac{|W_e - W_r|}{NPN_e \cdot W_d} \right) \cdot 100 \quad (10)$$

where  $W_e$  represents the windows in which a significant spectral change has been found,  $W_r$  stands for the windows

Instrument	stroke	Fisher		
		SPCs-KD	SPCs-ED	MFCCs
Guitar	up	93.88	78.88	72.22
	down	96.11	97.22	60.55
	overall	95.00	88.05	66.38
Piano	up	91.95	79.31	77.85
	down	97.81	84.36	81.42
	overall	94.88	81.83	79.64
Organ	up	90.00	89.16	78.33
	down	90.00	86.66	56.66
	overall	90.00	87.91	67.50
Autoharp	up	100	94.44	97.91
	down	100	86.80	79.86
	overall	100	90.62	88.88

**Table 1.** Success Rate (%) of UP and DOWN stroke classification using a Fisher linear classifier [23]. The features used by the classifier are: SPCs-KD (Spectral Centroid of selected Windows with known-delay), SPCs-ED (Spectral Centroid of selected Windows with estimated delay) and MFCCs (15 Mel Frequency Cepstral Coefficients).

Instrument	stroke	SVM		
		SPCs-KD	SPCs-ED	MFCCs
Guitar	up	91.57	87.77	58.44
	down	95.01	95.55	98.78
	overall	93.29	91.66	78.61
Piano	up	90.12	81.22	77.25
	down	96.45	82.84	83.63
	overall	93.28	82.03	80.44
Organ	up	89.16	90.52	90.83
	down	88.66	87.98	51.66
	overall	88.91	89.25	71.25
Autoharp	up	99.30	90.97	91.27
	down	97.91	95.14	90.89
	overall	98.61	93.05	91.08

**Table 2.** Success Rate (%) of UP and DOWN stroke classification using a linear SVM classifier [23]. The features used by the classifier are: SPCs-KD (Spectral Centroid of selected Windows with known-delay), SPCs-ED (Spectral Centroid of selected Windows with estimated delay) and MFCCs (15 Mel Frequency Cepstral Coefficients).

where the changes actually happen and  $W_d$  is number of windows between two consecutive  $W_r$  windows. Table 3 shows the obtained results.

The proposed method for the estimation of the number of notes and delays can be improved. This is a first approach to solve this problem. Our main goal has been to detect the up or down stroke direction which is useful to complete the transcription of the performance of certain instruments, specifically the autoharp. The performance attained in the detection of the stroke direction is satisfactory according to the results shown.

It is important to note, that even though  $Error_W$  seems to be quite high, this error is in most of cases positive, i.e., the change is detected one or two windows after the first

Instrument	stroke	$Error_{NPN}$	$Error_W$
Guitar	up	37.65	10.49
	down	33.33	15.92
	overall	35.49	13.20
Piano	up	30.72	28.38
	down	33.65	18.10
	overall	32.18	23.24
Organ	up	24.54	29.72
	down	36.52	26.12
	overall	30.53	27.92
Autoharp	up	53.06	10.46
	down	42.88	13.96
	overall	47.97	12.21

**Table 3.** Error (%) in the estimation of the number of notes played and in the estimation of the delay between consecutive played notes in chords.

Instrument	stroke	Fisher	
		SPCs-ED	MFCCs
Autoharp	up	65.21	43.47
	down	86.44	94.91
	overall	75.10	69.19
Autoharp		SVM	
		SPCs-ED	MFCCs
		up	73.77
down	89.83	81.52	
overall	77.52	72.18	

**Table 4.** Success Rate (%) of UP and DOWN stroke classification for real autoharp chords.

window that actually contains the change. This issue is not critical for the feature employed by the classifier because it is possible to observe the difference in the estimation of the  $SPC_m$  in (8).

Finally, Table 4 presents the results obtained for real chords played in an autoharp. We have used 100 chords for training and 230 chords for testing. The 330 autoharp chords recorded are equally distributed between UP and DOWN chords and in different tessituras. It can be observed that the proposed feature outperforms the baseline proposed based on the usage of MFCCs.

#### 4. CONCLUSIONS

In this paper, a new feature to detect the up or down direction of strokes and arpeggios has been presented. The developed method also provides information about the number of played notes and the stroke speed.

The system have been tested with four different instruments: classic guitar, piano, autoharp and organ and it has been shown how the new proposed feature outperforms the baseline defined for this task. The baseline makes use of the well known MFCCs as classification features so that the baseline scheme can be easily replicated. The Matlab files used to generate the data-set for piano, guitar and organ, the audio files of the autoharp and the ground truth are available upon request for reproducible research.

## 5. ACKNOWLEDGMENTS

This work has been funded by the Ministerio de Economía y Competitividad of the Spanish Government under Project No. TIN2013-47276-C6-2-R, by the Junta de Andalucía under Project No. P11-TIC-7154 and by the Ministerio de Educación, Cultura y Deporte through the Programa Nacional de Movilidad de Recursos Humanos del Plan Nacional de I-D+i 2008- 2011. Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

## 6. REFERENCES

- [1] A. M. Barbancho, I. Barbancho, B. Soto, and L.J. Tardón. Transcription of piano recordings. *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 377–380, 2011.
- [2] A. M. Barbancho, I. Barbancho, L. J. Tardón, and E. Molina. *Database of Piano Chords. An Engineering View of Harmony*. SpringerBriefs in Electrical and Computer Engineering, 2013.
- [3] A. M. Barbancho, A. Klapuri, L.J. Tardón, and I. Barbancho. Automatic transcription of guitar chords and fingering from audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 20:915–921, March 2012.
- [4] I. Barbancho, C. de la Bandera, A. M. Barbancho, and L. J. Tardón. Transcription and expressiveness detection system for violin music. *Proceedings of the IEEE conference on Acoustics, Speech, and Signal Proc. (ICASSP)*, pages 189–192, March 2009.
- [5] I. Barbancho, L.J Tardon, S. Sammartino, and A.M. Barbancho. Inharmonicity-based method for the automatic generation of guitar tablature. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(6):1857–1868, 2012.
- [6] J.P. Bello, L. Daudet, and M.B. Sandler. A tutorial on onset detection in music signals. *IEEE Trans. on Audio, Speech and Language Processing*, 14:1035–1047, September 2005.
- [7] E. Benetos, S. Dixon, D. Giannoulis, and H. Kirchhoff. Automatic music transcription: Breaking the glass ceiling. *ISMIR*, 2012.
- [8] Albert S Bregman. *Auditory scene analysis: The perceptual organization of sound*. MIT press, 1994.
- [9] D. Chaffaux, J.-L. Le Carrou, B. Fabre, and L. Daudet. Experimentally based description of harp plucking. *The Journal of the Acoustical Society of America*, 131:844, 2012.
- [10] M. Goto. Development of the RWC music database. In *18th Int. Con. on Acoustics*, volume I, pages 553–556, 2004.
- [11] A. Kirke and E. R. Miranda. An overview of computer systems for expressive music performance. In *Guide to Computing for Expressive Music Performance*, pages 1–47. Springer, 2013.
- [12] A. Klapuri and T. Virtanen. Automatic music transcription. In *Handbook of Signal Processing in Acoustics*, pages 277–303. Springer, 2009.
- [13] A.P. Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Trans. on Speech and Audio Processing*, 11:804–816, Nov. 2003.
- [14] K. Lee and M. Slaney. Acoustic chord transcription and key extraction from audio using key-dependent hmms trained on synthesized audio. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):291–301, 2008.
- [15] J. A. Moorer. On the transcription of musical sound by computer. *Computer Music Journal*, pages 32–38, 1977.
- [16] M.Schoeffler, F.R. Stoter, H.Bayerlein, B.Edler, and J.Herre. An experiment about estimating the number of instruments in polyphonic music: a comparison between internet and laboratory results. *ISMIR*, 2013.
- [17] M. Müller, D. P. W. Ellis, A. Klapuri, and G. Richard. Signal processing for music analysis. *IEEE Journal of Selected Topics in Signal Processing*, 5:1088–1110, October 2011.
- [18] T. H. Ozaslan, E. Guaus, E. Palacios, and J. L. Arcos. Identifying attack articulations in classical guitar. In *Computer Music Modeling and Retrieval. Exploring Music Contents. Lecture Notes in Computer Science*, pages 219–241. Springer-Verlag, 2011.
- [19] J.A. Paradiso, L.S Pardue, K.-Y. Hsiao, and A.Y. Benbasat. Electromagnetic tagging for electronic music interfaces. *Journal of New Music Research*, 32(4):395–409, 2003.
- [20] M. Peterson. *Mel Bays Complete Method for Autoharp or Chromaharp*. Mel Bay Publications, 1979.
- [21] W.H. Press, S. A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes: The Art of Scientific Computing. Third Edition*. Cambridge University Press, 2007.
- [22] L.J. Tardón, S.Sammartino, and I.Barbancho. Design of an efficient music-speech discriminator. *Journal of the Acoustical Society of America*, 1:271–279, January 2010.
- [23] S. Theodoridis and K. Koutroumbas. *Pattern Recognition, 4th Edition*. Academic Press, 2008.
- [24] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Trans. on Audio, Speech and Language Processing*, 10:293–302, 2002.

# PREDICTING EXPRESSIVE DYNAMICS IN PIANO PERFORMANCES USING NEURAL NETWORKS

**Sam van Herwaarden**

Austrian Research Institute for AI

samvherwaarden@gmail.com

**Maarten Grachten**

Austrian Research Institute for AI

<http://www.ofai.at/~maarten.grachten>

**W. Bas de Haas**

Utrecht University

w.b.dehaas@uu.nl

## ABSTRACT

This paper presents a model for predicting expressive accentuation in piano performances with neural networks. Using Restricted Boltzmann Machines (RBMs), features are learned from performance data, after which these features are used to predict performed loudness. During feature learning, data describing more than 6000 musical pieces is used; when training for prediction, two datasets are used, both recorded on a Bösendorfer piano (accurately measuring note on- and offset times and velocity values), but describing different compositions performed by different pianists. The resulting model is tested by predicting note velocity for unseen performances. Our approach differs from earlier work in a number of ways: (1) an additional input representation based on a local history of velocity values is used, (2) the RBMs are trained to result in a network with sparse activations, (3) network connectivity is increased by adding skip-connections, and (4) more data is used for training. These modifications result in a network performing better than the state-of-the-art on the same data and more descriptive features, which can be used for rendering performances, or for gaining insight into which aspects of a musical piece influence its performance.

## 1. INTRODUCTION

Music is not performed exactly the way it is described in score: a performance in which notes occur on a regular temporal grid and all notes are played equally loud is often considered dull. Depending on the instrument, performers have different parameters they use for modulating expression in their music [14]: time (timing, tempo), pitch, loudness and timbre. For some of these parameters composers add annotations to musical score describing how they should be varied, but for a large part performers are expected render the score according to tacit knowledge, and personal judgment. This allows performers to imbue on a performance their personal style, but this is not to say that music performance is arbitrary—it is often clear which

interpretations are (not) musically appropriate.

This article describes a number of modifications to the method for modeling expressive dynamics proposed by Grachten & Krebs [7], and is based on the MSc thesis work described in [17]. We show that, with an additional input representation and a different set-up of the machine learning approach, we achieve a statistically significant improvement on the prediction accuracy achieved in [7], with more descriptive features. Our achieved performance is also comparable with the work in [8]. In the following sections we first summarize previous work in this area, followed by an overview of the used machine learning architecture. We then describe the experiments, the results and the relevance of the findings.

## 2. PREVIOUS WORK

Two important aspects of music that affect the way it is to be performed are the musical structure, and the emotion that the performance should convey [13]. The last decades different methods for analyzing the structural properties of a piece of music have been proposed (e.g. [12, 15]), where the analysis tends to stress the relationship between structure on a local level (elements of pitch and rhythm) and their effect on the melodic expectancy of a listener. Emotional charge conveyed by a piece is more abstract and variable: trained musicians can play the same piece conveying different emotions, and in fact these emotions can be identified by listeners [5].

Because musical structure can be studied through inspection of the musical score, computational models of musical expression tend to focus on this. A number of different computational models of expression have been developed earlier, studying different expressive parameters (e.g. [1, 4]). Many models are rule-based, where the rules describing how expression should be applied are often hand-designed. Other models still focus on rules, but automatically extract them from performance data (e.g. [11, 18]). A performance model can also be based on the score annotations for the relevant parameter provided by the composer, as in [8] which uses information on note pitch, loudness annotations and other hand-crafted features.

Some recent studies model regularities in musical sequences using unsupervised techniques [2, 16], in the context of musical sequence prediction. Grachten & Krebs [7] apply unsupervised learning techniques to learn features from a simple input representation based on a piano roll



© S. van Herwaarden, M. Grachten, W.B. de Haas.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Sam van Herwaarden, Maarten Grachten, W. Bas de Haas. “Predicting expressive dynamics in piano performances using neural networks”, 15th International Society for Music Information Retrieval Conference, 2014.

representation of the symbolic score, in the context of predicting musical expression. The resulting learned features then describe common patterns occurring in the input data, which can be related to concepts from music theory and used for prediction of expressive dynamics. By using a simple input representation and network, the model remains relatively transparent with regard to its inner workings. It is shown that Restricted Boltzmann Machines (RBMs) learn the most effective model, and in this paper, we build on that approach.

An RBM is a type of artificial neural network, particularly suitable for unsupervised learning from binary input data. During training it learns a set of features that can efficiently encode the input data. The features are used to transform the input data non-linearly, which can be useful for further (supervised) learning. For a detailed explanation of RBMs the reader is referred to for example [9, 10].

### 3. ARCHITECTURE

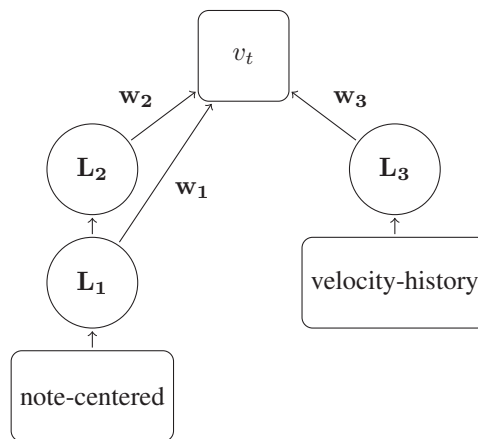
Figure 1 illustrates the setup of the network we use. As input the network sees the music data in two different representations: the score-based note-centered representation first developed by [7] and the new loudness-based velocity-history representation. The input data is transformed through a series of hidden unit activations (RBM feature activations) in  $L_1$ ,  $L_2$  and  $L_3$ . These feature activations are then used to estimate the output (normalized velocity). As is typical with neural networks, the model is blind to the meaningful ordering of the input nodes (we could change the ordering without affecting the results).

The set-up is different from that in [7] in a number of ways: (1) an additional input representation based on a local history of velocity values is used, (2) the RBMs are trained for sparse activations, (3) network connectivity is increased with skip-connections (i.e.  $w_1$  and  $w_2$  in Figure 1 can be used simultaneously), and (4) more data is used for training. The following sections cover these changes in more detail. First, we describe the data available for developing the model. We then describe the way these data are presented to our model as input and output, and finally the process of training and evaluating our model.

#### 3.1 Available data

Data from a number of sources is used for the experiments in this paper. We have *score* data, which describes musical score in a piano-roll fashion, and we have *performance* data, based on recordings from a computer-controlled Bösendorfer piano. For the performance data, accurate note on- and offsets are available as well as velocity values, and these values have been linked to corresponding score data. For all available performance data, score data is also available, the converse does not hold.

A number of (MIDI) score datasets is used: the JSB Chorales,<sup>1</sup> some MuseScore pieces,<sup>2</sup> the Mutopia



**Figure 1:** The used architecture. The rounded squares correspond to in- and outputs, the circles to layers of hidden units trained as Restricted Boltzmann Machines.  $w_1$  through  $w_3$  are the weights used to predict  $v_t$  based on the hidden unit activations in hidden layers  $L_1$  through  $L_3$ .  $w_1$  through  $w_3$  are determined with a least-squares fit.

database,<sup>3</sup> the Nottingham database,<sup>4</sup> the Piano-midi archive<sup>5</sup> and the Voluntocracy dataset<sup>6</sup>. These datasets are used during unsupervised learning with the note-centered representation only. The performance datasets we use have been developed at the Austrian Research Institute for AI (OFAI). One dataset contains performance data of all Chopin’s piano music played by Nikita Magaloff [3] ( $\sim 300.000$  notes in 155 pieces), the other contains all Mozart piano sonatas, performed by Roland Batik [18] ( $\sim 100.000$  notes in 128 pieces). These datasets have been used both for unsupervised and supervised learning.

#### 3.2 Note-centered representation

Score data is input into the network in one form in the *note-centered representation*, which is based on a piano-roll representation. For every note in a musical score, an input sample is generated with this note in the center, as illustrated in Figure 2b. The horizontal axis corresponds to score time and covers a span of 3 beats before the onset of the central note to 3 beats after the onset. Each beat is further divided into 8 equal units of time (effectively each column in the input corresponds to a 32<sup>nd</sup> note), and longer notes are wider. The vertical axis corresponds to relative pitch compared to the central note, and covers a span of  $-55$  to  $+55$  semi-tones. To allow the representation to distinguish between separate notes of the same pitch played consecutively, and a single long note at that pitch, note durations are represented as their score duration minus 32<sup>nd</sup> note duration (this was also done in [7]).

This approach is the same as the *duration coding* approach used in [7] with two exceptions: they experimented with time-spans of 1, 2 and 4 beats (with very small dif-

<sup>3</sup> [www.mutopiaproject.org](http://www.mutopiaproject.org)

<sup>4</sup> [www.chezfred.org.uk/University/music/database.htm](http://www.chezfred.org.uk/University/music/database.htm)

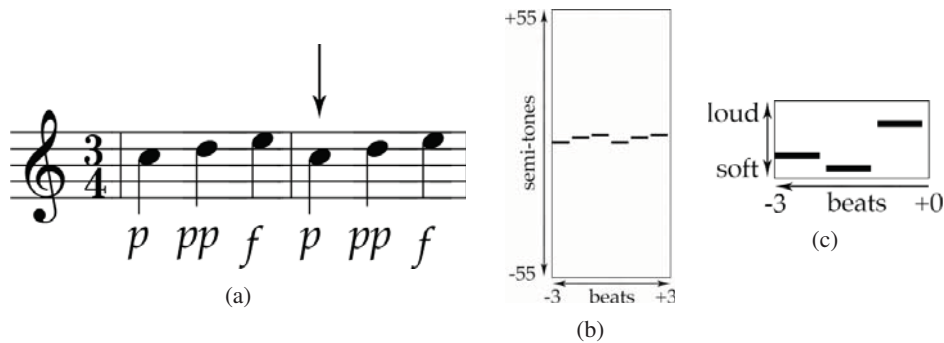
<sup>5</sup> [www.piano-midi.de](http://www.piano-midi.de)

<sup>6</sup> [www.voluntocracy.org](http://www.voluntocracy.org)

<sup>1</sup> [www.jsbchorales.net](http://www.jsbchorales.net)

<sup>2</sup> [www.musescore.org](http://www.musescore.org)





**Figure 2:** A short piece of score, and resulting network input for the note indicated by the arrow: (a) shows the score, where the annotations should be interpreted as *performed* loudness, not as annotated loudness directives, (b) shows the note-centered representation and (c) the velocity-history representation.

ferences in results between the 2 and 4 beats experiments), and used a pitch range of  $-87$  to  $+87$  semi-tones (so that always the entire piano keyboard is covered). In practice, the large pitch range is likely unnecessary and only increases the length of the network input vector (note combinations with such intervals are very rare and do not noticeably affect the learned features).

This choice of representation makes our system insensitive to absolute pitch: if all input notes are transposed by a few semi-tones in the same direction, the generated input samples will be identical. This also allows the system to learn about harmony based on relative pitch: for example certain chords will typically be represented in the same way regardless of their root tone. No additional information on absolute note pitch was included, to keep the model simple.

### 3.3 Velocity-history representation

When analyzing expressive parameters in existing performances, it is interesting to not only take into account direct harmonic and rhythmic structure around a note as is done with the note-centered representation, but also effects in continuity of musical phrases: for example, in many cases note loudness increases or decreases gradually over a number of notes. The precise accentuation of a note is than affected by the accentuation of preceding notes.

Our *velocity-history representation* is designed to encode this kind of information. Figure 2c illustrates this representation. Conceptually, it is similar to the note-centered representation, with a few differences: the vertical axis now represents relative velocity (normalized with respect to the mean  $\mu$  and standard deviation  $\sigma$  of the velocity in a piece, where the range from  $\mu - 2\sigma$  to  $\mu + 2\sigma$  is quantized into 12 discrete values), and the horizontal axis corresponds to the time preceding the current note (ranging from note onset  $-3$  beats to note onset  $+0$ ).

The velocity-history representation uses information from an actual performance during prediction. In a sense, the system is asked to predict the continuation of a musical phrase: given that the last notes were played in a certain way, how will the next note be played? When using this representation, experiments with our model aim to *explain*

how a note is performed in an existing performance, rather than *predict* it for a new piece of bare score (an actual performance needs to be available).

### 3.4 Velocity normalization

Since we use semi-supervised learning, at some point we need target values accompanying our input representations. We have exactly one sample for each note, and we are studying dynamics, so the logical parameter to base these target values on is note velocity. However, the different pieces described in our data have fairly diverse characteristics when it comes to dynamics. Some pieces are performed louder on average, or have stronger variations in dynamics. In this study we have chosen to focus on local effects within a single piece, and not so much on differences between pieces. For this reason we normalize our velocity target values so they have zero-mean and unit standard-deviation within a piece (we use these values both for supervised learning and for generating the velocity-history representation). This is slightly different from the normalization used in [7], where normalization was only used to obtain zero-mean within a piece.

### 3.5 Training and evaluation

The process of developing and testing the network can be separated into three phases: unsupervised learning, supervised learning and performance evaluation. We will now describe these in more detail.

#### 3.5.1 Unsupervised learning

During unsupervised learning, we train only hidden layers  $L_1$  through  $L_3$ . The layers are trained as RBMs on the full set of score data in the note-centered and velocity-history representations, where  $L_1$  and  $L_3$  are trained on the input representations directly, and  $L_2$  is trained on the feature activations in  $L_1$ .

In the note-centered representation samples consist of 5280 binary input values.  $L_1$  is trained with 512 hidden units (ensuring a significant bottleneck in the network), and  $L_2$  contains fewer hidden units again: 200 units. In the velocity-history representation samples consist of 288 input values, these are encoded in 120 hidden units in  $L_3$ .

We enforce sparse coding in the network, using the method proposed in [6], which allows us to not only control the average activation of hidden units in the network, but also the actual distribution of activations: we can force the RBM to represent each sample as a number of highly active features, improving inspectability.

### 3.5.2 Supervised learning

For supervised learning we use a simple approach: given the transformation of an input sample by  $L_1$  to  $L_3$ , we fit the hidden unit activations in these layers to the corresponding  $v_t$  (normalized velocity) values using least-squares. Exploratory experiments suggested that more advanced techniques do not yield much better results. Thus,  $w_1$  through  $w_3$  simply define a linear transformation from the features activations to a prediction of the normalized velocity.

### 3.5.3 Performance evaluation

To evaluate the performance of our model we use a leave-one-out approach: we cycle through all the pieces in the performance data, where every time a particular piece is left out during supervised learning, after which the trained network is used to predict the expressive dynamics of the left-out piece. The quality of the prediction is then quantified using the  $R^2$  measure (coefficient of determination). As mentioned before, the full set of data is used during unsupervised learning – because the objective function optimized during this phase has no relation to the velocity targets, we believe that this is an acceptable approach. As the final score after cycling through the whole dataset in this fashion, we use the weighted average  $R^2$ , where the number of notes in a piece is used as its weight.

## 4. EXPERIMENTS

In our experiments we vary two parameters: network connectivity, and training/testing datasets. Other experiments were also done but are not described in this paper, for these the interested reader is referred to [17].

### 4.1 Network connectivity

Different parts of our model describe information concerning different aspects of the input data. The note-centered representation corresponds to rhythmic and harmonic structure of the score surrounding a note, while the velocity-history representation relates more closely to expressive phrases. This distinction continues through the layers of feature activations. To get an impression of how strongly the expressive variation in velocity data corresponds to these different aspects, we experimented with the different layers in isolation and together. We will refer to the network configurations by the layers that were used during training and prediction, i.e.  $L_{1,2}$  means both of the layers on top of the note-centered representation were used, and  $L_3$  was not. Another way to see this would be that  $w_3$  is constrained to be a matrix of only 0's.

	no vel. inf.			with vel. inf.	
	$L_1$	$L_{1,2}$	$L_2$	$L_3$	$L_{1,2,3}$
<b>M.</b> → <b>M.</b>	.202	<u>.207</u>	.191	.315	.470
<b>B.</b> → <b>B.</b>	.366	.376	.357	.236	.532
<b>B.</b> → <b>M.</b>	.132	.126	.125	.286	.386
<b>M.</b> → <b>B.</b>	.291	.295	.283	.209	.457
<b>All</b> → <b>M.</b>	.198	.203	.186	.313	.466
<b>All</b> → <b>B.</b>	.341	.350	.329	.222	.503

**Table 1:**  $\bar{R}^2$  scores obtained on the test data.  $X \rightarrow Y$  indicates the model was trained on  $X$  and tested on  $Y$ , where **M.** is the Magaloff and **B.** the Batik dataset. Experiments with velocity information (vel. inf.) use the velocity-history representation as input. We use the underlined result for comparison with previous work ([7] and [8]).

## 4.2 Training datasets

Experimenting with different sets of training data is interesting for several reasons. One is that from a musicological perspective, the structure of music of different styles can be quite different. As an extreme example, a system trained on Jazz music would not be expected to reliably predict performances of piano music by Bach. Another reason is that we can use combinations of datasets to test the validity of our model: if a model trained on music from one set of recordings, still performs well on another set of recordings, this can give us some confidence that our model has learned something about music in a general sense, and not just about the particular dataset.

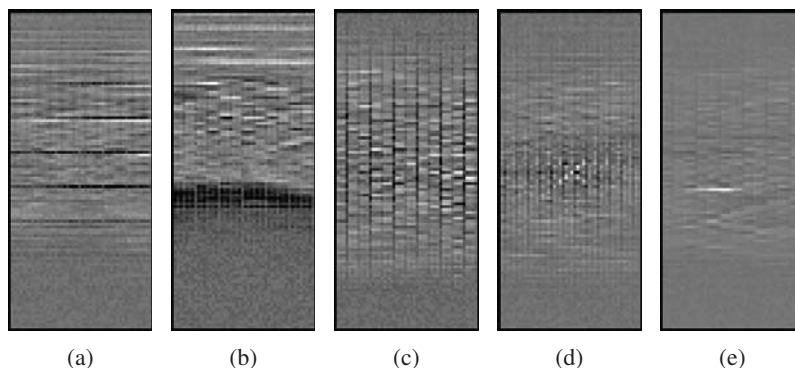
As mentioned before, we use two datasets: one describing performances of Chopin music and the other Mozart music. In all cases, during testing we kept the datasets separate. However, we varied the set of data used for training: we trained on the same dataset as used for testing, we trained on one dataset and tested on the other, and we tested a model trained on all data.

## 5. RESULTS

Table 1 lists the results obtained with our model. The model is more successful explaining the variance in the Batik (Mozart) data than in the Magaloff (Chopin) data – one possible explanation for this is that Chopin's music (from the Romantic period) has much more extreme variations in expression than Mozart's music (from the Classical period). It seems reasonable that a performance with more dynamic variation is harder to predict.

When comparing the different architectures, most information used by our model is encoded in  $L_1$  and  $L_3$ .  $L_2$  has less predictive value than  $L_1$ , and the score only improves by a little bit when these two layers are used together (suggesting there is a large amount of overlap in the information they encode).  $L_3$ , which is based on the velocity-history representation (which was not used in [7]) clearly contains a lot of information.

Interestingly,  $L_3$  contains most relevant information for



**Figure 3:** Some hand-selected features from  $L_1$  that are representative for the types of patterns learned from the note-centered representation (see Figure 2b). Dark values correspond to negative weights, light values to positive weights.

the Magaloff data, and  $L_1$  for the Batik data. This could be due to the difference between music from the Romantic period and that from the Classical period:  $L_1$  contains more information about harmony, whereas  $L_3$  contains more information about the expressive ‘flow’ of the piece.

Training on a single dataset has a positive effect on the prediction scores. This is likely due to the fact that the datasets are of a different nature in terms of musical style, and if we would want to predict performance parameters for a Mozart piece, training on Chopin music will not provide our model with the relevant ‘know-how’. This is also illustrated by the cross-training experiments, where we trained on one dataset and tested on the other: a drop in performance of around 0.08 in all cases is observed. Still, also a relatively large amount of the predictive capability remains, providing some confidence that our model generalizes over different datasets to some extent.

Because the velocity-history representation requires detailed performance data for predictions, we use the results from our  $L_{1,2}$  experiments when comparing our results to earlier work which does not use performance data. In [7] the best obtained  $\bar{R}^2$  score on the Magaloff data is .139, using a single dense RBM layer with 1000 hidden units (similar to our  $L_1$  model). Our  $L_{1,2}$  model achieved an  $\bar{R}^2$  of .207 on the same dataset. To keep statistical testing simple, we tested the statistical significance of the difference in *unweighted* average  $R^2$  of our model and the model in [7] using a Wilcoxon signed rank test. We chose the Wilcoxon test because the underlying distribution of the  $R^2$  data is unknown. We found that the unweighted average  $R^2$  of .199 of our  $L_{1,2}$  model is significantly different from the unweighted average  $R^2$  of .121 of the model in [7] ( $W = 11111, p < 2.2 \cdot 10^{-16}$ ). In [8], the maximal obtained prediction accuracy on the Magaloff dataset is an  $\bar{R}^2$  of .188. This model uses information our models have no access to, most importantly dynamic score annotations. Nevertheless, with an  $\bar{R}^2$  of .207 our  $L_1L_2$  model again seems more successful even though it does not take such annotations into account.<sup>7</sup> When we do use performance

<sup>7</sup>To perform the statistical test, detailed results from [7] were kindly provided by the authors. For the work in [8] these results were unfortunately unavailable, meaning we could not perform the same statistical analysis with this result.

data, the difference becomes more pronounced: our  $L_{1,2,3}$  model obtains an  $\bar{R}^2$  of .470 on the Magaloff data.

Something interesting to mention here is that in [17] we also experimented with limiting training data to a particular genre (i.e. training only on Nocturnes). These experiments suggested that the velocity-history representation encodes some genre-specific information, however due to space constraints we do not cover these results further here.

## 6. DISCUSSION

We discuss two properties of our model: the features that were learned from the musical data, and the performance achieved during prediction. Figure 3 illustrates a number of hand-selected features that have been learned from the note-centered representation, which were chosen to give an impression of the variety of learned features. Compared to the features learned by [7], there is a larger variety of features, where features represent sharper patterns.

### 6.1 Learned features

Figure 3 illustrates some of the learned features. The displayed features were selected so as to give the reader an impression of the diversity of the learned features. From a musicological perspective, it is interesting to see that there seem to be some remarkable patterns relating the features to music theory. The features learned from the velocity-history representation are harder to interpret musicologically, these are not further discussed in this paper.

Figure 3a shows clear horizontal banding, where interestingly the bands are exactly 12 rows apart – this corresponds to octaves. The feature in some locations displays a strong contrast between pitches one semi-tone apart, which is related to dissonance.

A common pattern is illustrated in Figure 3b, with a dark (inhibitive) band above or below a lighter region. This type of feature is also described by Grachten & Krebs [7], who argue this can be regarded as an accompaniment versus melody detector: the illustrated feature is strongly inhibited by notes in a sample that are below the central note, meaning that the feature activates more readily for bass notes. The opposite type of feature, with inhibitive regions above and excitatory regions below the central note (not

shown here), is active with a high probability for melody notes, where surrounding notes have lower pitch.

Another common pattern is the vertical banding illustrated by Figure 3c. There is some variation in the offset of the vertical bands from the edges (their phase) and how close they are together (their period). These features can convey information on the pace in the current part of the piece (predominantly short or long notes) and the temporal position of the note with respect to the beat.

A few features also display diagonal banding as illustrated by Figure 3d, although these are relatively rare. Still, we hypothesize that with these our model can deduce whether the central note is in an ascending or descending sequence.

A final common pattern is that in Figure 3e, with a sharp white band corresponding to a note at a certain relative pitch and time from the central note. It seems reasonable to suggest that these can be related to particular melodic steps – changes from one note to another with a particular relative pitch and timing.

## 6.2 Model performance

The performance of our model is an improvement compared to earlier work, particularly when the goal is to *explain* the structure of an existing performance rather than predict a performance for a new piece of score – in the former situation the velocity-history representation can be used to good effect. Still, when considering a purely predictive context (using no velocity information), an  $R^2$  of around 0.2 leaves room for improvement. There is of course a practical limit in terms of what score can be obtained: even the same pianist might not play a piece in exactly the same way on different occasions, meaning that an  $R^2$  close to 1.0 cannot be expected. A factor that limits our model is that it considers score structure at a local level only – structure at larger timescales is not considered, nor are loudness annotations, which of course also convey a lot of information about how loudly a particular piece of score is to be played. These omissions are opportunities for further work: including these components could improve performance further, for example loudness annotations could be included similarly to what was done in [8].

## 7. CONCLUSIONS

We showed that neural networks trained on relatively raw representations of musical score and musical performances can be used to predict expressive dynamics in piano performances. This was done before in [7], but we changed the learning architecture (using sparse RBMs and skip-connections), and developed a new input representation, resulting in better predictions and clearer features. We also showed that our model generalizes well to datasets on which it was not trained.

## 8. ACKNOWLEDGEMENTS

The research described in this article was sponsored by the Austrian Science Fund (FWF) under project Z159

(Wittgenstein Award), and by the European Commission under the projects Lrn2Cre8 (grant agreement no. 610859), and PHENICX (grant agreement no. 601166). The research was part of an MSc project resulting in a thesis [17], the contents of which overlap to some extent with those in this paper. W. Bas de Haas is supported by the Netherlands Organization for Scientific Research, through the NWO-VIDI-grant 276-35-001.

## 9. REFERENCES

- [1] E. Bisesi and R. Parncutt. An accent-based approach to automatic rendering of piano performance: Preliminary auditory evaluation. *Archives of Acoustics*, 36(2):283–296, 2010.
- [2] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proceedings of the Twenty-nine International Conference on Machine Learning*. ACM, 2012.
- [3] S. Flossmann, W. Goebel, M. Grachten, B. Niedermayer, and G. Widmer. The magaloff project: An interim report. *Journal of New Music Research*, 39(4):363–377, 2010.
- [4] A. Friberg, L. Fryden, L. Bodin, and J. Sundberg. Performance rules for computer-controlled contemporary keyboard music. *Computer Music Journal*, 15(2):49–55, 1991.
- [5] A. Gabrielsson and P.N. Juslin. Emotional expression in music performance: Between the performer’s intention and the listener’s experience. *Psychology of music*, 24(1):68–91, 1996.
- [6] H. Goh, N. Thome, and M. Cord. Biasing restricted boltzmann machines to manipulate latent selectivity and sparsity. In *NIPS workshop on deep learning and unsupervised feature learning*, 2010.
- [7] M. Grachten and F. Krebs. An assessment of learned score features for modeling expressive dynamics in music. *Transactions on multimedia: Special issue on music data mining*, 16(5):1–8, 2014.
- [8] M. Grachten and G. Widmer. Explaining musical expression as a mixture of basis functions. In *Proceedings of the 8th Sound and Music Computing Conference (SMC 2011)*, 2011.
- [9] G.E. Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):926, 2010.
- [10] G.E. Hinton and T.J. Sejnowski. Learning and relearning in boltzmann machines. *MIT Press, Cambridge, Mass*, 1:282–317, 1986.
- [11] H. Katayose and S. Inokuchi. Learning performance rules in a music interpretation system. *Computers and the Humanities*, 27(1):31–40, 1993.
- [12] F. Lerdahl and R.S. Jackendoff. *A generative theory of tonal music*. The MIT Press, 1983.
- [13] C. Palmer. Music performance. *Annual review of psychology*, 48(1):115–138, 1997.
- [14] R. Parncutt. Accents and expression in piano performance. *Perspektiven und Methoden einer Systemischen Musikwissenschaft*, pages 163–185, 2003.
- [15] H. Schenker. *Five graphic music analyses*, 1969.
- [16] A. Spiliopoulou and A. Storkey. Comparing probabilistic models for melodic sequences. In *Proceedings of the 2011 European conference on Machine learning and knowledge discovery in databases - Volume Part III, ECML PKDD’11*, pages 289–304, Berlin, Heidelberg, 2011. Springer-Verlag.
- [17] S. van Herwaarden. Teaching neural networks to play the piano. Master’s thesis, Utrecht University, 2014.
- [18] G. Widmer. Large-scale induction of expressive performance rules: First quantitative results. In *Proceedings of the International Computer Music Conference (ICMC’2000)*, pages 344–347, 2000.

# AN RNN-BASED MUSIC LANGUAGE MODEL FOR IMPROVING AUTOMATIC MUSIC TRANSCRIPTION

Siddharth Sigtia<sup>†</sup>, Emmanouil Benetos<sup>‡</sup>, Srikanth Cherla<sup>‡</sup>,  
Tillman Weyde<sup>‡</sup>, Artur S. d’Avila Garcez<sup>‡</sup>, and Simon Dixon<sup>†</sup>

<sup>†</sup> Centre for Digital Music, Queen Mary University of London

<sup>‡</sup> Department of Computer Science, City University London

<sup>†</sup> { s.s.sigtia, s.e.dixon}@qmul.ac.uk

<sup>‡</sup> { emmanouil.benetos.1, srikanth.cherla.1, t.e.veyde, a.garcez}@city.ac.uk

## ABSTRACT

In this paper, we investigate the use of Music Language Models (MLMs) for improving Automatic Music Transcription performance. The MLMs are trained on sequences of symbolic polyphonic music from the Nottingham dataset. We train Recurrent Neural Network (RNN)-based models, as they are capable of capturing complex temporal structure present in symbolic music data. Similar to the function of language models in automatic speech recognition, we use the MLMs to generate a prior probability for the occurrence of a sequence. The acoustic AMT model is based on probabilistic latent component analysis, and prior information from the MLM is incorporated into the transcription framework using Dirichlet priors. We test our hybrid models on a dataset of multiple-instrument polyphonic music and report a significant 3% improvement in terms of F-measure, when compared to using an acoustic-only model.

## 1. INTRODUCTION

Automatic Music Transcription (AMT) involves automatically generating a symbolic representation of an acoustic musical signal [4]. AMT is considered to be a fundamental topic in the field of music information retrieval (MIR) and has numerous applications in related fields in music technology, such as interactive music applications and computational musicology. The majority of recent transcription papers utilise and expand *spectrogram factorisation* techniques, such as non-negative matrix factorisation (NMF) [18] and its probabilistic counterpart, probabilistic latent component analysis (PLCA) [25]. Spectrogram factorisation techniques decompose an input spectrogram of the audio signal into a product of spectral templates (that typically correspond to musical notes) and component activations (that indicate whether each note is active at a given

time frame). Spectrogram factorisation-based AMT systems include the work by Bertin et al. [7], who proposed a Bayesian framework for NMF, which considers each pitch as a model of Gaussian components in harmonic positions. Benetos and Dixon [3] proposed a convolutive model based on PLCA, which supports the transcription of multiple-instrument music and supports tuning changes and frequency modulations (modelled as shifts across log-frequency).

In terms of connectionist approaches for AMT, Nam et al. [20] proposed a method where features suitable for transcribing music are learned using a deep belief network consisting of stacked restricted Boltzmann machines (RBMs). The model performed classification using support vector machines and was applied to piano music. Böck and Schedl used recurrent neural networks (RNNs) with Long Short-Term Memory units for performing polyphonic piano transcription [8], with the system being particularly good at recognising note onsets.

There is no doubt that a reliable acoustic model is important for generating accurate symbolic transcriptions of a given music signal. However, since music exhibits a fair amount of structural regularity much like language, it is natural for one to think of the possibility of improving transcription accuracy using a *music language model* (MLM) in a manner akin to the use of a language model to improve the performance of a speech recognizer [21]. In [9], the predictions of a polyphonic MLM were used to this end, which was further developed in [10], where an input/output extension of the RNN-RBM was proposed that learned to map input sequences to output sequences in the context of AMT. Both in [9] and [10], evaluations were performed using synthesized MIDI data. In [22], Raczyński et al. utilise chord and key information for improving an NMF-based AMT system in a post-processing step. A major advantage of using a hybrid acoustic + language model system is that the two models can be trained independently using data from different sources. This is particularly useful since annotated audio data is scarce while it is relatively easy to find MIDI data for training robust language models.

In the present work, we integrate a MLM with an AMT system, in order to improve transcription performance. Specifically, we make use of the predictions made by a Recurrent Neural Network (RNN) and a RNN-Neural Autore-



© S. Sigtia, E. Benetos, S. Cherla, T. Weyde, A. S. d’Avila Garcez, and S. Dixon.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** S. Sigtia, E. Benetos, S. Cherla, T. Weyde, A. S. d’Avila Garcez, and S. Dixon. “An RNN-based Music Language Model for Improving Automatic Music Transcription”, 15th International Society for Music Information Retrieval Conference, 2014.

gressive Distribution Estimator (RNN-NADE) based polyphonic MLM proposed in [9] to refine the transcriptions of a PLCA-based AMT system [2, 3]. Information from the MLM is incorporated into the PLCA-based acoustic model as a prior for pitch activations during parameter estimation. It is observed that combining the two models in this way boosts transcription accuracy by +3% on the Bach10 dataset of multiple-instrument polyphonic music [13], compared to using the acoustic AMT system only.

The outline of this paper is as follows. The PLCA-based transcription system is presented in Section 2. The RNN-based polyphonic music prediction system that is used as a music language model is described in Section 3. The combination of the two aforementioned systems is presented in Section 4. The employed dataset, evaluation metrics, and experimental results are shown in Section 5; finally, conclusions are drawn in Section 6.

## 2. AUTOMATIC MUSIC TRANSCRIPTION SYSTEM

For combining acoustic and music language information in an AMT context, we employ the model of [3], which supports the transcription of multiple-instrument polyphonic music and also supports pitch deviations and frequency modulations. The model of [3] is based on PLCA, which is a latent variable analysis method which has been used for decomposing spectrograms. For computational efficiency purposes, we employ the fast implementation from [2], which utilized pre-extracted note templates that are also pre-shifted across log-frequency, in order to account for frequency modulations or tuning changes. In addition, as was shown in [24], PLCA-based models can utilise priors for estimating unknown model parameters, which will be useful in this paper for informing the acoustic transcription system with symbolic information.

The transcription model takes as input a normalised log-frequency spectrogram  $V_{\omega,t}$  ( $\omega$  is the log-frequency index and  $t$  is the time index) and approximates it as a bivariate probability distribution  $P(\omega, t)$ .  $P(\omega, t)$  is decomposed into a series of log-frequency spectral templates per pitch, instrument, and log-frequency shifting (which indicates deviation with respect to the ideal tuning), as well as probability distributions for pitch, instrument, and tuning.

The model is formulated as:

$$P(\omega, t) = P(t) \sum_{p,f,s} P(\omega|s, p, f) P_t(f|p) P_t(s|p) P_t(p) \quad (1)$$

where  $p$  denotes pitch,  $s$  denotes the musical instrument source, and  $f$  denotes log-frequency shifting.  $P(t)$  is the energy of the log-spectrogram, which is a known quantity.  $P(\omega|s, p, f)$  denotes pre-extracted log-spectral templates per pitch  $p$  and instrument  $s$ , which are also pre-shifted across log-frequency. The pre-shifting operation is made in order to account for pitch deviations, without needing to formulate a convolutive model across log-frequency.  $P_t(f|p)$  is the time-varying log-frequency shifting distribution per pitch,  $P_t(s|p)$  is the time-varying source contribution per

pitch, and finally,  $P_t(p)$  is the pitch activation, which essentially is the resulting music transcription. As a time-frequency representation in the log-frequency domain we use the constant-Q transform (CQT) with a log-spectral resolution of 60 bins/octave [23].

The unknown model parameters ( $P_t(f|p)$ ,  $P_t(s|p)$ , and  $P_t(p)$ ) can be iteratively estimated using the expectation-maximisation (EM) algorithm [12]. For the *Expectation* step, the following posterior is computed:

$$P_t(p, f, s|\omega) = \frac{P(\omega|s, p, f) P_t(f|p) P_t(s|p) P_t(p)}{\sum_{p,f,s} P(\omega|s, p, f) P_t(f|p) P_t(s|p) P_t(p)} \quad (2)$$

For the *Maximization* step (without using any priors) unknown model parameters are updated using the posterior computed from the Expectation step:

$$P_t(f|p) \propto \sum_{\omega,s} P_t(p, f, s|\omega) V_{\omega,t} \quad (3)$$

$$P_t(s|p) \propto \sum_{\omega,f} P_t(p, f, s|\omega) V_{\omega,t} \quad (4)$$

$$P_t(p) \propto \sum_{\omega,f,s} P_t(p, f, s|\omega) V_{\omega,t} \quad (5)$$

We consider the sound state templates to be fixed, so no update rule for  $P(\omega|s, p, f)$  is applied. Using fixed templates, 20-30 iterations using the update rules presented in the present section are sufficient for convergence. The output of the system is a pitch activation which is scaled by the energy of the log-spectrogram:

$$P(p, t) = P(t) P_t(p) \quad (6)$$

After performing 5-sample median filtering for note smoothing, thresholding is performed on  $P(p, t)$  followed by minimum note duration pruning set to 40ms in order to convert  $P(p, t)$  into a binary piano-roll representation, which is the output of the transcription system, and is also used for evaluation purposes.

## 3. POLYPHONIC MUSIC PREDICTION SYSTEM

Taking inspiration from speech recognition, it has been shown that a good statistical model of symbolic music can help the transcription process [11]. However there are 2 main reasons for the use of MLMs in AMT not being more common.

1. Training models that capture the temporal structure and complexity of symbolic polyphonic music is not an easy task. In speech recognition, often simple language models like n-grams work extremely well. However, music has a more complex structure and simple statistical models like n-grams and HMMs fail to model these characteristics accurately even for music with simple structure [9].
2. There is no consensus on how to incorporate this prior information within the transcription system. However, recently there have been some successful attempts at using this prior information to improve the accuracy on AMT tasks [9, 10].

In this section we discuss the details of the music prediction system and the models used. In the next section we discuss how we incorporate the predictions from these models in a PLCA-based music transcription system.

### 3.1 Recurrent Neural Networks

A recurrent neural network (RNN) is a powerful model for time-series data which can account for long-term temporal dependencies, over multiple time-scales when trained effectively. Given a sequence of inputs  $v_1, v_2, \dots, v_T$  each in  $\mathbb{R}^n$ , the network computes a sequence of hidden states  $\hat{h}_1, \hat{h}_2, \dots, \hat{h}_T$  each in  $\mathbb{R}^m$ , and a sequence of predictions  $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_T$  each in  $\mathbb{R}^k$  by iterating the equations

$$\hat{h}_t = e(W_{\hat{h}x}v_t + W_{\hat{h}\hat{h}}\hat{h}_{t-1} + b_{\hat{h}}) \quad (7)$$

$$\hat{y}_t = g(W_{y\hat{h}}\hat{h}_t) \quad (8)$$

where  $W_{y\hat{h}}, W_{\hat{h}x}, W_{\hat{h}\hat{h}}$  are the weight matrices,  $b_{\hat{h}}$  is the bias and  $e$  and  $g$  are activation functions which are typically non-linear and applied element-wise.

An RNN can be trained using the gradient-based Back-Propagation Through Time algorithm [27] using the exactly computable error gradients in the network. However, 1<sup>st</sup> order gradient methods fail to correctly train RNNs for many real-world problems. This difficulty has been associated with what is known as the *vanishing/exploding gradients* phenomenon [6], where the errors exhibit exponential decay/growth as they are back-propagated through time. years [15, 16, 19].

However, recent work in the field of neural networks and deep learning has led to several improvements in gradient based optimization methods that make training of RNNs possible. Most notably, the Hessian Free (HF) optimization algorithm has been used to train RNNs successfully on several real world datasets, including symbolic polyphonic music data [19]. Apart from second order methods like HF, several modifications to first-order gradient based methods exist that currently form the state of the art in training RNNs [5].

### 3.2 Recurrent Neural Network-based models

One of the drawbacks of using RNNs to predict polyphonic symbolic music is that any output of the network,  $\hat{y}_i$  at time step  $t$ , is conditionally independent of  $\hat{y}_j, \forall j \neq i$  given the sequence of input vectors  $v_1, v_2, \dots, v_T$ . This is a severe constraint when used for modelling polyphonic music, where notes often appear in very correlated patterns within a frame. In order to overcome this limitation, models derived from RNNs have been proposed which are better at modelling high-dimensional sequences [9, 26].

The first RNN-based model that tried to model high-dimensional sequences is the Recurrent Temporal Restricted Boltzmann Machine (RTRBM) [26]. This model was extended to the more general RNN-RBM model, where the hidden states for the RBM and RNN were not constrained to be the same. For our prediction system, we make use of a variant of the RNN-RBM, called the RNN-NADE. The only difference is that the conditional distributions at each

step are modelled by a Neural Autoregressive Distribution Estimator (NADE) [17] as opposed to an RBM. As discussed in the next section, to combine the predictions with the transcription system, we need individual pitch activation probabilities at each time-step. Obtaining these probabilities from an RBM is intractable as it requires summing over all possible hidden states. However the NADE is a tractable distribution estimator and we can easily obtain these probabilities from the NADE. The NADE models the probability of occurrence of a vector  $p$  as:

$$P(p) = \prod_{i=1}^D P(p_i | \mathbf{p}_{<i}) \quad (9)$$

where  $p \in \mathbb{R}^D, p_i$  is the pitch activation and  $\mathbf{p}_{<i}$  is the vector containing all the pitch activations  $p_j$  such that  $j < i$ .

In our system we utilise each of the conditional probabilities  $P(p_i | \mathbf{p}_{<i})$  as probabilities of the pitch activations. Although the pitch activation probabilities are only conditioned on  $\mathbf{p}_{<i}$ , we hypothesize that this will be a better model than the RNN, where the pitch activation probabilities are completely independent. Another motivation for using the NADE is that the gradients can be computed exactly, and therefore we can employ HF optimization for training the RNN-NADE.

## 4. COMBINING TRANSCRIPTION AND PREDICTION

In this section, we describe the process for combining the acoustic model with the music language model for deriving an improved transcription. Firstly, the input music signal is transcribed using the process described in Section 2. The resulting piano-roll representation of the transcription system is considered to be a sequence  $p_1, p_2, \dots, p_T$  that is placed as input to the MLM presented in Section 3. For the RNN-NADE, we compute the probability  $P(p_i | \mathbf{p}_{<i})$  for all time frames, and use that as prior information for the combined model, with the prior information denoted as  $P_{MLM}(p, t)$ , where  $P_{MLM}(p = i, t) = P(p_i | \mathbf{p}_{<i})$ . For the RNN, the prediction output is directly denoted as  $P_{MLM}(p, t)$ , since pitch probabilities are independent.

As shown in [24], PLCA-based models use multinomial distributions; since the Dirichlet distribution is conjugate to the multinomial, a Dirichlet prior can be used to enforce structure on the pitch activation distribution  $P_t(p)$ . Following the procedure of [24], we define the Dirichlet hyperparameter for the pitch activation as:

$$\alpha_t(p) \propto P_t(p)P_{MLM}(p, t) \quad (10)$$

where  $\alpha_t(p)$  essentially is a pitch activation probability which is filtered through a pitch indicator function computed from the symbolic prediction step (the denominator is simply for normalisation purposes).

The recording is then re-transcribed, using as additional information the prior computed from the transcription step. The modified update for the pitch activation which replaces

(5) is given by:

$$P_t(p) \propto \sum_{\omega, f, s} P_t(p, f, s | \omega) V_{\omega, t} + \kappa \alpha_t(p) \quad (11)$$

where  $\kappa$  is a weight parameter expressing how much the prior should be imposed; as in [24], the weight decreases from 1 to 0 throughout the iterations. To summarize, the transcription creates a symbolic prediction, which in turn improves the subsequent re-transcription of the music signal. An overview of the complete transcription-prediction system architecture can be seen in Fig. 1.

## 5. EVALUATION

### 5.1 Dataset

For testing the transcription system, we employ the Bach10 dataset [13], which is a freely available multi-track collection of multiple-instrument polyphonic music. It consists of ten recordings of J.S. Bach chorales, performed by violin, clarinet, saxophone, and bassoon. Pitch ground truth for each instrument is also provided. Due to the tonal and homogeneous content of the dataset (single composer, single music language), it is suitable for testing the incorporation of music language models in a multiple-instrument transcription system. For training the transcription system, pre-extracted and pre-shifted spectral templates are extracted for the instruments present in the dataset, using isolated note samples from the RWC database [14].

For training the MLMs we use the Nottingham dataset<sup>1</sup>, a collection of 1200 music pieces in symbolic ABC format, which contain simple chord combinations and tunes. We trained the RNN and the RNN-NADE models using both Stochastic Gradient Descent (SGD) and HF to compare performance. The inputs to both the models are sequences of length 200 where each frame of the sequence is a binary vector of length 88 which covers the full piano note range. We train both the RNN and the RNN-NADE to predict the next vector given a sequence of input vectors. We train the models by minimizing the negative log-likelihood of the sequences using the cross-entropy  $\sum_i t_i \log p_i + (1 - t_i) \log(1 - p_i)$  where  $i$  sums over all the dimensions of the binary vector and  $t_i$  is the pitch target.

### 5.2 Metrics

For evaluating the performance of the proposed system for multi-pitch detection, we employ the precision (*Pre*), recall (*Rec*), and F-measure (*F*) metrics, which are commonly used in transcription evaluations [1]. As in the public evaluations on multi-pitch detection carried out through the MIREX framework [1], a detected note is considered correct if its pitch is the same as the ground truth pitch and its onset is within a 50ms tolerance interval of the ground-truth onset.

Model	<i>Pre</i>
RNN (SGD)	67.89%
RNN (HF)	69.61%
RNN-NADE (SGD)	68.89%
RNN-NADE (HF)	<b>70.61%</b>

**Table 1.** Validation results for MLMs

### 5.3 Results

To validate the performance of the MLMs, we calculate the prediction precision on unseen sequences of music from the Nottingham dataset of folk melodies. We utilise 694 tracks for training, 173 tracks for validation and 170 for testing<sup>2</sup>. For both the RNN and RNN-NADE models we sample 10 vectors from the conditional distribution at each time-step and calculate the expected precision against the ground truth. The reported precision is found by finding the mean over the predictions of every frame. Table 1 shows the results of the validation experiments. These results are of the same order as the prediction accuracies reported in [9]. We found that for both the models, HF optimization gave better precision than SGD. Training with HF was also easier as there were less hyper parameters to be tuned when compared to SGD, where learning rate needs to be updated to make sure training is effective. The RNN models had a hidden layer of size 150, while the RNN-NADE models had a hidden layer of size 100 and the NADE consisted of a hidden layer of size 150.

Multi-pitch detection experiments are performed using the proposed system, with various configurations. A first configuration only considers the transcription system from Section 2. A second configuration takes the output of the transcription system and gives it as input to the prediction system of Section 3, where the final piano-roll is the output of the prediction step. A third configuration (presented in Section 4), re-transcribes the recording, having the prediction as a prior information for estimating the pitch activations. For the prediction system, experiments were performed using both the RNN-NADE and the RNN.

Results using the various system configurations are displayed in Table 2. It can be seen that the best performance is achieved by the 3rd configuration when using the NADE-HF model for prediction, which surpasses the acoustic-only transcription system by more than 3%. In general, it can be seen that using the prediction system as a post-processing step (2nd configuration) always leads to an improvement over the acoustic-only model (1st configuration). A similar trend can be observed when integrating the prediction information as a prior in the transcription system (configuration 3) compared to just using the prediction system as post-processing (configuration 2); an improvement is always reported. Another observation can be made when comparing the RNN-NADE with the RNN, with the former providing a clear improvement. For comparative purposes, we also trained MLMs using 500 MIDI files of J.S. Bach chorales<sup>3</sup> and tested the models on the

<sup>1</sup> ifdo.ca/~seymour/nottingham/nottingham.html

<sup>2</sup> <http://www-etud.iro.umontreal.ca/~boulanni/icml2012>

<sup>3</sup> <http://www.jsbchorales.net/sets.shtml>



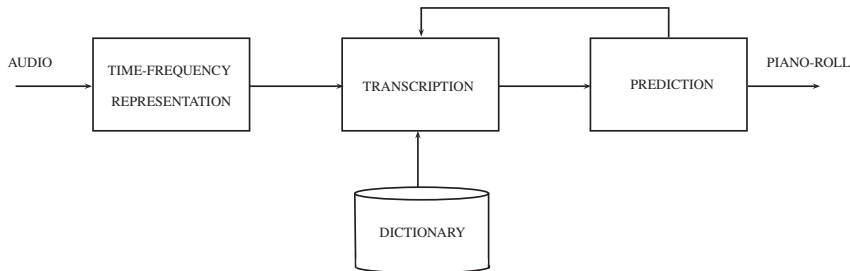


Figure 1. Proposed system diagram.

Configuration	$F$	$Pre$	$Rec$
Configuration 1	62.02%	58.51%	66.12%
Configuration 2 - NADE	62.62%	59.70%	65.92%
Configuration 3 - NADE	64.08%	61.96%	66.44%
Configuration 2 - RNN	62.29%	59.08%	65.98%
Configuration 3 - RNN	63.85%	61.14%	66.90%
Configuration 2 - NADE-HF	62.20%	59.14%	65.68%
Configuration 3 - NADE-HF	<b>65.16%</b>	<b>62.80%</b>	<b>67.78%</b>
Configuration 2 - RNN-HF	62.44%	59.28%	66.07%
Configuration 3 - RNN-HF	62.87%	60.03%	66.11%

Table 2. Transcription results using various system configurations.

Bach10 recordings. Using the Bach MLMs, the system reached  $F = 63.58\%$ , which is an improvement over the acoustic-only system, but is outperformed by the Nottingham language model.

Qualitatively, the MLMs are able to improve transcription performance by providing a rough estimate of which pitches are expected to appear in the recording (and which pitches are not expected to appear). The language models were trained using simple chord sequences (from the Nottingham dataset) that are representative of simple tonal music and are applicable as language models to the more complex Bach chorales. We believe that the reason for the J.S. Bach MLMs not performing as well as the Nottingham MLMs is due to the fact that predicting Bach’s music is a complex task (many exceptions, key changes, modulations), whereas a simple tonal model like the Nottingham dataset can work as a general-purpose language model in many types of music (this is also verified in [9]).

By comparing with the method of [13] (where the Bach10 dataset was first introduced), the proposed method using the frame-based accuracy metric reaches 74.3% for the NADE-HF using the 3rd configuration, whereas the method of [13] reaches 69.7% (with unknown polyphony). As an example of the proposed system’s performance, the spectrogram and raw output of the system using the 3rd configuration is displayed for a Bach10 recording Fig. 2, whereas the post-processed transcription output along with the ground truth for the same recording is shown in Fig. 3.

## 6. CONCLUSIONS

In this paper, we proposed a system for automatic music transcription which incorporated prior information from a polyphonic music prediction model based on recurrent

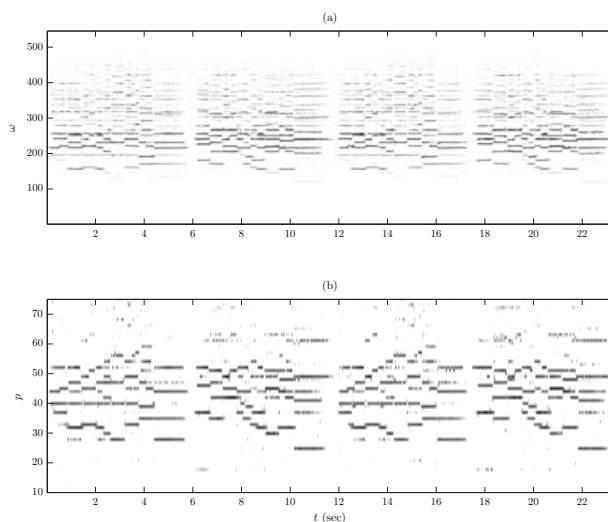
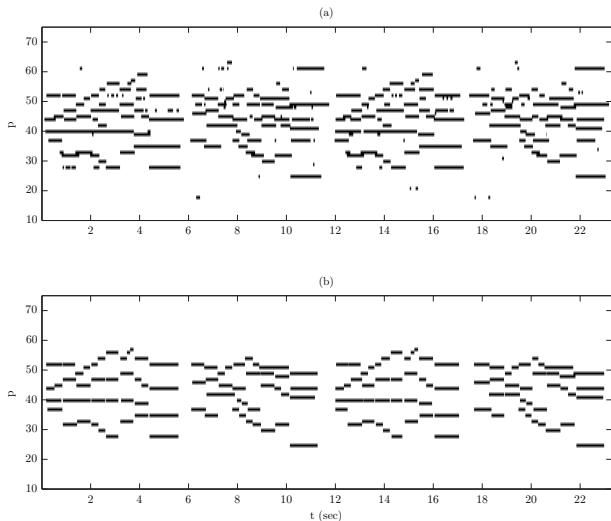


Figure 2. (a) The spectrogram  $V_{\omega,t}$  for recording “Ach Lieben Christen” from the Bach10 dataset. (b) The pitch activation  $P(p,t)$  using the transcription-prediction system using the 3rd configuration, with the NADE-HF.

neural networks. The acoustic transcription model was based on probabilistic latent component analysis, and information from the prediction system was incorporated using Dirichlet priors. Experimental results using the multiple-instrument Bach10 dataset showed that there is a clear and significant improvement (3% in terms of F-measure) by combining a music language model with an acoustic model for improving the performance of the latter. These results also demonstrate that the MLM can be trained on symbolic music data from a different source as the acoustic data, thus eliminating the need to acquire collections of symbolic and corresponding acoustic data (which are scarce).

In the current system, the language models are trained on only one dataset. In the future, we would like to evaluate the proposed system using language models trained from different sources to see if this helps the MLMs generalize better. We will also investigate different system configurations, to test whether iterating the transcription and prediction steps leads to improved performance. We will also investigate the effect of using different RNN architectures like Long Short Term Memory (LSTM) and bi-directional RNNs and LSTMs. Finally, we would like to extend the current models for high-dimensional sequences to better fit the requirements for music language modelling.



**Figure 3.** Transcription example for recording “Ach Lieben Christen” from the Bach10 dataset. (a) The post-processed output of the transcription-prediction system using the 3rd configuration, with the NADE-HF. (b) The pitch ground truth of the recording.

## 7. ACKNOWLEDGEMENT

SS is supported by a City University London Pump-Priming Grant and the Queen Mary University of London Postgraduate Research Fund. EB is supported by a City University London Research Fellowship. SC is supported by a City University London Research Studentship.

## 8. REFERENCES

- [1] Music Information Retrieval Evaluation eXchange (MIREX). <http://music-ir.org/mirexwiki/>.
- [2] E. Benetos, S. Cherla, and T. Weyde. An efficient shiftinvariant model for polyphonic music transcription. In *6th International Workshop on Machine Learning and Music*, 2013.
- [3] E. Benetos and S. Dixon. A shift-invariant latent variable model for automatic music transcription. *Computer Music Journal*, 36(4):81–94, 2012.
- [4] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, December 2013.
- [5] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu. Advances in optimizing recurrent networks. In *ICASSP*, pages 8624–8628, May 2013.
- [6] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks*, 5(2):157–166, 1994.
- [7] N. Bertin, R. Badeau, and E. Vincent. Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Trans. Audio, Speech, and Language Processing*, 18(3):538–549, March 2010.
- [8] S. Böck and M. Schedl. Polyphonic piano note transcription with recurrent neural networks. In *ICASSP*, pages 121–124, March 2012.
- [9] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *29th Int. Conf. Machine Learning*, 2012.
- [10] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. High-dimensional sequence transduction. In *ICASSP*, pages 3178–3182, May 2013.
- [11] A. T. Cemgil. *Bayesian Music Transcription*. PhD thesis, Radboud University of Nijmegen, 2004.
- [12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.
- [13] Z. Duan, B. Pardo, and C. Zhang. Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Trans. Audio, Speech, and Language Processing*, 18(8):2121–2133, November 2010.
- [14] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: music genre database and musical instrument sound database. In *ISMIR*, Baltimore, USA, October 2003.
- [15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [16] H. Jaeger. Adaptive nonlinear system identification with echo state networks. In *Advances in neural information processing systems*, pages 593–600, 2002.
- [17] H. Larochelle and I. Murray. The neural autoregressive distribution estimator. *Journal of Machine Learning Research*, 15:29–37, 2011.
- [18] D. D. Li and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, October 1999.
- [19] J. Martens and I. Sutskever. Learning recurrent neural networks with Hessian-free optimization. In *28th Int. Conf. Machine Learning*, pages 1033–1040, 2011.
- [20] J. Nam, J. Ngiam, H. Lee, and M. Slaney. A classification-based polyphonic piano transcription approach using learned feature representations. In *ISMIR*, pages 175–180, October 2011.
- [21] L. Rabiner and B.-H. Juang. *Fundamentals of speech recognition*. 1993.
- [22] S.A. Raczynski, E. Vincent, and S. Sagayama. Dynamic Bayesian networks for symbolic polyphonic pitch modeling. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(9):1830–1840, 2013.
- [23] C. Schörkhuber and A. Klapuri. Constant-Q transform toolbox for music processing. In *7th Sound and Music Computing Conf.*, Barcelona, Spain, July 2010.
- [24] P. Smaragdis and G. Mysore. Separation by “humming”: user-guided sound extraction from monophonic mixtures. In *IEEE WASPAA*, pages 69–72, October 2009.
- [25] P. Smaragdis, B. Raj, and M. Shashanka. A probabilistic latent variable model for acoustic modeling. In *Neural Information Processing Systems Workshop*, Whistler, Canada, December 2006.
- [26] I. Sutskever, G. E. Hinton, and G. W. Taylor. The recurrent temporal restricted Boltzmann machine. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2008.
- [27] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

# TOWARDS MODELING TEXTURE IN SYMBOLIC DATA

Mathieu Giraud

LIFL, CNRS

Univ. Lille 1, Lille 3

Florence Levé

MIS, UPJV, Amiens

LIFL, Univ. Lille 1

Florent Mercier

Univ. Lille 1

Marc Rigaudière

Univ. Lorraine

Donatien Thorez

Univ. Lille 1

{mathieu, florence, florent, marc, donatien}@algonus.fr

## ABSTRACT

Studying *texture* is a part of many musicological analyses. The change of texture plays an important role in the cognition of musical structures. Texture is a feature commonly used to analyze musical audio data, but it is rarely taken into account in symbolic studies. We propose to formalize the texture in classical Western instrumental music as melody and accompaniment layers, and provide an algorithm able to detect homorhythmic layers in polyphonic data where voices are not separated. We present an evaluation of these methods for parallel motions against a ground truth analysis of ten instrumental pieces, including the first movements of the six quatuors op. 33 by Haydn.

## 1. INTRODUCTION

### 1.1 Musical Texture

According to Grove Music Online, texture refers to *the sound aspects of a musical structure*. One usually differentiates *homophonic* textures (rhythmically similar parts) and *polyphonic* textures (different layers, for example melody with accompaniment or contrapuntal parts). Some more precise categorizations have been proposed, for example by Rowell [17, p. 158 – 161] who proposes eight “textural values”: orientation (vertical / horizontal), tangle (interweaving of melodies), figuration (organization of music in patterns), focus vs. interplay, economy vs. saturation, thin vs. dense, smooth vs. rough, and simple vs. complex. What is often interesting for the musical discourse is the *change of texture*: J. Dunsby, recalling the natural tendency to consider a great number of categories, asserts that “one has nothing much to say at all about texture as such, since all depends on what is being compared with what” [5].

*Orchestral texture*. The term *texture* is used to describe orchestration, that is the way musical material is laid out on different instruments or sections, taking into account registers and timbres. In his 1955 *Orchestration* book, W. Piston presents seven types of texture: orchestral unison, melody and accompaniment, secondary melody, part writing, contrapuntal texture, chords, and complex textures [15].

In 1960, Q. R. Nordgren [13] asks: “*Is it possible to measure texture?*”. He proposes to quantify the horizontal and vertical relationships of sounds making up the texture beyond the usual homophonic/polyphonic or light/heavy categories. He considers eight features, giving them numerical values: the number of instruments, their range, their register and their spacing, the proportion and register of gap, and doubling concentrations with their register. He then analyzes eight symphonies by Beethoven, Mendelssohn, Schumann and Brahms with these criteria, finding characteristic differences between those composers.

*Non-orchestral texture*. However, the term texture also relates to music produced by a smaller group of instruments, even of same timbre (such as a string quartet), or to music produced by a unique polyphonic instrument such as the piano or the guitar. As an extreme point of view, one can consider texture on a monophonic instrument: a simple monophonic sequence of notes can sound as a melody, but also can figure accompaniment patterns such as arpeggiated chords or Alberti bass.

*Texture in musical analysis*. Studying texture is a part of any analysis, even if texture often does not make sense on its own. As stated by J. Levy, “*although it cannot exist independently, texture can make the functional and sign relationships created by the other variables more evident and fully effective*” [10]. Texture plays a significant role in the cognition of musical structures. J. Dunsby attributes two main roles to texture: the illusion it creates and the expectation it arouses from the listeners towards familiar textures [5]. J. Levy shows with many examples how texture can be a sign in Classic and Early Romantic music, describing the role of accompaniment patterns, solos and unison to raise the attention of the listener before important structural changes [10].

### 1.2 Texture and Music Information Retrieval

Texture was often not as deeply analyzed and formalized as other parameters (especially melody or harmony). In the field of Music Information Retrieval (MIR), the notion of texture is often used in audio analysis, reduced to timbral description. Any method dealing with audio signals is somewhat dealing with timbre and texture [3, 9]. Based on audio texture, there were for example studies on segmentation. More generally, the term “sound texture” can be used to describe or synthesize non-instrumental audio signals, such as ambient sounds [18, 19].



© Mathieu Giraud, Florence Levé, Florent Mercier, Marc Rigaudière, Donatien Thorez. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Mathieu Giraud, Florence Levé, Florent Mercier, Marc Rigaudière, Donatien Thorez. “Towards modeling texture in symbolic data”, 15th International Society for Music Information Retrieval Conference, 2014.

Among the studies analyzing scores represented by symbolic data, few of them take texture into account. In 1989, D. Huron [7] explains that the three common meanings about the texture term are the volume, the diversity of elements used and the “surface” description, the first two being more easily formalizable. Using a two-dimensional space based on onset synchronization and similar pitch motion, he was able to capture four broad categories of textures: monophony, homophony, polyphony and heterophony. He found also that different musical genres occupy a different region of the defined space.

Some of the features of the jSymbolic library, used for classification of MIDI files, concern musical texture [11, 12]. “[They] relate specifically to the number of independent voices in a piece and how these voices relate to one another.” [11, p. 209]. The features are computed on MIDI files where voices are separated, and include statistical features on choral or orchestral music organization: maximum, average and variability of the number of notes, variability between features of individual voices (number of notes, duration, dynamics, melodic leaps, range), features of the loudest voice, highest and lowest line, simultaneity, voice overlap, parallel motion and pitch separation between voices.

More recently, Tenkanen and Gualda [20] detect articulative boundaries in a musical piece using six features including pitch-class sets and onset density ratios. D. Rafailidis and his colleagues segment the score in several textural streams, based on pitch and time proximity rules [2, 16].

### 1.3 Contents

As we saw above, there are not many studies on modeling or automatic analysis of texture. Even if describing musical texture could be done on a local level of a score, it requires some high-level musical understanding. We thus think that it is a natural challenge, both for music modeling and for MIR studies.

In this paper, we propose some steps towards the modeling and the computational analysis of texture in Western classical instrumental music. We choose here not to take into account orchestration parameters, but to focus on textural features given by local note configurations, taking into account the way these may be split into several layers. For the same reason, we do not look at harmony or at motives, phrases, or pattern large-scale repetition.

The following section presents a formal modeling of the texture and a ground truth analysis of first movements of ten string quartets. Then we propose an algorithm discovering texture elements in polyphonic scores where voices are not separated, and finally we present an evaluation of this algorithm and a discussion on the results.

## 2. FORMALIZATION OF TEXTURE

### 2.1 Modeling Texture as Layers

We choose to model the texture, by grouping notes into sets of “layers”, also called “streams”, sounding as a whole

grouped by perceptual characteristics. Auditory stream segregation was introduced by Bregman, who studied many parameters influencing this segregation [1]. Focusing on the information contained on a symbolic score, notes can be grouped in such layers using perceptual rules [4, 16]. The number of layers is not directly the number of actual (monophonic) voices played by the instruments. For instance, in a string quartet where all instruments are playing, there can be as few as only one perceived layer, several voices blending in homorhythmy. On the contrary, some figured patterns in a unique voice can be perceived as several layers, as in a Alberti bass.

More precisely, *we model the texture in layers according to two complementary views*. First, we consider two main roles for the layers, that is how they are perceived by the listeners: *melodic* (mel) layers (dominated by contiguous pitch motion), and *accompaniment* (acc) layers (dominated by harmony and/or rhythm). Second, we describe how each layer is composed.

- A melodic layer can be either a monophonic voice (solo), or two or more monophonic voices in homorhythmy (h), or within a tighter relation, such as (from most generic to most similar) parallel motion (p), octave (o) or unison (u) doubling. The h/p/o/u relations do not need to be exact: for example, a parallel motion can be partly in thirds, partly in sixths, and include some foreign notes (see Figure 1).
- An accompaniment layer can also be described by h/p/o/u relations, but it is often worth focusing on its rhythmic component: for example, such a layer can contain sustained, sparse or repeated chords, Alberti bass, pedal notes, or syncopation.

The usual texture categories can then be described as:

- **mel/acc** – the usual accompanied melody;
- **mel/mel** – two independent melodies (counterpoint, imitation...);
- **mel** – one melody (either solo, or several voices in h/p/o/u relation), no accompaniment;
- **acc** – only accompaniment, when there is no noticeable melody that can be heard (as in some transitions for example).

The formalism also enables to describe more layers, such as mel/mel/mel/mel, acc/acc, or mel/acc/acc.

*Limitations.* This modeling of texture is often ambiguous, and has limitations. The distinction between melody and accompaniment is questionable. Some melodies can contain repeated notes, arpeggiated motives, and strongly imply some harmony. Limiting the role of the accompaniment to harmony and rhythm is also over-simplified. Moreover, some textural gestures are not modeled here, such as upwards or downwards scales. Finally, what Piston calls “complex textures” (and what is perhaps the most interesting), interleaving different layers [15, p. 405], can not

1,	75	: mel/acc (SAp / TB)
8,	82	: mel/acc (SA / TBp)
9,	83	: mel/acc (SAp / T syncopation, B)
13,	87	: mel/acc (SAp / TB imitation)
19,	93	: mel/acc (SAo / TBh)
20,	94	: mel/acc (SAp / TBhr)
21,	95	: imitation/acc (SA / TB)
25,	99	: imitation/acc (SA / TB)
	102	: mel/acc (S / ATB)
29,	103	: mel/acc (S / ATBhr)
30		: mel/acc (S / ATBh)
	104	: mel/acc (SAh/ TBh)
31,	105	: mel/acc (S / ATBh sparse chords)
33,	107	: acc/mel/acc (S / ATp / B)
...		

**Figure 1.** Beginning of the string quartet K. 157 no. 4 by W. A. Mozart, with the ground truth analysis describing textures. We label as S / A / T / B (soprano / alt / tenor / bass) the four instruments (violin I / violin II / viola / cello). The first eight measures have a melodic layer “SAp” made by a parallel motion (with thirds), however the parallel motion has some exceptions (unison on *c*, strong beat on measures 1 and 8, and small interruption at the beginning of measure 5).

always be modeled by this way. Nevertheless, the above formalization is founded for most music of the Classical and of the Romantic period, and corresponds to a way of melody/accompaniment writing.

## 2.2 A Ground Truth for Texture

We manually analyzed the texture on 10 first movements of string quartets: the six quartets op. 33 by Haydn, three early quartets by Mozart (K. 80 no. 1, K. 155 no. 2 and K. 157 no. 4), and the quartet op. 125 no. 1 by Schubert. These pieces covered the textural features we wanted to elucidate. We segmented each piece into non-overlapping segments based only on texture information, using the formalism described above.

It is difficult to agree on the signifiacnce on short segments and on their boundaries. Here we choose to report the texture with a resolution of *one measure*: we consider only segments during at least one measure (or filling the most part of the measure), and round the boundaries of these segments to bar lines.

We identified 691 segments in the ten pieces, and Table 1 details the repartition of these segments. The ground truth file is available at [www.algomus.fr/truth](http://www.algomus.fr/truth), and Figure 1 shows the analysis for the beginning of the string quartet K. 157 no. 4 by Mozart.

The segments are further precised by the involved voices and the h/p/o/u relations. For example, focusing on the most represented category “mel/acc”, there are 254 segments labelled either “S / ATB” or “S / ATBh” (melodic layer at the first violin) and 81 segments labelled “SAp / TB” or “SAp / TBh” (melodic layer at the two violins, in a parallel move). Note that h/p/o/u relations were evaluated here in a subjective way. The segments may contain some small interruptions that do not alter the general perception of the h/p/o/u relation.

## 3. DISCOVERING SYNCHRONIZED LAYERS

We now try to provide a computational analysis of texture starting from a polyphonic score where voices are not separated. A first idea is to *first segment the score into*

*layers by perception principles, and then to try to qualify some of these layers.* One can for example use the algorithm of [16] to segment the musical pieces into layers (called “streams”). This algorithm relies on a distance matrix, which tells for each possible pair of notes whether they are likely to belong to the same layer. The distance between two notes is computed according to their synchronicity, pitch and onset proximity (among others criteria); then for each note, the list of its *k*-nearest neighbors is established. Finally, notes are gathered in clusters. A melodic stream can be split into several small chunks, since the diversity of melodies does not always ensure coherency within clusters; working on larger layers encompass them all. Even if this approach produces good results in segmentation, many layers are still too scattered to be detected as full melodic or accompaniment layers. Nonetheless, classification algorithms could label some of these layers as melodies or accompaniments, or even detect the type of the accompaniment.

The second idea, that we will develop in this paper, is to *detect directly noteworthy layers from the polyphonic data.* Here, we choose to focus on perceptually significant relations based on homorhythmic features. The following paragraphs define the notion of *synchronized layers*, that is sequences of notes related by some homorhythmy relation (h/p/o/u), and show how to compute them.

### 3.1 Definitions: Synchronized Layers

A *note*  $n$  is given as a triplet  $(n.pitch, n.start, n.end)$ , where  $n.pitch$  belongs to a pitch scale (that can be defined diatonically or by semitones), and  $n.start$  and  $n.end$  are two positions with  $n.start < n.end$ . Two notes  $n$  and  $m$  are *synchronized* (denoted by  $n \equiv_h m$ ) if they have the same start and the same end.

A *synchronized layer* (SL) is a set of two sequences of consecutive synchronized notes (in other words, these sequences correspond to two “voices” in homorhythmy). Formally, two sequences of notes  $n_1, n_2 \dots n_k$  and  $m_1, m_2 \dots m_k$  form a synchronized layer when:

- for all  $i$  in  $\{1, \dots, k\}$ ,  $n_i.start = m_i.start$

	tonality	length	mel/acc	mel/mel	acc/mel/acc	acc/mel	mel	acc	others	h	p	o	u
Haydn op. 33 no. 1	B minor	91m	38	0	8	1	0	0	5	19	21	1	0
Haydn op. 33 no. 2	E-flat major	95m	37	0	2	4	0	0	7	34	13	0	0
Haydn op. 33 no. 3	C major	172m	68	0	0	0	3	13	6	29	50	1	0
Haydn op. 33 no. 4	B-flat major	90m	25	0	1	0	0	0	6	16	6	0	0
Haydn op. 33 no. 5	G major	305m	68	0	3	4	7	0	5	56	45	6	0
Haydn op. 33 no. 6	D major	168m	58	0	1	3	15	0	29	43	42	0	2
Mozart K. 80 no. 1	G major	67m	36	4	6	0	2	0	3	5	33	3	0
Mozart K. 155 no. 2	D major	119m	51	0	0	0	1	0	0	21	32	4	1
Mozart K. 157 no. 4	C major	126m	29	0	3	6	2	0	7	18	22	2	0
Schubert op. 125 no. 1	E-flat major	255m	102	0	0	0	20	2	0	54	8	46	2
		1488m	512	4	24	18	50	15	68	295	272	63	5

**Table 1.** Number of segments in the ground truth analysis of the ten string quartets (first movements), and number of h/p/o/u labels further describing these layers.

- for all  $i$  in  $\{1, \dots, k\}$ ,  $n_i.end = m_i.end$
- for all  $i$  in  $\{1, \dots, k-1\}$ ,  $n_i.end = n_{i+1}.start$

This definition can be extended to any number of voices. As p/o/u relations have a strong musical signification, we want to be able to enforce them. One can thus restrain the relation  $\equiv_h$ , considering the pitch information:

- we denote  $n \equiv_\delta m$  if the interval between the two notes  $n$  and  $m$  is  $\delta$ . The nature of the interval  $\delta$  depends on the pitch model: for example, the interval can be diatonic, such as in “third” (minor or major), or an approximation over the semitone information, such as in “3 or 4 semitones”. Some synchronized layers with  $\equiv_\delta$  relations correspond to parallel motions;
- we denote  $n \equiv_o m$  if notes  $n$  and  $m$  are separated by any number of octaves;
- we denote  $n \equiv_u m$  where there is an exact equality of pitches (unison).

Given a relation  $\equiv \in \{\equiv_h, \equiv_\delta, \equiv_o, \equiv_u\}$ , we say that a synchronized layer respects the relation  $\equiv$  if its notes are pairwise related according to this relation. The relation  $\equiv_h$  is an equivalence relation, but the restrained relations do not need to be equivalence relations: Some  $\equiv_\delta$  relations are not transitive.

For example, in Figure 1, there is between voices S and A (corresponding to violins I and II), in the first two measures:

- a synchronized layer ( $\equiv_h$ ) on the two measures;
- and a synchronized layer ( $\equiv_{\text{third}}$ ) on the two measures, except the first note.

Note that this does not correspond exactly to the “musical” ground truth (parallel move on at least the first four measures) because of some rests and of the first synchronized notes that are not in thirds.

A synchronized layer is *maximal* if it is not strictly included in another synchronized layer. Note that two maximal synchronized layers can be overlapping, if they are not synchronized. Note also that the number of synchronized layers may grow exponentially with the number of notes.

### 3.2 Detection of a Unique Synchronized Layer

A very noticeable textural effect is when *all* voices use the same texture at the same time. For example, a sudden striking unison raises the listener’s attention. We can first check if all notes in a segment of the score belong to a unique synchronized layer (within some relation). For example, we consider that all voices are in octave doubling or unison if it lasts at least two quarters.

### 3.3 Detection of Maximal Synchronized Layers

In the general case, the texture has several layers, and the goal is thus to extract layers using *some* of the notes. Remember that we work on files where polyphony is not separated into voices: moreover, it is not always possible to extract voices from a polyphonic score, for example on piano music. We want to extract maximal synchronized layers. However, as their number may grow exponentially with the number of notes, we will compute only the start and end positions of maximal synchronized layers.

The algorithm is a kind of 1-dimension interval chaining [14]. The idea is as follows. Recursively, two voices  $n_1, \dots, n_k$  and  $m_1, \dots, m_k$  are synchronized if and only if  $n_1, \dots, n_{k-1}$  and  $m_1, \dots, m_{k-1}$  are synchronized,  $n_k$  and  $m_k$  are synchronized and finally  $n_{k-1}.end = n_k.start$ . Formally, the algorithm is described by the following:

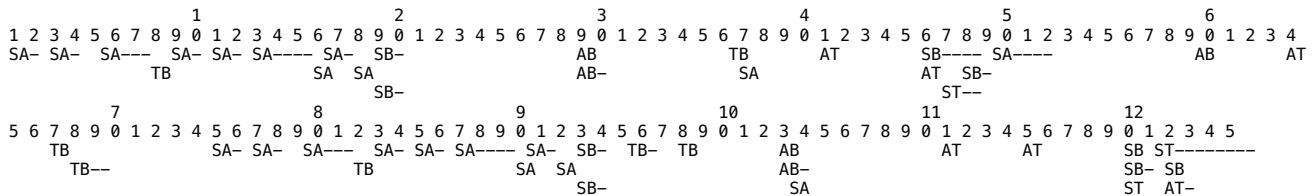
**Step 1.** Compute a table with left-maximal SL. Build the table  $leftmost\_start_{\equiv}[j]$  containing, for each ending position  $j$ , the leftmost starting position of a SL respecting  $\equiv$  ending in  $j$ . This can be done by dynamic programming with the following recurrence:

$$leftmost\_start_{\equiv}[j] = \begin{cases} \min\{leftmost\_start_{\equiv}[i] \mid i \in S_{\equiv}(j)\} & \text{if } S_{\equiv}(j) \text{ is not empty} \\ j & \text{if } S_{\equiv}(j) \text{ is empty} \end{cases}$$

where  $S_{\equiv}(j)$  is the set of all starting positions of synchronized notes ending at  $j$  respecting the relation  $\equiv$ :

$$S_{\equiv}(j) = \left\{ n.start \mid \begin{array}{l} \text{there are two different notes } n \equiv m \\ \text{such that } n.end = j \end{array} \right\}$$

**Step 2.** Output only (left and right) maximal SL. Output  $(i, j)$  with  $i = leftmost\_start_{\equiv}[j]$  for each  $j$ , such that  $j = \max\{j_o \mid leftmost\_start_{\equiv}[j_o] = leftmost\_start_{\equiv}[j]\}$



**Figure 2.** Result on the parallel move detection on the first movement of the string quartet K. 157 no. 4 by Mozart. The top lines display the measure numbers. The algorithm detects 52 synchronized layers respecting the  $\equiv_p$  relation. 39 of these 52 layers overlap layers identified in the truth with p/o/u relations. The parallel motions are further identified by their voices (S / A / T / B), but this information is not used in the algorithm which works on non-separated polyphony.

The first step is done in  $O(nk)$  time, where  $n$  is the number of notes and  $k \leq n$  the maximal number of simultaneously sounding notes, so in  $O(n^2)$  time. The second step is done in  $O(n)$  time by browsing from right to left the table  $leftmost\_start_{\equiv}$ , outputting values  $i$  when they are seen for the first time.

To actually retrieve intervals, we can store in the table  $leftmost\_start_{\equiv}[j]$  a pair  $(i, \ell)$ , where  $\ell$  is the list of notes/intervals from which the set of SL can be built (this set may be very large, but not  $\ell$ ). The time complexity is now  $O(n(k+w))$ , where  $w \leq n$  is the largest possible size of  $\ell$ . Thus the time complexity is still in  $O(n^2)$ . This allows, in the second step, to filter the SL candidates according to additional criteria on  $\ell$ .

Note finally that the definition of synchronized layer can be extended to include consecutive notes separated with rests. The same algorithm still applies, but the value of  $k$  rises to the maximum number of notes that can be linked in that way.

#### 4. RESULTS AND DISCUSSION

We tested the proposed algorithm to look for synchronized layers respecting  $\equiv_{\delta}$  relation (constant pitch interval, including parallel motion) on the ten pieces of our corpus given as .krn Humdrum files [8]. Although the pieces are string quartets, we consider them as non-separated polyphonic data, giving as input to the algorithm a single set of notes. The algorithm finds 434 layers. Figure 2 shows an example of the output of the algorithm. Globally, on the corpus, the algorithm labels 797 measures (that is 53.6% of the length) as synchronized layers.

*Evaluation against the truth.* There are in the truth 354 layers with p/o/u relations: mainly parallel moves, and some octave doubling and unisons. As discussed earlier, these layers reported in the truth correspond to a musical interpretation: they are not as formalized as our definition of synchronized layer. Moreover, less information is provided by the algorithm than in the ground truth: when a parallel motion is found, the algorithm cannot provide at which voice/instrument it appears, since we worked from polyphonic data with no voice separation.

Nevertheless, we compared the layers predicted by the algorithms with the ones of the truth. Results are summarized on Table 2. A computed layer is marked as true positive (TP) as soon as it overlaps a p/o/u layer of the truth.

356 of the 434 computed synchronized layers are overlapping the p/o/u layers of the truth, thus 82.0% of the computed synchronized layers are (at least partially) musically relevant. These 356 layers map to 194 p/o/u layers in the truth (among 340, that is a sensitivity of 58.0%): a majority of the parallel moves described in the truth are found by the algorithm.



**Figure 3.** Haydn, op. 33 no. 6, m. 28-33. The truth contains four parallel moves.

*Merged parallel moves.* If one restricts to layers where borders coincide with the ones in the truth (same start, same end, with a tolerance of 2 quarters), the number of truth layers found falls from 194 to 117. This is because the algorithm often merge consecutive parallel moves. An example of this drawback is depicted on Figure 3. Here a melody is played in imitation, resulting in parallel moves involving all voices in turn. The algorithm detects a unique synchronized layer, which corresponds to a global perception but gives less information about the texture. We should remember here that the algorithm compute boundaries of synchronized layers and not actual instances, which would require some sort of voice separation and possibly generate a large number of instances.

*False positives.* Only 78 false positives are found by the algorithm. Many false positives (compared to the truth) are parallel moves detected inside a homorhythmy  $\equiv_h$  relation between 3 or 4 voices. In particular, the algorithm detects a parallel move as soon as there are sequences of repeated notes in at least two voices. This is the case in in op. 33 no. 4 by Haydn which contains many homorhythmies in repeated notes, for which we obtain 30 false positives. Even focusing on layers with a real “move”, false positive could also appear between a third voice and two voices with repeated notes. Further research should be carried to discard these false positives either in the algorithm or at a later filtering stage.

	hits length	hits	TP	FP	truth-overlap	truth-exact
Haydn op. 33 no. 1	40m (44%)	37	32 (86.5%)	5	14/22 (63.6%)	7/22
Haydn op. 33 no. 2	21m (22%)	17	15 (88.2%)	2	7/13 (53.9%)	7/13
Haydn op. 33 no. 3	73m (42%)	48	44 (91.7%)	4	27/51 (52.9%)	15/51
Haydn op. 33 no. 4	19m (21%)	47	17 (36.2%)	30	5/6 (83.3%)	3/6
Haydn op. 33 no. 5	235m (77%)	58	47 (81.0%)	11	27/51 (52.9%)	11/51
Haydn op. 33 no. 6	63m (37%)	24	21 (87.5%)	3	19/44 (43.2%)	11/44
Mozart K. 80 no. 1	45m (67%)	27	26 (96.3%)	1	20/36 (55.6%)	14/36
Mozart K. 155 no. 2	76m (64%)	46	44 (95.7%)	2	27/37 (73.0%)	15/37
Mozart K. 157 no. 4	62m (49%)	52	39 (75.0%)	13	15/24 (62.5%)	8/24
Schubert op. 125 no. 1	163m (64%)	78	71 (91.0%)	7	33/56 (58.9%)	20/56
	797m (54%)	434	356 (82.0%)	78	194/340 (57.1%)	111/340

**Table 2.** Evaluation of the algorithm on the ten string quartets of our corpus. The columns TP and FP show respectively the number of true and false positives, when comparing computed parallel moves with the truth. The columns truth-overlap shows the number of truth parallel moves that were matched by this way. The column truth-exact restricts these matchings to computed parallel moves for which borders coincide to the ones in the truth (tolerance: two quarters).

## 5. CONCLUSION AND PERSPECTIVES

We proposed a formalization of texture in Western classical instrumental music, by describing melodic or accompaniment “layers” with perceptive features (h/p/o/u relations). We provided a first algorithm able to detect some of these layers inside a polyphonic score where tracks are not separated, and tested it on 10 first movements of string quartets. The algorithm detects a large part of the parallel moves found by manual analysis. We believe that other algorithms implementing textural features, beyond h/p/o/u relations, should be designed to improve computational music analysis. The corpus should also be extended, for example with music from other periods or piano scores.

Finally, we believe that this search of texture, combined with other elements such as patterns and harmony, will improve algorithms for music structuration. The ten pieces of our corpus have a *sonata form* structure. The tension created by the exposition and the development is resolved during the recapitulation, and textural elements contribute to this tension and its resolution [10]. For example, the medial caesura (MC), before the beginning of theme S, has strong textural characteristics [6]. Textural elements predicted by algorithms could thus help the structural segmentation.

## 6. REFERENCES

- [1] A. S. Bregman. *Auditory scene analysis*. Bradford, Cambridge, 1990.
- [2] E. Cambouropoulos. Voice separation: theoretical, perceptual and computational perspectives. In *Int. Conf. on Music Perception and Cognition (ICMPC)*, 2006.
- [3] R. B. Dannenberg and M. Goto. *Handbook of Signal Processing in Acoustics*, chapter Music Structure Analysis, pages 305–331. Springer, 2008.
- [4] D. Deutsch and J. Feroe. The internal representation of pitch sequences in tonal music. *Psychological Review*, 88(6):503–522, 1981.
- [5] J. M. Dunsby. Considerations of texture. *Music and letters*, 70(1):46–57, 1989.
- [6] J. Hepokoski and W. Darcy. The medial caesura and its role in the eighteenth-century sonata exposition. *Music Theory Spectrum*, 19(2):115–154, 1997.
- [7] D. Huron. Characterizing musical textures. In *Int. Computer Music Conf. (ICMC)*, pages 131–134, 1989.
- [8] D. Huron. Music information processing using the Humdrum toolkit: Concepts, examples, and lessons. *Computer Music J.*, 26(2):11–26, 2002.
- [9] A. Klapuri and M. Davy. *Signal Processing Methods for Music Transcription*. Springer, 2006.
- [10] J. M. Levy. Texture as a sign in classic and early romantic music. *J. of the American Musicological Society*, 35(3):482–531, 1982.
- [11] C. McKay. *Automatic music classification with jMIR*. PhD thesis, McGill University, 2010.
- [12] C. McKay and I. Fujinaga. jSymbolic: A feature extractor for MIDI files. In *Int. Computer Music Conf. (ICMC)*, pages 302–305, 2006.
- [13] Q. R. Nordgren. A measure of textural patterns and strengths. *J. of Music Theory*, 4(1):19–31, Apr. 1960.
- [14] E. Ohlebusch and M. I. Abouelhoda. *Handbook of Computational Molecular Biology*, chapter Chaining Algorithms and Applications in Comparative Genomics. 2005.
- [15] W. Piston. *Orchestration*. Norton, 1955.
- [16] D. Rafailidis, A. Nanopoulos, Y. Manolopoulos, and E. Cambouropoulos. Detection of stream segments in symbolic musical data. In *Int. Conf. on Music Information Retrieval (ISMIR)*, pages 83–88, 2008.
- [17] L. Rowell. *Thinking about Music: An Introduction to the Philosophy of Music*. Univ. of Massachusetts, 1985.
- [18] N. Saint-Arnaud and K. Popat. Computational auditory scene analysis. chapter Analysis and Synthesis of Sound Textures, pages 293–308. Erlbaum, 1998.
- [19] G. Strobl, G. Eckel, and D. Rocchesso. Sound texture modeling: a survey. In *Sound and Music Computing (SMC)*, 2006.
- [20] A. Tenkanen and F. Gualda. Detecting changes in musical texture. In *Int. Workshop on Machine Learning and Music*, 2008.



# COMPUTATIONAL MODELS FOR PERCEIVED MELODIC SIMILARITY IN A CAPPELLA FLAMENCO SINGING

**N. Kroher, E. Gómez**

Universitat Pompeu  
Fabra

emilia.gomez  
@upf.edu,  
nadine.kroher  
@upf.edu

**C. Guastavino**

McGill University  
& CIRMMT

catherine.guastavino  
@mcgill.ca

**F. Gómez**

Technical University  
of Madrid

fmartin  
@eui.upm.es

**J. Bonada**

Universitat Pompeu  
Fabra

jordi.bonada  
@upf.edu

## ABSTRACT

The present study investigates the mechanisms involved in the perception of melodic similarity in the context of a cappella flamenco singing performances. Flamenco songs belonging to the same style are characterized by a common melodic skeleton, which is subject to spontaneous improvisation containing strong prolongations and ornamentations. For our research we collected human similarity judgements from naïve and expert listeners who listened to audio recordings of a cappella flamenco performances as well as synthesized versions of the same songs. We furthermore calculated distances from manually extracted high-level descriptors defined by flamenco experts. The suitability of a set of computational melodic similarity measures was evaluated by analyzing the correlation between computed similarity and human ratings. We observed significant differences between listener groups and stimuli types. Furthermore, we observed a high correlation between human ratings and similarities computed from features from flamenco experts. We also observed that computational models based on temporal deviation, dynamics and ornamentation are better suited to model perceived similarity for this material than models based on chroma distance.

## 1. INTRODUCTION

The task of modeling perceived melodic similarity among music pieces is a multi-dimensional task whose complexity increases when human judgements are influenced by implicit knowledge about genre-specific musicological aspects and contextual information. Nevertheless, such computational models are of utmost importance for automatic similarity retrieval and recommendation systems in large music databases. Furthermore, analysis of melodic sim-

ilarity among large amounts of data can provide important clues for musicological studies regarding style classification, similarity and evolution. In the past, numerous approaches have focused on melodic similarity measures, mainly computed from automatically aligned score-like representations. For a complete review of symbolic note similarity measures we refer the reader to [1]. Several previous studies have related computational measures to human ratings. In an extensive study in [14], expert ratings of similarity between western pop songs and generated variants were compared to 34 computational measures. The best correlation was observed for a hybrid method combining various weighted distance measures, which is successfully used to automatically retrieve variants of a given melody from a folk song database. In similar studies, human similarity ratings were compared to transportation distances [16] and statistical descriptors related to tone, interval and note duration distribution [17]. In order to gain a deeper insight into the perception process of melodic similarity, Volk and van Kranenburg studied the relationship between musical features and human similarity-based categorization, where a large collection of folk songs was manually categorized into tune families [15]. Furthermore, human similarity judgement based on various musical facets were gathered. Results indicate that songs perceived as similar tend to show strong similarities in rhythm, pitch contour and contained melodic motifs, whereas the individual importance of these criteria varies among the data. When dealing with audio recordings for which no score is available, it seems natural to focus on the alignment and comparison of the time-frequency representation of the melodic contour. In the context of singing voice assessment, Molina et al. used *dynamic time warping* to align fundamental frequency contours and calculate melodic and rhythmic deviations between them [2].

Despite the growing interest in non-Western music traditions, most algorithms are designed and evaluated on Western commercial music. In a first genre-specific approach to melodic similarity in flamenco music, Cabrera et al. computed melodic similarity among a cappella singing performances from automatic descriptions [3]. The two standard distance measures implemented, the *edit* distance and



© N. Kroher, E. Gómez, C. Guastavino, F. Gómez, J. Bonada.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** N. Kroher, E. Gómez, C. Guastavino, F. Gómez, J. Bonada. “Computational Models for Perceived Melodic Similarity in A Cappella Flamenco Singing”, 15th International Society for Music Information Retrieval Conference, 2014.

the correlation between pitch and interval histograms, obtained rather poor results when compared to expert judgements. As proposed by Mora et al., better results for intra- and inter-style similarity can be obtained for a similarity measure based on manually extracted high-level features (i.e., the direction of melodic movement in a specific part of the performance) [4]. Such studies elucidate the need for exploration of particular characteristics of non-Western music genres and the adaptation of existing music information retrieval systems to such styles.

The present study addresses perceived melodic similarity in a cappella flamenco singing from different standpoints: with the aim of gaining insight into the mechanisms involved in perceiving melodies as more or less similar, we gathered similarity ratings among performances of the same style from naïve listeners as well as flamenco experts and analyzed them in terms of intra-subject and intra-group agreement. In order to isolate the melody from other variables such as lyrics, expression and dynamics, we gathered the same ratings for synthesized melodic contours. We furthermore evaluated three computational models for melodic similarity by analyzing the correlation between computed similarity and human ratings. We compared the results to distances computed from manually extracted high-level features defined by experts in the field. The rest of the paper is organized as follows. In Section 2 we provide background information on flamenco music and the *martinete* style, which is the focus of this study. We give a detailed description of the database used in the present experiment in Section 3. Section 4 summarizes the methodology of the listening experiments, the extracted high-level features and the implemented computational similarity models. We give the results of the correlation analysis in Section 5 and conclude our study in Section 6.

## 2. BACKGROUND

Flamenco is an oral tradition whose roots are as diverse as the cultural influences of its area of origin, Andalusia, a region in southern Spain. Its characteristics are influenced by music traditions of a variety of immigrants and colonizations that settled in the area throughout the past centuries, among them Visigoths, Arabs, Jews and to a large extend gypsies, who decisively contributed to shape the genre as we know it today. For a comprehensive and complete study on history and style, we refer to [5–7]. Flamenco germinated and nourished mainly from the singing tradition and until now the singing voice represents its central element, usually accompanied by the guitar and rhythmic hand-clapping. In the flamenco jargon, songs, but also styles, are referred to as *cantes*.

### 2.1 The flamenco singing voice

Flamenco singing performances are usually spontaneous and highly improvisational. Songs are passed from generation to generation and only rarely manually transcribed. Even though there is no distinct ideal for timbre and several



(a) Performance by Antonio Mairena



(b) Performance by Chano Lobato

**Figure 1.** Manual transcriptions of performances a *debla* “En el barrio de Triana”; Transcription: Joaquín Mora

voice types can be identified, the flamenco singing voice can be generally characterized as matt, breathy, and containing few high frequency harmonics. Moreover, singers usually lack the singer’s formant [13]. Melodic movements appear mainly in conjunct degrees within a small pitch range (*tessitura*) of a major sixth interval and are characterized by insistence on recitative notes. Furthermore, singers use a large amount melisma, microtonal ornamentation and pitch glides during note attacks [4].

### 2.2 The flamenco martinete

*Martinete* is considered one of the oldest styles and forms part of the sub-genre of the *tonás*, a group of unaccompanied singing styles, or *cantes*. As in other *cantes*, songs belonging to *martinete* style are characterized by a common melodic skeleton, which is subject to strong spontaneous ornamentation and expressive prolongations. The untrained listener might perceive two performances of the same *cante* as very different and the fact that they belong to the same style is not obvious at all. To illustrate this principle, Figure 1 shows the transcription of two a cappella performances in Western music notation, both belonging to the same style (*debla*) [4].

Furthermore, the *martinete* is characterized by a solemn performance in slow tempo with free rhythmic interpretation. Traditionally, the voice is accompanied by hammer strokes on an anvil. The tonality corresponds mainly to the major mode, whereas the third scale degree may be lowered occasionally, converting the scale to the minor mode.

## 3. MUSIC COLLECTION

In consultation with flamenco experts, we gathered 12 recordings of *martinete* performances, covering the most representative singers of this style. This dataset represents

Singer	Percussion
Antonio Mairena	No
Chano Lobato	No
Chocolate	Yes
Jacinto Almadén	No
Jesus Heredia	No
Manuel Simón	Yes
Miguel Vargas	No
Naranjito	No
Paco de Lucía	No
Talegón de Córdoba	Yes
Tomás Pavón	No
Turronero	No

**Table 1.** Dataset containing 12 *martinete* performances.

a subset of the *tonás*<sup>1</sup> dataset, which contains a total of 56 *martinete* recordings. The average duration of the extracted excerpts containing the first verse is approximately 20 seconds. We limited our study to such a small set, mainly due to the duration of the listening experiment. As an additional stimuli for the listening experiments, we furthermore created synthesized versions of all excerpts. We used the method described in [8] to extract fundamental frequency and energy envelopes and re-synthesize with a sinusoid.

We selected the first verse of each recording, containing the characteristic exposition of the melodic skeleton. Although some *martinete* recordings contain additional accompaniment (guitar, bowing string or wind instruments), we limited our selection to a cappella recordings without rhythmic accompaniment or with very sparse one, as it is found traditionally. We intentionally incorporated a wide range of interpretation characteristics, regarding richness in ornamentation, tempo, articulation and lyrics. Among the singers listed in Table 1, *Tomás Pavón* is to be mentioned as the most influential artist in the a cappella singing styles, performing the *martinete* in an exemplar way. Furthermore, *Antonio Mairena* and *Chocolate* are thought to be the main references for their singing abilities and knowledge of the singing styles. *Chano Lobato* omits some of the basic notes during the melodic exposition and the performance has been included as an example of strong deviation in the melodic interpretation.

## 4. METHODOLOGY

### 4.1 Human similarity ratings

In order to obtain a ground truth for perceived melodic similarity among the selected excerpts, we conduct a listening experiment in Montreal (Canada) with 24 naïve listeners with little or no previous exposure to flamenco and in Sevilla (Spain) with 3 experts, as described in [9]. After evaluating various experiment designs (i.e. pair-wise comparison), we decided to collect the similarity ratings in a

free sorting task [19]. Using the *sonic mapper*<sup>2</sup> software, subjects were asked to create groups of similar interpretations, leaving the number of groups open. The participants were explicitly instructed to focus on the melody only, neglecting differences in voice timbre, lyrics, percussion accompaniment and sound quality. Nevertheless, in order to isolate the melodic line as a similarity criterion, the experiment had also been conducted with the synthesized versions of the excerpts described above. For each excerpt we extracted the fundamental frequency as described in [8] with a window length of 33 ms and a hop size of 0.72 ms. The pitch contour was synthesized with a single sine wave. A similarity matrix was computed based on the number of times a pair of performances had been grouped together. We compared individual participants' similarity matrices using *Mantel* tests. The *Mantel* test can be considered as the most widely used method to account for distance correlations [12]. We used *zt*, a simple tool for *Mantel* test, developed by Bonnet and Vande Peer [18], and measured the correlation between participant matrices. We observed that the average correlation for novices is  $\mu = 0.0824$ , with a  $\sigma = 0.2109$  and the average p-value:  $\mu = 0.3391$ ,  $\sigma = 0.2139$  (min=0.002). This indicates a very low agreement among them, and indicates differences in perception of melodic similarity depending on the listener's background. Although we should take these results with caution given the small number of experts, we found higher correlation values among them, with an average correlation  $\mu = 0.1891$ , and  $\sigma = 0.1170$ . For a detailed description of the procedure and the analysis, we refer to [9].

### 4.2 Manually extracted high-level features

We manually extracted six high-level features defined by experts in the field. As illustrated above, two *cantes* having the same main notes and different ornamentation would be perceived as the same *cante* by a flamenco aficionado. This fact makes the automatic computation of the features unfeasible. Because of that, we had to rely on manual extraction.

The high-level features were the following.

1. Repetition of the first hemistich. A hemistich is half-line of a verse; the presence of this repetition is important in these *cantes*.
2. Clivis/flexa at the end of the first hemistich. A clivis is a descending melodic movement. Here it refers to a descending melodic contour between main notes. Again, the ornamentation is not taken into account when detecting the presence of the clivis.
3. Highest scale degree in the two first hemistichs. The highest scale degree reached during the *cante* is an important feature.
4. Frequency of the highest degree in the second hemistich. How many times that highest degree is reached is also significant.

<sup>1</sup> <http://mtg.upf.edu/download/datasets/tonas>

<sup>2</sup> <http://www.music.mcgill.ca/~gary/mapper/>

5. Final note of the second hemistich.
6. Duration (fast / regular / slow).

A distance matrix was obtained by calculating the Euclidean distance among the feature vectors. The feature vectors were mostly composed of categorical data and we used a standardized Euclidean distance. For a detailed explanation of the descriptors and their musicological background, the reader is referred to [4].

### 4.3 Computational similarity measures

We implemented three computational measures based on fundamental frequency envelopes and automatic transcriptions and evaluated their suitability for modeling the perceived melodic similarity by analyzing the correlation between computed distance matrices and human judgements. The fundamental frequency contours as well as the automatically generated symbolic note representations were obtained using the system described in [8].

#### 4.3.1 Dynamic time warping alignment

Similar to [2] we used a *dynamic time warping* algorithm to align melodies and estimate their rhythmic and pitch similarity. Since vocal vibrato and microtonal ornamentations strongly influence the cost matrix, we instead align continuous contours of quantized pitch values obtained with the automatic transcription described in [8]. The cost matrix  $M$  describes the squared frequency deviation between all possible combinations of time frames between the two analyzed contours  $f_{01}$  and  $f_{02}$ , where  $\alpha$  is a constraint limiting the maximum cost:

$$M_{i,j} = \min((f_{01}[i] - f_{02}[j])^2, \alpha) \quad (1)$$

The *dynamic time warping algorithm* determines the optimal path among the matrix  $M$  from first to last frame. The deviation of the slope of the path  $p$  with length  $N$  from the diagonal path gives a measure for temporal deviation ( $DTW_{temporal}$ ),

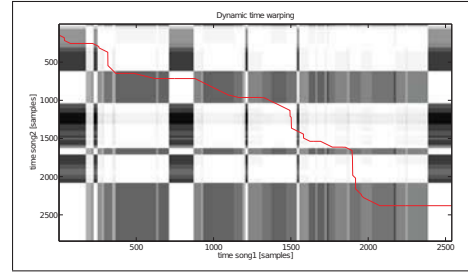
$$\Delta_{temp} = \frac{\sum_{i=1}^N (p[i] - p_{diag}[i])^2}{N} \quad (2)$$

while the average over its elements defines the pitch deviation ( $DTW_{pitch}$ ):

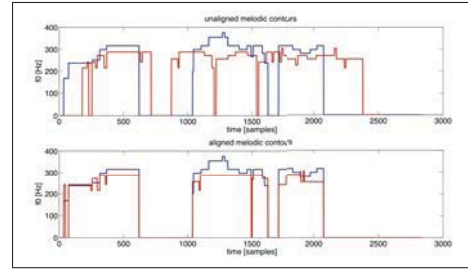
$$\Delta_{pitch} = \frac{\sum_{i=1}^N p[i]}{N}. \quad (3)$$

We used a *MATLAB* implementation<sup>3</sup>, which extends the algorithm with several restrictions in order to obtain a musically meaningful temporal alignment. Figure 2 shows the cost matrix and Figure 3 the unaligned and aligned pitch sequences.

<sup>3</sup> <http://www.ee.columbia.edu/~dpwe/resources/matlab/dtw/>



**Figure 2.** Dynamic time warping: Cost matrix and optimal path.



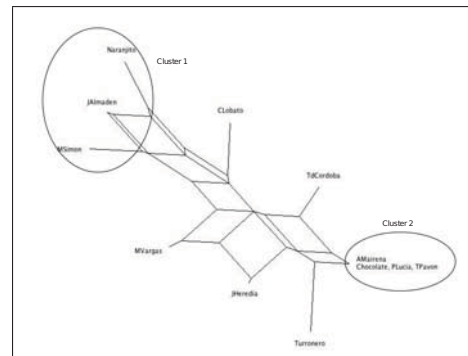
**Figure 3.** Unaligned (top) and aligned (bottom) melodic contours.

#### 4.3.2 Global performance descriptors

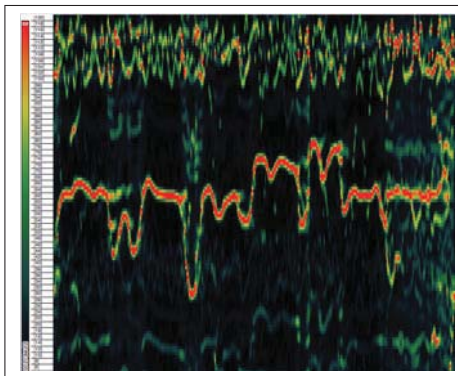
As described in [10], we extracted a total of 13 global descriptors from automatic transcriptions and computed a similarity matrix based on the Euclidean distance among feature vectors. In order to determine the most suitable descriptors for this task, we analyzed the *phylogenetic tree* (Figure 4) computed from the distance matrix of expert similarity ratings. Here, we identify two main clusters, at large distance from each other.

Using these two clusters as classes in a classification task, we perform a support vector machine (SVM) subset selection in order to identify the descriptors that are best suited to distinguish the two clusters. We accordingly extracted the six best ranked descriptors for all songs and computed the similarity matrix from the Euclidean distances among feature vectors. The extracted descriptors are summarized below:

1. **Amount of silence:** Percentage of silent frames.



**Figure 4.** Phylogenetic tree generated from expert similarity judgements.



**Figure 5.** Harmonic pitch class profile for a sung phrase with a resolution of 12 bins per semitone.

2. **Average note duration** in seconds.
3. **Note duration fluctuation:** Standard deviation of the note duration in seconds.
4. **Average volume** of the notes relative to the normalized maximum.
5. **Volume fluctuation:** standard deviation of the note volume relative to normalized maximum.
6. **Amount of ornamentation:** Average per-frame distance in [Hz] between the quantized note value and the fundamental frequency contour.

#### 4.3.3 Chroma similarity

We implemented a similarity measure presented in [11] in the context of cover identification: First, the harmonic pitch class profiles (HPCP) are extracted on a global and a frame basis. The resulting pitch class histogram describes the relative strength of the 12 pitch classes of the equal-tempered scale. HPCPs are robust to detuning as well as variation in timbre and dynamics. After adjusting the key of one sequence to the other, a binary similarity matrix is computed based on the frame-wise extracted HPCPs. Again, dynamic time warping was used to find the best possible path among the similarity matrix. For a detailed description of the algorithm, we refer the reader to [11].

#### 4.4 Evaluation

We evaluated the suitability of the computational models for this task by analyzing the correlation between computed similarity and human ratings. A common method to evaluate a possible relation between two distance matrices is the Mantel test [12]: first, the linear correlation between two matrices is measured with the *Pearson correlation*, which gives a value  $r$  between -1 and 1. A strong correlation is indicated by a value significantly different from zero. To verify that a relation exists, the value is compared to correlations to permuted versions of the matrices. Here, 10000 random permutations are performed. The *confidence value*  $p$  corresponds to the proportion of permutations giving a higher correlation than the original matrix.

Consequently, a confidence value close to zero confirms an existing correlation.

## 5. RESULTS

Figure 6 shows the comparison of the computed similarity measures by means of correlation  $r$  and confidence value  $p$  for the different participant groups and stimuli types. We first note that the distance measure obtained from manually extracted high-level descriptors seems to reflect best the perceived melodic similarity for both, expert and naïve listeners. Even though the computed similarity correlates strongly with the expert ratings, the also strong relation with the non-expert similarity judgments is still surprising, given the fact that the descriptors are based on rather abstract musicological concepts. We furthermore find a weaker, but still significant correlation between human ratings and the temporal deviation measure of the *dynamic time warping* algorithm as well as the vector distance among performance descriptors. On the other hand, we find no relation between human ratings and the pitch deviation from the dynamically aligned sequences, nor the chroma similarity measure. Given the fact that the selected performance descriptors are related to dynamic and temporal behavior and ornamentation and the temporal deviation measure does not consider the absolute pitch difference of the aligned sequences, we can speculate that for the given material these factors influence perceived similarity stronger than differences in the pitch progression. *Martinete* presents a particularly interesting case, since the skeleton of the melodic contour and at least its outer envelope is preserved throughout the performances. Notice also that in all cases the found correlation with the similarity ratings of real recordings is stronger than for the synthesized versions. Since none of the computational methods take voice timbre or lyrics into account, we can preclude that these factors influenced human judgement. It is however possible that it was more difficult for the listener to internalize these synthesized sequences compared to real recordings given their artificial nature and consequently judging similarity was more difficult and less precise.

## 6. CONCLUSIONS

The present study investigates the mechanisms involved in the perception of melodic similarity for the particular case of a cappella flamenco singing. We compared human judgements from experts and naïve listeners for audio recordings and synthesized melodic contours. Computational models are furthermore used to create distance matrices and evaluated based on their correlation with human ratings. We observed a significantly higher agreement among experts and a stronger correlation among computational models and the ratings based on real recordings than when comparing to ratings for synthesized melodies. Furthermore, we discover that models based on descriptors related to rhythm, dynamics and ornamentation are better suited to recreate similarity judgements than models based on absolute pitch distance. We obtained the highest corre-

Subject group	Expert listeners		Naive listener	
	real	synth	real	synth
Stimuli type				
<b>High-level</b>	r=0.585 p=0.001	r=0.424 p=0.001	r=0.429 p=0.000	r=0.202 p=0.054
<b>DTW temporal</b>	r=0.306 p=0.047	r=0.213 p=0.044	r=0.333 p=0.003	r=0.245 p=0.123
<b>DTW pitch</b>	r=-0.118 p=0.256	r=-0.094 p=0.224	r=-0.130 p=0.198	r=-0.096 p=0.204
<b>Perform. descrip.</b>	r=0.431 p=0.011	r=0.207 p=0.044	r=0.308 p=0.0123	r=0.176 p=0.061
<b>Chroma</b>	r=0.108 p=0.239	r=0.107 p=0.187	r=0.090 p=0.247	r=0.102 p=0.193

**Figure 6.** Correlation between computed similarity and human ratings. Statistically significant results are marked grey.

lation for both expert and non-expert ratings for a similarity measure computed from manually extracted high-level features. The problem of how to compute the high-level features automatically is still open. This problem is equivalent to that of automatically detecting ornamentation and main notes in a flamenco *cante*.

#### Acknowledgements

The authors would like to thank Joaquin Mora for providing the manual transcriptions and Joan Serrá for computing the chroma similarity measures. This research is partly funded by the COFLA (Proyectos de Excelencia de la Junta de Andalucía, P12-TIC-1362) and SIGMUS (Spanish Ministry of Economy and Competitiveness, TIN2012-36650) research projects as well as the PhD fellowship program of the Department of Information and Communication Technologies, Universitat Pompeu Fabra.

#### 7. REFERENCES

- [1] A. Marsden: "Interrogating Melodic Similarity: A Definitive Phenomenon or the Product of Interpretation?" *Journal of New Music Research*, Vol. 4, No. 44, pp. 323–335, 2012.
- [2] E. Molina, I. Barbancho, E. Gómez, A. M. Barbancho, and L. J. Tardón: "Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.
- [3] J. J. Cabrera, J. M. Díaz-Báñez, F. J. Escobar, E. Gómez, F. Gómez, and J. Mora: "Comparative Melodic Analysis of A Cappella Flamenco Cantes," *Proceedings of the Conference on Interdisciplinary Musicology*, 2008.
- [4] J. Mora, F. Gómez, E. Gómez, F. J. Escobar, and J. M. Díaz-Báñez: "Melodic Characterization and Similarity in A Cappella Flamenco Cantes," *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, 2010.
- [5] J. Blas Vega, and M. Ríos Ruiz: *Diccionario enciclopédico ilustrado del flamenco*, Cinterco, 1988.
- [6] J. L. Navarro, and M. Ropero: *Historia del flamenco*, Tartessos, 1995.
- [7] J. M. Gamboa: *Una historia del flamenco*, Espasa-Calpe, 2005.
- [8] E. Gómez, and J. Bonada: "Towards Computer-Assisted Flamenco Transcription: An Experimental Comparison of Automatic Transcription Algorithms as Applied to A Cappella Singing," *Computer Music Journal*, Vol. 37, No. 2, pp. 73-90, 2013.
- [9] E. Gómez, C. Guastavino, F. Gómez, and J. Bonada: "Analyzing Melodic Similarity Judgements in Flamenco A Cappella Singing," *Proceedings of the International Conference on Music Perception and Cognition*, 2012.
- [10] N. Kroher: *The Flamenco Cante: Automatic Characterization of Flamenco Singing by Analyzing Audio Recordings*, Master Thesis, Universitat Pompeu Fabra, 2013.
- [11] J. Serra, E. Gómez, P. Herrera, and X. Serra: "Chroma Binary Similarity and Local Alignment Applied to Cover Song Identification," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 16, No. 6, pp. 1138-1151, 2008..
- [12] N. Mantel, and R. S. Valand: "A technique of non-parametric multivariate analysis," *Biometrics*, Vol. 26, pp. 547-558, 1970.
- [13] J. Sundberg: "The acoustics of the singing voice," *Scientific American*, Vol. 236 (3), pp.104-116, 1977.
- [14] D. Muellensiefen, and K. Frieler: "Modelling experts' notions of melodic similarity," *Musicae Scientiae*, Discussion Forum 4A, pp.183-210, 2007.
- [15] A. Volk, and P. van Kranenburg: "Melodic similarity among folk songs: An annotation study on similarity-based categorization in music," *Musicae Scientiae*, 16 (3) pp.317-339, 2012.
- [16] R. Typke, and F. Wiering: "Transportation distances and human perception of melodic similarity," *Musicae Scientiae*, Discussion Forum 4A, pp.153-181, 2007.
- [17] T. Eerola, T. Jaervinen, J. Louhivuori, and P. Toivianen, : "Statistical Features and Perceived Similarity of Folk Melodies," *Music Perception*, 18 (3), pp.275-296, 2001.
- [18] E. Bonnet, and Y. Van de Peer : "zt: a software tool for simple and partial Mantel tests," *Journal of Statistical software*, 7 (10), pp.1-12, 2002.
- [19] B. Giordano, C. Guastavino, E. Murphy, M. Ogg, and B.K. Smith: "Comparison of Dissimilarity Estimation Methods". *Multivariate Behavioral Research*, 46, 1-33, 2011.

# THE VIS FRAMEWORK: ANALYZING COUNTERPOINT IN LARGE DATASETS

Christopher Antila and Julie Cumming

McGill University

christopher@antila.ca; julie.cumming@mcgill.ca

## ABSTRACT

The *VIS Framework for Music Analysis* is a modular Python library designed for “big data” queries in symbolic musical data. Initially created as a tool for studying musical style change in counterpoint, we have built on the `music21` and `pandas` libraries to provide the foundation for much more.

We describe the musicological needs that inspired the creation and growth of the VIS Framework, along with a survey of similar previous research. To demonstrate the effectiveness of our analytic approach and software, we present a sample query showing that the most commonly repeated contrapuntal patterns vary between three related style periods. We also emphasize our adaptation of typical  $n$ -gram-based research in music, our implementation strategy in VIS, and the flexibility of this approach for future researchers.

## 1. INTRODUCTION

### 1.1 Counterpoint

“The evolution of Western music can be characterized in terms of a dialectic between acceptable vertical sonorities on the one hand. . . and acceptable melodic motions on the other.” [12] A full understanding of polyphonic music (with more than one voice or part) requires description in terms of this dialectic, which is called counterpoint. Whereas music information retrieval research (such as [6]) typically describes polyphonic music only in terms of vertical (simultaneous or harmonic) intervals, musicologists interested in contrapuntal patterns also want to know the horizontal (sequential or melodic) intervals in each voice that connect the vertical intervals. Since counterpoint describes how pitches in independent voices are combined in polyphonic music, a computerized approach to counterpoint analysis of symbolic music can provide a wealth of information to musicologists, who have previously relied primarily on prose descriptions of musical style.<sup>1</sup>

<sup>1</sup>We wish to thank the following people for their contributions: Natasha Dillabough, Ichiro Fujinaga, Jane Hatter, Jamie Klassen, Alexander Morgan, Catherine Motuz, Peter Schubert. The ELVIS Project was

Figure 1 shows a musical score in bass clef, C major, 4/4 time. The score is annotated with vertical intervals above the notes and horizontal intervals below the notes. The first two beats are annotated with vertical intervals of 3 and horizontal intervals of +2 and -3. The next two beats are annotated with vertical intervals of 6 and 5, and horizontal intervals of +2 and -3. A dashed box highlights a common contrapuntal module in the final two beats, annotated with vertical intervals of 7, (6 5), 6, and 8, and horizontal intervals of 1 and -2.

**Figure 1.** Symbolic score annotated with vertical and horizontal intervals. A common contrapuntal module appears in the box.

Figure 1 shows the counterpoint between two voices in a fragment of music. We annotated the vertical intervals above the score, and the lower voice’s horizontal intervals below. Note that we show intervals by diatonic step size, counting number of lines and spaces between two notes, rather than semitones. We describe this contrapuntal module further in Section 2.1. By using intervals rather than note names, we can generalize patterns across pitch levels, so the same pattern may start on any pitch. For this article, we ignore interval quality (e.g., major or minor third) by using diatonic intervals (e.g., third), allowing generalization across mode and key. We do use interval quality for other queries—this is a choice available in VIS at runtime.

To allow computerized processing of contrapuntal patterns, we encode the counterpoint between two voices with alternating vertical and horizontal intervals. In Figure 1, the first two beats are “3 +2 3.” We call these patterns interval  $n$ -grams, where  $n$  is the number of vertical intervals. Our  $n$ -gram notation system is easily intelligible to music theorists and musicologists, and allows us to stay close to musicology.

### 1.2 Research Questions

Until recently, musicologists’ ability to accurately describe polyphonic textures was severely limited: any one person can learn only a limited amount of music in a lifetime, and the computer-based tools for describing or analyzing polyphonic music in detail are insufficiently precise for many repertoires. Descriptions of musical style and style change are often vague, derived from intuitive impressions and personal knowledge of repertoire rather than quantifiable

supported by the Digging into Data Challenge; the Canadian team responsible for the work described in this paper was additionally funded by the Social Sciences and Humanities Research Council of Canada.



© Christopher Antila and Julie Cumming.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Christopher Antila and Julie Cumming. “The VIS Framework: Analyzing Counterpoint in Large Datasets”, 15th International Society for Music Information Retrieval Conference, 2014.

evidence. Our project attempts the opposite by quantitatively describing musical style change using counterpoint.

We chose counterpoint not only because musicologists are already aware of its importance, but because it allows us to consider structure in all polyphonic music, which includes the majority of surviving Western music created after 1300. Our project’s initial goal is to find the most frequently-repeated contrapuntal patterns for different periods, genres, and composers, to help form detailed, evidence-based descriptions of style periods and style change by knowing which features change over time and when. In addition, statistical models will allow a fresh approach to attribution problems (determining the composer of a piece where it is not otherwise known), by enabling us to describe some of the factors that distinguish a composer’s style.

### 1.3 The VIS Framework

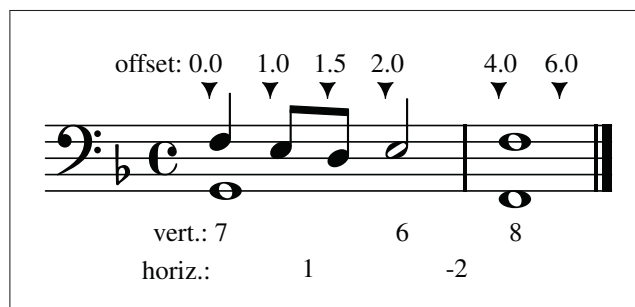
Our project’s most important accomplishment is the VIS Framework—the software we developed to answer the research questions described above. (VIS stands for “vertical interval successions,” which is a way to describe counterpoint). Currently VIS’s primary function is to find contrapuntal patterns in symbolic music, recording them with the notation described above in Figure 1 so they may be counted. However, we designed the framework to allow a much broader set of queries, and we intend to add support for additional musical dimensions (like meter and harmony) as well as more complicated statistical experiments (like Markov-chain modeling).

We used the *Counterpoint Web App*, a Web-based user interface for VIS’s counterpoint functionality, to run the analyses presented in this article.<sup>2</sup> Such Web-based software encourages musicologists to participate in data-driven analysis even if they are otherwise unable to program. The Web App’s visual design, the use of musicologist-friendly terms and user workflows, and the ability to output analysis results on musical scores are significant advantages. At the same time, programmers are encouraged to download and extend the VIS Framework using its well-documented Python API. While our Framework provides a guide for structuring analysis workflows, each analytic step benefits from our integration of the `music21` and `pandas` libraries. Together, these allow analytic approaches more amenable to musicians and statisticians, respectively.<sup>3</sup>

## 2. BACKGROUND

### 2.1 Contrapuntal Modules

A *contrapuntal module* is a repeated contrapuntal pattern made from a series of vertical (harmonic) and horizontal (melodic) intervals—a repeated interval  $n$ -gram. [11] We are primarily interested in the frequency and nature of two-voice contrapuntal modules. VIS allows us to computerize tedious score analysis previously done by hand, as when Peter Schubert identified modules in Palestrina. [13] While



**Figure 2.** “Cadence” contrapuntal module from Figure 1, with `music21` offset values.

two-voice contrapuntal modules are the primary structural element of much Renaissance music, we can find contrapuntal modules in nearly all polyphonic music, so our software and research strategies will be useful for a wide range of music. [3]

Figure 2 shows a representation of the “7 1 6 -2 8” interval 3-gram (a 3-gram because there are three vertical intervals). Using a half-note rhythmic offset, the first vertical interval is a seventh, the horizontal motion of the lower part is a unison (1), there is a vertical sixth, the lower part moves down by a second (-2), and the final vertical interval is an octave. In modal counterpoint, this is a highly conventionalized figure used to signal a cadence—a closing or concluding gesture for a phrase or piece. This is the same 3-gram as in the box in Figure 1.

Importantly, our analysis method requires that voicing information is encoded in our files. MIDI files where all parts are given in the same channel cannot be analyzed usefully with our software.

### 2.2 Previous Uses of $n$ -Grams in MIR

We have chosen to map musical patterns with  $n$ -grams partly because of their previous use in natural language processing.<sup>4</sup> Some previous uses of  $n$ -grams in music analysis, and computerized counterpoint analysis, are described below.

J. Stephen Downie’s dissertation presents a method for indexing melodic  $n$ -grams in a large set of folk melodies that will be searched using “Query by Humming” (QBH). [7] Downie’s system is optimized for what he calls “lookup,” rather than “analysis,” and he admits that it lacks the detail required by musicologists. Importantly, Downie only indexes horizontal intervals: melody rather than counterpoint.

Another QBH lookup system, proposed by Shyamala Doraisamy, adapts  $n$ -grams for polyphonic music. [5, 6] While this system does account for polyphony, it does not record horizontal intervals so it lacks the detailed contrapuntal information we seek. Furthermore, Doraisamy’s intervals are based on MIDI note numbers rather than the diatonic steps preferred by musicologists. Finally, the largest interval allowed by Doraisamy’s tokenization strategy is 26

<sup>2</sup> Visit [counterpoint.elvisproject.ca](http://counterpoint.elvisproject.ca).

<sup>3</sup> Refer to [pandas.pydata.org](http://pandas.pydata.org) and [mit.edu/music21](http://mit.edu/music21).

<sup>4</sup> As in the *Google Ngram Viewer*; refer to [books.google.com/ngrams](http://books.google.com/ngrams).



semitones—just over two octaves, and therefore narrower than the normal distance between outer voices even in Renaissance polyphony. Considering the gradual expansion of range in polyphonic music, to the extremes of the late 19th-century orchestra, the gradual appearance of large intervals may be an important style indicator.

Meredith has proposed geometric transformation systems for encoding multi-dimensional musical information in a computer-friendly way. [9, 10] We especially appreciate the multi-dimensional emphasis and the mathematical properties of these systems, and the software’s ability to work even when voicing information is not available in the symbolic file.

Finally, Jürgensen has studied accidentals in a large fifteenth-century manuscript of organ intabulations with an approach very similar to ours, but carried out using the Humdrum Toolkit. [8] She locates cadences by identifying a contrapuntal model and then records the use of accidentals at the cadence. While she searches only for specific contrapuntal modules, we identify all of the  $n$ -grams in a test set in order to determine the most frequently recurring contrapuntal patterns.

### 2.3 Multi-Dimensional $n$ -Grams in VIS

Considering these previous uses of  $n$ -grams and counterpoint in MIR, we designed our software with the flexibility to accommodate our requirements, as well as those of future analysis strategies. By tokenizing  $n$ -grams with strings that minimally transform the input, musicologists can readily understand the information presented in an  $n$ -gram. This strategy offers a further benefit to programmers, who can easily create  $n$ -grams that include different musical dimensions without necessarily developing a new token transformation system. Users may choose to implement any tokenization strategy on top of our existing  $n$ -gram-indexing module.

The example 3-gram shown in Figure 2 is tokenized internally as “7 1,” “6 -2,” “8 END.” Although there appear to be  $2n - 1$  tokens, we consider a vertical interval and its following horizontal interval as a combined unit—as though it were a letter in an  $n$ -gram as used in computational linguistics. The simplicity afforded by using strings as tokens, each of which may contain an arbitrary array of musical information, has been advantageous.

Indeed, the difficulty of determining a musical analogue to the letter, word, and phrase divisions used in computational linguistics may be one of the reasons that computer-driven research has yet to gain much traction in mainstream musicology. That music lacks an equivalent for space characters poses an even greater problem in this regard: while some music does use clear breaks between phrases, their exact placement can often be disputed among experts. Musicologists also wish to account for the multiple simultaneous melody lines of polyphonic music, which has no equivalent in natural language. These are the primary motivating factors behind our multi-dimensional interval  $n$ -gram tokens that encode both vertical and horizontal intervals. As our research continues, context models and multiple view-

point systems, in the style of Conklin and Witten, will partially obviate the questions of which  $n$  value to use, and of how best to incorporate varied musical elements. [1]

The popularity of Python within scientific computing communities allows us to benefit from any software that accepts *pandas* data objects. The easy-to-learn, object-oriented API of `music21`, along with the relatively high number of supported file formats, are also significant advantages. In the 1980s, a music analysis toolkit consisting of a collection of *awk* scripts was sensible, but Humdrum’s limitation to UNIX systems and a single symbolic file format pose undesirable limitations for a big data project.

## 3. EXPERIMENT

### 3.1 Data Sets

We present an experiment to quantitatively describe style change in the Renaissance period, providing a partial answer for our primary research question.<sup>5</sup> We assembled test sets for three similar style periods, named after a representative composer from the period: Ockeghem (1440–85), Josquin (1485–1521), and Palestrina (1540–85). The pieces in the test set were chosen to represent the style period as accurately as our project’s database allowed.<sup>6</sup> The twenty-year gap between the later periods is a result of less symbolic music being available from those decades. Each set consists of a mixture of sacred and secular vocal music, most with four parts, in a variety of genres, from a variety of composers. Though we analyzed  $n$ -grams between two and twenty-eight vertical intervals long, we report our results only for 3-grams because they are the shortest contrapuntal unit that holds meaning. Note that we include results from all possible two-part combinations, reflecting Renaissance contrapuntal thinking, where many-part textures are composed from a series of two-part structures. [3, 13]

The Ockeghem test set consists of 50 files: 28 in the MIDI format and 22 in *\*\*kern*. For the composers, 8 pieces were written by Busnoys, 32 by Ockeghem, and 10 are late works by Dufay. The longest repeated  $n$ -gram was a 25-gram.

The Josquin test set consists of 56 files: 18 MIDI, 23 *\*\*kern*, 9 MusicXML, and 6 NoteWorthy Composer. For the composers, 3 pieces were written by Agricola, 7 by Brumel, 6 by Compre, 2 by Fvin, 12 by Isaac, 19 by Josquin, 3 by Mouton, 2 by Obrecht, and 2 by la Rue. The longest repeated  $n$ -gram was a 28-gram.

Finally, the Palestrina test set consists of 53 files: 30 MIDI, 15 *\*\*kern*, 6 MusicXML, and 2 NoteWorthy Composer. For the composers, 15 pieces were written by Palestrina, 9 by Rore, 28 by Victoria, and 1 by Wert. The longest repeated  $n$ -gram was a 26-gram.

### 3.2 Methodology

The *VIS Framework* uses a modular approach to query design, dividing analysis tasks into a series of well-defined

<sup>5</sup> You may download our test sets from [elvisproject.ca/ismir2014](http://elvisproject.ca/ismir2014).

<sup>6</sup> Visit [database.elvisproject.ca](http://database.elvisproject.ca).

steps.<sup>7</sup> We intend the module break-down to be helpful for musicologists who wish to reason about and design their own queries. Thus, musicological concerns drove the creation of many of the analysis steps, such as the filtering modules described below. The interval  $n$ -gram frequency experiment in this article uses the following modules: *NoteRestIndexer*, *IntervalIndexer*, *HorizontalIntervalIndexer*, *FilterByOffsetIndexer*, *FilterByRepeatIndexer*, *NGramIndexer*, *ColumnAggregator*, and finally the *FrequencyExperimenter*.<sup>8</sup>

The *NoteRestIndexer* finds note names and rests from a `music21 Score`. The *IntervalIndexer* and *HorizontalIntervalIndexer* calculate vertical and horizontal intervals, respectively.

The *FilterByOffsetIndexer* uses a basic algorithm to filter weak-beat embellishing tones that otherwise obscure structural counterpoint. We regularize observations to a given rhythmic offset time interval using the `music21 offset`, measured in quarter lengths. Refer to Figure 2 as an example, where vertical intervals are filtered with a 2.0 offset. Events beginning on a multiple of that duration will be retained (like the notes at 0.0, 2.0, and 4.0). Events lasting for multiples of that duration will appear to be repeated (like the note at 4.0, which is also recorded at 6.0). Events not beginning on a multiple of the duration will be removed (like the notes at 1.0 and 1.5) or shifted to the following offset, if no new event occurs. For this study, we chose a half-note (2.0) offset interval in accordance with Renaissance notation practices, but this can be changed in VIS at runtime.

The *FilterByRepeatIndexer* removes events that are identical to the immediately preceding event. Because of its placement in our workflow for this experiment, subsequent vertical intervals will not be counted if they use the same pitches. Our interval  $n$ -grams therefore necessarily involve contrapuntal *motion*, which is required for proper pattern recognition. Such repeated events arise in musical scores, for example, when singers recite many words on the same pitch. The *FilterByOffsetIndexer* may also create repeated events, as at offset 6.0 in Figure 2. Users may choose not to run this module.

In this article, our *NGramIndexer* includes results from all pairs of part combination. Users may exclude some combinations at runtime, choosing to limit their query to the highest and lowest parts, for example. On receiving intervals from the *FilterByRepeatIndexer*, the *NGramIndexer* uses the gliding window technique to capture all possible overlapping interval  $n$ -grams. The indexer also accepts a list of tokens that prevent an  $n$ -gram from being counted. We use this feature to avoid counting contrapuntal patterns that include rests. Finally, the *NGramIndexer* may add grouping characters, surrounding “vertical” events in brackets and “horizontal” events in parentheses to enhance legibility of long  $n$ -grams. The 3-grams in this article are short enough that grouping characters are unnecessary; on

the other hand, the legibility of the “[10] (+2) [9] (1) [8] (+2) [7] (1) [6] (-2) [8]” 6-gram found 27 times in the Palestrina test set greatly benefits from grouping characters.

The *FrequencyExperimenter* counts the number of occurrences of each  $n$ -gram. These results, still specific to part combinations within pieces, are then combined with the *ColumnAggregator*.

On receiving a spreadsheet of results from VIS, we calculated the number of  $n$ -grams as the percentage total of all  $n$ -grams in each of the test sets. For each set, we also counted the total number of 3-grams observed (including all repetitions of all 3-grams), the number of distinct 3-gram types (whether repeated or not), and the number of 3-gram types that occur more than once; these are shown below in Table 1.

### 3.3 Results

Due to the limited time span represented in this study, we wish to suggest avenues for future exploration, rather than offer conclusive findings. We present a visualization of the experimental results in Figure 3, a hybrid between a Venn diagram, word cloud (i.e., a 3-gram cloud), and a timeline. The diagram includes interval 3-grams that constitute greater than 0.2% of the 3-grams in at least one of the test sets. When a 3-gram appears in an intersection of style periods, that 3-gram constitutes greater than 0.2% of the 3-grams in those sets. As in a world cloud, the font size is scaled proportionately to a 3-gram’s frequency in the test sets in which it is common. Most visually striking is the confirmation of musicologists’ existing experiential knowledge: certain contrapuntal patterns are common to all three style periods, including the cadence module (“7 1 6 -2 8”) and two other 3-grams that end with the “7 1 6” cadential suspension. These results make sense because cadences are an essential feature of musical syntax.

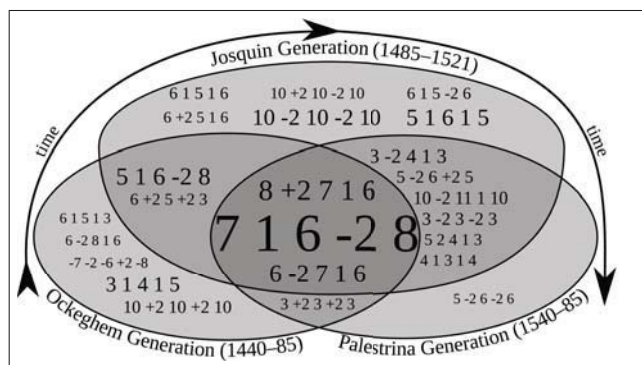
Test Set	Total	Types	Repeated Types
Ockeghem	30,640	10,644	4,509 (42%)
Josquin	31,233	9,268	4,323 (47%)
Palestrina	33,339	10,773	5,023 (47%)

**Table 1.** Summary of 3-gram repetitions in our query.

In addition to the common cadential patterns noted above, both Figure 3 and Table 1 show evidence of stylistic change over time. Most notably, the Josquin and Palestrina test sets show a higher level of repetition than the Ockeghem set. The number of 3-grams included in Figure 3 is higher in the Josquin test set (with seventeen 3-grams) than either the Ockeghem or Palestrina sets (both with eleven 3-grams). Yet Table 1 indicates the Josquin and Palestrina sets both have a higher percentage of 3-gram types that are repeated at least once (47% in both sets, compared to 42% in the Ockeghem set). These data suggest an increase in repetition of contrapuntal modules from the Ockeghem to the Josquin generations, which was retained in the Palestrina generation. Figure 3 only partially reinforces this suggestion: while five 3-grams are unique to the Ockeghem set, six are unique to the Josquin set, but only one is unique

<sup>7</sup> This section refers to the 2.x release series.

<sup>8</sup> For more information about the VIS Framework’s analysis modules and overall architecture, please refer to our Python API at [vis.elvisproject.ca](http://vis.elvisproject.ca).



**Figure 3.** Frequency of contrapuntal modules is different between temporally-adjacent style periods.

to the Palestrina set. Moreover, the “5 -2 6 -2 6” module, unique to the Palestrina set, is the least common 3-gram in Figure 3—how did contrapuntal repetition both decrease in Palestrina’s generation *and* remain the same?

Previous research by Cumming and Schubert may help us explain the data. In 2008, Cumming noted that exact repetition became much more common in Josquin’s lifetime than in Ockeghem’s. [2] Schubert showed that composers tended to repeat contrapuntal patterns in inversion during Palestrina’s lifetime, so that the lower voice is moved above the original upper voice. [13] Inversion changes a contrapuntal pattern’s vertical intervals in a consistent way that preserves, but switches, the horizontal intervals of the two parts. For example, “7 1 6 -2 8” inverts at the octave to “2 -2 3 +2 1.” While humans can recognize both forms as two versions of the same pattern, VIS currently shows only exact repetition; future enhancements will permit us to equate the original and the inversion. This decision may explain why our data show lower rates of repetition for the Palestrina test set.

We find further evidence of stylistic change in Figure 3: certain patterns that musicologists consider to be common across all Renaissance music are in fact not equally common in our three test sets. For example, motion by parallel thirds and tenths appears to be more common in certain style periods than others, and in a way that does not yet make sense. The Palestrina set shares ascending parallel thirds (“3 +2 3 +2 3”) with the Ockeghem and descending parallel thirds (“3 -2 3 -2 3”) with the Josquin set. Ascending parallel tenths (“10 +2 10 +2 10”) are more common in the Ockeghem set, and descending parallel tenths (“10 -2 10 -2 10”) in the Josquin set. In particular, descending parallel thirds are an order of magnitude less common in the Ockeghem test set than the Josquin or Palestrina (constituting 0.013%, 0.272%, and 0.225% of 3-grams in their test set, respectively). Conventional musicological wisdom suggests these 3-grams will be equally common in all three test sets, and that parallel tenths will be more common than parallel thirds in later style periods, as the range between voices expands. Since the reasons for such a deviation are not yet known, we require further investigation to study the changing nature of contrapuntal repetition during the Renaissance period. Yet even with these preliminary findings

it is clear that evidence-based research has much to offer musicology.

#### 4. FUTURE WORK

Our research will continue by extending VIS to add the option of equivalence classes that can group, for example, inversionally-related interval  $n$ -grams. We will also build on previous work with melody- and harmony-focussed multiple viewpoint systems to create an all-voice contrapuntal prediction model. [1, 14]

Our experiments will continue with larger test sets for increased confidence in our findings, also adding style periods earlier than the Ockeghem and later than the Palestrina sets, and subdividing our current style periods. This will help us reassess boundaries between style periods, and exactly what such a boundary entails. We will also compare results of single pieces with test sets of various sizes.

Finally, we will implement additional multi-dimensional  $n$ -gram tokens, for example by adding the note name of the lowest voice. This approach would encode Figure 2 as “7 F 1 6 F -2 8 E.” In Renaissance music, this type of  $n$ -gram will clarify the relationships between contrapuntal modules and a piece’s mode.

#### 5. CONCLUSION

The *VIS Framework for Music Analysis* is a musicologist-friendly Python library designed to analyze large amounts of symbolic musical data. Thus far, our work has concentrated on counterpoint—successions of vertical intervals and the horizontal intervals connecting them—which some scholars view as composers’ primary concern throughout the development of Western music. Our software uses multi-dimensional  $n$ -grams to find and count the frequency of repeated contrapuntal patterns, or modules. In particular, by retaining all inputted dimensions and using strings as tokens (rather than integers or characters), we simultaneously allow musicologists to quickly understand the content of an  $n$ -gram while also avoiding the challenge of developing a new tokenization strategy for every musical dimension added to the  $n$ -gram. We hope this flexibility and ease-of-use encourages musicologists and non-expert programmers, who would otherwise be discouraged from computer-based music analysis, to experiment more freely.

The results of our query presented in this article, which compares the most commonly-repeated contrapuntal modules in three Renaissance style periods, show the type of insight possible from computerized music research. The time-consuming effort required for previous work on contrapuntal modules is greatly reduced when analysts have access to specialized computer software. We analyzed more than 150 polyphonic compositions for interval  $n$ -grams between two and twenty-eight vertical intervals in length, which would have taken months or years for a human. Even with simple mathematical strategies like counting the frequency of interval  $n$ -grams to know which are most common, we can confirm existing intuitive knowledge about

the foundations of counterpoint while also suggesting avenues for future research on the nature of musical repetition.

## 6. REFERENCES

- [1] D. Conklin and I. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.
- [2] J. Cumming. From variety to repetition: The birth of imitative polyphony. In Bruno Bouckaert, Eugeen Schreurs, and Ivan Asselman, editors, *Yearbook of the Alamire Foundation*, number 6, pages 21–44. Alamire, 2008.
- [3] J. Cumming. From two-part framework to movable module. In Judith Peraino, editor, *Medieval music in practice: Studies in honor of Richard Crocker*, pages 177–215. American Institute of Musicology, 2013.
- [4] M. S. Cuthbert and C. Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 637–42, 2010.
- [5] S. Doraisamy. *Polyphonic Music Retrieval: The n-gram approach*. PhD thesis, University of London, 2004.
- [6] S. Doraisamy and S. Rger. Robust polyphonic music retrieval with n-grams. *Journal of Intelligent Information Systems*, 21(1):53–70, 2003.
- [7] J. S. Downie. *Evaluating a Simple Approach to Music Information Retrieval: Conceiving melodic n-grams as text*. PhD thesis, University of Western Ontario, 1999.
- [8] F. Jürgensen. Cadential accidentals in the Buxheim organ book and its concordances: A midfifteenth-century context for musica ficta practice. *Acta Musicologica*, 83(1):39–68, 2011.
- [9] D. Meredith. A geometric language for representing structure in polyphonic music. In *Proceedings of the International Society for Music Information Retrieval*, pages 133–8, 2012.
- [10] D. Meredith, K. Lemström, and G. Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 41(4):321–45.
- [11] J. A. Owens. *Composers at Work: The Craft of Musical Composition 1450–1600*. Oxford University Press, 1997.
- [12] P. Schubert. Counterpoint pedagogy in the renaissance. In T. Christensen, editor, *The Cambridge History of Western Music Theory*, pages 503–33. Cambridge University Press, 2002.
- [13] P. Schubert. Hidden forms in Palestrina’s first book of four-voice motets. *Journal of the American Musicological Society*, 60(3):483–556, 2007.
- [14] R. Whorley, G. Wiggins, C. Rhodes, and M. Pearce. Multiple viewpoint systems: Time complexity and the construction of domains for complex musical viewpoints in the harmonisation problem. *Journal of New Music Research*, 42(3):237–66, 2013.

# HIERARCHICAL APPROACH TO DETECT COMMON MISTAKES OF BEGINNER FLUTE PLAYERS

Yoonchang Han, Kyogu Lee

Music and Audio Research Group

Seoul National University, Seoul, Republic of Korea

{yoonchanghan, kglee}@snu.ac.kr

## ABSTRACT

Music lessons are a repetitive process of giving feedback on a student's performance techniques. The manner in which performance skills are improved depends on the particular instrument, and therefore, it is important to consider the unique characteristics of the target instrument. In this paper, we investigate the common mistakes of beginner flute players and propose a hierarchical approach to detect such mistakes. We first examine the structure and mechanism of the flute, and define several types of common mistakes that can be caused by incorrect assembly, poor blowing skills, or mis-fingering. We propose tailored algorithms for detecting each case by combining deterministic signal processing and deep learning, to quantify the quality of a flute sound. The system is structured hierarchically, as mis-fingering detection requires the input sound to be correctly assembled and blown to discriminate minor sound difference. Experimental results show that it is possible to identify different mistakes in flute performance using our proposed algorithms.

## 1. INTRODUCTION

The most important part of a music lesson is giving a student feedback on his or her performance, posture, and playing skills so that the student can play the sound correctly. Music lesson methods vary depending on the instrument being learned; therefore, audio signal processing for music education should make extensive use of prior knowledge regarding playing style, common mistakes, unique characteristics, and constraints of the target instrument. However, most existing music signal analysis techniques use a general-purpose model, and relatively little attention is paid to an instrument-specific approach. A general-purpose model is advantageous because it can be applied to various types of instruments. However, this model lacks the capability to capture instrument-specific sound characteristics. There are always common mistakes that beginners make, but little is known about how to detect these automatically.

The goal of this paper is to investigate common beginner's mistakes when playing a specific instrument—the flute, in this case—and to analyze the spectral characteristic of each case to give the student appropriate feedback on his or her performance. Because the sound of a musical instrument is affected by numerous factors, in our work, we first divide the factors that usually lead beginners to play the wrong sound into three parts: incorrect flute assembly, blowing skill, and fingering.

The rest of the paper is organized as follows: We briefly present existing works related to our proposed idea. Then, we investigate possible mistakes in flute performance by examining the structure and mechanism of the flute, and several types of common mistakes and the resulting sounds are explained. Next, we present an overall system structure to distinguish each mistake, along with a detail explanation of each proposed algorithm. We then present the experimental results to demonstrate the feasibility of the proposed system, followed by our conclusion and directions for future work.

## 2. RELATED WORK

The characteristics of musical instruments depend on their sound production mechanism. The characteristics of one instrument can greatly differ from those of others, and each instrument's characteristics may not be captured equally well as another even when using the same computational model [2]. However, there has been minimal research regarding an instrument-specific model. Some examples of instrument-specific approaches involve the use of a violin [8, 14-16], guitar [1], bells [9], and tabla [5]. For instance, the violin transcription system in [8] makes use of characteristics such as highest and lowest pitch, possible play style (e.g., upper octave duophony), vibrato, and loudness. The training system in [14] uses a common envelope style of violin sound for note segmentation prior to real-time pitch detection, and [9] uses the acoustic characteristics of a church bell, as well as the rules of a bell charming performance, for transcription and estimating the number of bells. In addition, a chord transcription system designed for guitar in [1] outperforms the non-guitar-specific method.

As shown above, using prior knowledge of the characteristics of a target instrument creates new possibilities in music signal processing, and can also improve the per-



© First author, Second author, Third author.

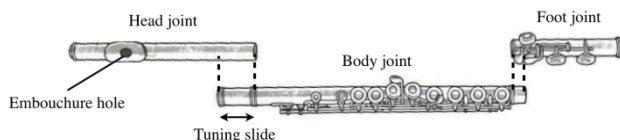
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** First author, Second author, Third author. "Paper Template For ISMIR 2014", 15th International Society for Music Information Retrieval Conference, 2014.

formance of the system. However, there are still many instruments to be studied, and the flute is one of them.

### 3. COMMON MISTAKES OF A FLUTE PLAYER

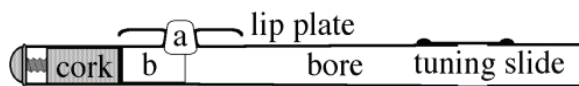
#### 3.1 Assembling the Flute

Like most woodwind instruments, the flute needs to be assembled before it is played. The flute consists of a head joint, body joint, and foot joint, as shown in Figure 1. The connecting part between the body and foot joint is very short, while the connecting part between the head joint and body joint is a few centimeters long. This intentionally designed adjustable part is called the tuning slide, and it can be used for changing the total length of the flute to various sizes, which affects the overall pitch of the flute. For instance, if the head joint is placed very deep into the tuning slide of the body, the pitch will be increased for every note. By contrast, if the head joint is pulled out too far, the overall pitch will drop owing to the longer wavelength.



**Figure 1.** Flute consists of head joint, body joint, and foot joint (modified after [11]).

Another method of pitch tuning is adjusting the cork part of the head joint, as shown in Figure 2. This can be adjusted by a screw. Pushing the cork will raise the pitch of all notes. However, this is beyond the scope of this paper, as this screw is normally not adjusted by flute performers but by flute technicians.



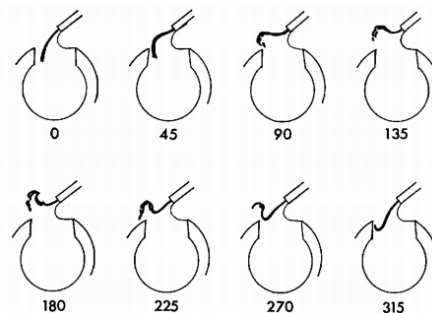
**Figure 2.** Schematic of a flute head joint [13].

Trained performers use this variable tuning slide for pitch tuning. The pitch of the flute is sensitive to the conditions of the surrounding environment, such as humidity and temperature. However, novice flutists are not sensitive to minor pitch shifting, and they may play the flute in the wrong overall pitch without recognizing it.

#### 3.2 Blowing Embouchure

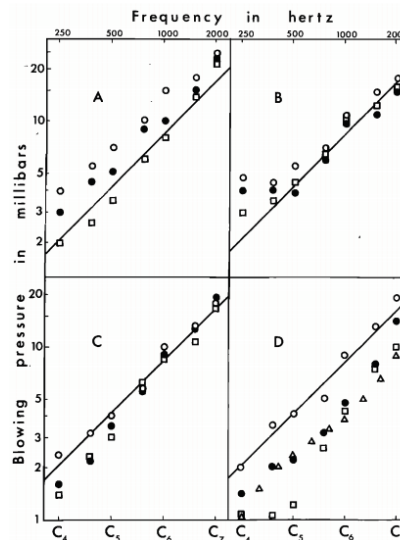
The flute generates sound by blowing a rapid air jet across the embouchure<sup>1</sup> hole, as shown in Figure 3. Hence, the quality of the generated sound is highly dependent on the blowing skill of the performer. Blowing skill involves lip position and the thickness/stability of the air jet. Clear tone production is challenging for beginners because the method of tone production for the flute

is not supported by mechanical parts; rather, it depends only on the player's blowing skill [4].



**Figure 3.** Airstream oscillation of the flute embouchure hole. The labels indicate the phase angles of the acoustic current at the hole [3].

Tone quality and octave of the sound are related to blowing skill. The flute has a range of three octaves, starting from middle C (C<sub>4</sub>), with several less-used notes in octaves 3 and 7. The blowing pressure determines the octave of the sound, as shown in Figure 4. Greater blowing pressure can be achieved by blowing a narrower and stronger air jet. To generate a stable and clean sound, it is important to keep this blowing pressure reasonably steady. Failure to do this will result in fluctuating sound and noise, which is highly unpleasant and typically the first hurdle for beginners to overcome in their training.



**Figure 4.** Air jet blowing pressure has a roughly linear relationship to fundamental frequency. A, B, C, and D are different performers, and different shapes represent different dynamics [12].

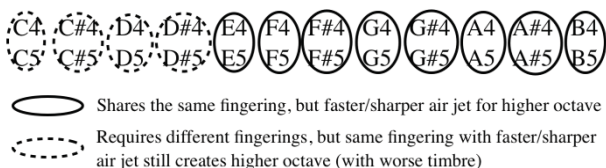
#### 3.3 Fingering

Novice flutists frequently make mistakes in fingering owing to their lack of familiarity with the irregular fingering rules of the flute. High-octave fingering is comparatively more complex than low-octave fingering [4], which is the reason why flute lessons usually start with the lowest octave and move step-by-step to higher octaves. Hence, we

<sup>1</sup> Mouthpiece of a musical instrument.

focus on octaves 4 and 5, which are the octaves that beginner flute players initially study.

Most of the octave 5 fingerings are identical to those of octave 4, as shown in Figure 3. However, the fingering for C and D, as well as the sharps of these notes, require different fingerings than those of octave 4. These notes can be played with octave 4 fingering using a faster and sharper air jet, but this results in a slightly airy timbre, compared to the sound when the flute is correctly fingered. As this airy timbre is not significantly noticeable, and most of the notes in octaves 4 and 5 share the same fingering, many beginners do not notice that they used octave 4 fingerings to play octave 5, unless the instructor spots it.



**Figure 5.** Fingering of octave 4 and 5 flute notes. Note that C, D, and sharps of these require different fingering, unlike E, F, G, A, and B.

Another fingering-related problem is the proper positioning of the fingers. The open-hole flute requires that the flutist use his or her fingers to block the holes in the keys. Most professional flutists prefer the open-hole flute owing to its advantages in tone production and intonation adjustment [4]. However, this is not considered in our system because beginners who have trouble with blocking open-hole keys can avoid this problem by putting plastic plugs in the holes until they get used to playing the open-hole flute.

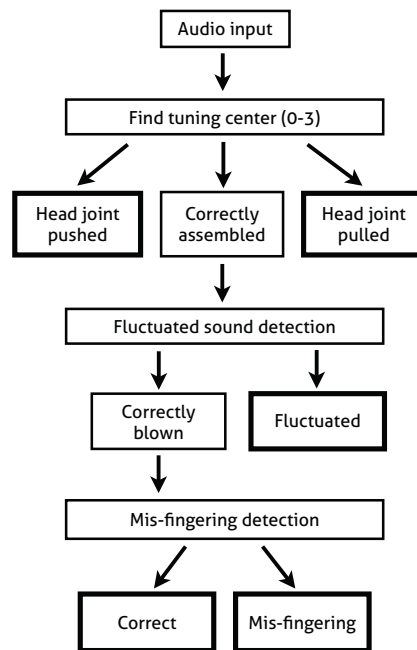
#### 4. PROPOSED SYSTEM & METRICS

The overall system comprises several steps. In the first step, the system determines whether the flute is assembled correctly using entire input audio. Next, once the flute sound is detected as coming from a correctly assembled flute, the system measures if sound of the each note is a clear, correctly blown sound or an airy-timbered sound. Finally, the properly blown sound is identified as sound generated from either correct fingering or incorrect fingering. The system is hierarchically structured, because mis-fingering detection does not work well for fluctuated sound or head joint pushed/pulled sound as it requires discriminating minor sound difference. The input audio is resampled to 16 kHz first, and the system architecture is shown in Figure 6.

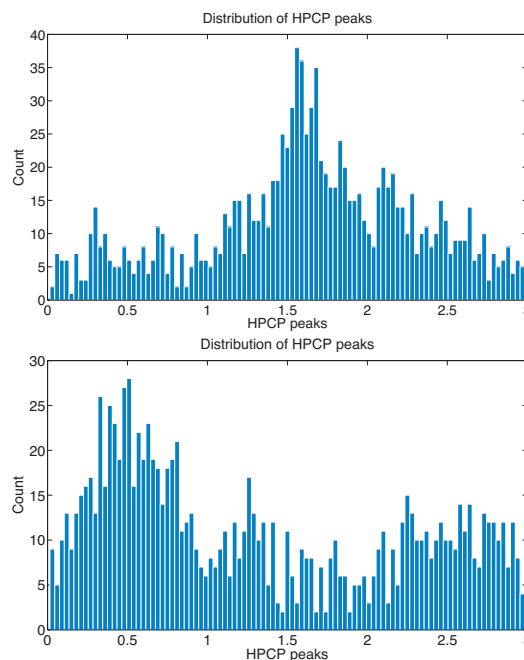
##### 4.1 Assembling Error Detection

Some mistakes can cause modifications to the overall pitch, and some mistakes result in poor timbre. The assembling error affects only the overall pitch of the generated sound. As mentioned in 3.1, the distance the head

joint is pushed in or pulled out from the tuning slide of the body joint determines the overall pitch. To this end, a quantized chromagram from Harte and Sandler is used to detect the tuning center [6].



**Figure 6.** Flow diagram of the overall system. The bold box indicates where the system sends feedback to the user.



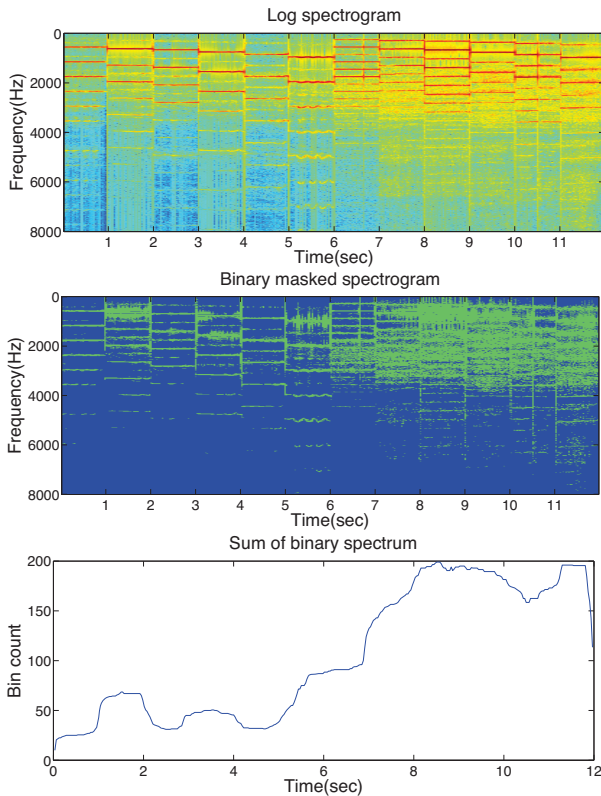
**Figure 7.** HPCP peaks histogram within a semitone for correctly (up) and loosely assembled (down) case.

To determine the tuning center, a spectrum of linear frequency spectra is Constant-Q transformed and summed across octaves to produce a harmonic pitch class profile (HPCP). A 36-bin quantized chromagram is used to determine the semitone center, and three bins were al-

located for each semitone. By observing the distribution of peak positions across the width of a semitone, as shown in Figure 7, it is possible to determine the tuning center of the instrument. Because three bins are allocated for each semitone, the tuning center of a perfectly tuned sound would ideally be 1.5. Therefore, the system will consider the input sound to be correctly tuned when the tuning center value is approximately 1.5. If the detected tuning center is too low (less than 1), the system sends feedback to the user that the head joint is too loosely assembled. Conversely, the system tells the user that the head joint is assembled more tightly than necessary when the tuning center is high (greater than 1).

## 4.2 Fluctuated Sound Detection

Incorrect lip position on the embouchure, along with an irregular stream of blown wind, results in a highly unpleasant and fluctuating tone. This sound contains many inharmonic partials in a spectrum, and it is clearly visible on a spectrogram. Performing binary masking on a spectrogram makes these inharmonic partials more obvious, as shown in the second row of Figure 8.



**Figure 8.** Log spectrogram, binary masked spectrogram, and sum of bins for each frame for D, E, F, G, A, and B of octave 5. Up to 6 second is correctly blown sound and from 6 to 12 second is fluctuated sound.

Binary masking is performed as follows:

$$X_b(k) = \begin{cases} 0 & X(k) < \theta \\ 1 & X(k) > \theta \end{cases} \quad (1)$$

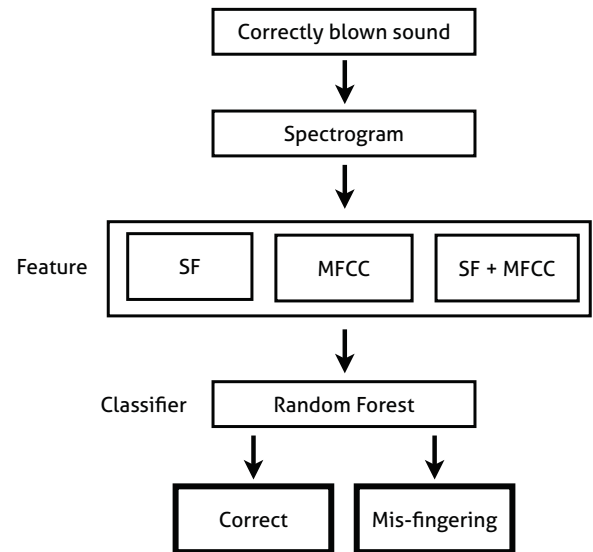
where  $X$  is the log spectrum,  $X_b$  is the binary masked spectrum, and  $\theta$  is the threshold constant. Empirically, a value between -20 and -30 works well for  $\theta$ , depends on recording environment. Note that these values are obtained when natural log multiplied by 20 is used for the log spectrum. Using this binary masked spectrogram, the sum of the number of positive valued bins of each spectrum can be used as a measurement for determining how the sound fluctuates owing to poor blowing skill. This can be expressed as follows:

$$F(k) = \sum_{k=0}^{N-1} X_b(k) \quad (2)$$

where  $F$  is the amount of fluctuation. The third row of Figure 8 is  $F$  value obtained from (2) with 1 second median filtering, and it is possible to observe the value is much higher for fluctuated sound than correctly blown sound.

## 4.3 Mis-fingering detection

As mentioned in 3.3, for C5, C#5, D5, and D#5, using octave 4 fingering with a faster and sharper air jet still generates octave 5 pitches even without correct fingering, although the timbre is slightly airy. To detect this timbral difference, we decided to use both the Mel-frequency cepstral coefficient (MFCC)—a widely used, hand-designed feature—and sparse filtering (SF) [10]—a deep-layered, unsupervised feature learning method. SF works by optimizing the sparsity of feature distribution, and it works well on a range of data modalities without specific tuning. Both single- and double-layered sparse filtering were used with 200 units for each layer. The obtained feature was classified into two classes (correct/incorrect) using a random forest (RF) classifier, which exhibits better performance than a support vector machine or back-propagation neural network in a variety of cases [7]. The flow diagram for mis-fingering detection is shown below.



**Figure 9.** Flow diagram for mis-fingering detection.



## 5. EXPERIMENT

### 5.1 Objective & Procedure

The goal of our experiment was to explore whether the proposed system and algorithms work well for detecting the mistakes of beginner flutists. Flute sound samples were obtained from two intermediates (who have played the flute for one to two years) and one expert (who holds an exam score of Grade 8 with a Distinction). Flutes used for the experiment were a B foot joint with open holes, and a silver head with nickel body and foot. The correct flute sound, fluctuating sound, head joint pulled, and head joint pushed sound were recorded for octaves between 4 and 5. The length of the collected audio was 30 seconds for each semitone. The case of correct and incorrect fingering for C5, C#5, D5, and D#5 was recorded for 10 minutes each to obtain sufficient training data. The input audio was recorded at 44.1 kHz mono and downsampled to 18 kHz. Tuning center was calculated from whole target audio as it is not time-varying characteristics. Meanwhile, fluctuating and mis-fingering detection was performed framewise. Different window and hop size were used for each experiment, as each mistake detection algorithm requires different spectral resolution.

### 5.2 Results

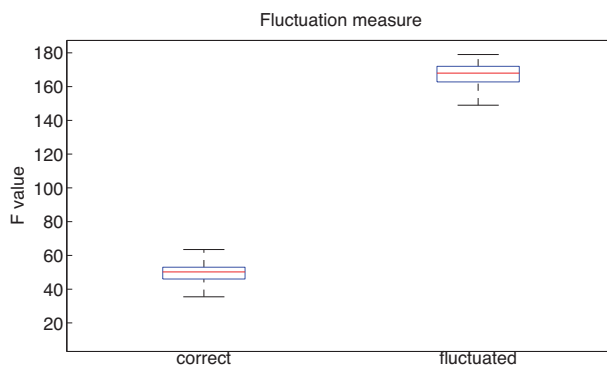
The experimental results show that the system successfully distinguishes each mistake. To find tuning center, a 74 ms window and 18 ms hop size were used. As shown in Table 1, the tuning center of a correctly played sample is close to 1.5, which is the exact center. Also, tuning center values for the head joint when it is pushed and pulled fell into the expected range, which were (0–1) and (2–3), respectively.

Mistake cases	Tuning center value (0 to 3)		
	Player 1	Player 2	Player 3
Correct	1.68	1.59	1.62
Head joint pushed	2.55	2.49	2.43
Head joint pulled	0.48	0.45	0.75

**Table 1.** Tuning center values of correct, head joint pushed, and head joint pulled flute sound of three different flutists.

Next, Figure 10 is a framewise distribution of fluctuation measure (1) for correct and fluctuated flute sound. A 64 ms window and 32 ms hop size were used, with  $\theta$  value of -25dB. The median value of the correct flute sound is 50, and most of the values fall between 63 and 37. The fluctuating flute sound has a median value of 167, and most of the values fall between 150 and 178. This means that these cases are clearly distinguishable using the proposed metric.

Finally, Table 2 shows the ten-fold cross-validation results of the proposed mis-fingering classification using single-layer SF, double-layer SF, and MFCC as a feature, and RF as a classifier. A 16 ms window and 10 ms hop size were used, and SF was used with 200 units per layer.



**Figure 10.** Box plot of fluctuation measurements. The central marks indicate the median, and the edges are the first and third quartiles.

Method	Accuracy (%)
Spectrogram + SF (single)	90.24
Spectrogram + SF (double)	90.02
MFCC	90.89
MFCC + SF (single)	90.33
MFCC + SF (double)	91.35

**Table 2.** Mis-fingering classification ten-fold cross-validation result using SF/ MFCC as a feature and RF as a classifier.

The result shows that the combination of the MFCC and double-layered SF performs the best; however, all of the approaches perform reasonably well within a not very meaningful margin. The result indicates that the MFCC, a handcrafted feature, is still useful in separating the timbral differences of the flute. Further, although SF is not designed for the purpose of timbre analysis, it works quite well without fine-tuning, as mentioned in [10]. In the experiment, single-layered SF worked better when the input is a spectrogram, but double-layered SF showed better performance when the input is MFCC.

## 6. CONCLUSION & FUTURE WORK

The objective of our work is to use audio signal analysis to give a student feedback on his or her flute performance to help fix mistakes, as a lesson teacher would do. To achieve this goal, we examined the mechanism and structure of the flute. We also investigated the common mistakes of beginner flute players. We determined several types of common mistakes and developed a hierarchical system to detect such cases by observing the tuning cen-

ter, fluctuation metric, and a mis-fingering detection algorithm. As a result, we have successfully identified common mistake cases from input audio, which can be used as feedback that would be provided by a lesson teacher-. Head-joint assembling errors were detected by determining the tuning center of the flute sound. Fluctuating sound caused by poor blowing skills was separated from the correct flute sound by measuring the amount of noisy harmonic contents. Finally, mis-fingering cases were detected by analyzing their timbre using MFCC and SF with an RF classifier.

There remain some problems to be tackled in this mistake detection algorithm for real-world user applications. First, the mis-fingering detection algorithm may be affected by the material or maker of the flute because the algorithm detects very minor changes in timbre. In the experiment, only two types of flute (silver head with nickel body, and foot) were used. However, the flute can be made of various types of metal, such as silver, gold, and platinum. Moreover, various flute makers have their own timbral characteristics, which may influence the classification results. Second, the experiment was done on the frame level, but the user perceives the score based on the note level. Hence, the system should be used along with appropriate onset-offset detection to give more user-friendly feedback.

We believe that this type of timbre-related and user-behavior-oriented feedback is highly important for the next-generation music transcription systems, especially those used for educational purposes. Playing the instrument with correct onset and pitch is not a very difficult part of being a good player, but making a beautiful timbre is what really takes time. This paper focuses only on the flute; however, our overall approach, including analyzing mistake cases and determining customized solutions, can be applied to various instruments in a similar way.

## 7. ACKNOWLEDGEMENTS

This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2013-H0301-13-4005).

## 8. REFERENCES

- [1] A. M. Barbancho, A. Klapuri, L. J. Tardon, and I. Barbancho: "Automatic Transcription of Guitar Chords and Fingering from Audio," *IEEE TASLP*, 20(3): 915–921, 2012.
- [2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchoff, and A. Klapuri: "Automatic Music Transcription: Breaking the Glass Ceiling," In *ISMIR*, 2012.
- [3] J. W. Coltman: "Sounding Mechanism of the Flute and Organ Pipe." *The Journal of the Acoustical Society of America* 44, 1968.
- [4] C. Delaney, *Teacher's Guide for the Flute*. Rev. 11/98, Selmer, 1969.
- [5] O. Gillet and G. Richard: "Automatic Labelling of Tabla Signals," In *ISMIR*, 2003.
- [6] C. Harte and M. Sandler: "Automatic Chord Identification using a Quantised Chromagram." In *Audio Engineering Society Convention 118*. 2005.
- [7] M. Liu, M. Wang, J. Wang, and D. Li: "Comparison of Random Forest, Support Vector Machine and Back Propagation Neural Network for Electronic Tongue Data Classification: Application to the Recognition of Orange Beverage and Chinese Vinegar," *Elsevier Sensors and Actuators*, pp. 970–980, Vol. 177, 2013.
- [8] A. Loscos, Y. Wang, and W. J. Boo: "Low Level Descriptors for Automatic Violin Transcription." In *ISMIR*, 2006.
- [9] M. Marolt: "Automatic Transcription of Bell Chiming Recordings." In *IEEE TASLP*, 20(3): pp. 844–853, 2012.
- [10] J. Ngiam, P. W. Koh, Z. Chen, S. Bhaskar and A. Y. Ng., "Sparse Filtering," In *NIPS*, 2011.
- [11] H. Pinksterboer, *Tipbook Flute and Piccolo: The Complete Guide*, Hal Leonard, 2009.
- [12] T. D. Rossing, F. Richard Moore, and P. A. Wheeler: *The Science of Sound*. Vol. 2. Massachusetts. Addison-Wesley, 1990.
- [13] J. Smith, J. Wolfe, and M. Green: "Head Joint, Embouchure Hole and Filtering Effects on the Input Impedance of Flutes." In *Proc. of the Stockholm Music Acoustics Conference*, pp. 295–298. 2003.
- [14] J. Wang, S. Wang, W. Chen, K. Chang, and H. Chen: "Real-Time Pitch Training System for Violin Learners," *Multimedia and Expo Workshops (ICMEW), IEEE*, 2012.
- [15] R. S. Wilson: "First Steps Towards Violin Performance Extraction using Genetic Programming," In John R. Koza, editor, *Genetic Algorithms and Genetic programming*, pp. 253–262, 2002.
- [16] J. Yin, Y. Wang, and D. Hsu: "Digital Violin Tutor: An Integrated System for Beginning Violin Learners," *ACM Multimedia*, Hilton, Singapore, 2005.

# ROBUST JOINT ALIGNMENT OF MULTIPLE VERSIONS OF A PIECE OF MUSIC

Siying Wang

Sebastian Ewert

Simon Dixon

Queen Mary University of London, UK

{siying.wang, s.ewert, s.e.dixon}@qmul.ac.uk

## ABSTRACT

Large music content libraries often comprise multiple versions of a piece of music. To establish a link between different versions, automatic music alignment methods map each position in one version to a corresponding position in another version. Due to the leeway in interpreting a piece, any two versions can differ significantly, for example, in terms of local tempo, articulation, or playing style. For a given pair of versions, these differences can be significant such that even state-of-the-art methods fail to identify a correct alignment. In this paper, we present a novel method that increases the robustness for difficult to align cases. Instead of aligning only pairs of versions as done in previous methods, our method aligns multiple versions in a joint manner. This way, the alignment can be computed by comparing each version not only with one but with several versions, which stabilizes the comparison and leads to an increase in alignment robustness. Using recordings from the Mazurka Project, the alignment error for our proposed method was 14% lower on average compared to a state-of-the-art method, with significantly less outliers (standard deviation 53% lower).

## 1. INTRODUCTION

Recent years have seen significant efforts to create large, comprehensive music collections. Music content providers (e.g. Spotify, iTunes, Pandora) rely on their existence, while national libraries and charitable organizations create and curate them in order to provide access to cultural heritage. For a given piece of music, large collections often contain various related recordings (cover songs, different interpretations), videos (official clip, live concert) and musical scores (in different formats such as MIDI and MusicXML, covering several editions). To identify and link these different versions, various automatic alignment methods have been proposed in recent years. Such *synchronization methods* have been used to facilitate navigation in large collections [1], to implement score following in real-time [2–5], to compare different interpretations of

a piece [6], to identify cover songs [7] or to simplify complex audio processing tasks [8].

In general, the goal of music synchronization is, given a position in one version of a piece of music, to locate the corresponding position in another version. To compute a synchronization, existing methods align two versions of a piece at a time, even if several relevant versions are available. For example, in [9, 10] a score of a piece is automatically aligned to a corresponding audio recording, while in [11] two acoustic realizations are being synchronized. As shown previously, current methods yield in many cases alignments of high accuracy [9–11]. However, musicians can interpret a piece in diverse ways, which can lead to significant local differences in terms of articulation and note lengths, ornamental notes, or the relative loudness of notes (balance). If such differences are substantial, the alignment accuracy of state-of-the-art methods can drop significantly.

To increase alignment robustness for difficult cases, the main idea in this paper is to exploit the fact that multiple versions of a piece are often available and can be aligned in a joint way. This way, we can exploit the additional information that each version provides about how a certain position in a piece can be realized by a musician. As a consequence, while two given recordings might be rather different and hard to align, both of them might actually be more similar to a third recording and including such a recording within the alignment process can lead to an increase in overall robustness. To compute our joint synchronization, we modify a multiple sequence alignment method typically employed in biological signal processing and combine it with strategies developed in a musical context based on Multiscale-DTW (FastDTW) and chroma-based onset features for increased computational efficiency and synchronization accuracy. In the following, we describe technical details of this method in Section 2. Then, we report on some of our experiments in Section 3. Conclusions and prospects for future work are given in Section 4.

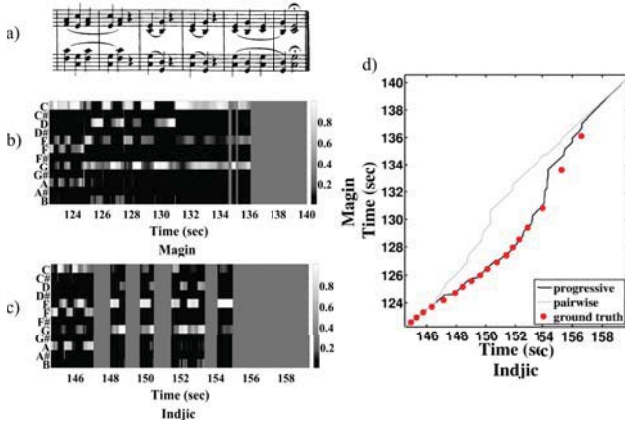
## 2. ALIGNMENT METHOD

Various methods have been proposed to align two given data sequences, including Dynamic Time Warping (DTW) and Hidden Markov Models (HMM) [2], Conditional Random Fields (CRF) [9], and Particle Filter / Monte-Carlo Sampling (MCS) based methods [4, 5]. With the exception of MCS methods, which are online methods, the remaining three methods operate in an offline fashion and are quite



© Siying Wang, Sebastian Ewert, Simon Dixon.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Siying Wang, Sebastian Ewert, Simon Dixon. “Robust Joint Alignment of Multiple Versions of a Piece of Music”, 15th International Society for Music Information Retrieval Conference, 2014.



**Figure 1.** Alignment of two interpretations of Chopin Op. 24 No. 2, measures 115-120: (a) Score for the six measures. (b)/(c) Chroma features for an interpretation by Magin and Indjic, respectively; chroma features with uniform energy distribution are the result of silence in the recording. (d) Alignment results for our baseline pairwise (gray) and proposed method (black).

similar from an algorithmic point of view. We describe our proposed method as an extension to DTW. However, the underlying ideas are applicable in HMM and CRF contexts as well.

## 2.1 Baseline Pairwise Alignment

To summarize DTW-based alignment, let  $X := (x_1, x_2, \dots, x_N)$  and  $Y := (y_1, y_2, \dots, y_M)$  be two feature sequences with  $x_n, y_m \in \mathcal{F}$ , where  $\mathcal{F}$  denotes a suitable feature space. Furthermore, let  $c : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}$  denote a local cost measure on  $\mathcal{F}$ . We define a resulting  $(N \times M)$  cost matrix  $C$  by  $C(n, m) := c(x_n, y_m)$ . An alignment between  $X$  and  $Y$  is defined as a sequence  $p = (p_1, \dots, p_L)$  with  $p_\ell = (n_\ell, m_\ell) \in [1:N] \times [1:M]$  for  $\ell \in [1:L]$  satisfying  $1 = n_1 \leq n_2 \leq \dots \leq n_L = N$  and  $1 = m_1 \leq m_2 \leq \dots \leq m_L = M$  (boundary and monotonicity condition), as well as  $p_{\ell+1} - p_\ell \in \{(1, 0), (0, 1), (1, 1)\}$  (step size condition). An alignment  $p$  having minimal total cost among all possible alignments is called an *optimal alignment*. To determine such an optimal alignment, one recursively computes an  $(N \times M)$ -matrix  $D$ , where the matrix entry  $D(n, m)$  is the total cost of the optimal alignment between  $(x_1, \dots, x_n)$  and  $(y_1, \dots, y_m)$ :

$$D(n, m) := \min \begin{cases} D(n-1, m-1) + w_1 C(n, m), \\ D(n-1, m) + w_2 C(n, m), \\ D(n, m-1) + w_3 C(n, m), \end{cases}$$

for  $n, m > 1$ . Furthermore,  $D(n, 1) := \sum_{k=1}^n w_2 C(k, 1)$  for  $n > 1$ ,  $D(1, m) = \sum_{k=1}^m w_3 C(1, k)$  for  $m > 1$ , and  $D(1, 1) := C(1, 1)$ . The weights  $(w_1, w_2, w_3) \in \mathbb{R}_+^3$  can be used to adjust the preference over the three step sizes. By tracking the choice for the minimum starting from  $D(N, M)$  back to  $D(1, 1)$ , an optimal alignment can be derived in a straightforward way [2]. In a musical context,  $\mathcal{F}$  typically denotes the space of normalized chroma features,  $c$  is usually a cosine (or Euclidean) distance with

weights set to  $(w_1, w_2, w_3) = (2, 1, 1)$  to remove a bias for the diagonal direction [2, 11].

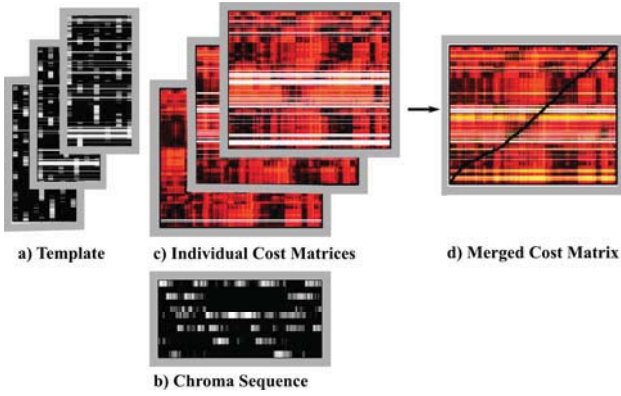
A main difficulty in aligning music stems from the degree of freedom a musician has in interpreting a score, in particular regarding the local tempo, balance (relative loudness of concurrent notes), articulation and playing style. If several differences occur together, standard alignment methods sometimes fail to identify the musically correct alignment. In Fig. 1(b)/(c), we see chroma features for two interpretations of Chopin Op. 24 No. 2 measures 115-120 (Fig. 1(a)) as performed by Magin and by Indjic, respectively. Besides the tempo, we see differences in the interpretation of pauses (the uniform energy distributions in the features correspond to silence), articulation and in the balance (relative loudness of notes). In this case, the differences are significant such that pairwise DTW-based approaches [10, 11] fail to compute the correct alignment, see upper path in Fig. 1(d). The red dots indicate corresponding beat positions in the two versions.

## 2.2 Joint Alignment of Multiple Versions

Comparing several versions of a piece, interpretations vary in different ways and to different extents. If several versions of a piece are available, each version provides an example of how a specific position in a piece can be realized, and this additional information can be used to stabilize the alignment for difficult sections. A straightforward strategy to compute a joint alignment could be to extend DTW to allow for more than two versions. For example, to align three versions, one can define an order-3 cost tensor in a straightforward way and apply the same dynamic programming techniques as used in DTW [12] (note that a cost matrix for two versions is an order-2 tensor). However, assuming that each feature sequence to be aligned is roughly of length  $N$ , the time and memory requirement to align  $K$  recordings would be in  $O(N^K)$ , which prohibits the alignment of more than a very few recordings.

In computational biology, multiple sequence alignment is a well-studied problem. Most popular are so called *profile-based methods* and *progressive alignment methods* [12]. Profile-based methods employ a specific type of HMM, which is trained via Expectation-Maximization (EM) on the set of feature sequences to be aligned. Each state of the resulting *profile-HMM* corresponds to a position in a so called average-sequence: the sequence of means of the observation probabilities of the HMM-states, see [12] for details. A multiple synchronization is then computed by aligning each sequence to the average-sequence via the Viterbi algorithm. This procedure has been attempted in a musical context with limited success [13]. We believe this is due to, using EM training, whereby aligned features are essentially averaged (with Gaussian observation probabilities), which results in a loss of information and can lead to a loss of alignment accuracy.

Using progressive alignment such averaging is not necessary. The underlying idea is to successively build a data structure referred to as a *template*, which provides efficient access to several aligned feature sequences, see Fig. 2(a).



**Figure 2.** Progressive alignment: Three aligned chroma sequences contained in the template (a) are compared to the chroma sequence (b). The resulting individual cost matrices (c) are merged into one (d), which is used to compute the alignment. The white lines in (a) and (c) indicate the positions of gap symbols.

By comparing a given feature sequence (Fig. 2(b)) to the sequences contained in the template, the alignment can be computed not only using one cost matrix (as in pairwise alignment) but several matrices in parallel - one for each sequence in the template (Fig. 2(c)). By suitably combining the information provided by each individual cost matrix, the influence of strong local differences on the alignment, that often only occur between specific pairs of versions, can be attenuated. As shown in Section 3, this can lead to a significant boost in alignment robustness.

To describe this procedure in more detail, we assume that we have  $K$  different versions of a piece and that their feature sequences are denoted by  $X^k = (x_1^k, \dots, x_{N_k}^k)$  for  $k \in [1 : K]$ . In each step of the progressive alignment, the template  $Z$  contains several of these feature sequences that have been stretched to have the same length. Initially,  $Z$  only consists of  $X^1$ . The remaining feature sequences are then successively aligned to  $Z$ , and after each alignment  $Z$  is updated by adding one more sequence. To this end, let  $\tilde{Z} = (\tilde{z}_1, \dots, \tilde{z}_{\tilde{L}})$  denote the current template which contains  $k - 1$  sequences of length  $\tilde{L}$  (i.e. each  $\tilde{z}_\ell$  contains  $k - 1$  features),  $X^k$  the sequence to be aligned, and  $p = (p_1, \dots, p_L) = ((n_1, m_1), \dots, (n_L, m_L))$  an alignment between  $\tilde{Z}$  and  $X^k$ . Intuitively, to add  $X^k$  to  $\tilde{Z}$ , we use  $p$  to stretch  $\tilde{Z}$  and  $X^k$  such that corresponding features are aligned and become part of the same element of  $Z$ . However, whenever features need to be copied to do the stretching (step sizes  $(1, 0)$  and  $(0, 1)$ ), we rather insert a special *gap* symbol instead of the features themselves. More precisely, let  $Z = (z_1, \dots, z_L)$  denote the updated template,  $z_n(k)$  denote the  $k$ -th feature in the  $n$ -th element of  $Z$ , and  $G$  denote the gap symbol<sup>1</sup>. Set  $z_1 = (\tilde{z}_1(1), \dots, \tilde{z}_1(k-1), x_1^k)$ , then for  $l = (2, 3, \dots, L)$ :

$$z_\ell = \begin{cases} (\tilde{z}_{n_\ell}(1), \dots, \tilde{z}_{n_\ell}(k-1), x_{m_\ell}^k), & p_\ell - p_{\ell-1} = (1, 1) \\ (\tilde{z}_{n_\ell}(1), \dots, \tilde{z}_{n_\ell}(k-1), G), & p_\ell - p_{\ell-1} = (1, 0) \\ (G, \dots, G, x_{m_\ell}^k), & p_\ell - p_{\ell-1} = (0, 1) \end{cases}$$

<sup>1</sup> Since chroma features contain only non-negative entries, the gap symbol can often be encoded as a pseudo-feature having negative entries.

The gap symbol and its influence will be further discussed in Section 3.

The alignment procedure itself is almost identical to standard DTW; only the local cost measure has to be adjusted to take the properties of the template into account. For a template  $Z$  comprising  $k - 1$  feature sequences and a feature sequence  $X$ , we define a template-aware cost function  $c_T : (\mathcal{F} \cup G)^{k-1} \times \mathcal{F} \rightarrow \mathbb{R}$  as

$$c_T(z_n, x_m) = \sum_{r=1}^{k-1} \begin{cases} c(z_n(r), x_m), & z_n(r) \neq G, \\ \mathcal{C}_G, & z_n(r) = G, \end{cases}$$

where  $\mathcal{C}_G > 0$  is a constant referred to as the *gap penalty*.

The influence a single additional recording can have using progressive alignment is illustrated in Fig. 1(d). Here, we included a third performance by Poblocka in the alignment, which could be considered as being “between” the two versions shown in Fig. 1 in terms of articulation style and balance. As we can see, the resulting path (black) follows the ground-truth markings (red dots) quite closely and improves significantly over the pairwise result.

### 2.3 Order of Alignments and Iterative Processing

The alignment of the first two versions in our progressive approach is equivalent to standard pairwise alignment. Errors in this first step influence to some degree all subsequent alignment steps. We discuss now two strategies that can help to increase the reliability of the first few alignments in our progressive approach. First, the order in which the alignments are computed is of importance, and we should start with recordings that are easy to align. In computational biology, a common approach to identify a reasonable order is referred to as the *guide tree approach* [12]. While there are various ways to implement such an approach, we consider the following procedure. First, for each pair of recordings, we compute the total cost of an optimal alignment between the pair to identify the pair having the lowest *average cost*, which is defined as the total cost of the alignment divided by its length  $L$ . We call the feature sequences for the recordings in this pair  $X^1$  and  $X^2$ . For the next recording, we identify the one being jointly closest to  $X^1$  and  $X^2$ . To this end, we sum for each of the remaining recordings the average cost of the alignments between the recording and  $X^1$ , and the recording and  $X^2$ . We call the feature sequence of the recording with the lowest sum  $X^3$ . We continue with this procedure until all recordings are in order. We refer to this strategy as *DTW-cost-based order*.

While this strategy leads to a useful order, its computational costs are significant. In our experiments, we found an alternative based on a much simpler strategy: We sorted the versions according to their length, starting with the shortest recordings. In the following, we refer to this strategy as *length-based order*. In Section 3, we compare both ordering strategies and discuss their behavior.

A second strategy to improve the reliability of the first alignments is referred to as *iterative progressive alignment*. The idea behind this strategy is, after all versions are aligned and included in the template, to remove one version from

ID	Piece	No. Rec.	No. Pairs
M17-4	Opus 17 No. 4	62	1891
M24-2	Opus 24 No. 2	62	1891
M30-2	Opus 30 No. 2	34	561
M63-3	Opus 63 No. 3	81	3240
M68-3	Opus 68 No. 3	49	1176

**Table 1.** Chopin Mazurkas and their identifiers used in our experiments. The last two columns indicate the number of performances available for the respective piece and the number of evaluated unique pairs.

the template and realign it, starting with the first version that was aligned. This way, errors made early in the progressive alignment can potentially be corrected. We implemented this extension as well and discuss it in Section 3.

### 2.4 Increasing the Computational Efficiency and Alignment Accuracy

Since progressive alignment shares its algorithmic roots with standard DTW, we can incorporate extensions that were successfully used with DTW-based methods. In particular, the methods described in [10, 14] employ a variant of DTW referred to as multiscale DTW (FastDTW) to increase the computational efficiency. The general idea is to recursively project an alignment computed at a coarse feature resolution level to a next higher resolution, and to refine the projected alignment on that resolution. This way, the matrix  $D$  only has to be evaluated around the projected path. This multiscale approach typically leads to a significant drop in runtime by up to a factor of 30, see [14].

Furthermore, the authors in [10] introduce a type of features that indicate onset positions separately for each chroma. These chroma-based onset features (DLNCO features) are then combined with normalized chroma features. As shown by the experiments in [10], these combined features can lead to a significant increase in alignment accuracy for pairwise methods. In the following, we employ the same features and cost measure as used in [10].

## 3. EXPERIMENTS

To illustrate the performance of our proposed method as well as the influence of certain parameters, we conducted a series of experiments using recordings taken from the Mazurka Project<sup>2</sup>, which compiled a database of over 2700 recorded performances by more than 130 distinct pianists for 49 Mazurkas composed by Frédéric Chopin. The recordings are dated between 1902 and today, and were made under strongly varying recording conditions. For our experiments, we employ a subset of five Mazurkas and 288 recordings, for which manually annotated beat positions are available, see Table 1. Performances with structural differences compared to the majority of recordings (such as additional repetitions of a part of a piece) were excluded from our experiments.

<sup>2</sup> <http://www.mazurka.org.uk>

### 3.1 Evaluation Measure

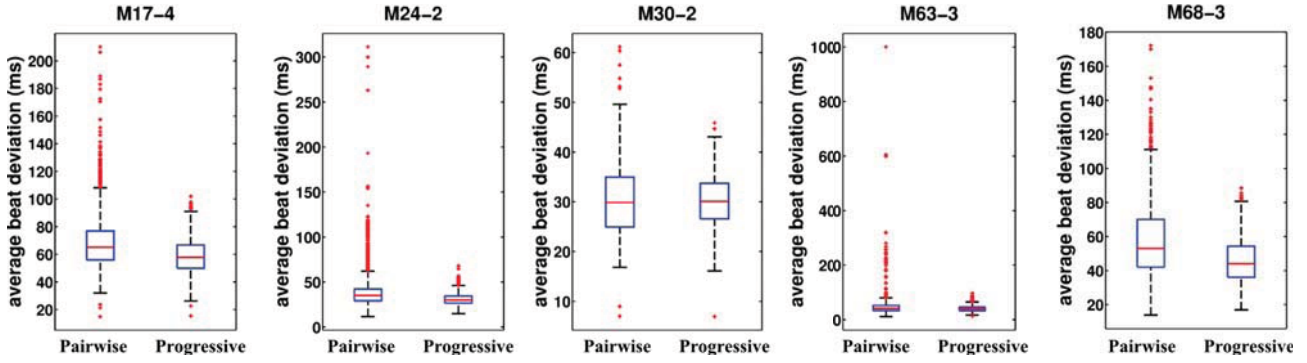
To evaluate the accuracy of an alignment between two different versions of a piece, we employ the beat annotations as ground truth. To this end, we use the alignment to locate for each annotated beat position in the one version a corresponding position in the other version. Using the manual beat annotations for the other version, we can then compute the absolute difference between the correct beat position and the one obtained from the alignment. By averaging these differences for all beats, we obtain the *average beat deviation (ABD)* for a given alignment, which we measure in milli-seconds. For our evaluation, we compute this measure for each Mazurka and each pair of recordings. For example, for M17-4 our setup contains 62 recordings, which results in  $\binom{62}{2} = 1891$  unique pairs and corresponding average beat deviation values, see Table 1.

### 3.2 Pairwise vs Progressive Alignment

In a first experiment, we compare the alignment accuracy for pairwise and progressive alignment. Since the pairwise method described in [10] employs the same features and cost measure as our proposed progressive method, we use [10] as a baseline (other pairwise methods [11] showed a similar behavior). In particular, we use a temporal resolution of 20ms for both chroma and onset-indicator (DLNCO) features. The DTW weights are set to  $(w_1, w_2, w_3) = (2, 1.5, 1.5)$ . As proposed in [10], we use the cosine distance for the chroma features and the Euclidean distance for the DLNCO features. Moreover, for our proposed progressive alignment, we use the length-based alignment order and set the gap penalty parameter to the highest value the cost measure  $c$  can assume. The distribution of the average beat deviation (ABD) values for all pairs is summarized for each of the five Mazurkas separately in the boxplots<sup>3</sup> shown in Fig. 3, as well as in column A and B in Table 2.

Comparing the results for pairwise and progressive alignment, we can see that the mean ABD drops slightly using the progressive approach for most examples. For example, the mean ABD for M17-4 drops from 68ms using pairwise alignment to 59ms using our progressive method (decrease by 13%). On average, the mean ABD drops by 14%. More importantly though, the progressive alignment is significantly more stable. In particular, the inter-quartile range is smaller for all five Mazurkas using the progressive alignment (Fig. 3). Further, the number of alignments with a very high ABD is significantly reduced. This can be measured by the standard deviation (std), which for M17-4 using pairwise alignment is 19ms, while progressive alignment leads to an std of 12ms. This difference is even greater for other Mazurkas (M24-2 and M63-3). On average, the std is reduced by more than 50%. So overall, while our proposed procedure also led to an increase in

<sup>3</sup> We use standard boxplots: the red bar indicates the median, the blue box gives the 25th and 75th percentiles ( $p_{25}$  and  $p_{75}$ ), the black bars correspond to the smallest data point greater than  $p_{25} - 1.5(p_{75} - p_{25})$  and the largest data point less than  $p_{75} + 1.5(p_{75} - p_{25})$ , and the red crosses are called outliers.



**Figure 3.** Comparison of the baseline pairwise alignment method with our proposed progressive alignment method. The boxplots illustrate the distribution of the average beat deviation values for each Mazurka separately.

M17-4	[A]	[B]	[C]	[D]	[E]	[F]	[G]
min	15	15	17	15	15	15	19
mean	68	59	68	63	76	80	91
max	210	102	118	116	789	129	252
std	19	12	13	13	94	13	22
M24-2							
min	12	15	17	12	15	16	11
mean	39	31	38	33	31	46	56
max	311	68	118	59	68	98	320
std	20	6	12	7	6	9	22
M30-2							
min	7	7	7	7	16	6	6
mean	30	30	31	29	31	40	43
max	61	46	49	53	46	64	80
std	8	5	6	6	5	7	9
M63-3							
min	11	13	15	12	13	14	9
mean	46	40	46	40	40	53	62
max	1000	97	99	99	97	109	1000
std	32	11	12	11	11	11	33
M68-3							
min	14	17	21	15	17	21	12
mean	58	46	57	53	46	71	86
max	172	89	144	105	89	179	335
std	23	13	18	15	13	21	34

**Table 2.** Statistics over the average beat deviation (ABD) values for the five Mazurkas and for 7 different alignment approaches (see text). [A]: Pairwise alignment. [B]: Proposed progressive alignment. [C]: Proposed without gap symbols. [D]: Proposed using DTW-cost-based alignment order. [E]: Proposed using iterative alignment. [F]: Proposed without DLNCO features. [G]: Pairwise without DLNCO features. All values in milli-seconds.

alignment accuracy on average, the main effect is a gain in robustness against strongly incorrect alignments.

### 3.3 Gap Penalties

In the next experiment, we investigate the influence of the gap penalty parameter by testing a slightly modified version of our proposed method. To this end, we modify the way the template is created by setting  $z_\ell = (\tilde{z}_{n_\ell}(1), \dots, \tilde{z}_{n_\ell}(k-1), x_{m_\ell}^k)$  for  $\ell \in [1:L]$ , i.e. we do not insert gap symbols but copy features as necessary to create the new template (comparing to Section 2.2). The results using this modification are shown in column C in Table 2. Comparing these values to our proposed method (column B) and the reference pairwise method (column A), we see

that this gap-less version typically improves over pairwise alignment in terms of maximum ABD values and the standard deviation, just as the proposed method. For example, for M17-4, the max ABD in column A is 210ms, while the max ABD in column C is 118ms. However, we do not observe a decrease in the mean ABD compared to pairwise alignment. For example, for M17-4, while using gaps the mean ABD drops from 68ms (column A) to 59ms (column B), it stays on a similar level in column C (68ms). The reason could be that by copying the features to create the template, some temporal precision is lost and this results in a minor loss of alignment accuracy.

### 3.4 Alignment Order

Next, we investigate the influence of the order in which we compute the progressive alignment, comparing the length-based and the DTW-cost-based strategy (see Section 2.3). The results are given in columns B and D of Table 2, respectively. As we can see, there are no significant differences between both strategies. For example, for M17-4, the mean ABD using the length-based strategy is 59ms (column B), while using the DTW-cost-based strategy the ABD slightly increases to 63ms. The other statistical values show a similar behavior. Since these results do not disclose any obvious advantages for the DTW-cost-based strategy, we therefore propose to simply use the length-based strategy. Interestingly, using the length-based strategy but starting with the longest recordings led to worse results.

Since (local) tempo differences can usually be handled quite well using DTW, it is not obvious why sorting by length yields a useful order. However, the fact that it does could indicate that there might be a correlation between the chosen tempo and other expressive parameters, such as articulation or balance, as strong differences in these parameters typically lead to difficulties for the alignment. Furthermore, the fact that according to our evaluation the shorter recordings were easier to align, could indicate that a high tempo could limit the range of possible realizations of expressive parameters in a performance. However, further studies would be necessary to confirm such theories.

### 3.5 Iterative Alignment

In a further experiment, we investigate whether iterative processing could further improve the alignment accuracy, compare Section 2.3. To this end, we use two iterations: the first iteration corresponds to progressive alignment, and in the second iteration, each version is removed from the template once and is then realigned. The results for this extension are given in column E of Table 2. Overall, the iterative variant led to a slight decrease in ABD in almost all examples, which is not even visible in Table 2 as we rounded all values. On the contrary, we observed a significant increase in ABD for M17-4 using the iterative variant. Here, the realignment led to a misalignment of several shorter recordings. Therefore, the results do not indicate any significant advantages of using iterative alignment.

### 3.6 Influence of Onset-Indicator Features

In a final experiment, we investigate the influence of the chroma-based onset-indicator (DLNCO) features [10] on the alignment accuracy when using progressive alignment. To this end, we disabled the DLNCO features in our proposed method, and computed the alignment only based on the normalized chroma features. The results of this experiment are given in column F in Table 2. As a further reference, we disabled the DLNCO features in our baseline pairwise method as well (column G).

As we can see, the minimum over the ABD values remains unaffected for most of the Mazurkas, which means that easy to align pairs can be aligned with chroma features alone just as well. For example, for M17-4, the minimum value in column F is identical to the one in column B. However, we see a significant increase in ABD in all other statistical values. For example, the mean ABD for M17-4 for our proposed method including DLNCO features is 59ms (column B), while disabling the DLNCO leads to a mean ABD of 80ms (column F). Similar observations can be made comparing the pairwise results. Overall, the results seem to indicate that including onset-indicator features indeed leads to a significant increase in alignment accuracy also for progressive alignments.

## 4. CONCLUSION

In this paper, we introduced a method for aligning multiple versions of a piece of music in a joint way. The availability of multiple versions to compare against during the alignment, stabilized the comparison for hard-to-align recordings and led to an overall increase in alignment accuracy and, in particular, in alignment robustness. Our experiments using real-world recordings from the Mazurka Project demonstrated that our proposed method can indeed be used to raise the alignment accuracy compared to previous methods that are limited to pairwise alignments. For the future, we plan to further investigate the behaviour of our procedure. In particular, we plan to analyze how other ordering strategies influence the alignment accuracy. We will also further explore different strategies to implement a cost for the gap symbol and to make it more adaptive.

**Acknowledgements:** This work was partly funded by the China Scholarship Council (CSC), EPSRC Grant EP/J010375/1, and the Queen Mary Postgraduate Research Fund (PGRF).

## 5. REFERENCES

- [1] M. Müller, M. Clausen, V. Konz, S. Ewert, and C. Fremerey, "A multimodal way of experiencing and exploring music," *Interdisciplinary Science Reviews (ISR)*, vol. 35, no. 2, pp. 138–153, 2010.
- [2] R. B. Dannenberg and C. Raphael, "Music score alignment and computer accompaniment," *Communications of the ACM, Special Issue: Music Information Retrieval*, vol. 49, no. 8, pp. 38–43, 2006.
- [3] A. Arzt, S. Böck, S. Flossmann, H. Frostel, M. Gasser, and G. Widmer, "The complete classical music companion v0.9," in *Proceedings of the AES International Conference on Semantic Audio*, London, UK, 18–20 2014, pp. 133–137.
- [4] N. Montecchio and A. Cont, "A unified approach to real time audio-to-score and audio-to-audio alignment using sequential Montecarlo inference techniques," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011, pp. 193–196.
- [5] Z. Duan and B. Pardo, "A state space model for online polyphonic audio-score alignment," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011, pp. 197–200.
- [6] G. Widmer, S. Dixon, W. Goebel, E. Pampalk, and A. Tobudic, "In search of the Horowitz factor," *AI Magazine*, vol. 24, no. 3, pp. 111–130, 2003.
- [7] J. Serrà, E. Gómez, P. Herrera, and X. Serra, "Chroma binary similarity and local alignment applied to cover song identification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, pp. 1138–1151, 2008.
- [8] S. Ewert, B. Pardo, M. Müller, and M. D. Plumbley, "Score-informed source separation for musical audio recordings: An overview," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 116–124, May 2014.
- [9] C. Joder, S. Essid, and G. Richard, "A conditional random field framework for robust and scalable audio-to-score matching," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2385–2397, 2011.
- [10] S. Ewert, M. Müller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, 2009, pp. 1869–1872.
- [11] S. Dixon and G. Widmer, "MATCH: A music alignment tool chest," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, London, GB, 2005, pp. 492–497.
- [12] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. New York, USA: Cambridge University Press, 1999.
- [13] H. I. Robertson, "Testing a new tool for alignment of musical recordings," Master's thesis, McGill University, 2013.
- [14] M. Müller, H. Mattes, and F. Kurth, "An efficient multiscale approach to audio synchronization," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Victoria, Canada, 2006, pp. 192–197.



# FORMALIZING THE PROBLEM OF MUSIC DESCRIPTION

**Bob L. Sturm**  
Aalborg University  
Denmark

bst@create.aau.dk

**Rolf Bardeli**  
Fraunhofer IAIS  
Germany

rolf.bardeli@iais.fraunhofer.de

**Thibault Langlois**  
Lisbon University  
Portugal

tl@di.fc.ul.pt

**Valentin Emiya**  
Aix-Marseille Université  
CNRS UMR 7279 LIF

valentin.emiya@lif.univ-mrs.fr

## ABSTRACT

The lack of a formalism for “the problem of music description” results in, among other things: ambiguity in what problem a music description system must address, how it should be evaluated, what criteria define its success, and the paradox that a music description system can reproduce the “ground truth” of a music dataset without attending to the music it contains. To address these issues, we formalize the problem of music description such that all elements of an instance of it are made explicit. This can thus inform the building of a system, and how it should be evaluated in a meaningful way. We provide illustrations of this formalism applied to three examples drawn from the literature.

## 1. INTRODUCTION

Before one can address a problem with an algorithm (a finite series of well-defined operations that transduce a well-specified input into a well-specified output) one needs to define and decompose that problem in a way that is compatible with the formal nature of algorithms [17]. A very simple example is the problem of adding any two positive integers. Addressing this problem with an algorithm entails defining the entity “positive integer”, the function “adding”, and then producing a finite series of well-defined operations that applies the function to an input of two positive integers to output the correct positive integer.

A more complex example is “the problem of music description.” While much work in music information retrieval (MIR) has proposed systems to attempt to address the problem of music description [4, 12, 29], and much work attempts to evaluate the capacity of these systems for addressing that problem [9, 20], we have yet to find any work that actually *defines* it. (The closest we have found is that of [24].) Instead, there are many allusions to the problem: predict the “genre” of a piece of recorded music [25]; label music with “useful tags” [1]; predict what a listener will “feel” when “listening” to some music [29]; find music “similar” to some other music [26]. These allusions are deceptively simple, however, since behind them lie many

problems and questions that have major repercussions on the design and evaluation of any proposed system. For example, What is “genre”? What is “useful”? How is “feeling” related to “listening”? “Similar” in what respects?

With respect to the problem of music description, some work in MIR discusses the meaningfulness, worth, and futility of designing artificial systems to describe music [28]; the idea of and the difficulty in “ground truth” [3, 6, 15]; the size of datasets [5], a lack of statistics [10], the existence of bias [16], and the ways such systems are evaluated [21, 22, 27]. Since a foundational goal of MIR is to develop systems that can imitate the human ability to describe music, these discussions are necessary. However, what remains missing is a formal definition of the problem of music description such that it can be addressed by algorithms, and relevant and valid evaluations can be designed.

In this work, we formalize the problem of music description and try to avoid ambiguity arising from semantics. This leads to a rather abstract form, and so we illustrate its aspects using examples from the literature. The most practical benefit of our formalization is a specification of all elements that should be explicitly defined when addressing an instance of the problem of music description.

## 2. FORMALISM

We start our formalization by defining the domain of the problem of music description. In particular, we discriminate between the music that is to be described and a recording of it since the former is intangible and the latter is data that a system can analyze. We then define the problem of music description, a recorded music description system (RMDS), and the analysis of such a system. This leads to the central role of the use case.

### 2.1 Domain

Denote a *music universe*,  $\Omega$ , a set of music, e.g., Vivaldi’s “The Four Seasons”, the piano part of Gershwin’s “Rhapsody in Blue”, and the first few measures of the first movement of Beethoven’s Fifth Symphony. A member of  $\Omega$  is intangible. One cannot hear, see or point to any member of  $\Omega$ ; but one can hear a performance of Vivaldi’s “The Four Seasons”, read sheet music notating the piano part of Gershwin’s “Rhapsody in Blue”, and point to a printed score of Beethoven’s Fifth Symphony. Likewise, a recorded performance of Vivaldi’s “The Four Seasons” is *not* Vivaldi’s “The Four Seasons”, and sheet music notating the piano part of Gershwin’s “Rhapsody in Blue” is *not* the piano part of Gershwin’s “Rhapsody in Blue”.



© Bob L. Sturm, Rolf Bardeli, Thibault Langlois, Valentin Emiya.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Bob L. Sturm, Rolf Bardeli, Thibault Langlois, Valentin Emiya. “Formalizing the Problem of Music Description”, 15th International Society for Music Information Retrieval Conference, 2014.

In the tangible world, there may exist tangible recordings of the members of  $\Omega$ . Denote the *tangible music recording universe* by  $\mathcal{R}_\Omega$ . A member of  $\mathcal{R}_\Omega$  is a recording of an element of  $\omega \in \Omega$ . A *recording* is a tangible object, such as a printed CD or score. Denote one recording of  $\omega \in \Omega$  as  $r_\omega \in \mathcal{R}_\Omega$ . There might be many recordings of an  $\omega$  in  $\mathcal{R}_\Omega$ . We say the music  $\omega$  is *embedded* in  $r_\omega$ ; it enables for a listener an indirect sense of  $\omega$ . For instance, one can hear a live or recorded performance of  $\omega$ , and one can read a printed score of  $\omega$ . The acknowledgment of and distinction between intangible music and tangible recordings of music is essential since systems cannot work with intangible music, but only tangible recordings.

## 2.2 Music Description and the Use Case

Denote a *vocabulary*,  $\mathcal{V}$ , a set of symbols or tokens, e.g., “Baroque”, “piano”, “knock knock”, scores employing common practice notation, the set of real numbers  $\mathbb{R}$ , other music recordings, and so on. Define the *semantic universe* as

$$\mathcal{S}_{\mathcal{V},A} := \{s = (v_1, \dots, v_n) | n \in \mathbb{N}, \forall 1 \leq i \leq n [v_i \in \mathcal{V}] \wedge A(s)\} \quad (1)$$

where  $A(\cdot)$  encompasses a semantic rule, for instance, restricting  $\mathcal{S}_{\mathcal{V},A}$  to consist of sequences of cardinality 1. Note that the *description*  $s$  is a sequence, and not a vector or a set. This permits descriptions that are, e.g., time-dependent, such as envelopes, if  $\mathcal{V}$  and  $A(\cdot)$  permit it. In that case, the order of elements in  $s$  could be alternating time values with envelope values. Descriptions could also be time-frequency dependent.

We define *music description* as pairing an element of  $\Omega$  or  $\mathcal{R}_\Omega$  with an element of  $\mathcal{S}_{\mathcal{V},A}$ . The *problem of music description* is to make the pairing *acceptable with respect to a use case*. A *use case* provides specifications of  $\Omega$  and  $\mathcal{R}_\Omega$ ,  $\mathcal{V}$  and  $A(\cdot)$ , and success criteria. *Success criteria* describe how music or a music recording should be paired with an element of the semantic universe, which may involve the sanity of the decision (e.g., tempo estimation must be based on the frequency of onsets), the efficiency of the decision (e.g., pairing must be produced under 100 ms with less than 10 MB of memory), or other considerations.

To make this clearer, consider the following use case. The music universe  $\Omega$  consists of performances by Buckwheat Zydeco, movements of Vivaldi’s “The Four Seasons”, and traditional Beijing opera. The tangible music recording universe  $\mathcal{R}_\Omega$  consists of all possible 30-second digital audio recordings of the elements in  $\Omega$ . Let the vocabulary  $\mathcal{V} = \{\text{“Blues”, “Classical”}\}$ ; and define  $A(s) := [|s| \in \{0, 1\}]$ . The semantic universe is thus,  $\mathcal{S}_{\mathcal{V},A} = \{(), (\text{“Blues”}), (\text{“Classical”})\}$ . There are many possible success criteria. One is to map all recordings of Buckwheat Zydeco to “Blues”, map all recordings of Vivaldi’s “The Four Seasons” to “Classical”, and map all recordings of traditional Beijing opera to neither. Another is to map no recordings of Buckwheat Zydeco and Vivaldi’s “The Four Seasons” to the empty sequence, and to map any recording of traditional Beijing opera to either non-empty sequence with a probability less than 0.1.

## 2.3 Recorded Music Description Systems

A *recorded music description system* (RMDS) is a map from the tangible music recording universe to the semantic universe:

$$\mathcal{S} : \mathcal{R}_\Omega \rightarrow \mathcal{S}_{\mathcal{V},A}. \quad (2)$$

*Building* an RMDS means making a map according to well-specified criteria, e.g., using expert domain knowledge, automatic methods of supervised learning, and a combination of these. An instance of an RMDS is a specific map that is already built, and consists of four kinds of components [21]: algorithmic (e.g., feature extraction, classification, pre-processing), instruction (e.g., description of  $\mathcal{R}_\Omega$  and  $\mathcal{S}_{\mathcal{V},A}$ ), operator(s) (e.g., the one inputting data and interpreting output), and environmental (e.g., connections between components, training datasets). It is important to note that  $\mathcal{S}$  is not restricted to map any recording to a single element of  $\mathcal{V}$ . Depending on  $\mathcal{V}$  and  $A(\cdot)$ ,  $\mathcal{S}_{\mathcal{V},A}$  could consist of sequences of scalars and vectors, sets and sequences, functions, combinations of all these, and so on.  $\mathcal{S}$  could thus map a recording to many elements of  $\mathcal{V}$ .

One algorithmic component of an RMDS is a *feature extraction algorithm*, which we define as

$$\mathcal{E} : \mathcal{R}_\Omega \rightarrow \mathcal{S}_{\mathbb{F},A'} \quad (3)$$

i.e., a map from  $\mathcal{R}_\Omega$  to a semantic universe built from the vocabulary of a feature space  $\mathbb{F}$  and semantic rule  $A'(\cdot)$ . For instance, if  $\mathbb{F} := \mathbb{C}^M$ ,  $M \in \mathbb{N}$ , and  $A'(s) := [|s| = 1]$ , then the feature extraction maps a recording to a single  $M$ -dimensional complex vector. Examples of such a map are the discrete Fourier transform, or a stacked series of vectors of statistics of Mel frequency cepstral coefficients. Another algorithmic component of an RMDS is a *classification algorithm*, which we define:

$$\mathcal{C} : \mathcal{S}_{\mathbb{F},A'} \rightarrow \mathcal{S}_{\mathcal{V},A} \quad (4)$$

i.e., a map from one semantic universe to another. Examples of such a map are  $k$ -nearest neighbor, maximum likelihood, support vector machine, and a decision tree.

To make this clearer, consider the RMDS named “RT GS” built by Tzanetakis and Cook [25].  $\mathcal{E}$  maps sampled audio signals of about 30-s duration to  $\mathcal{S}_{\mathbb{F},A'}$ , defined by single 19-dimensional vectors, where one dimension is spectral centroid mean, another is spectral centroid variance, and so on.  $\mathcal{C}$  maps  $\mathcal{S}_{\mathbb{F},A'}$  to  $\mathcal{S}_{\mathcal{V},A}$ , which is defined by  $\mathcal{V} = \{\text{“Blues”, “Classical”, “Country”, “Disco”, “Hip hop”, “Jazz”, “Metal”, “Pop”, “Reggae”, “Rock”}\}$ , and  $A(s) := [|s| = 1]$ . This mapping involves maximizing the likelihood of an element of  $\mathcal{S}_{\mathbb{F},A'}$  among ten multivariate Gaussian models created with supervised learning.

Supervised learning involves automatically building components of an  $\mathcal{S}$ , or defining  $\mathcal{E}$  and  $\mathcal{C}$ , given a *training recorded music dataset*: a sequence of tuples of recordings sampled from  $\mathcal{R}_\Omega$  and elements of  $\mathcal{S}_{\mathcal{V},A}$ , i.e.,

$$\mathcal{D} := \{(r_i, s_i) \in \mathcal{R}_\Omega \times \mathcal{S}_{\mathcal{V},A} | i \in \mathcal{I}\} \quad (5)$$

The set  $\mathcal{I}$  indexes the dataset. We call the sequence  $(s_i)_{i \in \mathcal{I}}$  the *ground truth* of  $\mathcal{D}$ . In the case of RT GS, its training

recorded music dataset contains 900 tuples randomly selected from the dataset *GTZAN* [22, 25]. These are selected in a way such that the ground truth of  $\mathcal{D}$  has no more than 100 of each element of  $\mathcal{S}_{\mathcal{V},\mathcal{A}}$ .

## 2.4 Analysis of Recorded Music Description Systems

Given an RMDS, one needs to determine whether it addresses the problem of music description. Simple questions to answer are: does  $\Omega$  and  $\mathcal{R}_\Omega$  of the RMDS encompass those of the use case? Does the  $\mathcal{S}_{\mathcal{V},\mathcal{A}}$  of the RMDS encompass that of the use case? A more complex question could be, does the RMDS meet the success criteria of the use case? This last question involves the design, implementation, analysis, and interpretation of valid experiments that are relevant to answering hypotheses about the RMDS and success criteria [21, 27]. Answering these questions constitutes an *analysis* of an RMDS.

Absent explicit success criteria of a use case, a standard approach for evaluating an RMDS is to compute a variety of *figures of merit* (FoM) from its “treatment” of the recordings of a testing  $\mathcal{D}$  that exemplify the input/output relationships sought. Examples of such FoM are mean classification accuracy, precisions, recalls, and confusions. An implicit belief is that the correct output will be produced from the input only if an RMDS has learned criteria relevant to describing the music. Furthermore, it is hoped that the resulting FoM reflect the real world performance of an RMDS. The *real world performance* of an RMDS are the FoM that result from an experiment using a testing recording music dataset consisting of *all* members in  $\mathcal{R}_\Omega$ , rather than a sampling of them. If this dataset is out of reach, statistical tests can be used to determine significant differences in performance between two RMDS (testing the null hypothesis, “neither RMDS has ‘learned better’ than the other”), or between the RMDS and that of picking an element of  $\mathcal{S}_{\mathcal{V},\mathcal{A}}$  independent of the element from  $\mathcal{R}_\Omega$  (testing the null hypothesis, “The RMDS has learned nothing”). These statistical tests are accompanied by implicit and strict assumptions on the measurement model and its appropriateness to describe the measurements made in the experiment [2, 8].

As an example, consider the evaluation of RT GS discussed above [25]. The evaluation constructs a testing  $\mathcal{D}$  from the 100 elements of the dataset *GTZAN* not present in the training  $\mathcal{D}$  used to create the RMDS. They treat each of the 100 recordings in the testing  $\mathcal{D}$  with RT GS, and compare its output with the ground truth. From these 100 comparisons, they compute the percentage of outputs that match the ground truth (accuracy). Whether or not this is a high-quality estimate of the real world accuracy of RT GS depends entirely upon the definition of  $\Omega$ ,  $\mathcal{R}_\Omega$ ,  $\mathcal{S}_{\mathcal{V},\mathcal{A}}$ , as well as the testing  $\mathcal{D}$  and the measurement model of the experiment.

There are many serious dangers to the interpretation of the FoM of an RMDS as reflective of its real world performance: noise in the measurements, an inappropriate measurement model [2], a poor experimental design and errors of the third kind [14], the lack of error bounds or error bounds that are too large [8], and several kinds of

bias. One kind of bias comes from the very construction of testing datasets. For instance, if the testing dataset is the same as the training dataset, and the set of recordings in the dataset is a subset of  $\mathcal{R}_\Omega$ , then the FoM of an RMDS computed from the treatment may not indicate its real world performance. This has led to the prescription in machine learning to use a testing dataset that is disjoint with the training dataset, by partitioning for instance [13]. This, however, may not solve many other problems of bias associated with the construction of datasets, or increase the relevance of such an experiment with measuring the extent to which an RMDS has learned to describe the music in  $\Omega$ .

## 2.5 Summary

Table 1 summarizes all elements defined in our formalization of the problem of music description, along with examples of them. These are the elements that must be explicitly defined in order to address an instance of the problem of music description by algorithms. Central to many of these are the definition of a use case, which specifies the music and music recording universe, the vocabulary, the desired semantic universe, *and* the success criteria of an acceptable system. (Note that “use case” is not the same as “user-centered.”) If the use case is not unambiguously specified, then a successful RMDS cannot be constructed, relevant and valid experiments cannot be designed, and the analysis of an RMDS cannot be meaningful. Table 1 can serve as a checklist for the extent to which an instance of the problem of music description is explicitly defined.

## 3. APPLICATION

We now discuss two additional published works in the MIR literature in terms of our formalism.

### 3.1 Dannenberg et al. [7]

The use cases of the RMDS employed by Dannenberg et al. [7] are motivated by the desire for a mode of communication between a human music performer and an accompanying computer that is more natural than physical interaction. The idea is for the computer to employ an RMDS to describe the acoustic performance of a performer in terms of several “styles.” Dannenberg et al. circumvent the need to define any of these “styles” by noting, “what really matters is the ability of the performer to consistently produce intentional and different styles of playing at will” [7]. As a consequence, the use cases and thus system analysis are centered on the performer.

One use case considered by Dannenberg et al. defines  $\mathcal{V} = \{\text{“lyrical”, “frantic”, “syncopated”, “pointillistic”, “blues”, “quote”, “high”, “low”}\}$ , and the semantic rule  $A(s) := [|s| \in \{1\}]$ . The semantic universe  $\mathcal{S}_{\mathcal{V},\mathcal{A}}$  is then all single elements of  $\mathcal{V}$ . The music universe  $\Omega$  is all possible music that can be played or improvised by the specific performer in these “styles.” The tangible music recording universe  $\mathcal{R}_\Omega$  is all possible 5-second acoustic recordings of the elements of  $\Omega$ . Finally, the success criteria of this particular problem of music description includes the following requirements: reliable for a specific performer in an interactive performance, classifier latency of under

Element (Symbol)	Definition	Example
music universe ( $\Omega$ )	a set of (intangible) music	{“Automatic Writing” by R. Ashley}
tangible music recording universe ( $\mathcal{R}_\Omega$ )	a set of tangible recordings of all members of $\Omega$	{R. Ashley, “Automatic Writing”, LCD 1002, Lovely Music, Ltd., 1996}
recording ( $r_\omega$ )	a member of $\mathcal{R}_\Omega$	a 1-second excerpt of the 46 minute recording of “Automatic Writing” from LCD 1002
vocabulary ( $\mathcal{V}$ )	a set of symbols	{“Robert”, “french woman”, “bass in other room”, “Moog”} $\cup$ [0, 2760]
semantic universe ( $\mathcal{S}_{\mathcal{V},A}$ )	$\{s = (v_1, \dots, v_n)   n \in \mathbb{N}, \forall 1 \leq i \leq n [v_i \in \mathcal{V}] \wedge A(s)\}$ , i.e., the set of all sequences of symbols from $\mathcal{V}$ permitted by the semantic rule $A(\cdot)$	{(“Robert”, 1), (“Robert”, “Moog”, 4.3), (“french woman”, 104.3), (“french woman”, “Moog”, 459), ...}
semantic rule ( $A(s)$ )	a Boolean function that defines when sequence $s$ is “permissible”	$A(s) := [( s  \in \{2, 3, 4, 5\}) \wedge (\{v_1, \dots, v_{ s -1}\} \subseteq \{\text{“Robert”, “french woman”, “bass in other room”, “Moog”} \cup \{\}\}) \wedge (v_{ s } \in [0, 2760])]$
music description	the pairing of an element of $\Omega$ or $\mathcal{R}_\Omega$ with an element of $\mathcal{S}_{\mathcal{V},A}$	label the events (character, time) in recording LCD 1002 of “Automatic Writing” by R. Ashley
the problem of music description	make this pairing acceptable with respect to the success criteria specified by the use case	make this pairing such that F-score of event “Robert” is at least 0.9
use case	specification of $\Omega$ , $\mathcal{R}_\Omega$ , $\mathcal{V}$ , $A(s)$ , and success criteria	see all above
system	a connected set of interacting and interdependent components of four kinds (operator(s), instructions, algorithms, environment) that together address a use case	system created in the Audio Latin Genre Classification task of MIREX 2013 by organizer from submission “API” and fold 1 of LMD [18]
operators	agent(s) that employ the system, inputting data, and interpreting outputs	Audio Latin Genre Classification organizer of MIREX 2013
instructions	specifications for the operator(s), like an application programming interface	MIREX 2013 input/output specifications for Train/Test tasks; “README” file included with “API”
algorithm	a finite series of well-defined ordered operations to transduce an input into an output	“Training.m” and “Classifying.m” MATLAB scripts in “API”, etc.
environment	connections between components, external databases, the space within which the system operates, its boundaries	folds 2 and 3 of LMD [18], MIREX computer cluster, local MATLAB license file, etc.
recorded music description system (RMDS) ( $\mathcal{S}$ )	$\mathcal{S} : \mathcal{R}_\Omega \rightarrow \mathcal{S}_{\mathcal{V},A}$ , i.e., a map from $\mathcal{R}_\Omega$ to $\mathcal{S}_{\mathcal{V},A}$	“RT GS” evaluated in [25]
feature extraction algorithm ( $\mathcal{E}$ )	$\mathcal{E} : \mathcal{R}_\Omega \rightarrow \mathcal{S}_{\mathbb{F},A'}$ , i.e., a map from $\mathcal{R}_\Omega$ to an element of a semantic universe based on the feature vocabulary $\mathbb{F}$ and semantic rule $A'(s)$	compute using [19] the first 13 MFCCs (including zeroth coefficient) from a recording
feature vocabulary ( $\mathbb{F}$ )	a set of symbols	$\mathbb{R}^{13}$
classification algorithm ( $\mathcal{C}$ )	$\mathcal{C} : \mathcal{S}_{\mathbb{F},A'} \rightarrow \mathcal{S}_{\mathcal{V},A}$ , i.e., a map from $\mathcal{S}_{\mathbb{F},A'}$ to the semantic universe	single nearest neighbor
recorded music dataset	$\mathcal{D} := (\{r_\omega \in \mathcal{R}_\Omega, s \in \mathcal{S}_{\mathcal{V},A}\}_{i \in \mathcal{I}})$ , i.e., a sequence of tuples of recordings and elements of the semantic universe, indexed by $\mathcal{I}$	GTZAN [22, 25]
“ground truth” of $\mathcal{D}$	$(s_i)_{i \in \mathcal{I}}$ , i.e., the sequence of “true” elements of the semantic universe for the recordings in $\mathcal{D}$	in GTZAN: {“blues”, “blues”, ..., “classical”, ..., “country”, ...}
analysis of an RMDS	answering whether an RMDS can meet the success criteria of a use case with relevant and valid experiments	designing, implementing, analyzing and interpreting experiments that validly answer, “Can RT GS [25] address the needs of user A?”
experiment	principally in service to answering a scientific question, the mapping of one or more RMDS to recordings of $\mathcal{D}$ , and the making of measurements	apply RT GS to GTZAN, compare its output labels to “ground truth”, and compute accuracy
figure of merit (FoM)	performance measurement of an RMDS from an experiment	classification accuracy of RT GS in GTZAN
real world performance of an RMDS	the figure of merit expected if an experiment with an RMDS uses all of $\mathcal{R}_\Omega$	classification accuracy of RT GS

**Table 1.** Summary of all elements defined in the formalization of the problem of music description, with examples.

5 seconds. The specific definition of “reliable” might include high accuracy, high precision in every class, or only in some classes.

Dannenberg et al. create an RMDS by using a training dataset of recordings curated from actual performances, as well as collected in a more controlled fashion in a laboratory. The ground truth of the dataset is created with

input from performers. The feature extraction algorithm includes algorithms for pitch detection, MIDI conversion, and the computation of 13 low-level features from the MIDI data. One classification algorithm employed is maximum likelihood using a naive Bayesian model.

The system analysis performed by Dannenberg et al. involve experiments measuring the mean accuracy of all sys-

tems created and tested with 5-fold cross validation. Furthermore, they evaluate a specific RMDS they create in the context of a live music performance. From this they observe three things: 1) the execution time of the RMDS is under 1 ms; 2) the FoM of the RMDS found in the laboratory evaluation is too optimistic for its real world performance in the context of live performance; 3) using the confidence of the classifier and tuning a threshold parameter provides a means to improve the RMDS by reducing its number of false positives.

### 3.2 Turnbull et al. [24]

Turnbull et al. [24] propose several RMDS that work with a vocabulary consisting of 174 unique “musically relevant” words, such as “Genre–Brit\_Pop”, “Usage–Reading”, and “NOT–Emotion–Bizarre/\_Weird”.  $A(s) := [|s| = 10 \wedge \forall i \neq j (v_i \neq v_j)]$ , and so the elements of  $S_{\mathcal{V},A}$  are tuples of ten unique elements of  $\mathcal{V}$ . The music universe  $\Omega$  consists of at least 502 songs (the size of the CAL500 dataset), such as “S.O.S.” performed by ABBA, “Sweet Home Alabama” performed by Lynyrd Skynyrd, and “Fly Me to the Moon” sung by Frank Sinatra. The tangible music recording universe  $\mathcal{R}_\Omega$  is composed of MP3-compressed recordings of entire music pieces. The RMDS sought by Turnbull et al. aims “[to be] good at predicting all the words [in  $\mathcal{V}$ ]”, or “produce sensible semantic annotations for an acoustically diverse set of songs.” Since “good”, “sensible” and “acoustically diverse” are not defined, the success criteria is ambiguous.  $\Omega$  is also likely much larger than 502 songs.

The feature extraction algorithm in the RMDS of Turnbull et al. maps a music recording to a semantic universe built from a feature vocabulary  $\mathbb{F} := \mathbb{R}^{39}$ , and the semantic rule  $A'(s) := [|s| = 10000]$ . That is, the algorithm computes from an audio recording 13 MFCC coefficients on 23ms frames, concatenates the first and second derivatives in each frame, and randomly selects 10000 feature vectors from all those extracted. The classification algorithm in the RMDS uses a maximum a posteriori decision criterion, with conditional probabilities of features modelled by a Gaussian mixture model (GMM) of a specified order. One RMDS uses expectation maximization to estimate the parameters of an 8-order GMM from a training dataset.

Turnbull et al. build an RMDS using a training dataset of 450 elements selected from CAL500. They apply this RMDS to the remaining elements of CAL500, and measure how its output compares to the ground truth. When the ground truth of a recording in CAL500 does not have 10 elements per the semantic rule of the semantic universe, Turnbull et al. randomly add unique elements of  $\mathcal{V}$ , or randomly remove elements from the ground truth of the recording until it has cardinality 10.

Turnbull et al. compute from an experiment FoM such as mean per-word precision. Per-word precision is, for a  $v \in \mathcal{V}$  and when defined, the percentage of correct mappings of the system from the recordings in the test dataset to an element of the semantic universe that includes  $v$ . Mean per-word precision is thus the mean of the  $|\mathcal{V}|$  per-word precisions. Turnbull et al. compare the FoM of the RMDS to other systems, such as a random classifier and

a human. They conclude that their best RMDS is slightly worse than human performance on “more ‘objective’ semantic categories [like instrumentation and genre]” [24]. The evaluation, measuring the amount of ground truth reproduced by a system (human or not) and not the sensibility of the annotations, has questionable relevance and validity to the ambiguous use case.

## 4. CONCLUSION

Formalism can reveal when a problem is not adequately defined, and how to explicitly define it in no uncertain terms. An explicit definition of a problem shows how to evaluate solutions in relevant and valid ways. It is in this direction that we move with this paper for the problem of music description, the spirit of which is encapsulated by Table 1. The unambiguous definition of the use case is central for addressing an instance of the problem of music description.

We have discussed several published RMDS within this formalism. The work of Dannenberg et al. [7] provides a good model since its use case and analysis are clearly specified — both center on a specific music performer — and through evaluating the system in the real world they actually complete the research and development cycle to improve the system [27]. The use cases of the RMDS built by Tzanetakis and Cook [25] and Turnbull et al. [24] are not specified. In both cases, a labeled dataset is assumed to provide sufficient definition of the problem. Turnbull et al. suggest a success criterion of annotations being “sensible,” but the evaluation only measures the amount of ground truth reproduced. Due to the lack of definition, we are thus unsure what problem either of these RMDS is actually addressing, or whether either of them is actually considering the music [23]. An analysis of an RMDS depends on an explicit use case. The definition of the use case in Dannenberg et al. [7] renders this question irrelevant: all that is needed is that the RMDS meets the success criteria of a given performer, which is tested by performing with it.

While we provide in this paper a formalization of the problem of music description, and a checklist of the components necessary to define an instance of such a problem, it does not describe how to solve any specific problem of music description. We do not derive restrictions on any of the components of the problem definition, or show how datasets should be constructed to guarantee an evaluation can result in good estimates of real world performance. Our future work aims in these directions. We will incorporate the formalism of the design and analysis of comparative experiments [2,21], which will help define the notions of relevance and validity when it comes to analyzing RMDS. We seek to incorporate notions of learning and inference [13], e.g., to specify what constitutes the building of a “good” RMDS using a training dataset (where “good” depends on the use case). We also seek to explain more formally two paradoxes that have been observed. First, though an RMDS is evaluated in a test dataset to reproduce a large amount of ground truth, it appears to not be a result of the consideration of characteristics in the music universe [20]. Second, though artificial algorithms have

none of the extensive experience humans have in music listening, description, and culture, they can reproduce ground truth consisting of extremely subjective and culturally centered concepts like genre [11].

## 5. ACKNOWLEDGMENTS

The work of BLS and VE is supported in part by l’Institut français du Danemark, and ARCHIMEDE Labex (ANR-11-LABX- 0033).

## 6. REFERENCES

- [1] J.-J. Aucouturier and E. Pampalk. Introduction – from genres to tags: A little epistemology of music information retrieval research. *J. New Music Research*, 37(2):87–92, 2008.
- [2] R. A. Bailey. *Design of comparative experiments*. Cambridge University Press, 2008.
- [3] M. Barthelet, G. Fazekas, and M. Sandler. Multidisciplinary perspectives on music emotion recognition: Implications for content and context-based models. In *Proc. CMMR*, 2012.
- [4] T. Bertin-Mahieux, D. Eck, and M. Mandel. Automatic tagging of audio: The state-of-the-art. In W. Wang, editor, *Machine Audition: Principles, Algorithms and Systems*. IGI Publishing, 2010.
- [5] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere. The million song dataset. In *Proc. ISMIR*, 2011.
- [6] A. Craft, G. A. Wiggins, and T. Crawford. How many beans make five? The consensus problem in music-genre classification and a new evaluation method for single-genre categorisation systems. In *Proc. ISMIR*, pages 73–76, 2007.
- [7] R. B. Dannenberg, B. Thom, and D. Watson. A machine learning approach to musical style recognition. In *Proc. ICMC*, pages 344–347, 1997.
- [8] E. R. Dougherty and L. A. Dalton. Scientific knowledge is possible with small-sample classification. *EURASIP J. Bioinformatics and Systems Biology*, 2013:10, 2013.
- [9] J. Stephen Downie, Donald Byrd, and Tim Crawford. Ten years of ISMIR: Reflections on challenges and opportunities. In *Proc. ISMIR*, pages 13–18, 2009.
- [10] A. Flexer. Statistical evaluation of music information retrieval experiments. *J. New Music Research*, 35(2):113–120, 2006.
- [11] J. Frow. *Genre*. Routledge, New York, NY, USA, 2005.
- [12] Z. Fu, G. Lu, K. M. Ting, and D. Zhang. A survey of audio-based music classification and annotation. *IEEE Trans. Multimedia*, 13(2):303–319, Apr. 2011.
- [13] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2 edition, 2009.
- [14] A. W. Kimball. Errors of the third kind in statistical consulting. *J. American Statistical Assoc.*, 52(278):133–142, June 1957.
- [15] E. Law, L. von Ahn, R. B. Dannenberg, and M. Crawford. Tagatune: A game for music and sound annotation. In *Proc. ISMIR*, pages 361–364, 2007.
- [16] E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classification. In *Proc. ISMIR*, pages 628–233, Sep. 2005.
- [17] R. Sedgewick and K. Wayne. *Algorithms*. Addison-Wesley, Upper Saddle River, NJ, 4 edition, 2011.
- [18] C. N. Silla, A. L. Koerich, and C. A. A. Kaestner. The Latin music database. In *Proc. ISMIR*, 2008.
- [19] M. Slaney. Auditory toolbox. Technical report, Interval Research Corporation, 1998.
- [20] B. L. Sturm. Classification accuracy is not enough: On the evaluation of music genre recognition systems. *J. Intell. Info. Systems*, 41(3):371–406, 2013.
- [21] B. L. Sturm. Making explicit the formalism underlying evaluation in music information retrieval research: A look at the MIREX automatic mood classification task. In *Post-proc. Computer Music Modeling and Research*, 2014.
- [22] B. L. Sturm. The state of the art ten years after a state of the art: Future research in music information retrieval. *J. New Music Research*, 43(2):147–172, 2014.
- [23] B. L. Sturm. A simple method to determine if a music information retrieval system is a “horse”. *IEEE Trans. Multimedia*, 2014 (in press).
- [24] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Trans. Audio, Speech, Lang. Process.*, 16, 2008.
- [25] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Process.*, 10(5):293–302, July 2002.
- [26] J. Urbano. *Evaluation in Audio Music Similarity*. PhD thesis, University Carlos III of Madrid, 2013.
- [27] J. Urbano, M. Schedl, and X. Serra. Evaluation in music information retrieval. *J. Intell. Info. Systems*, 41(3):345–369, Dec. 2013.
- [28] G. A. Wiggins. Semantic gap?? Schemantic schmap!! Methodological considerations in the scientific study of music. In *Proc. IEEE Int. Symp. Multimedia*, pages 477–482, Dec. 2009.
- [29] Y.-H. Yang and H. H. Chen. *Music Emotion Recognition*. CRC Press, 2011.

# AN ASSOCIATION-BASED APPROACH TO GENRE CLASSIFICATION IN MUSIC

**Tom Arjannikov**

University of Lethbridge  
tom.arjannikov@uleth.ca

**John Z. Zhang**

University of Lethbridge  
zhang@cs.uleth.ca

## ABSTRACT

Music Information Retrieval (MIR) is a multi-disciplinary research area that aims to automate the access to large-volume music data, including browsing, retrieval, storage, etc. The work that we present in this paper tackles a non-trivial problem in the field, namely music genre classification, which is one of the core tasks in MIR. In our proposed approach, we make use of association analysis to study and predict music genres based on the acoustic features extracted directly from music. In essence, we build an associative classifier, which finds inherent associations between content-based features and individual genres and then uses them to predict the genre(s) of a new music piece. We demonstrate the feasibility of our approach through a series of experiments using two publicly available music datasets. One of them is the largest available in MIR and contains real world data, while the other has been widely used and provides a good benchmarking basis. We show the effectiveness of our approach and discuss various related issues. In addition, due to its associative nature, our classifier can assign multiple genres to a single music piece; hopefully this would offer insights into the prevalent multi-label situation in genre classification.

## 1. INTRODUCTION

The recent advances in technology, such as data storage and compression, data processing, information retrieval, and artificial intelligence, facilitate music recognition, music composition, music archiving, etc. The Internet is further promoting the enormous growth of digital music collections. Millions of songs previously in physical formats are now readily available through instant access, stimulating and motivating research efforts in meeting new challenges. Among them is *Music Information Retrieval (MIR)*, an interdisciplinary area that attracts practitioners from information retrieval, computer science, musicology, psychology, etc. One of the main tasks in MIR is the design and implementation of algorithmic approaches to managing large collections of digital music, including automatic

tag annotation, recommendation, playlist generation, etc.

The work to be presented in this paper explores the feasibility of applying association analysis to music genre classification. Through our experience with music data, we have found that there are some inherent associations between audio characteristics and human assigned music genre labels. Accordingly, it would be desirable to see whether these associations, if found, can provide insight into genre classification of music. Our work in this paper is geared toward this target.

In a nutshell, our proposed approach uses music data itself by extracting useful information from it and conducting association analysis to make genre prediction. When we talk about the actual sound data of music, we refer to whatever is stored on various media, such as magnetic tapes and now in the digital format. We can extract useful information from this data via signal processing. This information represents the different characteristics of the actual sound stored on media [10]. We refer to it as *content-based features* and use it with our approach. To our knowledge, we are among the first to propose using association analysis for music genre classification in the MIR community.

## 2. PREVIOUS WORK

### 2.1 Classification in MIR

Classification is the process of organizing objects into pre-defined classes. It is a supervised type of learning, where we are given some labeled objects from which we form a computational model that can be used to classify new, previously unseen objects [15].

Classification is one of the core tasks in MIR, since it is usually the first step in many applications, such as on-line music retrieval, playlist recommendation, etc. In our work, we focus on genre classification, which is concerned with categorizing music audio into different genres. Tzanetakis and Cook [18] are among the first to work on this problem, where the task is to label an unknown piece of music with a correct genre name. They show that this is a difficult problem even for humans and report that college students achieve no more than 70% accuracy.

Previous works in MIR along this direction include the following. DeCoro *et al.* [5] use *Bayesian Model* to aid in hierarchical classification of music by aggregating the results of multiple independent classifiers and, thus, perform error correction and improve overall classification accu-



© Tom Arjannikov, John Z. Zhang.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Tom Arjannikov, John Z. Zhang. "AN ASSOCIATION-BASED APPROACH TO GENRE CLASSIFICATION IN MUSIC", 15th International Society for Music Information Retrieval Conference, 2014.

racy. Recent examples of using *Support Vector Machines (SVM)* for music genre classification include an investigation of Meng and Shawe-Taylor [13], where they explore different kernels used in a support vector classifier. Li and Sleep [9] extend normalized information distance into kernel distance for SVM and demonstrate classification accuracy comparable to others. In addition, recently, Anglade *et al.* [3] use *Decision Tree* for music genre classification by utilizing frequent chord sequences to induce context free definite clause grammars of music genres.

## 2.2 Association Analysis in MIR

Association analysis attempts to discover the inherent relations among data objects in an application domain. These relations are represented as association rules. An example of such application domain is the shopping basket analysis in supermarkets, where one tries to discover relations among the items purchased by customers. For example, the association rule  $\{milk, eggs\} \rightarrow \{bread\}$  implies that, if *milk* and *eggs* are bought together by a customer, then *bread* is likely to be bought as well, i.e., they have some inherent statistical relationships [7].

We consider the so-called itemsets, such as  $\{milk, eggs, bread\}$  in the above example, to be frequent if they appear in many transactions. The *support* of an itemset represents the percentage of transactions that contain the itemset and *minimum support* is the threshold that separates the frequent itemsets from the infrequent ones. A frequent itemset can produce an association rule of the form  $A \rightarrow B$ , where  $A$  and  $B$  are non-empty itemsets and  $A \cap B = \phi$ . An association rule holds for a dataset with some minimum support and *confidence*, which is the percentage of transactions containing  $A$  that also contain  $B$  [7].

A formal treatment of applying association analysis in MIR is in Section 3. Within the context of MIR, each track or music piece is represented using a set of content-based features derived from its digitized data. Together, a set of these features place the given track in a discrete location in the feature space. Intuitively, the tracks that are very similar to each other may share the same neighborhood. This could help with organizing music collections for effective data retrieval. When grouped together, the features contain some patterns. We would like to look for these patterns and use them for music genre classification.

Kuo *et al.* [8] propose a way to recommend music based on the emotion that it conveys and look for associations in data that contains information perceived only by humans. Similarly, Xiao *et al.* [19] use a parameterized statistical model to look for associations between timbre and perceived tempo. Liao *et al.* [12] use a dual-wing harmony model to discover association patterns between MTV video clips and the music that accompanies those clips. Neubarth *et al.* [14] present a method of association rule mining with constraints and discover rules in the form of  $A \rightarrow B$ , telling that either region implies genre or genre implies region. Arjannikov *et al.* [4] use association analysis to verify tag annotation in music, though their approach is based on textual music tags and is not content-based.

Our work to be presented below is different from the above and is among the initial efforts to apply association analysis to content-based music genre classification.

## 3. CLASSIFYING MUSIC INTO GENRES VIA ASSOCIATION ANALYSIS

Our work in this paper is focused on the music genre tags. As stated in [6, 10, 10], any discrete set of tags that are not correlated can be used as categories, or classes, into which we could split a collection of music pieces. Arjannikov *et al.* [4] show that association analysis reveals patterns in music textual tags. This motivates our investigation of association analysis in content-based music features.

### 3.1 Notation

Association analysis requires discrete items, however, most content-based music features are not. Thus, when given a set of features  $F = \{f_1, f_2, f_3, \dots, f_k\}$ , we discretize each feature into a predetermined number of bins  $b$ , where  $b > 1$ , and derive a new feature set  $F' = \{f'_{1_1}, f'_{1_2}, \dots, f'_{1_b}, f'_{2_1}, f'_{2_2}, \dots, f'_{2_b}, \dots, f'_{k_1}, f'_{k_2}, \dots, f'_{k_b}\}$ . Then, from the set of music pieces  $M$ , we derive a transactional style dataset  $D = \{d_1, d_2, \dots, d_r\}$ , where  $r = |M|$ . Each transaction  $d_i = \{a_1, a_2, \dots, a_k\}$  corresponds to a music piece and each  $a_j$  in  $d_i$  is a feature item in the literal form  $F_p B_q$ , where  $p$  corresponds to the feature number in  $F'$  and  $q$  corresponds to the bin number, into which the feature for the particular music piece falls. For example, if the first content-based feature is a number between 0 and 1, and it is discretized into 10 equidistant bins, then, given a particular music piece, whose first feature value is 0.125, its corresponding  $d_i$  would contain the label  $F_1 B_2$ .

When we formulate our problem as described above, the music set  $M$ , becomes a transactional set  $D$  suitable for association mining.

### 3.2 Proposed Approach

We call our proposed approach *association-based music genre classifier (AMGC)*. Figure 1 depicts the whole process of using AMGC, which is detailed below.

#### 3.2.1 AMGC

We start by preparing our data during the pre-processing stage. First, we acquire content-based features from music; in this paper, we use the features that have already been extracted and published for the purpose of comparing different classifiers on even ground [16, 17]. Then, we discretize any continuous features. It is worth noting that obtaining optimal discretization is an open problem in machine learning. In our work, we use feature discretization based on equal width of bins, for its simplicity, to avoid any possible bias based on class labels. Then we form transactional style datasets, as described in Section 3.1, and split the training dataset into subsets, one for each genre. Finally, we remove any items that appear in all transactions with a certain *frequency threshold (FRQ)*, which is the percent of transactions containing the item.



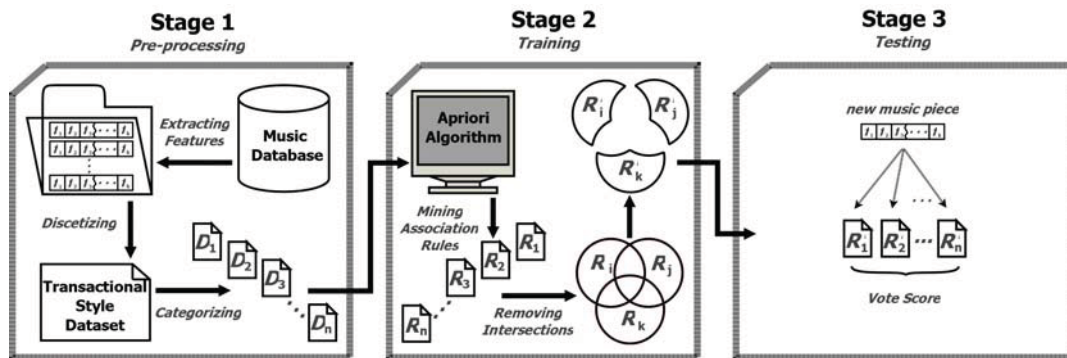


Figure 1. The three stages of our proposed association-based approach to classify music into genres.

During the training stage, we invoke the *Apriori* algorithm [1, 2] and mine frequent itemsets from each genre's sets of items at some minimum support. From these we generate classification rules of the form  $A \rightarrow B$ , where  $A$  is the frequent itemset and  $B$  is the genre associated with that itemset. Then, we remove any itemsets that appear in two or more genres. The resulting rules uniquely represent their respective genres and we use them for classification during the last stage.

### 3.2.2 Scoring Method

To obtain a classification score for each genre, we use the following four components. *Itemset Percentage* (IP) is the percent of itemsets that a given music piece matches for a given genre out of all itemsets matched from that genre. *Support Sum* (SS) is the sum of the matched itemsets' minimum support divided by the sum of all itemsets' minimum support for the given genre. *Confidence Sum* (CS) is the current genre's confidence sum of the matched itemsets divided by the sum of all itemsets' confidence. Finally, *Length Sum* (LS), the sum of cardinalities of the matched itemsets divided by the sum of cardinalities of all itemsets for the given genre.

We score each music piece against each genre's set of rules as following. First, we create a voting vector, whose cardinality is equal to the number of genres, and compute the corresponding component's value for each genre. Then, the genre with the highest value is voted as a candidate of that component, and its element in the voting vector is incremented by 1. Thus, the four components result in four votes and the genre with the highest number of votes is declared as winner and becomes the predicted genre of the given music piece.

### 3.2.3 Accuracy Evaluations

In our work, we use the following classification measures. *Recall*, also known as *sensitivity*, represents the percentage of correctly classified instances for that genre [7]. *Precision* reflects the percentage of correctly classified instances from all instances that are perceived as belonging to that genre by the classifier [7]. Finally, *accuracy* is calculated by dividing the number of all correctly classified instances for all genres by the total number of predictions made [7].

Because AMGC can assign multiple genre labels to a single music piece, we compute the *Multi-Labeling Rate* (MLR) by dividing the total number of predicted labels by the number of all test instances of a genre. MLR falls into the range between 1 and the total number of genres with frequent itemsets. The closer it is to 1, the fewer multi-label assignments were made, which indicates that AMGC is performing more like a single-label classifier. If MLR is equal to the total number of genres, then the results of classification are least useful. Furthermore, if MLR is below 1, then there are music pieces, whose genres could not be predicted.

### 3.3 Goals

Our aim is to test whether the classification rules obtained from music content-based features by AMGC can be used to categorize music into genres. For this, we designate three goals: ( $G_1$ ) AMGC achieves a classification accuracy that is better than choosing genres at random; ( $G_2$ ) AMGC is stable - when given similar datasets, it should achieve similar classification accuracy; ( $G_3$ ) AMGC attains higher accuracy with better quality data and fewer genres.

## 4. EXPERIMENT RESULTS AND DISCUSSIONS

### 4.1 Data Preparation

The classification task at hand requires content-based features paired with genre tags and we find two datasets that fit this requirement.

The *Latin Music Database* [17], denoted as  $D_{LMD}$ , is popular in the music genre classification task despite its small size. There are many classification results available in the literature, which are based on a set of features that has already been extracted and circulated as part of  $D_{LMD}$ . Thus, we can test the feasibility of our approach without introducing variance based on difference in feature extraction techniques. Moreover,  $D_{LMD}$  usually results in high classification accuracy for many methods [17]. We use one of the three sets of features included with it, which is extracted from the beginning 30 seconds of each music piece.

The *Million Song Dataset Benchmarking* [16], denoted as  $D_{MSDB}$ , is much larger than  $D_{LMD}$  and boasts several sets of content-based features. We use five of these sets

and the genre labels, which were originally obtained from Allmusic [16]. Additionally, we restrict the number of tracks to 1000 per genre, in order to balance the number of training and testing examples among genres.

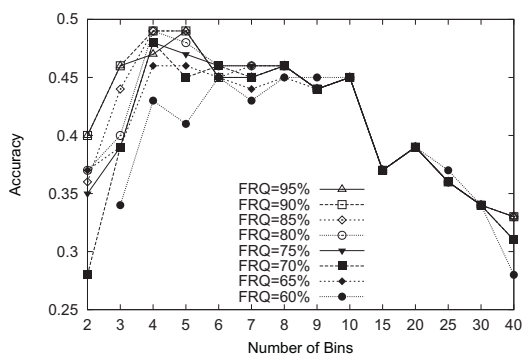
Dataset name	Number of songs	Number of genres	Number of features	Type of Features
$D_{LMD}$	3000	10	30	MFCC
$D_{MSDB-1}$	1500	15	10	MM
$D_{MSDB-2}$	1500	15	16	Spectral
$D_{MSDB-3}$	1500	15	20	LPC
$D_{MSDB-4}$	1500	15	20	AM
$D_{MSDB-5}$	1500	15	26	MFCC

**Table 1.** Music genre datasets and their statistics.

We include some statistical information about the datasets in Table 1 and label them accordingly. We split each one into two equal-sized partitions at random, while maintaining the genres balanced; each genre is represented by equal number of tracks in both partitions. One of the partitions becomes the training set and the other becomes the testing set. If there are too many music pieces belonging to one genre as compared to others, we remove the extra tracks at random. If a genre is represented by fewer pieces than 300 for  $D_{LMD}$  and 1000 for  $D_{MSDB}$ , then we do not use that genre in our experiments. This reduces the original  $D_{MSDB}$  dataset to 17 genres from 25. Moreover, during Stage 2 of our proposed approach, when we mine frequent itemsets, two of the genres produce none; therefore, only 15 genres persist, as reported in Table 1.  $D_{LMD}$  remains at 10 genres because it was originally balanced at 300 music pieces per genre.

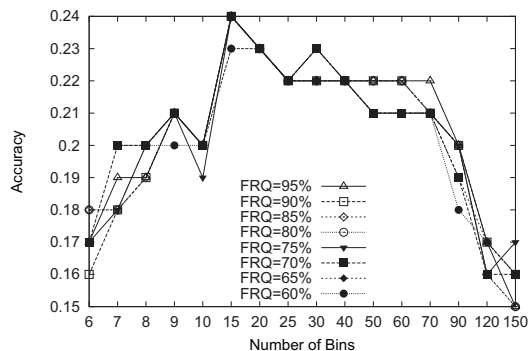
In the following section, we demonstrate through our experiment results how we achieve the three goals formulated in Section 3.3.

## 4.2 Results and Discussions



**Figure 2.**  $D_{LMD}$  at minimum support = 20%.

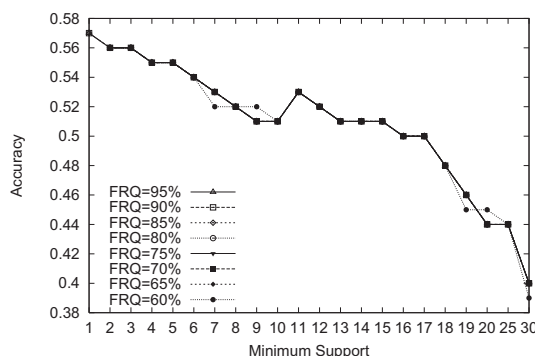
During our experiments, we observe that our proposed parameters affect the classification accuracy, and thus, they are effective. It is evident from Figures 2 and 3 that the number of discretization bins affects the classification accuracy for both  $D_{LMD}$  and  $D_{MSDB}$ . Figure 4 demonstrates how the classification accuracy is affected by the minimum support parameter. We also note that AMGC performs



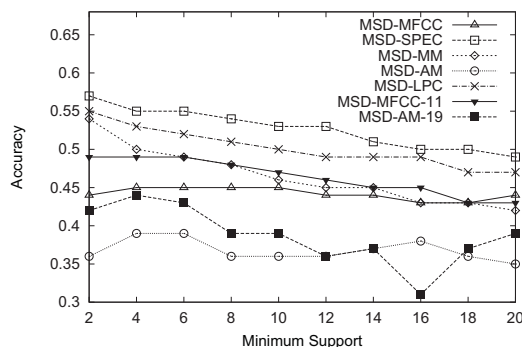
**Figure 3.**  $D_{MSDB-2}$  at minimum support = 2%.

much better than if we were to choose genres at random. Thus, we confirm that AMGC works for some parameter settings and conclude our work towards  $G_1$ .

As demonstrated in the literature, the classification accuracy usually increases when the number of classes is reduced [11]. Thus, we reduce the number of genres for both  $D_{LMD}$  and  $D_{MSDB}$  to 5 and observe that AMGC performs better. Therefore, we report only the results for the smaller set of genres in Figures 4 through 9. We also observe that  $D_{LMD}$  achieves higher accuracy than  $D_{MSDB}$  as can be seen in Figures 2 and 3. This concludes our work towards  $G_3$ , as AMGC performs better with a better quality dataset, moreover, it performs better on a reduced set of genres.



**Figure 4.**  $D_{MSDB-2}$  with number of bins = 20.



**Figure 5.** All five  $D_{MSDB}$  datasets compared, with number of bins = 13, unless otherwise specified.

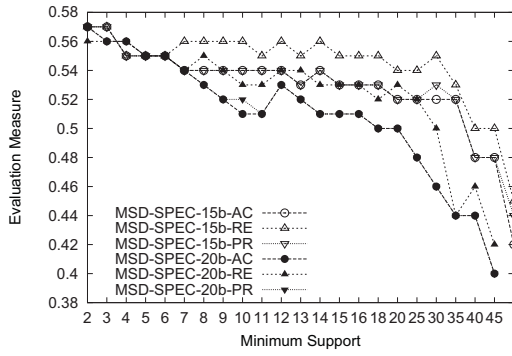


Figure 6.  $D_{MSDB-2}$  across different minimum support.

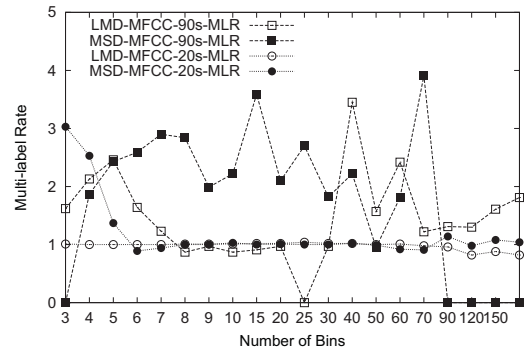


Figure 9. 5 genres of  $D_{LMD}$  and  $D_{MSDB-5}$  at  $FRQ = 95$ .

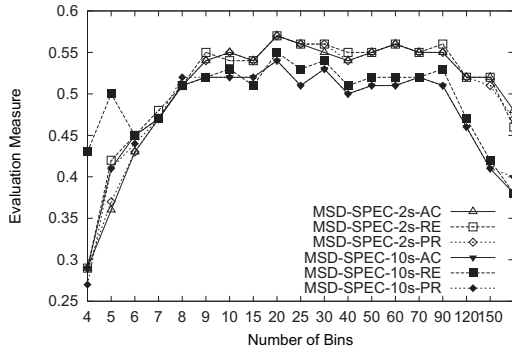


Figure 7.  $D_{MSDB-2}$  across different number of bins.

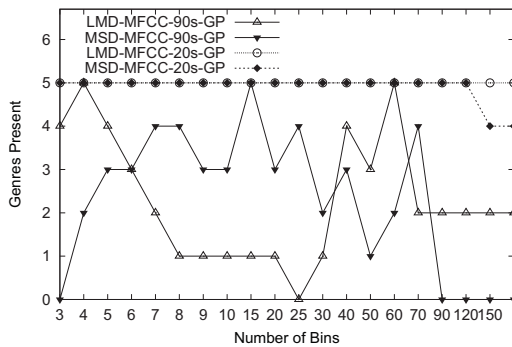


Figure 8. 5 genres of  $D_{LMD}$  and  $D_{MSDB-5}$  at  $FRQ = 95$ .

It is clear from Figures 2, 3 and 4, that the  $FRQ$  parameter does not significantly affect the classification accuracy, although, it produces highest accuracy overall when set to 95%. We use this setting in all of the experiment results in Figures 5 through 9.

During our experiments, we observe that  $D_{MSDB}$  datasets perform best at lower minimum support and number of bins settings. We set the number of bins to 13 and perform a sweep across minimum support values between 2 and 20. As can be seen in Figure 5, among all five,  $D_{MSDB-2}$  performs the best and  $D_{MSDB-4}$  the worst. Three of the five datasets achieve their highest accuracy when the number of bins is set to 13; however,  $D_{MSDB-4}$  performs better at 19 bins, and  $D_{MSDB-5}$  at 11 bins. Thus, we include the corresponding results in Figure 5.

We observe that all three evaluation measures, *recall*, *precision*, and *accuracy*, obtain very similar values to each other in our experiments, as can be seen in Figures 6 and 7. It can also be seen in Figures 2 through 7, that AMGC does not behave arbitrarily, when given different datasets or different parameter settings. This confirms that our approach is stable and concludes our work towards  $G_2$ .

During our experiments, we notice that for some values of minimum support and for some numbers of bins, AMGC performs much better than choosing genre assignment at random. However, with other values of these parameters, AMGC predicts majority of music to be of one genre. Moreover, sometimes it votes for all genres equally, where MLR becomes equal to the number of genres. Furthermore, we encountered certain parameter settings, when some or all genres were not represented by any classification rules. We investigate the behaviour of MLR and the number of genres present in both  $D_{LMD}$  and  $D_{MSDB}$  through further experiments and report our findings in Figures 8 and 9. Here, we set the minimum support to 20 and then to 90 for both datasets. As can be seen in Figure 8, at the higher minimum support, some genres are discarded, due to removal of intersections during Stage 2 of our approach. Meanwhile, Figure 9 illustrates that AMGC behaves as a single label classifier, because we remove rules that are found among any genre-pair, thus, the remaining rules are representative of a single genre.

When experimenting with our approach on music genre classification using different features in  $D_{MSDB}$ , we use the same genre assignment and alternate the features. This helps us confirm that difference in content-based features result in different classification performance. Hence, different features are more or less useful for the genre classification task, which is reflected by the *feature selection* task in MIR.

In our experiments, we notice that it may take a long time to pre-process the data and train the classifier. However, the resulting classification model is very fast, where its speed can be expressed as the number of classification rules multiplied by the number of music pieces to be classified.

## 5. CONCLUSION

In this paper, we introduce a novel approach to MIR, namely, using association analysis to help music genre classification. Association analysis looks for frequent patterns in music data, which represent the similarity of all music pieces in a given genre.

Through experiments, we demonstrate the effectiveness of our approach and confirm that association analysis can be applied to music data. However, there is still room for improvement, which includes feature extraction, feature selection and discretization. We believe that as they improve, our method will also improve. We can also take some immediate steps to improve our classifier by tuning the two parameters, minimum support for mining frequent items and the number of discretization bins. Our experiments demonstrate that these two parameters are directly related to the performance of our classifier, and they vary depending on the data. Hence, tuning these parameters to each specific dataset will improve the classification accuracy. We leave these to our future work.

## 6. REFERENCES

- [1] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, volume 22, pages 207–216. ACM, 1993.
- [2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, volume 1215, pages 487–499. Morgan Kaufmann Publishers Inc., 1994.
- [3] A. Anglade, R. Ramirez, and S. Dixon. Genre classification using harmony rules induced from automatic chord transcriptions. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 669–674. ISMIR, 2009.
- [4] T. Arjannikov, C. Sanden, and J. Z. Zhang. Verifying tag annotations through association analysis. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 195–200. ISMIR, 2013.
- [5] C. DeCoro, Z. Barutcuoglu, and R. Fiebrink. Bayesian aggregation for hierarchical genre classification. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 77–80. ISMIR, 2007.
- [6] Z. Fu, G. Lu, K. M. Ting, and D. Zhang. A survey of audio-based music classification and annotation. *IEEE Transactions on Multimedia*, 13(2):303–319, 2011.
- [7] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., the second edition, 2006.
- [8] F.-F. Kuo, M.-F. Chiang, M.-K. Shan, and S.-Y. Lee. Emotion-based music recommendation by association discovery from film music. In *Proceedings of the 13th Annual ACM International Conference on Multimedia*, pages 507–510. ACM, 2005.
- [9] M. Li and R. Sleep. Genre classification via an lz78-based string kernel. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 252–259. ISMIR, 2005.
- [10] T. Li, O. Mitsunori, and G. Tzanetakis, editors. *Music Data Mining*. CRC Press, 2012.
- [11] T. Li, M. Ogihara, and Q. Li. A comparative study on content-based music genre classification. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 282–289. ACM, 2003.
- [12] C. Liao, P. Wang, and Y. Zhang. Mining association patterns between music and video clips in professional mtv. In B. Huet, A. Smeaton, K. Mayer-Patel, and Y. Avrithis, editors, *Advances in Multimedia Modelling*, volume 5371 of *Lecture Notes in Computer Science*, pages 401–412. Springer Berlin Heidelberg, 2009.
- [13] A. Meng and J. Shawe-Taylor. An investigation of feature models for music genre classification using the support vector classifier. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 604–609. ISMIR, 2005.
- [14] K. Neubarth, I. Goienetxea, C. Johnson, and D. Conklin. Association mining of folk music genres and toponyms. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 7–12. ISMIR, 2012.
- [15] S. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson education Inc., the third edition, 2010.
- [16] A. Schindler, R. Mayer, and A. Rauber. Facilitating comprehensive benchmarking experiments on the million song dataset. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 469–474. ISMIR, 2012.
- [17] C. N. J. Silla, C. A. A. Kaestner, and A. L. Koerich. The latin music database. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 451–456. ISMIR, 2008.
- [18] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.
- [19] L. Xiao, A. Tian, W. Li, and J. Zhou. Using a statistic model to capture the association between timber and perceived tempo. In *Proceedings of the International Society for Music Information Retrieval*, pages 659–662. ISMIR, 2008.

# MULTIPLE VIEWPOINT MELODIC PREDICTION WITH FIXED-CONTEXT NEURAL NETWORKS

Srikanth Cherla<sup>1,2</sup>, Tillman Weyde<sup>1,2</sup> and Artur d’Avila Garcez<sup>2</sup>

<sup>1</sup>Music Informatics Research Group, Department of Computer Science, City University London

<sup>2</sup>Machine Learning Group, Department of Computer Science, City University London

{srikanth.cherla.1, t.e.veyde, a.garcez}@city.ac.uk

## ABSTRACT

The *multiple viewpoints* representation is an event-based representation of symbolic music data which offers a means for the analysis and generation of notated music. Previous work using this representation has predominantly relied on  $n$ -gram and variable order Markov models for music sequence modelling. Recently the efficacy of a class of distributed models, namely restricted Boltzmann machines, was demonstrated for this purpose. In this paper, we demonstrate the use of two neural network models which use fixed-length sequences of various viewpoint types as input to predict the pitch of the next note in the sequence. The predictive performance of each of these models is comparable to that of models previously evaluated on the same task. We then combine the predictions of individual models using an entropy-weighted combination scheme to improve the overall prediction performance, and compare this with the predictions of a single equivalent model which takes as input all the viewpoint types of each of the individual models in the combination.

## 1. INTRODUCTION

We are interested in the computational modelling of melodies available in symbolic music data formats such as MIDI and KERN. For this purpose, we chose to work with a representation of symbolic music first proposed in [9] in relation to *multiple viewpoints for music prediction* (which we refer to here as the “multiple viewpoints representation”). The multiple viewpoints representation is an event-based representation extracted from symbolic music data where a given piece of music is decomposed into parallel streams of features, known as *viewpoint types*. Each viewpoint type is either a directly observable musical dimension such as *pitch* and *note duration*, or an abstract one derived from them such as *inter-onset interval* or *pitch contour*. In order to analyse musical structure using this representation, one can train a machine learning model on sequences of

viewpoint types and apply it to tasks such as music generation [6] and classification [3, 7]. This representation has also been the focus of more recent work related to music cognition [14, 17]. The novelty of this approach is in its extension of previous work in language modelling to music with an information theoretic backing which facilitates an objective evaluation of models for music prediction. Approaches based on information theory have been of interest in musicology to understand structure and meaning in music in terms of its predictability [10, 11, 13].

In the original work on multiple viewpoints [9] and that which followed [15, 21], Markov models were exclusively employed for music modelling using this framework. While this is a reasonable choice, Markov models are often faced with a problem related to data sparsity known as the *curse of dimensionality* [2]. This refers to the exponential rise in the number of model parameters to be estimated with the length of the modelled sequences. Models which employ distributed architectures such as neural networks tend to avoid this problem, as they do not require enumerating all state transition probabilities, but rather the weights of the network encode only those dependencies necessary to minimize prediction error. It was demonstrated more recently in [4] how a distributed model — the restricted Boltzmann machine, is a suitable alternative in this context. It was also suggested in [8] that neural networks might be suitable alternatives to  $n$ -gram models for music modelling with multiple viewpoints but no actual research in this direction has ensued.

In this paper, we first present two neural networks for modelling sequences of musical pitch. The first is a simple feed-forward neural network [20], and the second is the musical extension of the Neural Probabilistic Language Model [2] — a deeper feed-forward network with an added weight-sharing layer between the input and hidden layers. The latter was originally proposed for learning distributed representations of words in language modelling. Both models predict a probability distribution over the possible values of the next pitch given a fixed-length context as input. Their predictive performance is comparable to or better than previously evaluated melody prediction models in [4, 16]. The second network is further extended to make use of additional viewpoint types extracted from the context, as inputs for the same task of predicting musical pitch. We then combine the predictions of individual models with different viewpoint types as their respective



© Srikanth Cherla<sup>1,2</sup>, Tillman Weyde<sup>1,2</sup> and Artur d’Avila Garcez<sup>2</sup>.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Srikanth Cherla<sup>1,2</sup>, Tillman Weyde<sup>1,2</sup> and Artur d’Avila Garcez<sup>2</sup>. “Multiple Viewpoint Melodic Prediction with Fixed-Context Neural Networks”, 15th International Society for Music Information Retrieval Conference, 2014.

inputs using an entropy-weighted combination scheme to improve the overall prediction performance, and compare this with the predictions of a single model which takes as input all the viewpoint types of each of the individual models in the combination.

We begin with an overview of the multiple viewpoints representation in Section 2. This is followed by a description of the two neural networks which are used with this representation, in Section 3. Section 4 presents an evaluation of the predictive performance of the two models along with a comparison to previous work. Finally, directions for future research are outlined in Section 5.

## 2. MULTIPLE VIEWPOINT SYSTEMS

In order to explain music prediction with multiple viewpoints, the analogy to natural language is used here. In statistical language modelling, the goal is to build a model that can estimate the joint probability distribution of subsequences of words occurring in a language  $L$ . A statistical language model (SLM) can be represented by the conditional probability of the next word  $w_T$  given all the previous ones  $[w_1, \dots, w_{(T-1)}]$  (written here as  $w_1^{(T-1)}$ ), as

$$P(w_1^T) = \prod_{t=1}^T P(w_t | w_1^{(t-1)}). \quad (1)$$

The most commonly used SLMs are  $n$ -gram models, which rely on the simplifying assumption that the probability of a word in a sequence depends only on the immediately preceding  $(n-1)$  words [12]. This is known as the Markov assumption, and reduces (1) to

$$P(w_1^T) = \prod_{t=1}^T P(w_t | w_{(t-n+1)}^{(t-1)}). \quad (2)$$

Following this approach, musical styles can be interpreted as vast and complex languages [9]. In predicting music, one is interested in learning the joint distribution of *musical event* sequences  $s_1^T$  in a *musical language*  $S$ . Much in the same way as an SLM, a system for music prediction models the conditional distribution  $p(s_t | s_1^{(t-1)})$ , or under the Markov assumption  $p(s_t | s_{(t-n+1)}^{(t-1)})$ . For each prediction, context information is obtained from the events  $s_{(t-n+1)}^{(t-1)}$  immediately preceding  $s_t$ . Musical events have a rich internal structure and can be expressed in terms of directly observable or derived musical features such as pitch, note duration, inter-onset interval, or a combination of two or more such features. The framework of multiple viewpoint systems for music prediction [9] was proposed in order to efficiently handle this rich internal structure of music by exploiting information contained in these different musical feature sequences, while at the same time limiting the dimensionality of the models using these features. In the interest of brevity, we limit ourselves to an informal discussion of multiple viewpoint systems for monophonic music prediction and refer the reader to [9] for a more detailed explanation.

A musical event  $s$  refers to the occurrence of a note in a melody. A *viewpoint type* (or simply *type*)  $\tau$  refers to any of a set of musical features that describe an event. The domain of a *type*, denoted by  $[\tau]$  is the set of possible values of that type. A *basic type* is a directly observable or given feature such as *pitch*, *note duration*, *key-signature* or *time-signature*. A *derived type* can be derived from any of the basic types or other derived types. Two or more types can be “linked” by taking the Cartesian product of their respective domains, thus creating a *linked viewpoint type*. A *multiple viewpoints system* (MVS) is a set of models, each of which is trained on subsequences of one *type*, whose individual predictions are combined in some way to influence the prediction of the next event in a given event sequence. Given a context  $s_{(t-n+1)}^{(t-1)}$  and an event  $s_t$ , each viewpoint  $\tau$  in an MVS must compute the probability  $p_\tau(s_t | s_{(t-n+1)}^{(t-1)})$ .

In order to input the viewpoint type sequences to the neural network models, we first convert each input type value into a binary one-hot encoding. When a context event is missing or undefined, each element of the vector is initialized to  $1/|S|$ . When there is more than one input type, one-hot vectors corresponding to all the input types for a musical event are concatenated to obtain an input vector for that event. As we are dealing with models of fixed context-length  $l$ , the final input feature vector input to the model is a concatenation of  $l$  such vectors. In doing so, we are effectively bypassing the need to compute a Cartesian product to link viewpoint types before using them as input to a single model which has been the practice when using  $n$ -gram and variable order Markov models.

Each model in an MVS relies on a different source of information (its respective input types) to make a prediction about the target viewpoint type. The accuracy of the prediction depends on how informative these input types are of the target type. It is possible to combine the information provided by different input types for possibly better predictive performance. Here, we consider two ways of doing this - *implicitly* in a single model which is trained using a set of input types, and *explicitly* by combining the probability distributions of multiple models, each of which is trained separately on a mutually exclusive subset of these input types. While the former is only a special case of what has been described so far, we provide an explanation of the latter below in Section 2.1.

### 2.1 Combining Multiple Models

It was demonstrated in [9, 15] that an entropy-weighted combination of the predictions of two or more  $n$ -gram or variable order Markov models typically results in ensembles with better predictive performance than any of the individual models. As it is the predicted distributions which are combined, this approach is independent of the types of models involved. Here, we briefly describe two approaches for creating such ensembles. Let  $M$  be a set of models and  $p_m(s)$  be the probability assigned to symbol  $s \in [\tau_{tgt}]$  by model  $m$ , where  $[\tau_{tgt}]$  is the domain of the target type. The first approach involves taking a weighted arithmetic mean of their respective predictions. This is the *mixture-*

of-experts combination, and is defined as

$$p(s) = \frac{\sum_{m \in M} w_m p_m(s)}{\sum_{m \in M} w_m}$$

where each of the weights  $w_m$  depends on the entropy of the distribution generated by the corresponding model  $m$  in the combination such that greater entropy (and hence uncertainty) is associated with a lower weight [5]. The weights are given by the expression  $w_m = H_{rel}(p_m)^{-b}$ , where the relative entropy  $H_{rel}(p_m)$  is

$$H_{rel}(p_m) = \begin{cases} H(p_m)/H_{max}(p_m), & \text{if } H_{max}([\tau_{tgt}]) > 0 \\ 1, & \text{otherwise} \end{cases}$$

The best value of the bias  $b$  is determined through cross-validation. The quantities  $H$  and  $H_{max}$  are respectively the entropy of the prediction and the maximum entropy of predictions over the symbol space  $[\tau_{tgt}]$ , and are defined as

$$H(p) = - \sum_{s \in [\tau_{tgt}]} p(s) \log_2 p(s). \quad (3)$$

$$H_{max}(p) = \log_2 |S|.$$

where  $p(s \in [\tau_{tgt}]) = p(\chi = s)$  is the probability mass function of a random variable  $\chi$  distributed over the discrete alphabet  $[\tau_{tgt}]$  such that the individual probabilities are independent and sum to 1.

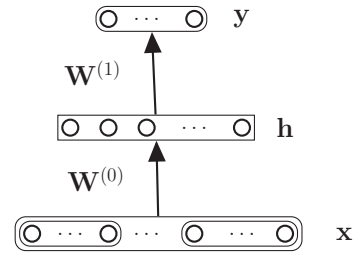
The second combination method — *product-of-experts*, is computed similarly as the weighted geometric mean of the probability distributions. This is given by

$$p(s) = \frac{1}{R} \left( \prod_{m \in M} p_m(s)^{w_m} \right)^{\frac{1}{\sum_{m \in M} w_m}}$$

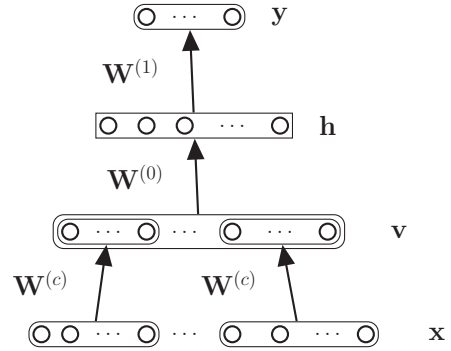
where  $R$  is a normalisation constant which ensures that the resulting distribution over  $S$  sums to unity. The weights  $w_m$  in this case are obtained in the same manner as for the mixture-of-experts case. It was observed in a previous application of these two combination methods to melody modelling [15], that product-of-experts resulted in a greater improvement in predictive performance.

### 3. FIXED-CONTEXT NEURAL NETWORKS

In this section, we provide a brief overview of the two fixed-context neural network models which we employed for the task of predicting the pitch of the next note in a melody, given a viewpoint type context which leads up to it. These are (1) a feed-forward neural network, and (2) a neural probabilistic melody model. The key difference between the two is the presence of an additional weight-sharing layer in the latter which transforms the binary representation of the viewpoint types into lower-dimensional real-valued vectors before passing these on as inputs to a feed-forward network (much like the former).



(a) Feed-forward Neural Network



(b) Neural Probabilistic Melody Model

**Figure 1:** The two models employed for multiple viewpoint melodic prediction in this paper (biases ignored in the illustration). A concatenation of the fixed-length input type context is presented to each model in its visible layer and the predictions are made in the output layer.

#### 3.1 Feed-forward Neural Network

In its simplest form, a feed-forward neural network (Figure 1) consists of an input layer  $\mathbf{x} \in \mathbb{R}^n$ , a hidden layer  $\mathbf{h} \in \mathbb{R}^m$  and an output layer  $\mathbf{y} \in \mathbb{R}^l$ . The input layer is connected to the hidden layer by a weight-matrix  $W^{(0)}$  and likewise, the hidden layer to the output layer by a matrix  $W^{(1)}$ . Each unit in the hidden layer typically applies a non-linear function to the input it receives from the layer below it. Similarly, each unit of the output layer applies a function to the input it receives from the hidden layer immediately preceding it. In a network with a single hidden layer, this happens according to the following equations

$$\mathbf{u}^{(0)} = \mathbf{b}^{(0)} + W^{(0)}\mathbf{x} \quad (4)$$

$$\mathbf{h} = f^{(0)}(\mathbf{u}) \quad (5)$$

$$\mathbf{u}^{(1)} = \mathbf{b}^{(1)} + W^{(1)}\mathbf{h} \quad (6)$$

$$\mathbf{y} = f^{(1)}(\mathbf{u}) \quad (7)$$

where  $\mathbf{b}^{(0)}$  and  $\mathbf{b}^{(1)}$  are the hidden and output layer biases,  $f^{(0)}$  and  $f^{(1)}$  are functions applied to the input received by each node in the hidden and output layers respectively. Thus, for a given input  $\mathbf{x}$ , the output  $\mathbf{y}$  is calculated as

$$\mathbf{y} = f^{(1)}(\mathbf{b}^{(1)} + W^{(1)} \cdot f^{(0)}(\mathbf{b}^{(0)} + W^{(0)}\mathbf{x})) \quad (8)$$

In the present case,  $f^{(0)}$  is the logistic sigmoid function and  $f^{(1)}$  is the softmax function. The network can

be trained in a supervised manner using the backpropagation algorithm [20]. This algorithm applies the chain rule of differentiation to propagate the error between the target output and the output of the model backwards into the network, and use these derivatives to appropriately update the model parameters (the network weights and biases).

### 3.2 Neural Probabilistic Melody Model

Next we consider the neural probabilistic melody model (NPMM), which was originally introduced in [2] as a language model for word sequences. It consists of a feed-forward network such as the one described in Section 3.1, with an additional *embedding* layer below it (Figure 1). This model takes as input a concatenation of binary viewpoint type vectors (*cf.* Section 3) which represent a fixed-length context. The first layer of the network maps each of these sparse binary vectors to lower-dimensional dense real-valued vectors which make up the input layer of what is essentially a feed-forward network above it. This mapping is determined by a shared weight matrix  $W^{(c)}$  which is learned from data, and is given by

$$\mathbf{v} = W^{(c)}\mathbf{x}. \quad (9)$$

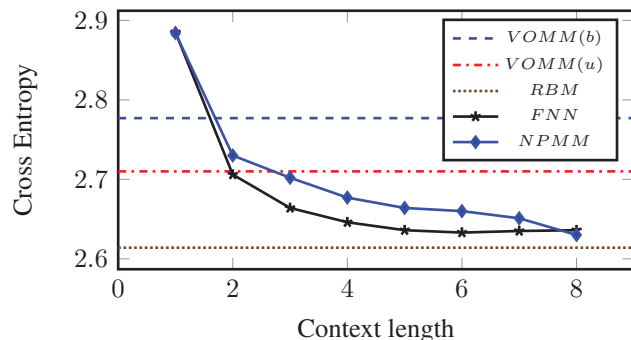
The hidden layer in the case of the NPMM consists of hyperbolic-tangent activation units. The output layer contains softmax units. The model is trained with backpropagation using gradient descent as in the case of a standard feed-forward neural network.

## 4. EVALUATION

The first goal of this paper is to demonstrate the suitability of fixed-context neural networks for multiple viewpoint melodic prediction. To this end, we compare the two models described in Section 3 with variable-order Markov models (VOMMs) and restricted Boltzmann machines (RBMs). It was observed that the predictive performance of each of the neural network models is either comparable to or better than that of the best VOMMs of both bounded and unbounded order [16], while slightly worse than the RBM of [4] (Figure 2). Second, we wish to compare the predictions of a single neural network which uses multiple input types with that of an ensemble of networks with smaller input dimensions, each of which uses a subset of the input types of the former, and combined with the entropy-based weighting scheme described in 2.1. We found that, while the addition of viewpoint types does improve the predictive performance in both cases, that of the single network is slightly worse than the ensemble (Figure 3). Moreover, the extent of this improvement diminishes with an increase in context length.

### 4.1 Dataset

Evaluation was carried out on a corpus of monophonic MIDI melodies that cover a range of musical styles. It consists of 4 datasets - Bach chorale melodies, and folk melodies from Canada, China and Germany, with a total



**Figure 2:** Comparison between the predictive performances of the best bounded and unbounded variable-order Markov models (VOMM(b) and VOMM(u) respectively), the best restricted Boltzmann machine (RBM), the feed-forward neural network (FNN) and the neural probabilistic melody model (NPMM) averaged over the datasets.

of 37,229 musical events. These were also used to evaluate RBMs and variable order Markov models for music prediction in [4, 16]. To facilitate a direct comparison, the melodies are not transposed to a default key.

### 4.2 Evaluation Measure

In order to evaluate the proposed prediction models, we turn to a previous study of Markov models for music prediction in [16]. There, *cross entropy* was used to measure the information content of the models. This is a quantity related to *entropy* (3). The value of entropy, with reference to a prediction model, is a measure of the uncertainty of its predictions. A higher value reflects greater uncertainty. In practice, one rarely knows the true probability distribution of the stochastic process and uses a model to approximate the probabilities in (3). An estimate of the goodness of this approximation can be measured using cross entropy ( $H_c$ ) which represents the divergence between the entropy calculated from the estimated probabilities and the source model. This quantity can be computed over all the subsequences of length  $n$  in the test data  $\mathcal{D}_{test}$ , as

$$H_c(p_{mod}, \mathcal{D}_{test}) = \frac{-\sum_{s_1^n \in \mathcal{D}_{test}} \log_2 p_{mod}(s_n | s_1^{(n-1)})}{|\mathcal{D}_{test}|} \quad (10)$$

where  $p_{mod}$  is the probability assigned by the model to the last pitch in the subsequence given its preceding context. Cross-entropy approaches the true entropy as the number of test samples ( $|\mathcal{D}_{test}|$ ) increases.

### 4.3 Model Selection

Different neural network configurations were evaluated by a grid search over the learning rate  $\eta = \{0.05, 0.1\}$ , the number of hidden units  $n_{hid} = \{25, 50, 100, 200, 400\}$ , number of embedding units  $n_{emb} = \{10, 20\}$  (only for the NPMM), and weight decay  $w_{decay} = \{0.0000, 0.0001, 0.0005\}$ . Each model was trained using mini-batch gradient descent up to a maximum of 1000 epochs with a batch size of 100 samples. Early-stopping [19] and weight-decay



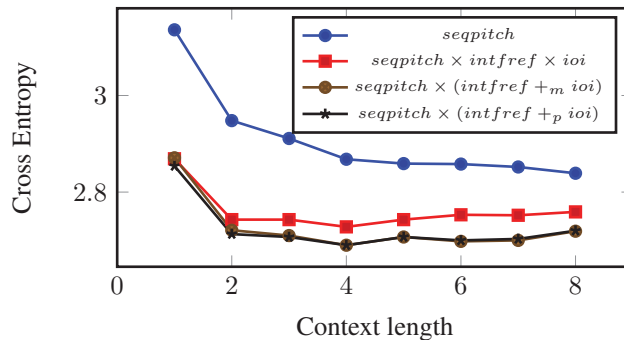
were also incorporated to counter overfitting. The momentum parameter  $\mu$ , was set to 0.5 during the first five epochs and then increased to 0.9 for the rest of the training. Each model was evaluated with 10-fold cross-validation, with folds identical to those used in [4, 16] for the sake of comparison.

#### 4.4 Model Comparison

We carried out a comparison between the predictive performance of the two neural network models presented here, and models previously evaluated on the same datasets [4, 16]. It is to be noted that, since neither of our models is updated online during prediction, the comparison with the variable order Markov models of [16] is limited to their best performing Long-Term Models. These are of order bound 2 and unbounded order (labelled there as  $C^*I$ ). It is evident from Figure 2 that both the neural network models are able to take advantage of information in longer contexts than the bounded order  $n$ -gram models. This is also a feature of the RBM, whose best case of context-length 5 outperforms the rest of the models in the plot. The slight deterioration in the performance of the feed-forward network for longer contexts is possibly due to poor optimization of its parameters. This is considering the fact that weight-decay and early-stopping were implemented in the training algorithm to prevent overfitting. While it was not possible to incorporate further steps for better parameter optimization in this paper, the results are still illustrative of the networks' suitability at the given task and the improvement in performance with context consistent with each other and with that of the RBMs. Possible optimizations have been left as future work, and will be discussed in Section 5.

#### 4.5 Model Combination

In order to evaluate the combination of viewpoint types, we selected one type which is related to the “what” in music — scale-degree (*intfref*), and another which is related to the “when” — inter-onset interval (*ioi*), from the several possible choices that exist. Furthermore, this experiment was performed using the NPMM and only on the Chinese folk melody dataset for the purpose of illustration, with the assumption that a similar trend would be observed with the other model and datasets. As our target viewpoint type i.e. the one being predicted, is musical pitch (*seqpitch*), the first model has the input types *seqpitch* and *intfref* and the second one *seqpitch* and *ioi*. The additional viewpoints are incorporated as explained in Section 2. The predictions of these two models are combined *explicitly* using the mixture- and product-of-experts schemes. On the other hand, the *implicit* combination of these two is a single model whose input types are *seqpitch*, *intfref* and *ioi*. Figure 3 compares the predictions of the pitch-only version of the NPMM and the three models using the additional input types. It can be seen that each of these three models has a better predictive performance than its pitch-only counterpart, thus confirming the relevance of the added viewpoint types to musical pitch prediction. Both the mixture- and product-of-experts combination schemes (*seqpitch*  $\times$



**Figure 3:** Comparison between the predictive performances, on the Chinese folk melody dataset, of the pitch-only NPMM, its extension which uses the *intfref* and *ioi* types as additional input, and ensembles each of which combines two models of input types (a) *seqpitch* and *intfref* (b) *seqpitch* and *ioi* using the mixture (+<sub>m</sub>) and product (+<sub>p</sub>) combination schemes.

(*intfref* +<sub>m</sub> *ioi*) and *seqpitch*  $\times$  (*intfref* +<sub>p</sub> *ioi*) respectively in the plot) result in very similar predictive performance, with the latter working only slightly better for shorter context-lengths of 1, 2 and 3. Moreover, both these explicit combinations of viewpoint types perform better than the single implicit combination of types (*seqpitch*  $\times$  *intfref*  $\times$  *ioi* in the plot). One will, however, notice that the cross entropy of the predictions slightly worsens at longer context-lengths, and that the discrepancy between the implicit and explicit combinations gradually increases in these cases. As mentioned earlier, we attribute this to the optimization of the network parameters, which is to be dealt with in future work.

## 5. CONCLUSIONS & FUTURE WORK

The two neural network models for melodic prediction presented here have been found to have a predictive performance comparable to or better than previously evaluated VOMMs, but slightly worse than that of RBMs. Predictive performance can be further improved by the addition of viewpoint types to the same model, or by combining multiple models using an entropy-weighted combination scheme. In our experiments, the latter tended to be better.

One open issue that remains is the parameter optimization in the two networks presented here. It was observed that, particularly when the input layer of a network is large and the dataset relatively small, the predictive performance does not improve as expected with context-length and the addition of viewpoint types. We note here that the results presented have been generated with models implemented in-house<sup>1</sup> for use with the Python machine learning library *scikit-learn* [18], and were thus limited in the various initialization and optimization strategies used in their learning algorithms. We also suspect this to be the reason for the limited success of the NPMM which exhibited relatively more promising results in its language

<sup>1</sup> Code available upon request.

modelling application in [2]. Many more measures to improve generalization and overall prediction accuracy (such as dropout, different weights initialization strategies and layer-wise pre-training) have been suggested in [1]. Incorporating these measures (or using an existing neural network library which does) can further improve the results.

Apart from this, there are three other aspects which are of immediate interest to us. The first is the incorporation of a short-term element in the prediction model which updates its parameters as data is presented to it, and has been shown to result in improved prediction performance and human-like predictions [15]. Secondly, while the number of parameters of the fixed-context models presented here increases linearly with the context-length (assuming a fixed number of hidden units), we are at present experimenting with recurrent networks where this problem does not arise due to their recurrent connections. And finally, the extension of the said models to polyphonic multiple viewpoints representations is also an open issue at the moment which we hope to address in the future.

## 6. ACKNOWLEDGEMENTS

Srikanth Cherla is supported by a PhD studentship from City University London. The authors would like to thank Marcus Pearce for his valuable advice, and the anonymous reviewers for their feedback on the submission.

## 7. REFERENCES

- [1] Yoshua Bengio. Practical Recommendations for Gradient-Based Training of Deep Architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. 2012.
- [2] Yoshua Bengio, Rejean Ducharme, Pascal Vincent, and Christian Jauvin. A Neural Probabilistic Language Model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.
- [3] Srikanth Cherla, Artur d’Avila Garcez, and Tillman Weyde. A neural probabilistic model for predicting melodic sequences. In *International Workshop on Machine Learning and Music*, 2013.
- [4] Srikanth Cherla, Tillman Weyde, Artur d’Avila Garcez, and Marcus Pearce. A distributed model for multiple viewpoint melodic prediction. In *International Society for Music Information Retrieval Conference*, pages 15–20, 2013.
- [5] Darrell Conklin. Prediction and entropy of music. 1990.
- [6] Darrell Conklin. Music generation from statistical models. In *AISB Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pages 30–35, 2003.
- [7] Darrell Conklin. Multiple viewpoint systems for music classification. *Journal of New Music Research*, 42(1):19–26, 2013.
- [8] Darrell Conklin and John G Cleary. Modelling and generating music using multiple viewpoints. 1988.
- [9] Darrell Conklin and Ian H Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.
- [10] Greg Cox. On the relationship between entropy and meaning in music: An exploration with recurrent neural networks. *Proceedings of the 32nd Annual Cognitive Science Society*. Austin TX: CSS, 2010.
- [11] David Brian Huron. *Sweet anticipation: Music and the psychology of expectation*. MIT press, 2006.
- [12] Christopher D Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.
- [13] Leonard B Meyer. Meaning in music and information theory. *The Journal of Aesthetics and Art Criticism*, 15(4):412–424, 1957.
- [14] Diana Omigie, Marcus T Pearce, Victoria J Williamson, and Lauren Stewart. Electrophysiological correlates of melodic processing in congenital amusia. *Neuropsychologia*, 2013.
- [15] Marcus T Pearce. *The construction and evaluation of statistical models of melodic structure in music perception and composition*. PhD thesis, City University London, 2005.
- [16] Marcus T Pearce and Geraint A Wiggins. Improved methods for statistical modelling of monophonic music. *Journal of New Music Research*, 33(4):367–385, 2004.
- [17] Marcus T Pearce and Geraint A Wiggins. Expectation in melody: The influence of context and learning. *Music Perception*, 23(5):377–405, June 2006.
- [18] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [19] Lutz Prechelt. Early Stopping But When? In *Neural Networks: Tricks of the Trade*, pages 55–69. 2012.
- [20] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 1988.
- [21] Raymond P Whorley. *The Construction and Evaluation of Statistical Models of Melody and Harmony*. PhD thesis, 2013.

# VEROVIO: A LIBRARY FOR ENGRAVING MEI MUSIC NOTATION INTO SVG

**Laurent Pugin**

Swiss RISM Office

laurent.pugin@rism-ch.org

**Rodolfo Zitellini**

Swiss RISM Office

rodolfo.zitellini@rism-ch.org

**Perry Roland**

University of Virginia

pdr4h@virginia.edu

## ABSTRACT

Rendering symbolic music notation is a common component of many MIR applications, and many tools are available for this task. There is, however, a need for a tool that can natively render the Music Encoding Initiative (MEI) notation encodings that are increasingly used in music research projects. In this paper, we present Verovio, a library and toolkit for rendering MEI. A significant advantage of Verovio is that it implements MEI's structure internally, making it the best suited solution for rendering features that make MEI unique. Verovio is designed as a fast, portable, lightweight tool written in pure standard C++ with no dependencies on third-party frameworks or libraries. It can be used as a command-line rendering tool, as a library, or it can be compiled to JavaScript using the Emscripten LLVM-to-JavaScript compiler. This last option is particularly interesting because it provides a complete in-browser music MEI typesetter. The SVG output from Verovio is organized in such a way that the MEI structure is preserved as much as possible. Since every graphic in SVG is an XML element that is easily addressable, Verovio is particularly well-suited for interactive applications, especially in web browsers. Verovio is available under the GPL open-source license.

## 1. INTRODUCTION

A few decades ago, rendering music notation by computer almost exclusively targeting printed output, most often in Postscript or PDF formats. Today, partly in response to the development of MIR applications, rendering of music notation can be necessary in a wide range of contexts, for example within standalone desktop applications, in server-side web application scenarios, or directly in a web browser. For example, music notation might need to be rendered for displaying search results or for visualizing analysis outputs. Another example is score-following applications, where the passage currently played needs to be displayed and possibly highlighted. Rendering music notation by computer, however, is a complex task. Powerful music notation rendering engines exist in commercial and

open-source notation editors, but these are usually not very modular and cannot easily be integrated within other applications. Other rendering engines, such as LilyPond [13] or Mup [1], can be used; however, they usually require the encoding to be converted to a particular typesetting input syntax. Their architectures and dependencies also often limit the contexts in which their use is possible.

In recent years, the Music Encoding Initiative (MEI) has been increasingly adopted for music research projects [6]. Its large scope (MEI can be used to encode a wide range of music notations, from medieval neumes to common Western music notation), modularity, rich metadata header and numerous other features, including alignment with audio files or performance annotations, make it appropriate for a wide range of MIR applications. Unfortunately, most of the solutions currently available for rendering MEI involve a conversion to another format, either explicitly or internally in the software application used for rendering.

In this paper, we present the Verovio project, a library and toolkit for rendering MEI natively in SVG. Its purpose is to provide a self-contained typesetting engine that is capable of creating high-quality graphical output and that can also be used in different application contexts. In the following section, we describe previous work and existing solutions for rendering MEI and the use of SVG for music notation. We then introduce Verovio, describe the MEI structure on which it is built, outline its programming architecture, and highlight features currently available. We then present possible uses and output examples and conclude the paper with the future work that is planned for Verovio.

## 2. PREVIOUS WORK

One currently available option for rendering MEI is conversion to another format in order to use existing tools that do not support MEI. For software applications or rendering engines that support the import of the MusicXML interchange format, MEI can be converted with the mei2musicxml XSL stylesheet [9]. Another option is to convert MEI directly to a typesetting format, such as Mup. Mup is a C rendering engine that was made open-source in 2012. It uses its own typesetting syntax and produces high quality Postscript output. The conversion of MEI to Mup can be achieved in one step using the mei2mup XSL stylesheet [8]. A similar approach is pos-



© Laurent Pugin, Rodolfo Zitellini, Perry Roland.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Laurent Pugin, Rodolfo Zitellini, Perry Roland. "Verovio: A Library for Engraving MEI Music Notation into SVG", 15th International Society for Music Information Retrieval Conference, 2014.

sible for rendering MEI in a web browser, using a conversion to the ABC format. ABC is an encoding format primarily targeting material with fairly limited notational features, such as folk and traditional melodies. It can be rendered in a web browser with the abcjs renderer [15], and the conversion from MEI to ABC can be achieved with the mei2abc converter [5]. There is also a new JavaScript library, MEItoVexflow [18], that makes it possible to render MEI directly in web browsers using the Vexflow API [12]. Another tool for rendition of MEI online is Neon.js [3]. The tool not only renders, but also acts as a full online editor for neumatic medieval notation.

SVG for music notation has been used in several projects. One early attempt was made in 2003 for converting MusicXML to SVG using XSLT [14]. A framework with an editor was also developed for outputting SVG from GuidoXML notation as part of a dissertation thesis [2]. With MEI, SVG rendering was used for the first time in the DiMusEd project, a critical edition of songs of Hildegard von Bingen (1098-1179) [11]. In this web-based edition of neumatic notation, MEI rendering is performed on the server side with a custom rendering engine. There are also attempts to use XSLT to transform MEI to SVG directly in the browser. This approach is used in mono:di, the transcription software of the Corpus Monodicum editorial project sponsored by the Akademie der Wissenschaften und der Literatur in Mainz, also focused on medieval notation [4]. Finally, SVG is a possible back-end for the aforementioned Vexflow API in conjunction with the Raphael JavaScript library.

These solutions all have strengths and drawbacks in terms of compatibility, usability, speed, output quality, and music notation features available. Many of them, however, have limitations when the format to which MEI is converted for rendering does not support some features encoded in the MEI source or has a different structure, with the consequence that part of the encoding will be lost in conversion, or not rendered appropriately.

### 3. VEROVIO

#### 3.1 MEI structure

The MEI schema provides multiple options for structuring the musical content. The most widely-used option is the score-based structure, where all the parts of a musical score are encoded together in the same XML sub-tree. The MEI schema also includes a part-based option, where each part is stored in a separate XML sub-tree. The choice between these options can depend not only on the type of document being encoded but also on the type of application. The Verovio library was designed as a direct implementation of the MEI structure. However, since it is rendering-focused, it is built around another content organization of MEI, a page-based customization more appropriate for graphical display. In a rendering task, the

page (or more generically, the rendering surface) is a required high-level entity on which elements can be laid out by the rendering process. The page-based customization is a more fitting alternative data organization that provides a page top-level entity. It prioritizes the hierarchy that is treated as secondary when encoded with milestone elements `<pb>` in other MEI representations.

In the page-based customization, the content of the music is encoded in `<page>` elements that are themselves contained in a `<pages>` element within `<mdiv>` as shown in Figure 1. A `<page>` element contains `<system>` elements. From then on, the encoding is identical to standard MEI. That is, a `<system>` element will contain `<measure>` elements or `<staff>` elements that are both un-customized, depending on whether or not the music is measured or un-measured, respectively. Since the modifications introduced by the customization are very limited, the Verovio library can be used to render un-customized MEI files. When loading un-customized MEI documents, some MEI elements are loaded by Verovio and converted to a page-based representation. Typically, `<pb>` milestone elements are converted to `<page>` container elements. Conversely, `<section>` container elements are converted to `<secb>` milestone elements.

```

<body>
  <mdiv>
    <pages>
      <page page.width="2108" page.height="2970" page.leftmar="20">
        <system system.leftmar="50" system.rightmar="50">
          <scoreDef>
            <staffGrp symbol="line">
              <staffDef n="1" clef.line="2" clef.shape="G" />
            </staffGrp>
          </scoreDef>
          <measure n="1">
            <staff n="1">
              <layer n="1">
                <!-- ... -->
              </layer>
            </staff>
          </measure>
        </system>
      </page>
    </pages>
  </mdiv>
</body>

```

**Figure 1.** The page-based MEI structure used by Verovio. The `<mdiv>` element contains `<pages>`, `<page>` and `<system>` elements.

#### 3.1.1 Layout and positioning

In addition to making rendering simpler and faster, the idea of the page-based customization is also to make it possible to encode the positioning of elements directly in the content tree without having to refer to the facsimile sub-tree. The latter traditional approach remains available with the page-based customization for more detailed and more complex referencing to facsimile images. However, the page-based customization introduces a lightweight positioning and referencing system that can be useful when the encoding represents a single source with one

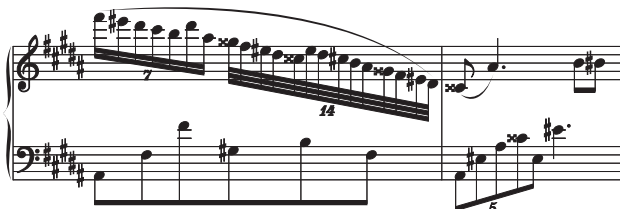
image per page. This is typically the case with optical music recognition applications for which the encoding of the position of each encoded element is necessary. Another possible use is the creation of overlay images to be displayed on top of facsimile images where the position of each symbol also needs to be encoded. Verovio supports both positioned elements and automatic layout. Automatic layout will be executed when un-customized MEI files are rendered.

### 3.1.2 Additional supported formats

In addition to MEI, Verovio can render Plain and Easy (PAE) code [7] and DARMS code [16]. PAE and DARMS encodings are widely used for encoding incipits, including those for the Répertoire International des Sources Musicales (RISM) project. In Verovio, these formats are converted to MEI internally, which means that the toolkit can also be used to convert them to MEI for purposes other than rendering.

## 3.2 SVG output

One significant advantage of SVG rendering over other formats (e.g., Postscript or PDF) is that it is rendered natively in most modern web browsers with no plug-in required. Because SVG is XML, it has an advantage over raster image formats that every graphical element is addressable, making it well-suited for interactive applications. In a web environment, this makes it easy to highlight notes or symbols, for example. In addition, since SVG is a vector format, the output can also be used for high-quality printing.



**Figure 2.** The output of Verovio for two bars. The built-in layout engine of Verovio avoids symbol collisions as much as possible.

One interesting feature of Verovio is that the SVG is organized in such a way that the MEI structure is preserved as much as possible. For example, a `<note>` element with an `@xml:id` attribute in the MEI file will have a corresponding `<g>` element in the SVG with an `@class` attribute equal to "note" and an `@id` attribute corresponding to the `@xml:id` of the MEI note. This makes interaction with the SVG very easy. The hierarchy of the elements is also preserved. For example, in MEI, a `<beam>` can be the child element of a `<tuplet>`, but the opposite is also possible. The hierarchy is fully preserved in the SVG as shown in Figure 3.

```

<tuplet xml:id="t1" num="3" numbase="2">
  <beam xml:id="b1">
    <note xml:id="n1" pname="d" oct="5" dur="8" />
    <note xml:id="n2" pname="e" oct="5" dur="16" dots="1"/>
    <note xml:id="n3" pname="d" oct="5" dur="32" />
    <note xml:id="n4" pname="c" oct="5" dur="8" accid="s"/>
  </beam>
</tuplet>
<beam xml:id="b2">
  <tuplet xml:id="t2" num="3" numbase="2">
    <note xml:id="n5" pname="d" oct="5" dur="8" />
    <note xml:id="n6" pname="e" oct="5" dur="16" dots="1"/>
    <note xml:id="n7" pname="f" oct="5" dur="32" accid="s"/>
    <note xml:id="n8" pname="e" oct="5" dur="8" />
  </tuplet>
</beam>

```

```

<g class="tuplet" id="t1" >
  <g class="beam" id="b1" >
    <g class="note" id="n1" >...</g>
    <g class="note" id="n2" >...</g>
    <g class="note" id="n3" >...</g>
    <g class="note" id="n4" >...</g>
  </g>
</g>
<g class="beam" id="b2" >
  <g class="tuplet" id="t2" >
    <g class="note" id="n5" >...</g>
    <g class="note" id="n6" >...</g>
    <g class="note" id="n7" >...</g>
    <g class="note" id="n8" >...</g>
  </g>
</g>

```

**Figure 3.** Comparison of MEI and SVG file structures. The hierarchy of the MEI is preserved in the SVG.

## 3.3 Programming architecture

Verovio is designed as a fast, portable, lightweight tool usable as a one-step conversion program. It is written in pure standard C++ with no dependencies on third-party frameworks and libraries. This ensures maximum portability of the codebase. Verovio implements its own rendering engine, which can produce SVG with all the musical symbols embedded in it. The musical glyphs are themselves SVG graphics that are included in the Verovio output. This means that no external font needs to be included in the SVG generated from Verovio, limiting dependencies and reducing as far as possible any potential compatibility issues between SVG rendering engines.

The Verovio rendering engine itself is defined as an abstract class, and the SVG output is the default concrete class. This makes it relatively easy to implement a rendering back-end different from SVG (e.g., PDF, or HTML Canvas), if necessary.

The Verovio toolkit has several options for controlling the output. These include options for defining the page size (i.e., the surface, or `<svg>` element size), for setting the amount of zoom, and for choosing whether layout information contained in the MEI file must be taken into account. When there is no layout information provided in the MEI file (no system or page breaks, for example), or when the option for ignoring them is selected, Verovio will extrapolate the necessary layout information.

## 3.4 Features

Verovio currently supports the basic features of simple common Western music notation and mensural notation.

Table 1 shows a list of music notation snippets rendered with Verovio. Figure 4 illustrates how the SVG output of Verovio can be used as facsimile overlay when the positioning feature of the MEI page-based customization is used. The example also illustrates mensural music notation support.

Beams and triplets



Measure rests and key and time signature changes



Clef changes



Trills and fermata



Ties



Grace notes (acciaciature)



Grace notes (appoggiature)



**Table 1.** A list of music notation snippets rendered with the Verovio toolkit. The basic features of simple common Western music notation are accounted for.



**Figure 4.** An example of the output of Verovio placed back on top of a facsimile image and acting as transcription overlay. In this case, positioning information was available in the page-based MEI encoding.

## 4. USE OF VEROVIO

### 4.1 C++ tools and library

Several use cases can be imagined for the Verovio toolkit. First of all, it can be built and used as a standalone command-line tool. This option is well-suited to scripting environments and applications. The command-line tool can be used to render music notation files (in MEI, PAE or DARMS) into SVG. It can also be used to convert DARMS or PAE to MEI. Another option is to use Verovio as a music notation rendering library that can be statically or dynamically linked to full applications. In such cases, it is also relatively easy to implement another drawing back-end for the corresponding C++ graphic environment for the music to be rendered directly on the screen. This is the case with the Aruspix optical music recognition software application where Verovio provides a screen rendering using a wxWidgets back-end instead of the standard SVG one. This approach is conceivable for any C++ graphical environments, be they cross-platform, like the Qt or JUCE toolkits, or platform specific.

### 4.2 JavaScript toolkit

The Verovio toolkit can also be compiled to JavaScript using the Emscripten LLVM-to-JavaScript compiler [19]. In this case, it behaves similarly to the command-line tool but in the web browser context. This approach is particularly interesting because it provides a complete in-browser music MEI typesetter that can be easily integrated into web-based applications.

Emscripten does not directly translate C++ into JavaScript. Instead it takes the LLVM (Low Level Virtual Machine) byte code generated by the Clang compiler from the C++ code as a base for the conversion to JavaScript. This has several advantages. Most importantly, the level of completeness in terms of C++ language feature support is extremely high since the idiomatic features of C++ did not have to be explicitly translated into JavaScript in the Emscripten compiler (only the translation from LLVM was necessary). In fact, for the Verovio toolkit, the Emscripten compiler is applied on exactly the same codebase as the C++ compiler, and no change to the code had to be done for this to work. Only the compilation makefile is different.

Another advantage of this approach is that the JavaScript produced is very fast because it benefits from all the code optimization performed by the Clang compiler when generating the LLVM byte code. Furthermore, in addition to standard JavaScript, Emscripten can also generate asm.js code, a subset of JavaScript that has the advantage of being highly optimizable. On web browsers that support asm.js (currently Firefox, Chrome and Opera), the execution speed is only up to about 1.6 times slower than with the native C++ executable. Table 2 shows the system time required to load an MEI file of 120 pages of

music (7 MB) and for displaying the first page with the native executable and with three web browsers. The figures are the median value of the operation repeated 100 times.

	Native	Firefox	Chrome	Safari
<b>System time in sec.</b>	0.657	1.054	1.364	1.811
<b>Comparison to native</b>	-	1.6	2.1	2.8

**Table 2.** The system time in seconds for loading an MEI files (120 pages, 7 MB) and for displaying the first page. The second line gives the ratio with the native executable time for the three web browsers used for comparison.

The JavaScript version of the Verovio toolkit is easy to use in web environments. It is packed in one single file which size is only about 1.2 MB. It is available as a JavaScript class, and all the options of the command-line version are supported in the toolkit. The options can be passed to the toolkit in JSON format, and the SVG output can be directly fed to HTML objects for display. The Figure 5 shows a HTML and Javascript code snippet for loading an MEI file using a jQuery HTTP GET request.

```

<script src="verovio-toolkit.js"></script>
<!-- The div where we are going to insert the SVG -->
<div id="output">
<script type="text/javascript">
  /* Create the Verovio toolkit instance */
  var vrvToolkit = new verovio.toolkit();
  /* Load the file using HTTP GET */
  $.get( "mei-file.mei", function( data ) {
    /* Render the data */
    var svg = vrvToolkit.renderData(
      data + "\n",
      JSON.stringify({ inputFormat: 'mei' } ) );
    /* Insert the data as content of the #output div */
    $("#output").html(svg);
  });
</script>

```

**Figure 5.** A JavaScript example for loading an MEI file. The toolkit parameters can be set using JSON.

The layout of the MEI data is performed on loading. Once the file is loaded into memory, it remains accessible in the toolkit instance. The class provides methods for getting the number of pages or for navigating through them, making it convenient to integrate the toolkit in a JavaScript application.

The Figure 6 shows a screenshot of a web application where the toolkit was turned into an online MEI file viewer. The application works on desktop computers but also on tablets and mobile devices. The JavaScript toolkit has been tested with recent versions of the most widely used web-browsers. Internet Explorer requires at least version 10.



**Figure 6.** An example of a web-based MEI viewer built with the Verovio toolkit. Large MEI files can be loaded and displayed in the web browser in a very convenient way.

## 5. CONCLUSION AND FUTURE WORK

Verovio is a toolkit for rendering MEI in SVG that can be used in different application environments, including online. It is designed with MEI in mind, making it the right basis for implementing encoding features that are specific to MEI. It will avoid problematic situations that occur when using rendering engines based on other formats and that implement a different data structure. Even if at this stage, the supported features can be in some cases more limited than with other rendering options, Verovio already implements many important features for rendering both common Western music notation and mensural notation.

Current work on Verovio includes the adoption of the Standard Music Font Layout (SMuFL) [17] for supporting other fonts converted to SVG glyphs, the improvement of the SVG structure and adding support for additional MEI elements and attributes. The priority is given to features specific to MEI. The future work will include the development of a prototype for making Verovio a possible basis for an online MEI editor. It will also include the creation of an MEI application profile for Verovio using the TEI One Document Does-it-all (ODD) approach. The corresponding XSL stylesheets for converting to it other MEI profiles will also be provided. Adding the import of other encoding formats is also envisaged in the future.

## 6. AVAILABILITY

Verovio can be downloaded from <http://www.verovio.org> and is available under the GPLv3 open-source license. The website also includes documentation on currently available features.

## 7. REFERENCES

- [1] Arkkra Enterprises, *Mup*. <<http://www.arkkra.com>>
- [2] G. A. Bays: *ScoreSVG: A New Software Framework for Capturing the Semantic Meaning and Graphical Representation of Musical Scores Using Java2D, XML, and SVG*. Diss. Georgia State Univ., 2005.
- [3] G. Bulet, A. Porter, A. Hankinson, and I. Fujinaga: “Neon.js: Neume Editor Online,” *Proceedings of the 13th International Society on Music Information Retrieval Conference*, pp. 121–6, 2012.
- [4] Corpus Monodicum, *mono:di*. <<http://monodi.corpus-monodicum.de>>
- [5] Ediom, *mei2abc*. <<https://github.com/ediorom/mei2abc>>
- [6] A. Hankinson, P. Roland, and I. Fujinaga: “The Music Encoding Initiative as a document-encoding framework,” *Proceedings of the 12th International Society on Music Information Retrieval Conference*, pp. 293–8, 2011.
- [7] J. Howard: “Plaine and Easie code: A code for music bibliography,” in Selfridge-Field, E. (Ed.), *Beyond MIDI: The Handbook of Musical Codes*. The MIT Press, Cambridge, pp. 362–72, 1997.
- [8] MEI, *mei2mup*. <<http://code.google.com/p/music-encoding/source/browse/trunk/tools/mei2mup>>
- [9] MEI, *mei2musicxml*. <<https://code.google.com/p/music-encoding/source/browse/trunk/tools/mei2musicxml>>
- [10] MEI-incubator, *page-based customization*, <<https://code.google.com/p/mei-incubator/source/browse/page-based>>
- [11] S. Morent: “Digitale Edition älterer Musik am Beispiel des Projekts Tübingen,” in *Digitale Edition zwischen Experiment und Standardisierung. Musik – Text – Codierung*, pp. 89–109, 2009.
- [12] M. Muthanna, *VexFlow*. <<https://github.com/0xfe/vexflow>>
- [13] H. W. Nienhuys and J. Nieuwenhuizen: “LilyPond, a system for automated music engraving,” *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, pp. 167–72, 2003.
- [14] L. O’Shea: “Stirring XML: Visualizations in SVG: MusicML2SVG,” *Proceedings of the SVGOpen2003 Conference*, pp. 2–6, 2003.
- [15] P. Rosen, *abcjs*. <<http://github.com/paulrosen/abcjs>>
- [16] E. Selfridge-Field: “DARMS, its dialects, its uses,” in Selfridge-Field, E. (Ed.), *Beyond MIDI: The Handbook of Musical Codes*. The MIT Press, Cambridge, pp. 163–74, 1997.
- [17] Steinberg, *Standard Music Font Layout*. <<http://www.smufi.org>>
- [18] TEI Music SIG, *MEItoVexFlow*. <<http://github.com/tei-music-sig/meitovexflow>>
- [19] A. Zakai: “Emscripten: an LLVM-to-JavaScript compiler,” *Companion to the 26th Annual ACM OOPSLA Conference*, pp. 301–12, 2011.



# MUSIC CLASSIFICATION BY TRANSDUCTIVE LEARNING USING BIPARTITE HETEROGENEOUS NETWORKS

Diego F. Silva, Rafael G. Rossi, Solange O. Rezende, Gustavo E. A. P. A. Batista  
 Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo  
 {diegoasilva,ragero,solange,gbatista}@icmc.usp.br

## ABSTRACT

The popularization of music distribution in electronic format has increased the amount of music with incomplete metadata. The incompleteness of data can hamper some important tasks, such as music and artist recommendation. In this scenario, transductive classification can be used to classify the whole dataset considering just few labeled instances. Usually transductive classification is performed through label propagation, in which data are represented as networks and the examples propagate their labels through their connections. Similarity-based networks are usually applied to model data as network. However, this kind of representation requires the definition of parameters, which significantly affect the classification accuracy, and presents a high cost due to the computation of similarities among all dataset instances. In contrast, bipartite heterogeneous networks have appeared as an alternative to similarity-based networks in text mining applications. In these networks, the words are connected to the documents which they occur. Thus, there is no parameter or additional costs to generate such networks. In this paper, we propose the use of the bipartite network representation to perform transductive classification of music, using a bag-of-frames approach to describe music signals. We demonstrate that the proposed approach outperforms other music classification approaches when few labeled instances are available.

## 1. INTRODUCTION

The popularity of online music services has dramatically increased in the last decade. The revenue of online services, such as music streaming, has more than triplicate in the last three years. Online services already account for a significant 40% of the overall industry trade revenues [1]. However, the popularization of music and video clips distribution in electronic format has increased the amount of music with incomplete metadata. The incompleteness of

the data can hamper some important tasks such as indexing, retrieval and recommendation.

For instance, users of music services commonly define their preferences based on genre information. A recommendation system can make use of such information to suggest other music conditional to the expressed preferences. The lack of genre information on music imposes limits to the capability of the recommendation systems to correctly identify consume patterns as well as to recommend a diverse set of music. Similar statements can be made for music indexing and retrieval.

Due to the academic and commercial importance of digital music, we have witnessed in the last decade a tremendous increase of interest for Music Information Retrieval (MIR) tasks. Most of the proposed MIR methods are based on supervised learning techniques. Supervised learning usually requires a substantial amount of correctly labeled data in order to induce accurate classifiers. Although labeled data can be obtained with human supervision, such process is usually expensive and time consuming. A more practical approach is to employ methods that can avail of both a small number of labeled instances and a large amount of unlabeled data.

Transductive learning directly estimates the labels of unlabeled instances without creating a classification model. A common approach to perform transductive classification is label propagation, in which the dataset is represented as a network and the labels of labeled instances are propagated to the unlabeled instances through the network connections. Similarity-based networks are usually applied to represent data as networks for label propagation [19]. However, they present a high cost due to the computation of the similarities among all dataset instances, and require the definition of several graph construction parameters that can significantly affect the classification accuracy [11].

Bipartite heterogeneous networks have appeared as an alternative to similarity-based networks in sparse domains, such as text mining [9, 10]. In these networks, words are connected to documents in which they occur. Thus, there are no parameters or additional costs to generate such networks. In a similar way, we can represent music collections as a bipartite network though the use of a bag-of-frames (BoF) representation. The BoF is a variation of bag-of-words (BoW) representation used in text analysis and has been applied in studies of genre recognition, music similarity, and others [12].



© Diego F. Silva, Rafael G. Rossi, Solange O. Rezende, Gustavo E. A. P. A. Batista.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Diego F. Silva, Rafael G. Rossi, Solange O. Rezende, Gustavo E. A. P. A. Batista. "Music Classification by Transductive Learning Using Bipartite Heterogeneous Networks", 15th International Society for Music Information Retrieval Conference, 2014.

In this paper, we propose the use of the bipartite network representation to perform transductive classification of music, using a BoF approach to describe music signals. We demonstrate that the proposed approach outperforms other music classification approaches when few labeled instances are available.

## 2. BACKGROUND & RELATED WORK

Transductive classification is a useful way to classify all dataset instances when just few labeled instances are available [19]. Perhaps the most common and intuitive way to perform transductive classification is through label propagation, which is commonly made by using similarity-based networks to represent the data. Usual ways to generate similarity-based networks are [17–19]: (i) fully connected-network, in which every pair of instances are connected; (ii)  $k$  nearest neighbor, in which an instance is connected with its  $k$  most similar instances; and (iii)  $\epsilon$  network, in which two instances are connected if their distance is above a threshold.

Bipartite networks have appeared as an alternative to similarity-based networks in sparse domains such as texts [9, 10]. The use of bipartite networks to represent text collections and the use of algorithms which perform label propagation in bipartite networks obtained results as good as the obtained by similarity-based networks [10]. However, the computation cost to generate similarity-based networks is  $O(|I|^2 \times |A|)$ , in which  $|I|$  is the number of instances and  $|A|$  is the number of attributes of a dataset, while the computational cost to generate bipartite networks is  $O(|I| \times |A|)$ . Moreover, the generation of bipartite networks is parameter-free.

We can represent music collections as a bipartite network though the generation of a bag-of-frames (BoF). Methods using BoF has become common in different MIR tasks, including similarity, genre, emotion and cover song recognition [12]. Such strategies basically consist of three main steps: feature extraction, codebook generation and learning/classification.

Probably, the most simple and commonly used strategy for the codebook generation is the Vector Quantization (VQ). Basically, the VQ uses clustering algorithms on the frame-level features and consider the center of clusters as the words of a dictionary. The simple  $k$ -means is, probably, the most used algorithm in this step and showed to achieve similar results to other methods [8].

Recently, new tools have emerged for creating codebooks. Specifically, strategies based on Sparse Coding [5, 15] and Deep Belief Networks [3, 7] have been widely used. However, even though these strategies often improve the results in different domains, they can present similar performance to simple strategies such as VQ in certain tasks [7].

## 3. PROPOSED FRAMEWORK: MC-LPBN

In this paper we propose a framework called MC-LPBN (Music Classification through Label Propagation in Bi-

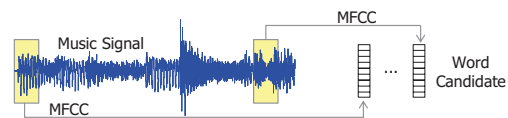


Figure 1. Word candidates generation process

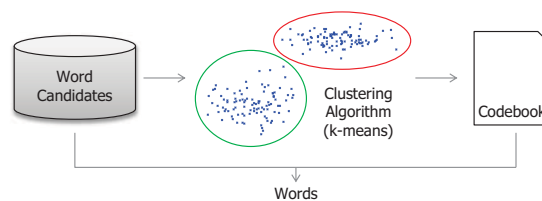


Figure 2. The word candidates are clustered and each centroid is elected as a codeword. The word frequency is directly related to the candidates count in each cluster

partite Networks) to perform transductive classification of musics. The proposed framework has three main steps: (i) codebook generation, (ii) network generation for transductive classification, and (iii) transductive classification using bipartite heterogeneous networks. In the next subsections we present the details of each step.

### 3.1 Codebook Generation and Bag-of-Frames

In order to represent a music collection as a BoF it is necessary to extract a set of words. Such procedure starts with the extraction of word candidates. A word candidate is a set of features extracted from a single window. As a sliding window swipes across the entire music signal, each music gives origin to a set of word candidates. In this work we use MFCC as feature extraction procedure. Figure 1 illustrates the word candidates generation process.

The next step is the creation of a codebook. A codebook is a set of codewords used to associate the word candidates to a finite set of words. The idea is to select the most representative codeword for each word candidate. To do this, we use a clustering algorithm with the word candidates and consider the center of each cluster as a codeword. So, each candidate is associated to the codeword that represents the cluster it belongs. In this step, we used the ubiquitous  $k$ -means algorithm, due to its simplicity and efficiency. Figure 2 illustrates this procedure.

Finally, there is a step to the generation of a BoF matrix. In such a matrix, each line corresponds to a music recording, each column corresponds to a word and the cells correspond to the frequency of occurrence of the word in the music. The BoF allows the generation of bipartite networks for transductive classification, as we discuss in the next subsection.

### 3.2 Network Generation for Transductive Classification

Formally, a network is defined by  $N = \langle \mathcal{O}, \mathcal{E}, \mathcal{W} \rangle$ , in which  $\mathcal{O}$  represents the set of objects (entities) of a problem,  $\mathcal{E}$  represents the set of connections among the objects and  $\mathcal{W}$  represents the weights of the connections. When

$\mathcal{O}$  is composed by a single type of object, the network is called homogeneous network. When  $\mathcal{O}$  is composed by  $h$  different types of objects, i.e.,  $\mathcal{O} = \mathcal{O}_1 \cup \dots \cup \mathcal{O}_h$ , the networks is called heterogeneous network [6].

The music collection can be represented by a bipartite heterogeneous network with  $\mathcal{O} = \mathcal{M} \cup \mathcal{T}$ , in which  $\mathcal{M} = \{m_1, m_2, \dots, m_n\}$  represents the set of music and  $\mathcal{T} = \{t_1, t_2, \dots, t_l\}$  represents the set of words.  $\mathcal{M}$  is composed by labeled ( $\mathcal{M}^L$ ) and unlabeled ( $\mathcal{M}^U$ ) music, i.e.,  $\mathcal{M} = \mathcal{M}^L \cup \mathcal{M}^U$ . A music  $m_i \in \mathcal{M}$  and a word  $t_j \in \mathcal{T}$  are connected if  $t_j$  occurs in  $m_i$ . The weight of the relation between  $m_i$  and  $t_j$  ( $w_{m_i, t_j}$ ) is the frequency of  $t_j$  in  $m_i$ . Thus, only the words and their frequencies in the music are needed to generate the bipartite network.

For transductive classification based on networks, let  $\mathcal{C} = \{c_1, c_2, \dots, c_l\}$  be the set of possible labels, and let  $\mathbf{f}_{o_i} = \{f_1, f_2, \dots, f_{|\mathcal{C}|}\}^T$  be the weight vector of an object  $o_i$ , which determines its weight or relevance for each class. Hence, it is also referred as class information vector. The class information of an object  $m_i \in \mathcal{M}$  or an object  $t_j \in \mathcal{T}$  is denoted respectively by  $\mathbf{f}_{m_i}$  and  $\mathbf{f}_{t_j}$ . All the class information of objects in  $\mathcal{M}$  or  $\mathcal{T}$  is denoted by the matrices  $\mathbf{F}(\mathcal{M}) = \{\mathbf{f}_{m_1}, \mathbf{f}_{m_2}, \dots, \mathbf{f}_{m_{|\mathcal{M}|}}\}^T$  and  $\mathbf{F}(\mathcal{T}) = \{\mathbf{f}_{t_1}, \mathbf{f}_{t_2}, \dots, \mathbf{f}_{t_{|\mathcal{T}|}}\}^T$ . The class information of a labeled music  $m_i$  is stored in a vector  $\mathbf{y}_{m_i} = \{y_1, y_2, \dots, y_{|\mathcal{C}|}\}^T$ , which has the value 1 to the corresponding class position and 0 to the others. The weights of connections among objects are stored in a matrix  $\mathbf{W}$ . A diagonal matrix  $\mathbf{D}$  is used to store the degree of the objects, i.e., the sum of the connection weights of the objects. Thus the degree of a music  $m_i$  in a bipartite network is ( $d_{m_i} = d_{i,i} = \sum_{t_j \in \mathcal{T}} w_{m_i, t_j}$ ).

### 3.3 Transductive Classification Using Bipartite Heterogeneous Networks

The main algorithms for transductive classification in data represented as networks are based on regularization [19], which have to satisfy two assumptions: (i) the class information of neighbors must be close; and (ii) the class information assigned during the classification process must be close to the real class information. In this paper we used three regularization-based algorithms: (i) Tag-based classification Model (TM) [16], (ii) Label Propagation based on Bipartite Heterogeneous Networks (LPBHN) [10], and (iii) GNetMine (GM) [6].

TM algorithm minimizes the differences among (i) the real class information and the class information assigned to music ( $\mathcal{M}$ ), (ii) the real class information and the class information assigned to and objects from other domains that aid the classification ( $\mathcal{A}$ ), and (iii) the class information among words ( $\mathcal{T}$ ) and objects in ( $\mathcal{M}$ ) or ( $\mathcal{A}$ ), as presented in Equation 1.

$$Q(\mathbf{F}) = \alpha \sum_{a_i \in \mathcal{A}} \|\mathbf{f}_{a_i} - \mathbf{y}_{a_i}\|^2 + \beta \sum_{m_i \in \mathcal{M}^L} \|\mathbf{f}_{m_i} - \mathbf{y}_{m_i}\|^2 \quad (1)$$

$$+ \gamma \sum_{m_i \in \mathcal{M}^U} \|\mathbf{f}_{m_i} - \mathbf{y}_{m_i}\|^2 + \sum_{o_i \in \mathcal{M} \cup \mathcal{A}} \sum_{t_j \in \mathcal{T}} w_{o_i, t_j} \|\mathbf{f}_{o_i} - \mathbf{f}_{t_j}\|^2$$

The parameters  $\alpha$ ,  $\beta$  and  $\gamma$  control the importance given for each assumption of TM. Objects are classified using class mass normalization [18].

LPBHN is a parameter-free algorithm based on the Gaussian Fields and Harmonic Functions (GFHF) algorithm [18], which performs label propagation in homogeneous networks. The difference is that LPBHN considers the relations among different types of objects. The function to be minimized by LPBHN is:

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{m_i \in \mathcal{M}} \sum_{t_j \in \mathcal{T}} w_{m_i, t_j} \|\mathbf{f}_{m_i} - \mathbf{f}_{t_j}\|^2 \quad (2)$$

$$+ \lim_{\mu \rightarrow \infty} \mu \sum_{m_i \in \mathcal{M}^L} \|\mathbf{f}_{m_i} - \mathbf{y}_{m_i}\|^2,$$

in which  $\mu$  tending to infinity means that  $\mathbf{f}_{m_i} \equiv \mathbf{y}_{m_i}$ , i.e., the information class of labeled musics do not change.

The GM framework is based on the Learning with Local and Global Consistency (LLGC) algorithm [17], which performs label propagation in homogeneous networks. The difference between the algorithms is that GM considers the different types of relations among the different types of objects. For the problem of music classification using bipartite networks, GM minimizes the differences of (i) the class information among music and words and (ii) the class information assigned to labeled music during the classification and their real class information. The function to be minimized by GM is:

$$Q(\mathbf{F}) = \sum_{m_i \in \mathcal{M}} \sum_{t_j \in \mathcal{T}} w_{m_i, t_j} \left\| \frac{\mathbf{f}_{m_i}}{\sqrt{d_{m_i}}} - \frac{\mathbf{f}_{t_j}}{\sqrt{d_{t_j}}} \right\|^2 \quad (3)$$

$$+ \sum_{m_i \in \mathcal{M}} \mu \|\mathbf{f}_{m_i} - \mathbf{y}_{m_i}\|^2,$$

in which  $0 < \mu < 1$ .

We highlight that all the algorithms presented above have iterative solutions to minimize the respective equations. This allows to obtain similar results to the closed solutions with a lower computational time.

## 4. EXPERIMENTAL EVALUATION

To illustrate the generality of our approach, we evaluate our framework in two different scenarios. In this section, we describe the tasks we considered, the experimental setup used in our experiments, as well as the results obtained and a short discussion about them.

### 4.1 Tasks Description

We evaluate our framework in genre recognition and cover song recognition scenarios. The remaining of this section contains a brief description of each task and the datasets used to each end.

#### 4.1.1 Genre Recognition

Genre recognition is an important task in several applications. Genre is a quality created by the human beings to

intuitively characterize music [14]. For humans, the classification of music by genre is relatively simple task, and can be done by listening to a short excerpt of a music.

Therefore, most of the existing data for this task considers a short duration excerpt for each recording. In this work, we use the GTZAN<sup>1</sup> and Homburg<sup>2</sup> datasets. The first has 1,000 snippets of 30 seconds of ten different genres. The number of instances of each class is equally distributed. The Homburg dataset, in turn, has ten seconds sections of 1,886 recordings, belonging to nine genres. In this case, the genre with fewer instances has only 47 examples, while the largest has 504.

#### 4.1.2 Cover Song Recognition

Cover song may be defined as distinct performances of the same music with differences in tempo, instrumentation, style or other characteristics [4]. Finding reinterpreted music is an important task mainly to commercial ends. For example, it can be used to ensure copyright in websites which allow users to create content.

In this paper, we evaluate our framework in a task similar to the cover song recognition. But, instead find the original recording of a query music, we consider all different interpretations of the same music as the same class.

To evaluate our proposal we used the Mazurkas Project data<sup>3</sup>, in which each music has several versions. This dataset contains 2914 recordings of 49 Chopin's mazurkas for piano (from 43 to 97 versions per class).

## 4.2 Experimental Setup

We evaluated our framework considering different configurations for the 1<sup>st</sup> and 3<sup>rd</sup> steps. For the 1<sup>st</sup> step, we consider variations of parameters of the feature extraction and codebook generation phases. In this work, we use 20 MFCC as frame-level features. This number is a common choice in MIR papers [2]. We use 5 different window sizes, with an overlap of 50% between them: 0.0625, 0.125, 0.25, 0.5 and 0.75 seconds. Finally, we applied the k-means using  $k \in \{100, 200, 400, 800, 1600, 3200\}$ .

For the 3<sup>rd</sup>, we consider the algorithms presented in Section 3.3: Label Propagation using Bipartite Heterogeneous Networks (LPBHN), Tag-based Model (TM), and GNetMine (GM). For GM we use the parameter  $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ . For TM we use  $\alpha = 0$ , since there are no objects from different domains,  $\beta \in \{0.1, 1.0, 10, 100, 1000\}$ , and  $\gamma \in \{0.1, 1.0, 10.0, 100.0, 1000.0\}$ . The iterative solutions proposed by the respective authors were used for all the algorithms. The maximum number of iterations is set to 1000 since this is a common limit value for iterative solutions.

We also carried out experiments using two algorithms for label propagation in similarity-based networks, Learning with Local and Global Consistency (LLGC) [17] and Gaussian Fields and Harmonic Functions (GFHF) [18],

and two classical supervised learning algorithms for comparison with the proposed approach,  $k$  Nearest Neighbors ( $k$ NN) and Support Vector Machines (SVM) [13].

We build similarity-based networks using the fully connected approach with  $\sigma = 0.5$  [19] and we set  $\alpha = \{0.1, 0.3, 0.5, 0.7, 0.9\}$  for LLGC algorithm. For  $k$ NN we set  $k = 7$  and weighted vote, and for SVM we used linear kernel and  $C = 1$  [9].

The metric used for comparison was the classification accuracy, i.e., the percentage of correctly classified music recordings. The accuracies are obtained considering the average accuracies of 10 runs. In each run we randomize the dataset and select  $x$  examples as labeled examples. The remaining  $|\mathcal{M}| - x$  examples are used for measuring the accuracy. We carried out experiments using  $x = \{1, 10, 20, 30, 40, 50\}$  to analyze the trade-off between the number of labeled documents and classification accuracy. The best accuracies obtained by some set of parameters of the algorithms are used for comparison.

## 4.3 Results and Discussion

Given the large amount of results obtained in this work, their complete presentation becomes impossible due to space constraints. Thus, we developed a website for this work, where detailed results can be found<sup>4</sup>. In this section, we summarize the results from different points of view.

### 4.3.1 Influence of Parameters Variation

Our first analysis consists in evaluating the influence of the variation of the codebook generation step parameters. Figure 3 presents the variation of accuracy for both genre recognition dataset and each window size according to a different number of words in the dictionary. To do this, we fixed the number of labeled examples in 10. This number represents a good trade-off between the classification accuracy of the algorithms and the human effort to label music. But, we note that the behaviors are similar to other numbers of properly labeled examples.

The results show that the transductive learning methods can achieve similar or even superior results than the obtained by using inductive models. In the case of GTZAN data, there is a clear increasing pattern when the number of words varies. Using higher values to it, both strategies perform well, but the transductive learning obtained the higher accuracies. The results obtained by similarity-based networks were slightly better in most of configurations. But, as mentioned before, similarity-based networks has a high cost to calculate the similarities between all the examples and require the setting of several parameters to construct the network. In the Homburg dataset, transductive learning is better independently of the parameter configuration. In this case, there are no significant differences between bipartite network approaches, but they performed better than the similarity-based networks in most of cases.

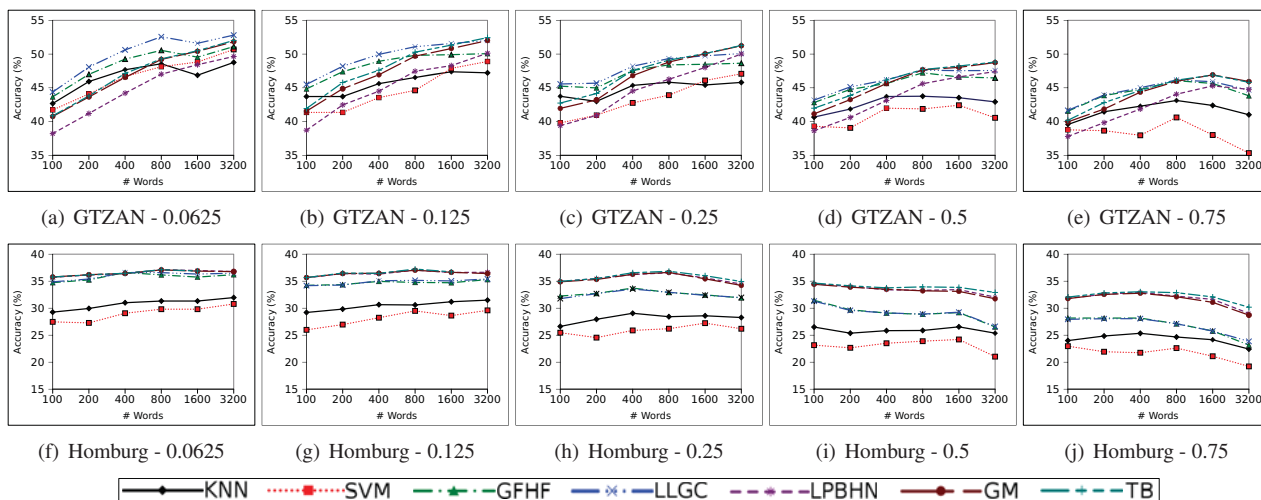
In order to evaluate our framework in the cover song recognition, we used the LPBHN algorithm, that achieved

<sup>1</sup> [http://marsyas.info/download/data\\_sets/](http://marsyas.info/download/data_sets/)

<sup>2</sup> <http://www-ai.cs.uni-dortmund.de/audio.html>

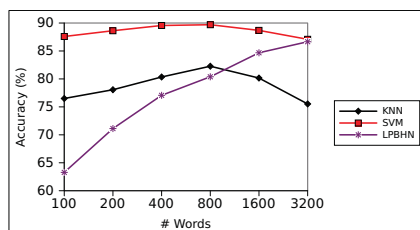
<sup>3</sup> <http://www.mazurka.org.uk/>

<sup>4</sup> <http://sites.labicc.icmc.usp.br/dfs/ismir2014>



**Figure 3.** Accuracy for genre recognition by varying the number of words in the codebook. The number of labeled examples per class is fixed in 10.

similar result to other transductive methods and has the advantage of being parameter-free, and both, SVM and  $k$ NN, inductive approaches. The results show a high increasing pattern when transductive learning is used, and a more stable pattern to supervised methods. Figure 4 shows the accuracy achieved by fixing the number of labeled examples in 10. We fixed the window size to 0.75 seconds, since Mazurkas is the larger dataset used in this work and this is the fastest configuration to the feature extraction phase. We believe that the performance of transductive learning can overcome the SVM if we increase the number of words.



**Figure 4.** Accuracy for Mazurkas by varying the number of words in the codebook. The number of labeled examples is fixed in 10 and the window length in 0.75 seconds.

#### 4.3.2 Number of Labeled Examples Variation

The evaluation of the performance variation according to the number of labeled examples is an important analysis in the context of transductive learning. Figure 5 shows the behavior of accuracy in genre recognition task when there is a variation in the number of labeled examples that belong to each class. The results were obtained by fixing the number of words in 3200, in which good results were achieved in several configurations, and a window size of the middle value of 0.25 seconds, since the results were similar than the obtained with other values to this parameter.

To analyze these graphs, it is interesting to know the proportion that the number of labeled examples represents in each dataset. For example, in the case of GTZAN set,

50 examples correspond to exactly 50% of the examples in each classes. In the case of Homburg, it represents 100% of the minority class, but less than 10% of the majority.

In both cases, the behavior of the accuracies was similar. As the number of labeled samples increases, the performance becomes better. The transductive learning methods performed better across the curve. As the proportion of the number of labeled instances increases, the tendency is that the performance of inductive algorithms approaches the performance of transductive algorithms.

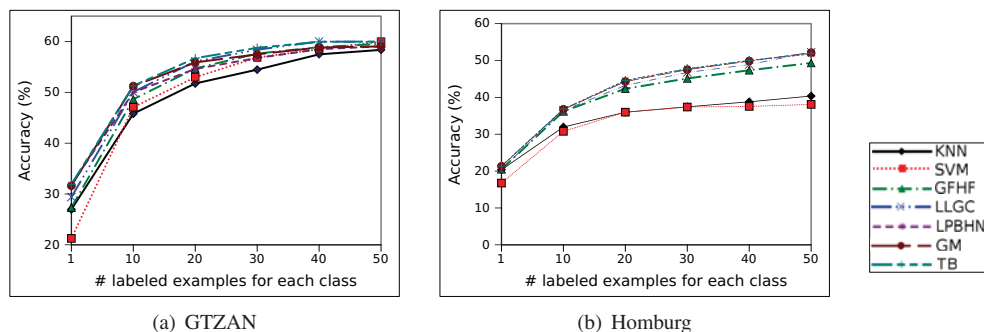
For sake of space limitations, we omitted the results for the cover song recognition task. However, we point out that the behavior of accuracy rates were very similar to obtained in the other task.

## 5. CONCLUSION

In this paper, we presented a framework for transductive classification of music using bipartite heterogeneous networks. We show that we can have a better performance by using this approach instead the traditional inductive learning. Our results were close or superior to the obtained by similarity-based networks. This kind of network, however, requires several parameters and has a high cost due to the calculation of the similarity between the instances.

We should note that the accuracy rates achieved in this paper are worse than some results presented in related works. For example, there are some papers that achieved accuracy higher than 80% to the GTZAN dataset using BoF approaches. However, these results were probably obtained due to the choice of specific features and parameters. Moreover, these papers used inductive learning approaches, with labels for the entire dataset. Nevertheless, we demonstrated that, for the same parameter set, the use of bipartite heterogeneous network achieved the best results.

As future work we will investigate a better feature configuration and different codebook generation strategies.



**Figure 5.** Accuracy of genre recognition by varying the numbers of labeled examples for each class. The number of words and the windows length are fixed in 3200 and 0.25 seconds, respectively.

**ACKNOWLEDGEMENTS:** We would like to thank the financial supports by the grants 2011/12823-6, 2012/50714-7, 2013/26151-5, and 2014/08996-0, Sao Paulo Research Foundation (FAPESP).

## 6. REFERENCES

- [1] Recording industry in numbers. Technical report, International Federation of the Phonographic Industry, 2014.
- [2] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.
- [3] S. Dieleman, P. Brakel, and B. Schrauwen. Audio-based music classification with a pretrained convolutional network. In *ISMIR*, pages 669–674, 2011.
- [4] D. P. W. Ellis and T. Bertin-Mahieux. Large-scale cover song recognition using the 2d fourier transform magnitude. In *ISMIR*, pages 241–246, 2012.
- [5] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun. Unsupervised learning of sparse features for scalable audio classification. In *ISMIR*, pages 681–686, 2011.
- [6] M. Ji, Y. Sun, M. Danilevsky, J. Han, and J. Gao. Graph regularized transductive classification on heterogeneous information networks. In *Proc. Eur. Conf. on Machine Learning and Knowledge Discovery in Databases*, pages 570–586. Springer-Verlag, 2010.
- [7] J. Nam, J. Herrera, M. Slaney, and J. O. Smith. Learning sparse feature representations for music annotation and retrieval. In *ISMIR*, pages 565–570, 2012.
- [8] M. Riley, E. Heinen, and J. Ghosh. A text retrieval approach to content-based audio retrieval. In *ISMIR*, pages 295–300, 2008.
- [9] R. G. Rossi, de T. P. Faleiros, de A. A. Lopes, and S. O. Rezende. Inductive model generation for text categorization using a bipartite heterogeneous network. In *Proc. Intl. Conf. on Data Mining*, pages 1086–1091. IEEE, 2012.
- [10] R. G. Rossi, A. A. Lopes, and S. O. Rezende. A parameter-free label propagation algorithm using bipartite heterogeneous networks for text classification. In *Proc. Symp. on Applied Computing*, pages 79–84. ACM, 2014.
- [11] C. A. R. Sousa, S. O. Rezende, and G. E. A. P. A. Batista. Influence of graph construction on semi-supervised learning. In *Proc. Eur. Conf. Machine Learning and Knowledge Discovery in Databases*, pages 160–175. Springer-Verlag, 2013.
- [12] L. Su, C.-C.M. Yeh, J.-Y. Liu, J.-C. Wang, and Y.-H. Yang. A systematic evaluation of the bag-of-frames representation for music information retrieval. *IEEE Transactions on Multimedia*, 16(5):1188–1200, Aug 2014.
- [13] P.-N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.
- [14] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE transactions on Speech and Audio Processing*, 10(5):293–302, 2002.
- [15] C. C. M. Yeh, L. Su, and Y. H. Yang. Dual-layer bag-of-frames model for music genre classification. In *Intl. Conf. on Acoustics, Speech and Signal Processing*, 2013.
- [16] Z. Yin, R. Li, Q. Mei, and J. Han. Exploring social tagging graph for web object classification. In *Proc. Intl. Conf. on Knowledge Discovery and Data Mining*, pages 957–966, 2009.
- [17] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, volume 16, pages 321–328, 2004.
- [18] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. Intl. Conf. on Machine Learning*, pages 912–919. AAAI Press, 2003.
- [19] X. Zhu, A. B. Goldberg, R. Brachman, and T. Dietterich. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers, 2009.

# AUTOMATIC MELODY TRANSCRIPTION BASED ON CHORD TRANSCRIPTION

Antti Laaksonen

Department of Computer Science

University of Helsinki

ahslaaks@cs.helsinki.fi

## ABSTRACT

This paper focuses on automatic melody transcription in a situation where a chord transcription is already available. Given an excerpt of music in audio form and a chord transcription in symbolic form, the task is to create a symbolic melody transcription that consists of note onset times and pitches. We present an algorithm that divides the audio into segments based on the chord transcription, and then matches potential melody patterns to each segment. The algorithm uses chord information to favor melody patterns that are probable in the given harmony context. To evaluate the algorithm, we present a new ground truth dataset that consists of 1.5 hours of audio excerpts together with hand-made melody and chord transcriptions.

## 1. INTRODUCTION

Melody and chords have a strong connection in Western music. The purpose of this paper is to exploit this connection in automatic melody transcription. Given a chord transcription, we can use it in melody transcription to constrain the set of possible melodies. Both the rhythm and the pitches of the melody should match the chords in a successful melody transcription.

For example, let us consider the melody in Figure 1. The melody consists of 16 bars, each annotated with a chord symbol. The first observation is that chord boundaries divide the melody into segments of approximately equal length. Each of the segments has a simple rhythmical structure. In this case the chord boundaries exactly match the bar lines, and each segment contains up to three melody notes. Of course, many melodies are more complex than this, but the underlying segmentation is still usually apparent.

Let us now consider the pitches of the melody. The key of the melody mainly determines what pitches typically occur in the melody. In this example the key of the melody is C major, and almost all melody pitches belong to the C major scale. However, there are two exceptions: the G#

**Figure 1:** A melody from Disney’s *Snow White and the Seven Dwarfs*. The chord transcription consists of 16 chords, and the melody transcription consists of 30 notes.

note in the second bar and the C# note in the sixth bar. Thus, although the melody follows the C major scale, the individual chords also have an effect on the pitches. In this case the major thirds of E major and A major chords are so predominant that the melody adapts to the harmony.

Human transcribers routinely use this kind of musical knowledge in music transcription. If the melody does not match the chords, or the chords do not match the melody, the transcription cannot be correct. However, in automatic music transcription, chord extraction and melody extraction have been studied separately for the most part.

Currently, the best automatic systems for chord transcription produce promising results, while melody transcription seems to be a more challenging problem. For this reason, we approach automatic melody transcription with the assumption that a chord transcription has already been done. We present an algorithm that divides the audio data into segments based on the chord boundaries. After this, the algorithm assigns each segment a melody pattern that matches both the audio data and the chord information.

### 1.1 Problem statement

Given an excerpt of music in audio form and a chord transcription in symbolic form, the task is to produce a melody transcription in symbolic form. We concentrate on typical Western music, and assume that the pitches of the notes are given in semitones.

We assume that the audio data  $A$  is divided into  $n_A$  frames of equal length using some preprocessing method. For each audio frame  $k$  ( $1 \leq k \leq n_A$ ), we are given values  $A[k].begin$  and  $A[k].end$  that are time values in seconds



© Antti Laaksonen.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Antti Laaksonen. “AUTOMATIC MELODY TRANSCRIPTION BASED ON CHORD TRANSCRIPTION”, 15th International Society for Music Information Retrieval Conference, 2014.

when the frame begins and ends. In addition, for each possible melody note  $q$  we are given a real number  $A[k][q]$  in the range  $[0, 1]$ . This value estimates the strength of the note  $q$  in frame  $k$ .

The chord transcription  $C$  consists of  $n_C$  chord changes. For each chord change  $k$  ( $1 \leq k \leq n_C$ ), we are given a value  $C[k].time$  that is the time when the chord changes. In addition, we are given a value  $C[k].chord$  that is the name of the chord. We restrict ourselves to triads (major, minor, diminished and augmented chords that consist of three notes), which results in a total of 48 possible chords.

Finally, the outcome of the algorithm should be a melody transcription  $M$  that consists of  $n_M$  melody notes. For each melody note  $k$  ( $1 \leq k \leq n_M$ ), the algorithm should produce values  $M[k].time$  and  $M[k].pitch$  that denote the onset time of the note and the pitch of the note.

Throughout the paper, we use MIDI note numbers to refer to the pitches. Thus, every pitch has a unique integer value and the interval of pitches  $a$  and  $b$  is  $|a-b|$  semitones. Pitch C4 (261.6 Hz) is associated with MIDI value 60.

## 1.2 Related work

Automatic melody transcription has been studied actively during the last decade. Detailed reviews of the proposed methods can be found in [16] and [20].

The usual first step in automatic melody transcription is to detect potential melody notes in the audio signal. The most popular method for this is to calculate a saliency function for the audio frames using the discrete Fourier transform or a similar technique (e.g. [2, 6, 15, 19]). Other proposed approaches for audio data processing include signal source separation [3] and audio frame classification [5].

After this, the final melody is selected according to some criterion. One technique for this is to construct a hidden Markov model (HMM) for note transitions and use the Viterbi algorithm for tracking the most probable melodic line [3, 5, 19]. An alternative to this is to use a set of local rules that describe typical properties of melody notes and outlier notes [7, 15, 20]. In addition, some systems [2, 6] feature agents that follow potential melody lines.

The idea of providing additional information to facilitate the melody transcription has also been considered in previous studies. A usual approach for this is to gather information from users. For example, users can determine which instruments are present [8], help in the source separation process [4] or create an initial version of the transcription [9]. The drawback of these systems is, of course, that the transcription is not fully automatic.

There are also some previous studies that combine key, chord and pitch estimation. In [18], the key and the chords of the piece are estimated simultaneously. Multiple pitch transcription systems that exploit key and chord information in pitch estimation include [1] and [17].

Most of the previous work on automatic melody transcription focuses on a slightly different problem than the topic of this paper, namely how to determine the melody frequency in the audio signal frame-by-frame. In [19] and [22], the output of the algorithm is similar to ours.

## 2. ALGORITHM

In this section we present our melody transcription algorithm that uses a chord transcription as a starting point for the transcription. The algorithm first divides the audio data into segments so that the boundaries of the segments correspond to the boundaries of the chords in the chord transcription. After this, the key of the piece is estimated using the chord transcription. Finally, the algorithm assigns each segment a pattern of notes that matches both the audio data and the chord transcription.

The input and the output of the algorithm are as described in Section 1.1. Thus, the algorithm is given  $n_A$  audio frames in array  $A$  and a chord transcription of  $n_C$  chord changes in array  $C$ , and the algorithm produces a melody transcription of  $n_M$  notes in array  $M$ .

### 2.1 Segmentation

The first step in the algorithm is to divide the audio data into segments. The segments will be processed separately in a later phase in the algorithm. The idea is to choose the boundaries of the segments so that the harmony in each segment is stable. This is accomplished using the chord boundaries in the chord transcription.

Let

$$l(k) = C[k+1].time - C[k].time$$

for each  $k$  where  $1 \leq k \leq n_C - 1$  and

$$f(k, x) = (l(k)/x) / \lfloor l(k)/x \rfloor$$

where  $x$  is a real value. Thus,  $l(k)$  is the length of the segment between chord changes  $k$  and  $k+1$ , and  $f(k, x)$  is an estimate how evenly  $x$  divides that segment into smaller segments. Finally, let

$$g(k, x) = \begin{cases} 1 & \text{if } f(k, x) \leq 1 + \epsilon \\ 0 & \text{otherwise} \end{cases}$$

and

$$s(x) = \sum_{i=1}^{n_C-1} g(i, x).$$

If  $f(k, x) \leq 1 + \epsilon$  for some small  $\epsilon$ , our interpretation is that  $x$  divides the segment evenly. Thus,  $g(k, x)$  indicates if the segment is divided evenly, and  $s(x)$  is the number of segments that are divided evenly if  $x$  was chosen. In this paper we use value  $\epsilon = 0.1$ .

The algorithm chooses a value of  $x$  such that  $x$  is in the range  $[min_x, max_x]$  and  $s(x)$  is as large as possible. The value  $x$  will be used as a unit length in the segmentation. The values  $min_x$  and  $max_x$  denote the minimum and maximum unit length; in this paper we use values  $min_x = 0.5$  and  $max_x = 3$ .

Finally, the algorithm produces a segment division  $S$  of  $n_S$  segments by dividing each chord segment  $k$  into  $l(k)/x$  smaller segments of equal length (the number of segments is rounded to the nearest integer). For each new segment  $u$  ( $1 \leq u \leq n_S$ ) the algorithm assigns the values  $S[u].begin$  and  $S[u].end$  as described above, and  $S[u].chord$  denotes the name of the chord in the segment.



## 2.2 Key estimation

After determining the segments, the algorithm estimates the key of the piece. The estimated key will be used later in the algorithm to favor melody notes that agree with the key. The key estimation is done using a simple method that is based on the chord information.

The algorithm goes through all segments in  $S$  and maintains a counter for each pitch class (a total of 12 counters). Initially, all the counters are zero. For each segment  $k$ , the algorithm increases the value of each counter that corresponds to  $S[k].chord$ . For example, if  $S[k].chord$  is G major, the algorithm increases counters that correspond to notes G, B and D.

Finally, the algorithm determines the key using the counters as follows. There are 24 possible keys, 12 major keys and 12 minor keys. For each key, the algorithm calculates the sum of counters that correspond to tonic, mediant and dominant in that key. The key whose sum is the highest is selected as the key of the piece.

This method for key estimation is more simple than methods used in previous studies involving chord and key estimation from music audio [11,18]. However, this method produces results that are considered accurate enough for this purpose.

## 2.3 Pattern matching

For each segment, the algorithm chooses a melody pattern that matches both the audio data within the segment and the chord and key information. Each segment is processed independently, and the final melody transcription consists of all melody patterns in the segments.

The algorithm divides each segment into  $d$  note slots where  $d$  is a preselected constant for all segments. Each note slot can contain either one melody note or rest in the melody pattern. The idea is to select  $d$  so that most rhythms can be represented using  $d$  note slots, but at the same time  $d$  is small enough to restrict the number of melody notes. In practice, small integers that are divisible by 2 and/or 3 should be good choices for  $d$ .

An optimal melody pattern is selected according to a scoring function. The scoring function should give high scores for melody patterns that are probable choices for the segment. Depending on the scoring function, there are three ways to construct the optimal melody pattern:

- Greedy construction: If the melody slots are independent of one another, we can select the optimal melody note for each slot and combine the results to get the optimal melody pattern.
- Dynamic programming: If the melody slots are not independent but the score of a slot only depends on the previous slot, we can use dynamic programming to construct the optimal pattern.
- Complete search: If the score of a melody pattern cannot be calculated before all melody notes are selected, we have to go through all possible note patterns and select the optimal one.

The methods involving greedy construction and dynamic programming are efficient in all practical situations. However, in complete search we need to check  $q^d$  melody patterns where  $q$  is the number of possible choices for a melody slot. In practice,  $q \approx 50$ , so complete search can be used only when  $d \leq 4$  to keep the algorithm efficient.

## 2.4 Scoring function

The scoring function that we use in this paper is primarily based on the key information and favors melody notes that match the estimated key of the excerpt. In addition, the notes that belong to the chord of the segment have an increased probability to be selected to the melody. Thus, if an E major chord occurs in the C major key, the note G# is a strong candidate for the melody note even if it does not belong to the C major scale.

Let  $s(k, a, q)$  denote the score for an event where  $a$ 'th note slot of segment  $k$  contains note  $q$ . We calculate the score using the formula

$$s(k, a, q) = z \cdot b(k, a, q) - x \cdot c(k, a, q)$$

where  $b(k, a, q)$  is a base score calculated from the audio data,  $c(k, a, q)$  is a penalty for selecting a note that does not appear in the audio data, and  $z$  and  $x$  are parameters that control the balance of the base score and the penalty.

Let  $I$  be a set that contains the indices of all audio frames that are inside the current note slot. Now we define

$$b(k, a, q) = \sum_{i \in I} A[i][q]$$

and

$$c(k, a, q) = \sum_{i \in I} e(i, q)$$

where

$$e(i, q) = \begin{cases} 1 & \text{if } A[i][q] = 0 \\ 0 & \text{otherwise.} \end{cases}$$

The parameter  $z$  favors melody notes that match the chord transcription, and it should depend on the key of the excerpt and the chord in segment  $k$ . We set  $z = 2$  if  $q$  belongs to the current chord,  $z = 1$  if  $q$  belongs to the key of the excerpt and otherwise  $z = 0$ . The parameter  $x$  controls the effect of adding a note to the melody that does not appear strongly in the audio data, and we study the effect of that constant in Section 3.

Finally, we select the note pattern greedily so that each note slot will be assigned the note that maximizes the score for that slot. If no note produces a score greater than 0, we leave that slot empty.

We also experimented with dynamic programming scoring functions that favor small intervals between consecutive notes, but the results remained almost unchanged. Consequently, we chose the more simple greedy construction approach.

## 3. EVALUATION

In this section we present results concerning the accuracy of the transcriptions produced by our algorithm, using real-world inputs. We evaluated our algorithm using a dataset

of Western popular music. We used both hand-made and automatic chord transcriptions as additional input for the algorithm.

Audio Melody Extraction task is an established part of the MIREX evaluation [13]. However, in the MIREX evaluation each audio frame is assigned a melody note frequency, and those results cannot be compared with our melody transcriptions that consist of note onset times and pitches in semitones.

### 3.1 Dataset

The evaluation dataset consists of 1,5 hours of audio excerpts from Western popular music. The length of each excerpt in the dataset is between 20 and 60 seconds. For each excerpt, we manually created time-aligned melody and chord transcriptions. We chose the excerpts so that the content of each excerpt is unique i.e. repetitions of verses and choruses are not included in the dataset.

The dataset can be found on our web site<sup>1</sup>, and is available for free for use in research. For each excerpt, the dataset includes an audio file in WAV format, and chord and melody transcriptions in text format. Each chord transcription is a list of chord change times and chord symbols, and each melody transcription is a list of note onset times and pitches. Thus, the transcriptions in the dataset correspond to the definitions in Section 1.1.

### 3.2 Evaluation method

To evaluate a melody transcription, we calculate two values: the precision and the recall. Precision is the ratio of the number of correct notes in the transcription and the total number of notes in the transcription. Recall is the ratio of the number of correct notes in the transcription and the total number of notes in the ground truth.

Let  $X$  be a melody transcription of  $n_X$  notes created by the algorithm, and let  $G$  be the corresponding melody transcription of  $n_G$  notes in the ground truth. Both transcriptions consists of a list of melody note onset times and pitches as described in Section 1.1.

To evaluate the precision and the recall of  $X$ , we first align the transcriptions. Let  $n_D$  be the maximum integer value such that we can create lists  $D_X$  and  $D_G$  as follows. List  $D_X$  consists of  $n_D$  note indices in  $X$ , and list  $D_G$  consists of  $n_D$  note indices in  $G$ . In addition, for each  $k$  ( $1 \leq k \leq n_D$ )  $X[D_X[k]].pitch = G[D_G[k]].pitch$  and  $|X[D_X[k]].time - G[D_G[k]].time| \leq \alpha$  where  $\alpha$  is a small constant. In other words, we require that lists  $D_X$  and  $D_G$  align a set of notes where all pitches match each other and the onset times of the notes do not differ more than  $\alpha$  from each other.

Finally, let

$$\text{precision}(X, G) = n_D / n_X$$

and

$$\text{recall}(X, G) = n_D / n_G.$$

In practice, we calculate the value  $n_D$  efficiently using dynamic programming. The technique is similar to calculating the Levenshtein distance between two strings [14].

This evaluation method corresponds with that used in [19] and [22], however, the previous papers do not specify how the notes in the two transcriptions are aligned.

### 3.3 Experiments

We implemented our algorithm as described in Section 2. For calculating array  $A$  we used an algorithm by Salamon and Gómez that estimates potential melody contours in the audio signal. We used the Vamp plugin implementation of the algorithm ("all pitch contours"). Note that this algorithm already restricts the set of possible melodies considerably. We converted each pitch frequency to a MIDI note number assuming that the frequency of A4 is 440 Hz.

We used four chord transcriptions in the evaluation:

- A random chord transcription where the time between two chord changes is a random real number in the range  $[0.5, 2]$  and each chord is randomly selected from the set of 48 possible triads.
- A simple automatic chord transcription created by our own algorithm. We used the standard technique of constructing a hidden Markov model and tracking the optimal path using the Viterbi algorithm [21].
- An advanced automatic chord transcription created using the Chordino tool [12].
- The chord transcription in the ground truth.

Random chord transcriptions were used in an effort to understand the actual role of the chord information and how the algorithm works if the chord information does not make sense at all.

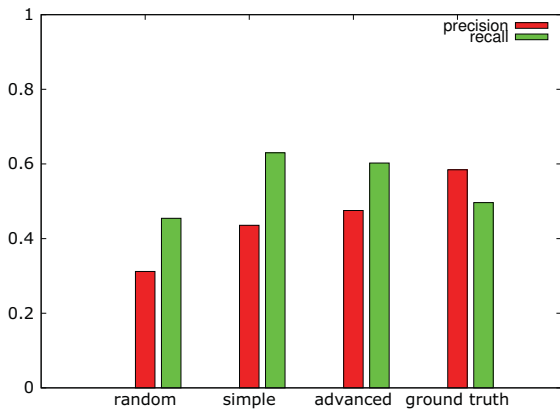
Finally, there are three parameters that we varied during the evaluation:

- $d$ : the number of note slots in a segment as described in Section 2.3 (default value: 6),
- $x$ : the cost for assigning a note to a frame without a note as described in Section 2.4 (default value: 1.00),
- $\alpha$ : the maximum onset time difference in the evaluation as described in Section 3.2 (default value: 0.25).

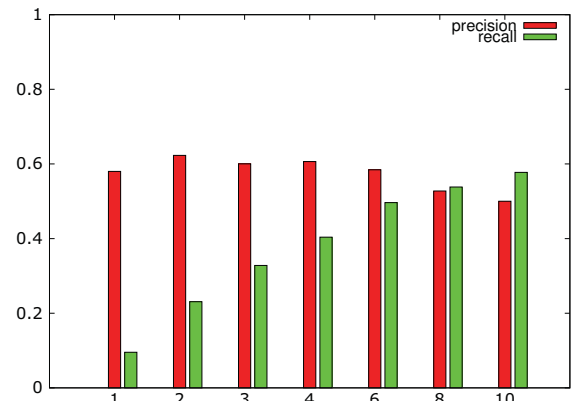
The default values were chosen so that they produce good results on the evaluation dataset.

In each experiment in the evaluation, we varied one parameter and kept the remaining parameters unchanged. We used the ground truth chord transcription as the default chord transcription. We created melody transcriptions for all excerpts in the dataset and calculated average precision and recall values.

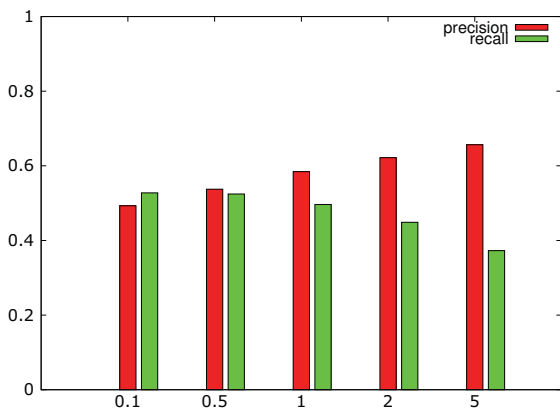
<sup>1</sup> <http://cs.helsinki.fi/u/ahslaaks/fpds/>



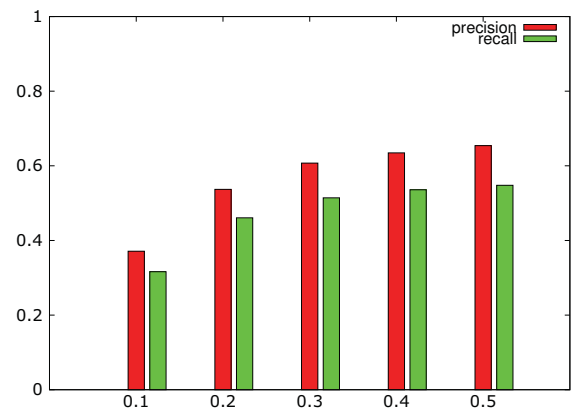
(a) The results using random chord transcription, simple automatic transcription, advanced automatic transcription, and ground truth transcription.



(b) The results varying the parameter  $d$ : the number of available note slots in a segment.



(c) The results varying the parameter  $x$ : the cost for assigning a note to a frame without a note.



(d) The results varying the parameter  $\alpha$ : the maximum onset time difference in seconds in the evaluation.

**Figure 2:** The results of the experiment.

### 3.4 Results

In the first experiment (Figure 2a) we studied how the quality of the chord transcription affects the results. As expected, the better the chord transcription, the better the precision of the melody transcription. However, recall was highest when using automatic chord transcriptions. One possible reason for this is that there were more chord changes in automatic transcriptions than in the ground truth transcription. Therefore more melody notes were selected using the automatic chord transcriptions.

In the second experiment (Figure 2b) we varied the parameter  $d$ : the number of note slots in a segment. Our findings suggest that 6 note slots is a good trade-off between the precision and the recall. This can be explained by the fact that 6 is divisible by both 2 and 3, and thus segments of 6 note slots are suitable for both 3/4 time and 4/4 time music. Interestingly, when  $d \leq 6$  the precision of the transcription remained nearly unchanged.

In the third experiment (Figure 2c) we varied the parameter  $x$ : the cost for assigning a note to a frame without a note. This was an important parameter, and the results

were as expected. Increasing the parameter  $x$  improves the precision because melody notes are only selected if they appear strongly in the audio data. At the same time, this decreases the recall because fewer uncertain notes are included in the melody transcription.

Finally, in the fourth experiment (Figure 2d) we varied the parameter  $\alpha$ : the maximum note onset time difference in the evaluation. Of course, the greater the parameter  $\alpha$ , the better the results. Interestingly, after reaching a value of approximately 0.25, increasing  $\alpha$  did not affect the results considerably. The probable reason for this is that if the melody note pitches in the transcription are not correct, the situation cannot be rescued by allowing more error for the onset times.

Previous studies also present some results about the precision and the accuracy of the algorithms. However, findings from these studies cannot be directly compared with the new results because the evaluation dataset is different in each study. In [19] precision 0.49 and recall 0.61 was reported using a database of 84 popular songs. In [22] the melody transcription was evaluated using a small set of 11 songs with precision 0.68 and recall 0.63.

#### 4. CONCLUSIONS

In this paper we presented an automatic melody transcription algorithm that uses a chord transcription for selecting melodies that match the harmony of the music. We evaluated the algorithm using a collection of popular music excerpts, and the results of the evaluation suggest that the chord information can be successfully used in melody transcription of real-world inputs.

Our new evaluation dataset consists of 1,5 hours of audio excerpts of popular music together with melody and chord annotations. The dataset can be used at no cost for research purposes, for example as evaluation material for other chord transcription and melody transcription systems.

Our future work aims to use the harmony information provided by the chord transcription more extensively in melody transcription. Currently our algorithm uses only information about chord notes to constrain the pitches of melody notes, but using more advanced musical knowledge should yield better results.

#### 5. ACKNOWLEDGEMENTS

This work has been supported by the Helsinki Doctoral Programme in Computer Science and the Academy of Finland (grant number 118653).

#### 6. REFERENCES

- [1] E. Benetos, A. Jansson and T. Weyde: "Improving automatic music transcription through key detection," *AES 53rd International Conference on Semantic Audio*, 2014.
- [2] K. Dressler: "An auditory streaming approach for melody extraction from polyphonic music," *12th International Society for Music Information Retrieval Conference*, 19–24, 2011.
- [3] J.-L. Durrieu et al: "Source/filter model for unsupervised main melody extraction from polyphonic audio signals," *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3), 564–575, 2010.
- [4] J.-L. Durrieu and J.-P. Thiran: "Musical audio source separation based on user-selected F0 track," *10th International Conference on Latent Variable Analysis and Signal Separation*, 2012.
- [5] D. Ellis and G. Poliner: "Classification-based melody transcription," *Machine Learning*, 65(2–3), 439–456, 2006.
- [6] M. Goto: "A real-time music scene description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Communication*, 43(4), 311–329, 2004.
- [7] S. Joo, S. Park, S. Jo and C. Yoo: "Melody extraction based on harmonic coded structure," *12th International Society for Music Information Retrieval Conference*, 227–232, 2011.
- [8] H. Kirchhoff, S. Dixon and A. Klapuri: "Shift-variant non-negative matrix deconvolution for music transcription," *37th International Conference on Acoustics, Speech and Signal Processing*, 2012.
- [9] A. Laaksonen: "Semi-automatic melody extraction using note onset time and pitch information from users," *SMC Sound and Music Computing Conference*, 689–694, 2013.
- [10] M. Lagrange et al: "Normalized cuts for predominant melodic source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2), 278–290, 2008.
- [11] K. Lee and M. Slaney: "A unified system for chord transcription and key extraction using hidden Markov models," *8th International Conference on Music Information Retrieval*, 245–250, 2007.
- [12] M. Mauch and S. Dixon: "Approximate note transcription for the improved identification of difficult chords," *11th International Society for Music Information Retrieval Conference*, 135–140, 2010.
- [13] MIREX Wiki: Audio Melody Extraction task, <http://www.music-ir.org/mirex/wiki/>
- [14] G. Navarro: "A guided tour to approximate string matching," *ACM Computing Surveys*, 33(1): 31–88, 2001.
- [15] R. Paiva, T. Mendes and A. Cardoso: "Melody detection in polyphonic musical signals: exploiting perceptual rules, note salience, and melodic smoothness," *Computer Music Journal*, 30(4), 80–98, 2006.
- [16] G. Poliner et al: "Melody transcription from music audio: approaches and evaluation," *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4), 1247–1256, 2007.
- [17] S. Raczynski, E. Vincent, F. Bimbot and S. Sagayama: "Multiple pitch transcription using DBN-based musicological models," *11th International Society for Music Information Retrieval Conference*, 363–368, 2010.
- [18] T. Rocher et al: "Concurrent estimation of chords and keys from audio," *11th International Society for Music Information Retrieval Conference*, 141–146, 2010.
- [19] M. Rynänen and A. Klapuri: "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music Journal*, 32(3), 72–86, 2008.
- [20] J. Salamon and E. Gómez: "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6), 1759–1770, 2012.
- [21] A. Sheh and D. Ellis: "Chord segmentation and recognition using EM-trained hidden Markov models," *4th International Conference on Music Information Retrieval*, 183–189, 2003.
- [22] J. Weil et al: "Automatic generation of lead sheets from polyphonic music signals," *10th International Society for Music Information Retrieval Conference*, 603–608, 2009.

# AUDIO-TO-SCORE ALIGNMENT AT NOTE LEVEL FOR ORCHESTRAL RECORDINGS

Marius Miron, Julio José Carabias-Orti, Jordi Janer

Music Technology Group, Universitat Pompeu Fabra

marius.miron, julio.carabias, jordi.janer@upf.edu

## ABSTRACT

In this paper we propose an offline method for refining audio-to-score alignment at the note level in the context of orchestral recordings. State-of-the-art score alignment systems estimate note onsets with a low time resolution, and without detecting note offsets. For applications such as score-informed source separation we need a precise alignment at note level. Thus, we propose a novel method that refines alignment by determining the note onsets and offsets in complex orchestral mixtures by combining audio and image processing techniques. First, we introduce a note-wise pitch salience function that weighs the harmonic contribution according to the notes present in the score. Second, we perform image binarization and blob detection based on connectivity rules. Then, we pick the best combination of blobs, using dynamic programming. We finally obtain onset and offset times from the boundaries of the most salient blob. We evaluate our method on a dataset of Bach chorales, showing that the proposed approach can accurately estimate note onsets and offsets.

## 1. INTRODUCTION

Audio-to-score alignment concerns synchronizing the notes in a musical score with the corresponding audio rendition. An additional step, alignment at the note level, aims at adjusting the note onsets, in order to further minimize the error between the score and audio. In the context of orchestral music, this task is challenging; first, because of the complex polyphonies, and, second, because of the timing expressivity of classical music.

As possible applications of note alignment, deriving the exact locations of the note onsets and offsets could improve tasks as score-informed source separation [6], [2], [7].

State-of-the-art score alignment methods use Non-negative matrix factorization (NMF) [14], [11], template adaptation through expectation maximization [9], dynamic time warping (DTW) [3], and Hidden Markov Models (HMM) [4, 6]. The method described in [11, p. 103] is the only one addressing explicitly the topic of fine note

alignment as a post-processing step. A factorization is performed to obtain the onsets of the anchor notes. The basis vectors are trained with piano pitches models, and the onsets are obtained from the activations matrix. Furthermore, an additional step is performed in order to look for onsets between anchors.

However, the methods listed above have certain limitations. First, accurately detecting the offset of the note is a challenging problem and none of these methods claim to solve it. Second, the scope of the NMF-based systems is solely piano recordings. Third, except [11], the algorithms consider a large window to evaluate detected onsets. Note that the MIREX Real-time Audio-to-Score Alignment task considers a 2000 ms window size.

With respect to image processing techniques deployed in music information research, a system to link audio and scores for makam music is presented in [13]. In this case, Hough transform is used for picking the line corresponding to the most likely path from a binarized distance matrix. Additionally, the same transform is used in [1] to find repeating patterns for audio thumbnailing.

In this paper we propose a novel method for audio-to-score alignment at the note level, which combines audio and image processing techniques. In comparison to classical audio-to-score alignment methods, we aim to detect the offset of the note, along with its onset. Additionally, we do not assume a constant delay between score and audio, thus we do not use any information regarding the beats, tempo or note duration, in order to adjust the onsets. Therefore, our method can align notes when dealing with variable delays, as the ones resulting from automatic score alignment or the ones yielded by manually aligning the score at the beat level.

The proposed method is based on two stages. First, the audio processing stage involves filtering the spectral peaks in time and frequency for every note. Consequently, the filtering occurs in the time interval restricted for each note and in the frequency bands of the harmonic partials corresponding to its fundamental frequency. Furthermore, we decrease the magnitudes of the peaks which are overlapping in time and frequency with the peaks from other notes. Using the filtered spectral peaks, we compute the pitch salience for each note using the harmonic summation algorithm described in [10]. Second, we detect the boundaries of the note using an image processing algorithm. The pitch salience matrix associated to each note is binarized. Then, blobs, namely boundaries and shapes, are detected using



© Marius Miron, Julio José Carabias-Orti, Jordi Janer.

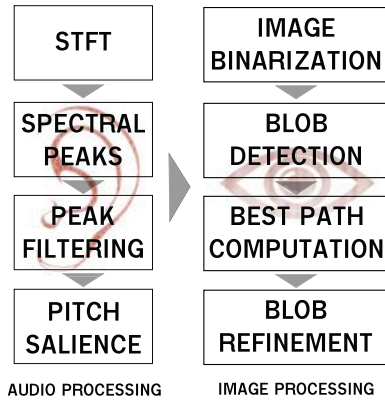
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Marius Miron, Julio José Carabias-Orti, Jordi Janer. "Audio-to-score alignment at note level for orchestral recordings", 15th International Society for Music Information Retrieval Conference, 2014.

the connectivity rules described in [12, p. 248]. From all the blobs candidates associated to every note, we pick the best combination of consecutive blobs using dynamic programming. The image processing part has the advantage that the blob boundaries will define the note onsets along with the corresponding offsets.

The remainder of this paper is structured as follows. In the first section we describe the note-wise pitch salience computation, followed by the blob selection using image processing methods. Then, we evaluate our algorithm on a dataset of Bach chorales [6] and we discuss the results.

## 2. METHOD

The proposed method aims to detect the onsets and offsets of the notes from a monaural audio recording, where the score is assumed to be automatically or manually aligned a priori, assuming an error up to 200 ms.



**Figure 1.** The two main sections of our method: audio and image processing, and the corresponding steps.

Figure 2 shows the block diagram of the proposed method. As can be seen, the method is subdivided in two stages. First, in the audio processing stage, a filtered pitch salience matrix is obtained for each of the notes in the score, and for every instrument. Second, in the image processing stage, the pitch salience matrix is regarded as a greyscale image, and blobs are detected in the binarized image. Moreover, we construct a graph with all the blobs and we pick the best combination of blobs by using Dijkstra’s algorithm to find the best path in the graph. Finally, we refine the time boundaries for the blobs that overlap, using an adaptive threshold binarization.

### 2.1 Note-wise pitch salience computation

For each input signal, we first compute the Short time Fourier transform (STFT) and we extract the spectral peaks. Then, we analyze each single note in the score and we select only the spectral peaks in the frames around its approximate time location and the frequency bands associated to its harmonic partials (i.e. multiples of the fundamental frequency). Finally, we compute the pitch salience, using the harmonic summation algorithm described in [10].

To select the time intervals at which we are going to look for the note onsets and offsets, we analyze the pre-aligned score that we want to refine. We start from the assumption that the note onsets are played with an error lower than 200 ms from the actual onset in the score. In other words, we set the search interval to  $\pm 200$  ms from the note onset at the score. Additionally, in the case of the offset, we extend the possible duration of a note in the score by 200 ms or until another note in the score appears. In the rest of the paper, this search interval will be referred to as  $T_{on}(n)$  and  $T_{off}(n)$ .

Then, we analyze the spectral peaks within the time interval defined for each note, and we filter them according to the harmonic frequencies of the MIDI note  $\hat{F}_n(i)$ , where  $\hat{F}_n(0)$  is the fundamental frequency of note  $n$ . Namely, we take the first 16 of the harmonic partials of this frequency,  $\hat{F}_n(i)$  with  $i \in [0, \dots, 15]$ . Taking into account vibratos, we set a 1.4 semitone interval around each of the harmonic partials. Consequently, we select a set of candidate peaks  $P_n(k)$  and the associated amplitudes  $A_n(k)$  for note  $n$  at frame  $k$  such that  $P_n(k) \in [\hat{F}_n(i) - \hat{L}_n(i), \dots, \hat{F}_n(i) + \hat{L}_n(i)]$ , where  $\hat{L}_n(i)$  is a frequency band equivalent to 0.7 of a semitone.

As a drawback, some of the selected peaks could overlap in time and frequency. To overcome this problem, we distribute the amplitude  $A_n(k)$  of the overlapped peaks  $P_n(k)$  using a factor  $g_i(P_n(k), P_m(k))$ , where  $n$  and  $m$  are the overlapped notes,  $g_i$  is a gaussian centered at the corresponding frequency  $\hat{F}_n(i)$  of the note  $n$  and the harmonic partial  $i$ . The standard deviation equals to  $\frac{\hat{L}_n(i)}{2}$ , thus:

$$g_i(x) = w * 0.8^i * e^{-\frac{(x - \hat{F}_n(i))^2}{\frac{\hat{L}_n(i)^2}{2}}} \quad (1)$$

Note that the magnitude of the gaussian decreases with the order of the harmonic,  $i$ , and is proportional to  $w$ , the weight of the rest of the instruments in current audio file, or the coefficient extracted from a pre-existing mixing matrix. For example, if we align using solely a monaural signal in which all four instruments have the same weight, 0.25 for all four instruments, the coefficient will be  $w = 0.75$ .

The factor  $g_i$  penalizes frequencies which are in the allowed bands but are further away from the central frequencies. In this way, we eliminate transitions to other notes or energy which can add up noise later on in the blob detection stage.

Finally, for each note  $n$  and its associated  $P_n(k)$  and  $A_n(k)$  where  $k \in [T_{on}(n), \dots, T_{off}(n)]$ , we use the pitch salience function described in [10]. The algorithm calculates a salience measure for each pitch candidate, starting at  $\hat{F}_n(0) - \hat{L}_n(0)$ , based on the presence of its harmonics and sub-harmonics partials, and the corresponding magnitudes. Finally, the salience function for each time window is quantized into cent bins, thus the resulting matrix  $S_n$  has the dimensions  $(T_{off}(n) - T_{on}(n), Q)$ , where  $Q$  is the number of frequency bins for the six octaves. In our case, we experimentally choose  $Q = 600$  bins.

## 2.2 Blob selection using image processing

The goals of the image processing stage are to obtain the location of the note onset and offset by binarizing the note-wise pitch salience, and to detect shapes and contours in the binarized image.

Accounting that the image binarization is not a robust process [12], different results are expected as a function of the amount of time overlap between notes, the salience of the pitch and its fundamental frequency. Therefore, as the shape and contour detection heavily relies on this step, we need a robust binarization, which would finally give us the best information for detecting the boundaries of the note.

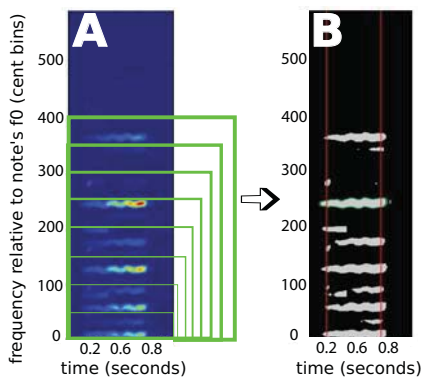
Previous approaches to improve binarization rely on background subtraction or local binarization [12]. Therefore, we propose a binarization method similar to the local binarization, but adapted to our context: the pitch salience matrix. On the assumption that the bins closer to the fundamental frequency,  $\hat{F}_n(0)$ , are more salient than the ones at higher frequencies, we split the binarization areas in sub-areas related to the harmonic partials  $\hat{F}_n(i)$ . Thus, the salience matrix  $\mathbf{S}_n$  is binarized gradually and locally, obtaining a binary matrix  $\mathbf{B}_n$ . Moreover, we consider  $l$  as the binarization step, moving gradually from 50 to 600 in steps of 50 bins.

Furthermore, we compute  $\mathbf{B}_n$  in  $l$  steps, each time only for the columns in the interval  $[l - 50 \dots l]$ .

$$\mathbf{B}_n(i, j) = \begin{cases} 0, & \mathbf{S}_n(i, j) < \text{mean}(\mathbf{S}_n^l) \\ 1, & \mathbf{S}_n(i, j) \geq \text{mean}(\mathbf{S}_n^l) \end{cases} \quad (2)$$

where  $i \in [T_{on}(n), \dots, T_{off}(n)]$ ,  $j \in [l - 50 \dots l]$ , and  $\mathbf{S}_n^l$  is a submatrix of  $\mathbf{S}_n$ , obtained by extracting the columns of  $\mathbf{S}_n$  in the interval  $[0..l]$ .

As an example, a pitch salience matrix  $\mathbf{S}_n$  for a bassoon note is plotted in the Figure 2A. The green rectangles mark the submatrices  $\mathbf{S}_n^l$  for various values of  $l$ . The resulting binarized image is depicted in Figure 2B.



**Figure 2.** Binarizing the spectral salience matrix (figure A) and detecting the blobs in the resulting image (figure B). Binarization is done gradually and locally, relative to the green squares areas in figure A. The ground truth onset and offset of the note are marked by vertical red lines.

The next step is detecting boundaries and shapes on the binarized image. We use the connectivity rules described

in [12, p. 248] in order to detect regions and the boundaries of these regions, namely the blobs. Thus, we want to label each pixel of the matrix  $\mathbf{B}_n$  with a number from 0 to  $r$ , where  $r$  is the total number of detected blobs.

Having a pixel  $(i, j)$  with  $i \in [T_{on}(n), \dots, T_{off}(n)]$  and  $j \in [0, \dots, Q]$ , where  $Q$  is the number of frequency bins, we need to consider all the neighboring pixels and we have to take into account their connectivity with the current pixel. The 4-way connectivity rules account for the immediate neighbors, as compared to 8-way connectivity which account for all the surrounding pixels. Because we are not interested in modeling transitions between notes, we discard diagonal shapes by using the 4-way connectivity rules. Hence, the connectivity matrix, which determines the neighborhood of the pixel  $(i, j)$ , can be written as:

$$\mathbf{M} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

For the matrix  $\mathbf{M}$ , the central pixel with the coordinates (2,2) represents the origin pixel  $(i, j)$ , and all the other non-zero pixels are the considered positions for the neighbors.

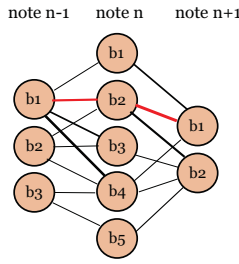
The algorithm, described in [12, p. 251], takes one pixel at a time and visits its non-zero neighbors. Then, we move sequentially from one pixel to its neighbors, setting boundaries for the pixels having neighbors equal to zero. Finally, the shape is enclosed when the algorithm reaches the pixel of origin.

Furthermore, once we have detected a set of blobs  $b_n$  for each note  $n$ , we need to compute the best combination of the blobs for all notes. Because search intervals for consecutive notes can overlap in time, choosing the best combination of blobs is not as trivial as picking the best blob in terms of area or salience, and the decisions that we take for a current note, should take into account the decisions we take for the previous and the next note. This kind of problem, which chains up a set of decisions can be solved with dynamic programming.

Consequently, we consider the blobs to be the vertices of an oriented graph, in which the edges are assigned a cost depending on the area of the two blobs and the overlapping between them, as seen in Figure 3. Basically, blobs with bigger area and little overlapping will have a lower cost, which makes them ideal candidates when we find the best path in the graph. Additionally, we can have an edge only between blobs of consecutive notes, and we can remove the edges between blobs which overlap more than 50% in time.

Therefore, we compute the area of each blob of the note  $n$  by summing up the values in the binarized matrix  $\mathbf{B}_n$ , enclosed by the corresponding blob contours. Additionally, we exclude the blobs which have the duration less than 100 ms, and the ones starting after the allowed interval for the attack time.

The normalized area of blob  $i$  for the note  $n$  is  $H(b_n^i)$  and is a value inversely proportional with the actual area, because we want the larger blobs to have a lower cost,



**Figure 3.** A sample of the graph between three consecutive notes.  $b_{[1..5]}$  are the blobs detected for each note. Thicker lines represent lower costs. The red line represents the best path in the graph.

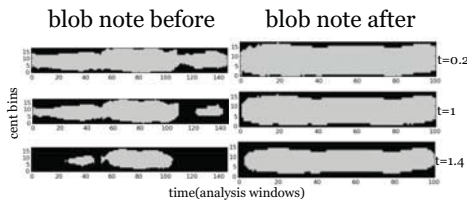
when picking the best path. In the same manner, we must increase the cost as the overlapping between the blobs increases. Thus, for two adjacent notes  $n$  and  $n + 1$ ,  $O(b_n^i, b_{n+1}^j)$  has cost 1 if there is no overlapping, and an increased value summing up the ratio of the area of the two overlapping blobs. For instance, if 20% of the area of the first blobs overlaps with 70% of the area of the second blob,  $O = 1 + 0.2 + 0.7 = 1.9$ .

Thus, the cost for the edges has the expression

$$\text{cost}(b_n^i, b_{n+1}^j) = O(b_n^i, b_{n+1}^j) * (H(b_n^i) + H(b_{n+1}^j))$$

In order to find the shortest path between the vertices of the first note in the score and the last one, we use Dijkstra's algorithm described in [5]. The algorithm finds the shortest path for a graph with non-negative edges by assigning a tentative distance to each of the vertices and progressively advancing by visiting the neighboring nodes.

Additionally, after the best path is computed, we can face the situation where two consecutive blobs overlap in time due to the inaccuracy in binarization and the fact that the minimum cost path does not guarantee no overlapping. Because the melody for a particular instrument is considered to be monophonic, we do not allow overlapping between two consecutive notes. Thus, we ought to find a splitting point between the starting point of the blob associated with the next note and the ending point of the blob associated with the current note.



**Figure 4.** Blob refinement using adaptive threshold binarization of two consecutive overlapping blobs in the best path. The minimum overlapping is achieved for threshold  $t = 1.4$

Having two consecutive blobs from the best path,  $b_n$  and  $b_{n+1}$ , we take the image patches surrounding their boundaries and we adaptively increase the threshold of bi-

narization until the minimum overlapping is achieved. Consequently, we consider the submatrices  $\hat{\mathbf{S}}_n$  and  $\hat{\mathbf{S}}_{n+1}$  of the corresponding pitch salience matrices  $\mathbf{S}_n$  and  $\mathbf{S}_{n+1}$ , and for a variable threshold  $t = [0.2..2]$ , we compute the binary matrices  $\hat{\mathbf{B}}_n^t$  and  $\hat{\mathbf{B}}_{n+1}^t$ .

$$\hat{\mathbf{B}}_n^t(i, j) = \begin{cases} 0, & \hat{\mathbf{S}}_n(i, j) < t * \text{mean}(\hat{\mathbf{S}}_n) \\ 1, & \hat{\mathbf{S}}_n(i, j) \geq t * \text{mean}(\hat{\mathbf{S}}_n) \end{cases} \quad (3)$$

As seen in Figure 4, the higher the threshold  $t$ , the less pixels are assigned to value 1 in the binary matrices, thus we increase the threshold gradually until no overlapping is achieved.

Finally, the note onset and offset are extracted from the leftmost and the rightmost pixels of the refined blobs in the best path.

### 3. EVALUATION

#### 3.1 Experimental setup

The dataset used to evaluate our proposal consists of 10 human played J.S. Bach four-part chorales, and is commonly known as Bach10. The audio files are sampled from real music performances recorded at 44.1 kHz that are 30 seconds in length per file. Each piece is performed by a quartet of instruments: violin, clarinet, tenor saxophone and bassoon. Each musician's part was recorded in isolation. Individual lines were then mixed to create 10 performances with four-part polyphony. More information about this dataset can be found in [6].

We observe that the dataset has a few particularities. First, every recording presents fermatas, where the final duration of the note is left at the discretion of the performer or the conductor, making it more difficult to detect the onset and offsets of the notes. Second, the chorales have a peculiar homophonic texture. Third, the annotated note onsets and offsets in the ground truth can have more or less notes than the actual score. We discovered that this mismatch comes from repeating notes, which in the original score are represented by a single larger note. This step also makes the detection of the note offsets more difficult.

In order to perform alignment at the note level, we generate a misaligned score by introducing onset and offset time deviations for all the notes and all the instruments in the ground-truth score. The deviations are randomly and uniformly distributed in the intervals  $[-200, \dots, -100]$  and  $[100, \dots, 200]$  ms. Moreover, we aim at refining the alignment of the algorithm proposed by [3]. Thus, we correct the onset times and we detect the offsets around the beginning of the next note. For both of these tasks we consider the interval  $[-200, \dots, 200]$  ms.

Furthermore, the STFT is computed using a Blackman-Harris 92dB window with a size of 128 ms and, a hop size of 6 ms. Additionally, we zero-pad the window by three times its length. Moreover, frequencies and magnitudes of the spectral peaks are extracted with the algorithm described in [8], which uses parabolic interpolation to accurately detect positive slopes in the spectrum computed at the previous step.



### 3.2 Results

We aim at correctly aligning the onsets and offsets of the misaligned score described in Section 3.1 and we add up 200 ms before and after the note boundaries in order to search for the exact starting and ending point of the note. Thus, our algorithm can have up to 400 ms in error for the onsets, and a larger error for the offset, because we are not constraining the duration of the note to any interval.

For each piece, aligned rate (AR) or precision is defined as the proportion of correctly aligned notes in the score and ranges from 0 to 1. A note is said to be correctly aligned if its onset does not deviate more than a threshold from the reference alignment. To test the reliability of our method, we tried different threshold values ranging from 15 to 140 ms. Other measures as the average offset (i.e. average absolute-valued time offset between a reported note onset by the score follower and its real onset in the reference file) and the std offset (i.e. standard deviation of sign-valued time offset) are also considered.

As illustrated in figure 5, the proposed system is able to accurately align more than the 30% of the onsets with a detection threshold lower than 15 ms. Furthermore, more than 80% of the onsets are accurately detected with a threshold of 60 ms. Because the search time interval for the note allows for error larger than 200 ms, the AR for the onset does not reach 100% in  $t = 200ms$ , as less than 2% of the onsets have larger errors.

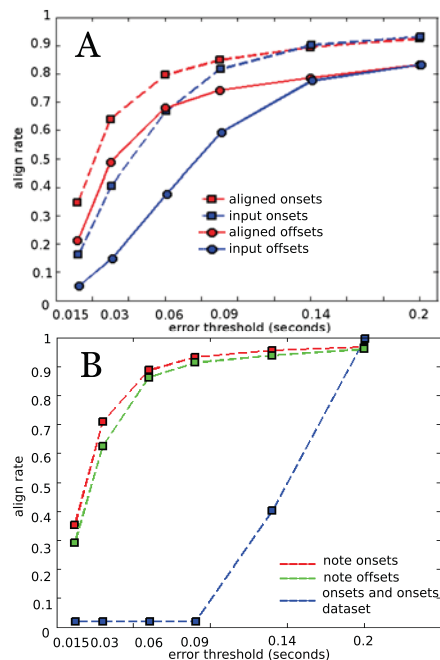
Furthermore observe that we less accurate in detecting the offsets, particularly when we do not know the approximate note offset and we estimate it around the onset of the next note, as when we take as input the alignment of the algorithm proposed by [3]. The drop in performance of the offset detection can also be explained by the fact that the energy of a note can decay below a threshold, thus excluding it when binarization is performed.

Figure 6 shows boxplots of the average offset and the std error for each instrument, and for the note onset and offset, for the misaligned dataset. The lower and upper lines of each box show 25th and 75th percentiles of the sample. The line in the middle of each box is the average offset. The lines extending above and below each box show the extent of the rest of the samples, excluding outliers. Outliers are defined as points over 1.5 times the interquartile range from the sample median and are shown as crosses.

We observe that performance is lower for violin compared to the other instrument. This can be explained by the fact that for this dataset the violin has noisier or soft attacks, which do not yield a high enough value in terms of pitch salience, and is lost when binarizing the image.

Moreover, the fact that we are able to detect most of the onsets in the interval 0.06 seconds, which is an acceptable interval for the attack of the instruments aligned, point us on some limitation on using the pitch salience function, which is not able to be accurate enough with noisier attacks, as it happened for the violin.

Furthermore, we want more insight on how the errors are distributed across the time range. Thus, we plot the 2-d histogram of the onset errors, as seen in Figure 7. We



**Figure 5.** The proposed system improves the align rate of (A) the system proposed by [3] and of (B) the misaligned dataset, for onset errors, as well as offset errors

observe that even though the original dataset had large errors, our method was able to detect the note onsets within a small time frame, as most of the errors are in the bin centered at zero.

Moreover, our method is better at fixing the delays in the note onsets. In comparison, we can commit more errors if the onset of the note is thought to be before the actual onset, because the window in which we have to look for it overlaps more with the previous note, hence we have more interference.

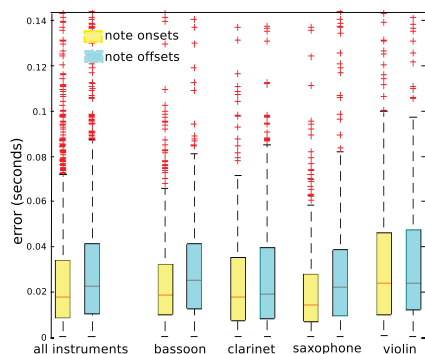
Additionally, for every note and every instrument, we compute the percentage of correctly detected frames with respect to ground truth. Our algorithm is able to correctly detect 89% of the frames of the ground truth notes. In comparison, the notes in the misaligned dataset have a degree of 66% correctly detected frames.

Finally, we compute the percentage of frames which are erroneously detected as part of the notes. We observe that solely 0.07% of frames from the notes we refine are outside the boundaries of the ground truth notes, compared to the misaligned dataset, for which 34% of the frames are displaced outside the time boundaries of the notes.

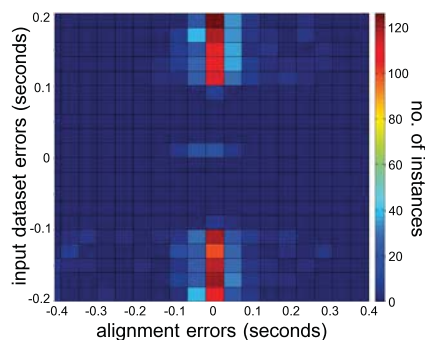
Therefore, our algorithm is more likely to shorten the notes, rather than making erroneous decisions regarding their time frame. This is due to the way we are picking the best sequence of blobs, which penalizes the overlapping, thus picking blobs which have a smaller area but less overlapping with the blobs from neighboring notes.

## 4. CONCLUSIONS

We proposed a method to refine the alignment of onsets and offsets in orchestral recordings, using audio and im-



**Figure 6.** The average offset and the std offset in terms of 25th and 75th percentile of the proposed system for bassoon, clarinet saxophone, and violin, for note onsets, as well as note offsets



**Figure 7.** The histogram of error distribution in the onset alignment

age processing techniques. We compute a note-wise pitch salience function and we binarize it. Moreover, we detect blobs in the binarized image, and we pick the best blob candidate for each note by finding the best path in the associated graph. Furthermore, as offset detection is regarded as a more difficult problem, the proposed method addresses this issue by detecting image blobs to simultaneously label note onsets and offsets.

The evaluation shows that our method is able to refine the alignment in a misaligned dataset, having detected more than 80% of the onsets with an error of 60 ms. Moreover, we analyzed the performance across all four instruments, and we discovered that the accuracy drops for a violin, as being higher for the other instruments. Thus, as a future step, we need to analyze what limitation has the algorithm regarding certain instrument classes. Additionally, the proposed method should be tested with another dataset, with more complex polyphonies and tempo variations.

Furthermore, our method can be improved by using timbre models when filtering the spectral peaks and decreasing their magnitude. Additionally, choosing the best sequence of blobs can be improved by using a better cost function for the Dijkstra's algorithm. In addition, one could use image processing with other data obtained by audio processing means, as the spectrogram or come with a more robust approach than the pitch salience which does not capture noisy note attacks or noisy spectrum.

Finally, the note refinement can be used to improve the performance of score informed source separation, in the situation where the score is not well aligned with the audio.

## 5. ACKNOWLEDGEMENTS

This work was supported by the European Commission, FP7 (Seventh Framework Programme), STREP project, ICT-2011.8.2 ICT for access to cultural resources, grant agreement No 601166. Phenix Project

## 6. REFERENCES

- [1] J.-J. Aucouturier and M. Sandler. Finding repeating patterns in acoustic musical signals. *VIRTUAL, SYNTHETIC, AND ENTERTAINMENT AUDIO*, pages 412–421, 2002.
- [2] J.J. Bosch, K. Kondo, R. Marxer, and J. Janer. Score-informed and timbre independent lead instrument separation in real-world scenarios. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 2417–2421, Aug 2012.
- [3] J.J. Carabias-Orti, P. Vera-Candeas, F.J. Rodriguez-Serrano, and F.J. Canadas-Quesada. A RealTime Audio to Score Alignment System using Spectral Factorization and Online Time Warping. *IEEE Transactions on Multimedia(submitted)*, 2014.
- [4] A. Cont. A coupled duration-focused architecture for real-time music-to-score alignment. *Pattern Anal. Mach. Intell. IEEE ...*, 32:974–987, 2010.
- [5] E. W. Dijkstra. A note on two problems in connexion with graphs. *NUMERISCHE MATHEMATIK*, 1(1):269–271, 1959.
- [6] Z. Duan and B. Pardo. Soundprism: An online system for score-informed source separation of music audio. *Selected Topics in Signal Processing, IEEE ...*, pages 1–12, 2011.
- [7] S. Ewert and M. Muller. Using score-informed constraints for NMF-based source separation. *Acoustics, Speech and Signal Processing (...)*, 2012.
- [8] J. O. Smith Iii and X. Serra. Parshl: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation . 1987.
- [9] C. Joder and B. Schuller. Off-line refinement of audio-to-score alignment by observation template adaptation. *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 206–210, 2013.
- [10] A. Klapuri. Multiple fundamental frequency estimation by summing harmonic amplitudes. In *ISMIR*, pages 216–221, 2006.
- [11] B. Niedermayer. *Accurate Audio-to-Score Alignment Data Acquisition in the Context of Computational Musicology*. PhD thesis, Johannes Kepler Universität, 2012.
- [12] M. Nixon. *Feature Extraction and Image Processing*. Elsevier Science, 2002.
- [13] S. Senturk, A. Holzapfel, and X. Serra. Linking Scores and Audio Recordings in Makam Music of Turkey. *Journal of New Music Research*, pages 35–53, 2014.
- [14] T.M. Wang, P.Y. Tsai, and A.W.Y. Su. Score-informed pitch-wise alignment using score-driven non-negative matrix factorization. In *Audio, Language and Image Processing (ICALIP), 2012 International Conference on*, pages 206–211, July 2012.

# A COMPOSITIONAL HIERARCHICAL MODEL FOR MUSIC INFORMATION RETRIEVAL

**Matevž Pesek**  
University of Ljubljana  
Faculty of computer  
and information science  
*matevz.pesek@fri.uni-lj.si*

**Aleš Leonardis**  
Centre for Computational  
Neuroscience and Cognitive Robotics  
School of Computer Science  
University of Birmingham  
*ales.leonardis@fri.uni-lj.si*

**Matija Marolt**  
University of Ljubljana  
Faculty of computer  
and information science  
*matija.marolt@fri.uni-lj.si*

## ABSTRACT

This paper presents a biologically-inspired compositional hierarchical model for MIR. The model can be treated as a deep learning model, and poses an alternative to deep architectures based on neural networks. Its main features are generativeness and transparency that allow clear insight into concepts learned from the input music signals. The model consists of multiple layers, each is composed of a number of parts. The hierarchical nature of the model corresponds well with the hierarchical structures in music. Parts in lower layers correspond to low-level concepts (e.g. tone partials), while parts in higher layers combine lower-level representations into more complex concepts (tones, chords). The layers are unsupervisedly learned one-by-one from music signals. Parts in each layer are compositions of parts from previous layers based on statistical co-occurrences as the driving force of the learning process. We present the model's structure and compare it to other deep architectures. A preliminary evaluation of the model's usefulness for automated chord estimation and multiple fundamental frequency estimation tasks is provided. Additionally, we show how the model can be extended to event-based music processing, which is our final goal.

## 1. INTRODUCTION

The field of music information retrieval (MIR) has reached a significant expansion in tasks and solutions in the short timespan of its existence [3, 10]. The tasks include extraction of high-level music descriptors from music, such as melody, chords and rhythm, as well as highly perceptual tasks involving mood estimation, genre recognition and artist influence. Solutions have not come to a perfect one for any of the described tasks yet; however, numerous approaches proposed each year are improving the

state-of-the-art rapidly. Recently, deep belief networks as an alternative single model for a variety of tasks, have been successfully introduced to the field.

This paper presents a biologically-inspired compositional hierarchical model for music information retrieval. The proposed model poses an alternative to recent deep learning architecture approaches [6, 9]. Its main difference from the latter is in its transparent structure, thus allowing representation and interpretation of the signal's information extracted on different levels. We show the usefulness of our proposed approach in a preliminary evaluation of the model for the tasks of automated chord estimation and multiple fundamental frequency estimation. We also show how the model can be extended to event-based music processing, and point out how the model's transparency enables other applications of the model, e.g. for music analysis, synthesis and visualization.

## 2. DEEP ARCHITECTURES FOR MIR

The concept of deep learning has grown in popularity in the fields of signal processing [15], audio processing [9] and MIR. Lee [7] presented one of the first attempts of using deep belief networks (DBNs) on audio signals, where convolutional DBNs were applied to the speaker identification task. A DBN was used as a feature extractor, and a support vector machine for classification.

Later, Hamel and Eck [5], evaluated DBNs for genre recognition using a five-layer DBN with three hidden layers for feature extraction. The support vector machine was used for classification, where as raw spectral data was used as input to the DBN. DBNs show great potential for many tasks that involve high-level feature extraction, such as emotion recognition, since there is usually no trivial spectral or temporal feature that could be used to model the high-level representation in question. Schmidt and Kim [13] showed promising results by using a 5-layer DBN for extraction of emotion-based acoustic features. Other approaches modeled temporal aspects of the audio signal. Conditional DBNs were used by Battenberg and Wessel [1] for drum pattern analysis. Schmidt [12] took a step further and showed that DBNs can be trained for discriminating rhythm and melody.

Overall, recent research has shown great interest and



© Matevž Pesek, Aleš Leonardis, Matija Marolt.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Matevž Pesek, Aleš Leonardis, Matija Marolt. "A compositional hierarchical model for music information retrieval", 15th International Society for Music Information Retrieval Conference, 2014.

success in using features learned from music signals, in contrast to previously used hand-crafted features. The research reviewed in this subsection took place only in the last few years; thus, there is a vast expansion of deep learning in MIR to be expected, as anticipated by Humphrey [6].

### 3. THE COMPOSITIONAL HIERARCHICAL MODEL

#### 3.1 Motivation and concept

DBNs brought an improvement to many MIR tasks with their unsupervised learning of features and generative modeling. However, they require a large set of hidden units per layer, and consequently, large training sets. Also, the hidden nature of units offers no clear explanation of the underlying feature extraction process and the meaning of extracted features. It is our goal to overcome these limitations by developing a white-box compositional hierarchical model with shareable parts, thus reducing the number of parts and learning data needed, as well as reaching transparency in terms of interpretable internal structure of the model.

The proposed model provides a hierarchical representation of the audio signal, from the signal components on the lowest level, up to individual musical events on the highest levels. It is built on the assumption that a complex signal can be decomposed into a hierarchy of building blocks - *parts*. These parts exist at various levels of granularity and represent sets of entities describing the signal. According to their complexity, parts can be structured across several layers from less to the more complex. Parts on higher layers are expressed as compositions of parts on lower layers (e.g.: a chord is composed of several pitches, each pitch of several harmonics etc.). A part can therefore describe individual frequencies in a signal, their combinations, as well as pitches, chords and temporal patterns, such as chord progressions.

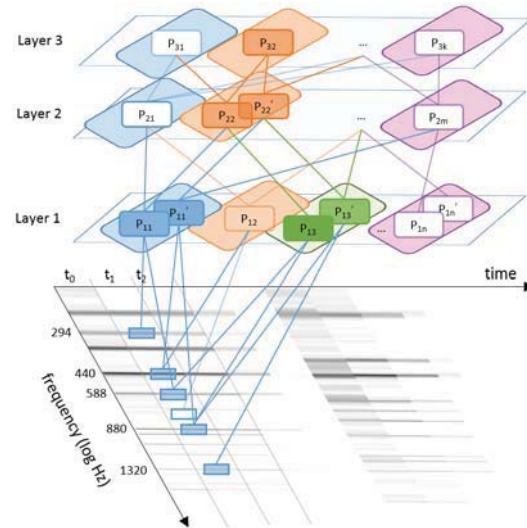
The structure of our model is inspired by work in computer vision, specifically the hierarchical compositional model presented by Leonardis and Fidler [8]. Their model represents objects in images in a hierarchical manner, structured in layers from simple to complex image parts. The model is learned from the statistics of natural images and can be employed as a robust statistical engine for object categorization and other computer vision tasks. We believe that such approach can also be used for music representation and analysis, however the transformation of the model to a different domain is not trivial.

#### 3.2 Model structure

The compositional hierarchical model consists of several layers. Each layer contains a set of parts. A part is a composition of two or more parts from a lower layer and may itself be part of any number of compositions on a higher layer. Thus, the compositional model forms a hierarchy of parts, where each part represents a composition of lower-layer parts, as seen in Figure 1. Connections in the figure represent compositions of parts.

##### 3.2.1 Input layer

The input layer of the model is derived from the time-frequency representation of the music signal. We denote this layer as layer  $\mathcal{L}_0$ . It contains a single atomic part, which is activated (produces output) at locations of all frequency components in the signal at a given time instance. An example is given in Figure 1, although not all activations are shown for clarity. More formally, a part's activation is defined by two values: location  $L_P$  that corresponds to frequency, and magnitude  $A_P$ , that corresponds to magnitude of the frequency component.



**Figure 1.** Compositional hierarchical model. Parts on the input layer correspond to signal components in the time-frequency representation. Parts on higher layers are compositions of lower-layer parts (denoted as links in the figure). A part may be contained in several compositions, e.g.  $P_{11}$  on the first layer is part of compositions  $P_{21}$ ,  $P_{22}$  and  $P_{2m}$  on the second layer. Several depictions of the same part (e.g. part instances  $P_{11}$  and  $P'_{11}$ ) denote several activations of the part on different locations (all instances of a part on a layer are marked with the same outlined color). Parts activated in  $t_1$  are shown filled with color.

Any time-frequency representation can be used for the input layer, although logarithmic frequency spacing produces more compact models due to the relative nature of part compositions on higher layers (as described further on).

##### 3.2.2 Subsequent layers

Higher layers of the model  $\mathcal{L}_n$  contain sets of *compositions* - parts composed of parts from lower layers. Each composition can contain any number of parts from the lower layers (for clarity we only use two-part compositions to explain the model). A composition can be part of any number of compositions on higher layers. Compositions are denoted as links between parts in Figure 1.

Composition  $i$  on layer  $\mathcal{L}_n$  can be formally defined as a structure containing parts from a layer below: a central part  $C$ , and a secondary part  $S$ . We name the parts forming

a composition subparts. A composition can be defined as:

$$P_{n,i} = \{C_{n-1,j}, S_{n-1,k}, (\mu_{n,i}, \sigma_{n,i})\}, \quad (1)$$

where  $C_{n-1,j}$  and  $S_{n-1,k}$  are the central and secondary subparts from layer  $n - 1$ , while  $\mu_{n,i}$  and  $\sigma_{n,i}$  define a Gaussian limiting the difference between locations of subpart activations (see definition of activation below). For clarity, we shall omit subscripts in the following equations and use  $P, C, S, \mu$  and  $\sigma$  to denote a part and its components.

A composition is *activated* (propagates output to higher layers) when all of its subparts are activated. This strict condition can be softened with hallucination, as explained in section 3.3. Part activation is composed of two values: activation location  $L_P$ , which represents the location (frequency) at which the part is activated, and activation magnitude  $A_P$ , which represents the strength of activation. The location of part's activation is defined simply as the location of activation of its central subpart:

$$L_P = L_C. \quad (2)$$

Thus, central parts of compositions on different layers propagate their locations upwards through the hierarchy. The magnitude of activation is defined as:

$$A_P = \tanh[G(L_C - L_S, \mu, \sigma) \cdot (A_C + A_S)], \quad (3)$$

where  $\tanh$  stands for the hyperbolic tangent function that limits the magnitude to  $[0,1]$  and  $G$  represent the Gaussian that limits the difference in locations of the central part and the subpart according to  $\mu$  and  $\sigma$ . As an example,  $P_{2,2}$  in Figure 1 is defined as

$$P_{2,2} = \{P_{1,1}, P_{1,3}, (1200, 25)\}, \quad (4)$$

where  $\mu$  and  $\sigma$  are given in cents. Therefore, it will be activated whenever  $P_{1,1}$  and  $P_{1,3}$  will be activated at locations approximately one octave (1200 cents) apart. Two such activations are shown in the figure, one at 294 Hz and one at 440 Hz.

### 3.3 Inference

The model can be used as a feature extractor over any desired dataset. An audio signal, transformed into a time-frequency representation, serves as input for layer  $\mathcal{L}_0$ . Activations are then calculated layer-by-layer according to Equations 2 and 3. Additionally, two biologically-inspired mechanisms govern the inference process and increase robustness of the model: *hallucination* and *inhibition*.

Before we define both mechanisms, we need to introduce the concept of coverage. Coverage  $c(P, L_P)$  of part  $P$  active at location  $L_P$  represents all signal information (frequency components) covered by the part and its subtree of parts. It is calculated top-down from an active part to  $\mathcal{L}_0$  as:

$$c(P, L_P) = \bigcup \{c(C, L_P), c(S, L_P + \mu)\}. \quad (5)$$

For the  $\mathcal{L}_0$  layer, coverage is defined as the set of parts with positive activations  $A_P > 0$ , thus representing the

set of covered frequency components. An example from Figure 1: the coverage of  $P_{2,2}$  active at 294 Hz is the set of frequencies:  $\{294Hz, 588Hz, 880Hz\}$ .

#### 3.3.1 Hallucination

Hallucination deals with filling-in the missing or damaged information in the signal and is implemented by enabling part activation in presence of incomplete input. The missing information in the signal can be replaced with knowledge encoded in the model during learning by allowing activations of parts most fittingly covering the information present. This allows the model to produce hypotheses in situations with no straight result. Hallucination also boosts alternative explanations of input data, thus increasing its explanation power and robustness.

Hallucination is governed by parameter  $\tau_1$  which can be defined per layer and modified during the inference. It changes the conditions under which a part may be activated. The default condition, as explained in section 3.2, is that activation of a part is possible when all of its subparts are active. With hallucination, a part  $P$  may be activated at location  $L_P$ , when the number of frequency components it covers  $|c(P, L_P)|$ , divided by the maximal number of components it may cover is larger than  $\tau_1$ . For example, a  $\tau_1$  of 0.75 means that  $\frac{3}{4}$  of all possible frequency components must be covered by the part for it to be activated. A  $\tau_1$  of 1 represents the default behavior.

#### 3.3.2 Inhibition

The second biologically-inspired mechanism provides a balancing factor by reducing redundant activations, similar to lateral inhibition performed by the human auditory system. Inhibition refines the set of parts that yield competing hypotheses of the same fragments of information in the input signal. Parts with greater activation magnitudes are retained and weaker activations inhibited. Inhibition also reduces activations that result from noise in the signal.

Activation of part  $P$  at  $L_P$  is inhibited, when another part  $Q$  with activation  $L_Q$  on the same layer (or a set of parts) covers the same fragments of information in the input signal, but with higher activation. The condition can be expressed as:

$$\exists Q : \frac{|c(P, L_P) \setminus c(Q, L_Q)|}{|c(P, L_P)|} < \tau_2 \wedge A_Q > A_P, \quad (6)$$

where  $\tau_2$  defines the amount of inhibition. For example, a value of 0.5 means that activation of  $P$  is inhibited if half of its coverage is already covered by another, stronger part.

To sum up: inference yields a set of activations on all model layers by calculating activations considering hallucination and inhibition over all layers in a bottom-up order and over all time-frames of the input signal. Resulting activations represent model features and can be directly interpreted or used as inputs for discriminative tasks.

### 3.4 Learning

The model is learned in an unsupervised manner on a set of input signals. It is constructed layer-by-layer, similar

to other deep architectures. The learning process relies on statistics of part activations, thus signal regularities are the driving force of the learning process.

When building layer  $\mathcal{L}_n$ , co-occurrences of activations of parts on  $\mathcal{L}_{n-1}$  are observed. Compositions are formed from parts that frequently activate together at similar distances. All such parts are joined into compositions and added to the set of candidate compositions  $\mathcal{P}$ . When forming a composition of two frequently co-occurring parts, the part at the lower location represents the central part of the composition, while parameters  $\mu$  and  $\sigma$  are estimated from all co-occurring activations of the two parts.

To reduce the number of compositions on each layer and keep only the most informative ones, the set of candidates  $\mathcal{P}$  is refined. The goal of refinement is to reduce the number of compositions in the learned layer while maintaining sufficient coverage of information in the learning set.

Refinement is implemented with a greedy approach, where in each iteration, a part that contributes most to the coverage of information in the learning set, is selected and added to the layer. Refinement is concluded when one of the following two criteria are reached: a sufficient percentage of information in the learning set is covered (according to threshold  $\tau_3$ ), or no part remaining in the candidate set adds to the cumulative coverage of information. Algorithm 1 outlines the described approach.

---

**Algorithm 1** Greedy approach for selection of compositions from the candidate set  $\mathcal{P}$ . Parts that add most to the coverage of information in the learning set are preferred. Function *perc* calculates the percentage of information covered in the learning set by the given set of parts.

---

```

1: procedure REFINED( $\mathcal{P}$ )
2:    $prevCov \leftarrow 0$ 
3:    $coverages \leftarrow \emptyset$ 
4:    $\mathcal{L}_n \leftarrow \emptyset$ 
5:   repeat
6:     for  $P \in \mathcal{P}$  do
7:        $coverages[P] \leftarrow perc(\mathcal{L}_n \cup P)$ 
8:        $Chosen \leftarrow \underset{P}{\operatorname{argmax}}(coverages)$ 
9:        $\mathcal{L}_n \leftarrow \mathcal{L}_n \cup Chosen$ 
10:       $\mathcal{P} \leftarrow \mathcal{P} \setminus Chosen$ 
11:      if  $coverages[Chosen] = prevCov$  then
12:        break //No added coverage - finish
13:       $prevCov \leftarrow coverages[Chosen]$ 
14:   until  $prevCov > \tau_3 \vee \mathcal{P} = \emptyset$ 

```

---

### 3.5 Time

The model presented so far is time-independent. It operates on a frame-by-frame basis, where each time frame in the time-frequency representation is treated independently from others. Music, however, evolves in time and models that operate on such bases often fail to reflect the evolution of sound properly.

The proposed model can be naturally extended to include the time dimension. Our first step towards extending the model for time-dependent processing was to implement a short-time automatic gain control mechanism, similar to the automatic gain control contrast mechanism in human and other animal perceptual systems. The mechanism inte-

grates part activations at similar locations over time. When a new part activation appears and persists, its value is initially boosted to accentuate the onset and later suppressed towards a stable value.

The mechanism operates on all layers, and has a short-term effect on lower layers, and longer-term effect on higher layers due to the upward propagation of activations. Its end effect is that it stabilizes activations, reduces noise, produces smoother model output and boosts event onsets.

### 3.6 Relation to Deep Architectures

The compositional hierarchical model shares a great deal of similarities with other deep learning architectures. The structure of the model is similar in terms of learning a variety of signal abstractions on several layers of granularity. The model is learned in an unsupervised generative manner, thus, no annotated data is needed. The learning procedure is similar: the structure is built layer-by-layer. The proposed model can also be used for discriminative tasks by observing activations of parts on multiple layers.

We see the biggest advantage of the proposed compositional hierarchical model over other established deep architectures in its transparency. As parts are compositions of subparts, their activations are directly observable and interpretable. This opens the model up for a variety of interesting usages, as it not only produces features that can be used, but features that can be interpreted and explained. In addition, the inhibition and hallucination mechanisms make it possible to produce alternative explanations of the input by suppressing the winning explanation and search for alternatives. In comparison to DBNs, where the outputs of each layer can only be interpreted during the evaluation, the proposed model offers a deeper analysis of results by tracing the higher layer activations over all layers and investigating the impact of each subpart.

Another difference in comparison to DBNs is the shareability and *relativeness* of parts, which both lead to a small number of parts needed to represent complex signals. A part in the proposed model is defined by the relative distance between its subparts and can thus be activated on different locations along the frequency axis. Thus, the large amount of layer units that DBNs need to cover the entire spectrum is not necessary and is replaced by reusing the available parts. This relativeness is accompanied with the concept of part shareability: parts on a layer may be shared by many compositions on higher layers. For example, a chord is composed of at least three pitches which may be identical in their representation in our model.

We show the usefulness of the described model's features in the evaluation section, where the model is used as both feature extractor and a classifier. Other possible applications exploiting the the model's structure are presented in section 5.

## 4. EVALUATION OF THE MODEL

The presented model is applicable to different MIR tasks. To present the model's usefulness, we built a three-layer

model and evaluated it on two tasks: automated chord estimation and multiple fundamental frequency estimation.

The input layer was the same for both tasks. A constant-Q transform was used to transform music signals onto 345 frequency bins between 55 and 8000 Hz, with a step size of 50 ms and maximal window size of 100 ms. Two layers of compositions  $\mathcal{L}_1$  and  $\mathcal{L}_2$  were learnt as described previously. Due to the shareability of parts, they contain only 23 and 12 parts respectively. The small number of parts in the model should mean that the model could be trained on a small learning set. We tested this hypothesis and trained the model on large and small datasets, and observed few differences. We were therefore able to build the model by using only a small set of 88 piano key samples as our learning set. We used the  $\mathcal{L}_2$  layer for the task of multiple fundamental frequency estimation. For the task of automated chord estimation, we provided an additional  $\mathcal{L}_3$  octave-invariant layer. The latter consists of 48 parts, where  $\mathcal{L}_3$  activations correspond to octave-invariant activations of the  $\mathcal{L}_2$ .

#### 4.1 Automated Chord Estimation

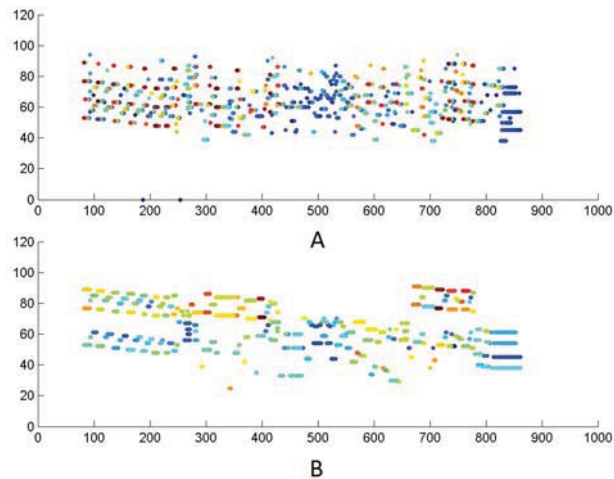
The time-independent model was tested for the task of automated chord estimation on the standard *Beatles* dataset, kindly provided by C. Harte. We used activations of the octave-invariant  $\mathcal{L}_3$  layer as features and made the classification by using a hidden Markov model (HMM) with 24 states, each representing a chord, as described by [2]. We used cross-validation for evaluation; one album was used for HMM training and the rest of the dataset for estimation.

Our per-frame classification accuracy on the given dataset was 67.14 % with 0.1525 standard deviation. Compared to other per-frame approaches, we find our results slightly lower than for example [11], which also used per-frame technique for feature extraction. Nevertheless, we performed the evaluation as a proof of concept with time-independent feature extraction and no fine-tuning of the model, its learning, nor tuning of HMM parameters. We anticipate significant results increase by extending the model to time-dependent evaluation, using the whole hierarchy for classification and parameter tuning.

#### 4.2 Multiple fundamental frequency estimation

The model was also tested for the task of multiple fundamental frequency estimation (MFEE) on the two subsets of MAPS (MIDI Aligned Piano Sounds) dataset, provided by [4]. Activations of layer  $\mathcal{L}_2$  were directly used as fundamental frequency estimations with no further processing.

The following metrics were used for evaluation: per-frame precision and recall, precision and recall without penalising for octave errors, and pitch-class precision and recall. Results are shown in Table 1. Our results are significantly lower when compared to recent approaches, e.g. [14] which reported 77.1% classification accuracy on the subsets. However, the mentioned approach differs significantly from ours, as a severely larger dataset (approx. 4 times larger than the test sets) was used for training the support vector machine (SVM) classifier. In comparison,



**Figure 2.** Hypotheses produced by our model for the task of multiple fundamental frequency estimation (A) and the ground truth (B). X axis represent time (in frames), and y midi pitches. Although the model produces many possible hypotheses per frame, only the ones with the highest magnitudes are used for comparison. Colors represent the magnitudes of activations in Fig. A or the MIDI velocity in Fig. B.

our model was trained only on a small set of piano key samples, so no parts of the MAPS dataset were used for training. It is also worth to mention that for this task, our model was used as a feature extractor and a classifier at the same time. We expect that accuracy would be improved if a classifier such as a SVM would be added on top of our model and would take features extracted on all layers for inputs. Our intention for this paper, however, is to present the general applicability of the model for multiple tasks and to avoid fine-tuning.

**Table 1.** Classification accuracy (CA) using all hypotheses provided by the model, precision (Pr) and recall (Re) values over a part of the MAPS dataset. Results without penalising octave errors and considering only pitch classes are marked with *O* and *PC* subscripts respectively.

Folder name	CA	Pr	Re
<i>AkPnBcht</i>	56.53 %	19.40 %	55.69 %
<i>AkPnBsd</i>	66.17 %	22.05 %	61.27 %
<i>AkPnBcht<sub>O</sub></i>	67.08 %	35.37 %	64.55 %
<i>AkPnBsd<sub>O</sub></i>	71.16 %	46.10 %	68.83 %
<i>AkPnBcht<sub>PC</sub></i>	86.20 %	51.83 %	86.59 %
<i>AkPnBsd<sub>PC</sub></i>	88.23 %	58.68 %	70.99 %

## 5. OTHER APPLICATIONS OF THE MODEL

Our intention with developing the proposed model is to make an interpretable model that overcomes some of the limitations of DBNs and can be used for tackling various MIR tasks. Its transparency, however, also makes other

uses of the model possible.

The hierarchical approach presented in this paper fits well with the hierarchical structure of music in frequency as well as in time domains. Each part of the model represents an explainable entity (e.g. tone partial, pitch, chord). In contrast to the DBNs, each part of the model can be visualized. Visualization not only exposes the layered structure of the model, but also discloses information processed by the observed part and its influence on other parts and their activations. This insight into the music signal can be used in several scenarios — music visualization, music analysis and music synthesis.

We have developed a real-time visualization of the model, enabling deeper understanding of the processed information. When observing an inferred audio signal, the output of all layers of the model is presented by visualizing activations of parts. This insight enables detailed analysis of each event in the music signal and may bring additional event details to light. For example, a chord inversion can be observed by looking into the activated subtree of the chord from top layers to bottom-ones. Thus, visualization of our model offers an innovative user interface for music analysis.

The transparency of the model can also be exploited for music processing and synthesis. Parts across all layers form a variety of harmonic structures, and can be used for signal manipulation and synthesis. By activating a set of parts at different locations, a new spectral representation is produced. Although the interface may not provide a sufficient amount of features for a standalone music performance, it can be used as a sound generator in a combination with a music instrument, e.g. a MIDI keyboard. The interface thus serves as an advanced tool for spectral modification, while the instrument provides the interface for performance.

## 6. CONCLUSION AND FUTURE WORK

This paper presents a compositional hierarchical model as an alternative to deep learning architectures based on neural networks. The model shares a great deal of similarities with other deep architectures, including a multi-layer structure, unsupervised generative learning and suitability for discriminative tasks. Furthermore, the white-box structure of the model offers new utilizations of the model. We highlighted three possible applications: feature extraction for MIR tasks, music visualization and music analysis/synthesis.

The model's internals rely on findings in the fields of neurobiology and cognitive sciences. By implementing biologically-inspired mechanisms into the model, we made an attempt to build a model which partially resembles a subset of functions of the human auditory system. We intend to retain and further develop this aspect of the model with an intention to bring the computational modeling closer to human auditory perception.

The paper presents an initial development of our model. We plan to further extend it with the focus on temporal modeling. Parts can namely be extended into the time do-

main, thus bringing their activations closer to event-based modeling. We also plan to tackle temporal tasks, such as onset detection, as well as beat tracking and tempo estimation. The proposed model is also going to be evaluated for pattern analysis of symbolic data, including discovery of repeated themes, and symbolic melodic similarity.

## 7. REFERENCES

- [1] Eric Battenberg and David Wessel. Analyzing Drum Patterns using Conditional Deep Belief Networks. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 37–42, 2012.
- [2] Juan P. Bello and Jeremy Pickens. A robust mid-level representation for harmonic content in music signals. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 304–311, London, 2005.
- [3] J. Stephen Downie, Andreas F. Ehmann, Mert Bay, and M. Cameron Jones. The Music Information Retrieval Evaluation eXchange: Some Observations and Insights. In Wieczorkowska A.A. and Ras Z.W., editors, *Advances in Music Information Retrieval*, pages 93–115. Springer-Verlag, Berlin, 2010.
- [4] Valentin Emiya, Roland Badeau, and Bertrand David. Multipitch Estimation of Piano Sounds Using a New Probabilistic Spectral Smoothness Principle. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6):1643–1654, August 2010.
- [5] Philippe Hamel and Douglas Eck. Learning Features from Music Audio with Deep Belief Networks. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 339–344, 2010.
- [6] Eric J. Humphrey, Juan P. Bello, and Yann LeCun. Moving beyond feature design: deep architectures and automatic feature learning in music informatics. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Porto, 2012.
- [7] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y. Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in Neural Information Processing Systems*, pages 1096–1104, 2009.
- [8] Aleš Leonardis and Sanja Fidler. Towards scalable representations of object categories: Learning a hierarchy of parts. *Computer Vision and Pattern Recognition, IEEE*, pages 1–8, 2007.
- [9] Abdel-rahman Mohamed, George E. Dahl, and Geoffrey Hinton. Acoustic Modeling using Deep Belief Networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):14–22, 2010.
- [10] Nicola Orio. Music Retrieval: A Tutorial and Review. *Foundations and Trends® in Information Retrieval*, 1(1):1–90, 2006.
- [11] Helene Papadopoulos and Geoffroy Peeters. Large-case Study of Chord Estimation Algorithms Based on Chroma Representation and HMM. *Content-Based Multimedia Indexing*, 53–60, 2007.
- [12] Eric M. Schmidt and Youngmoo E. Kim. Learning Rhythm and Melody Features with Deep Belief Networks. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 21–26, 2013.
- [13] Erik M. Schmidt and Youngmoo E. Kim. Learning emotion-based acoustic features with deep belief networks. In *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 65–68. IEEE, October 2011.
- [14] Felix Weninger, Christian Kirst, Bjorn Schuller, and Hans-Joachim Bungartz. A discriminative approach to polyphonic piano note transcription using supervised non-negative matrix factorization. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6–10, Vancouver, 2013.
- [15] Dong Yu and Li Deng. Deep Learning and Its Applications to Signal and Information Processing [Exploratory DSP]. *IEEE Signal Processing Magazine*, 28(1):145–154, January 2011.



# AN ANALYSIS AND EVALUATION OF AUDIO FEATURES FOR MULTITRACK MUSIC MIXTURES

Brecht De Man<sup>1</sup>, Brett Leonard<sup>2,3</sup>, Richard King<sup>2,3</sup>, Joshua D. Reiss<sup>1</sup>

<sup>1</sup>Centre for Digital Music, Queen Mary University of London

<sup>2</sup>The Graduate Program in Sound Recording, Schulich School of Music, McGill University

<sup>3</sup>Centre for Interdisciplinary Research in Music Media and Technology

b.deman@qmul.ac.uk, brett.leonard@mail.mcgill.ca,

richard.king@mcgill.ca, joshua.reiss@qmul.ac.uk

## ABSTRACT

Mixing multitrack music is an expert task where characteristics of the individual elements and their sum are manipulated in terms of balance, timbre and positioning, to resolve technical issues and to meet the creative vision of the artist or engineer. In this paper we conduct a mixing experiment where eight songs are each mixed by eight different engineers. We consider a range of features describing the dynamic, spatial and spectral characteristics of each track, and perform a multidimensional analysis of variance to assess whether the instrument, song and/or engineer is the determining factor that explains the resulting variance, trend, or consistency in mixing methodology. A number of assumed mixing rules from literature are discussed in the light of this data, and implications regarding the automation of various mixing processes are explored. Part of the data used in this work is published in a new online multitrack dataset through which public domain recordings, mixes, and mix settings (DAW projects) can be shared.

## 1. INTRODUCTION

The production of recorded music involves a range of expert signal processing techniques applied to recorded musical material. Each instrument or element thereof exists on a separate audio ‘track’, and this process of manipulating and combining these tracks is normally referred to as mixing. Strictly creative processes aside, each process can generally be classified as manipulating the dynamic (balance and dynamic range compression), spatial (stereo or surround panning and reverberation), and spectral (equalisation) features of the source material, or a combination thereof [1, 4, 8, 15].

Recent years have seen a steep increase in research on automatic mixing, where some of the tedious, routine tasks in audio production are automated to the benefit of the inexperienced amateur or the time constrained professional.



© Brecht De Man, Brett Leonard, Richard King and Joshua D. Reiss.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Brecht De Man, Brett Leonard, Richard King and Joshua D. Reiss. “An Analysis and Evaluation of Audio Features for Multitrack Music Mixtures”, 15th International Society for Music Information Retrieval Conference, 2014.

Most research is concerned with the validation of a mixing rule based on knowledge derived from practical literature or expert interviews [2, 6, 7, 9], usually through an experiment where a method based on this assumption is compared to a set of alternative methods. Furthermore, some research has been done on machine learning systems for balancing and panning of tracks [13]. In spite of these efforts, the relation between the characteristics of the source material and the chosen processing parameters, as well as the importance of subjective input of the individual versus objective or generally accepted target features, is still poorly understood. Recurring challenges in this field include a lack of research data, such as high-quality mixes in a realistic but sufficiently controlled setting, and tackling the inherently high cross-adaptivity of the mixing problem, as the value of each processing parameter for any given track is usually dependent on features and chosen processing parameters associated with other tracks as well.

In this work, we conduct an experiment where a group of mixing engineers mix the same material in a realistic setting, with relatively few constraints, and analyse the manipulation of the signals and their features. We test the validity of the signal-dependent, instrument-independent model that is often used in automatic mixing research [6, 7], and try to identify which types of processing are largely dependent on instrument type, the song (or source material), or the individual mixing engineer. Consequently, we also identify which types of processing are not clearly defined as a function of these parameters, and thus warrant further research to understand their relation to low-level (readily extracted) features or high-level properties (instrument, genre, desired effect) of the source audio. We discuss the relevance of a number of audio features for the assessment of music production and the underlying processes as described above. This experiment also provides an opportunity to validate some of the most common assumptions in autonomous mixing research.

## 2. EXPERIMENT

The mixing engineers in this experiment were students of the MMus in Sound Recording at the Schulich School of Music at McGill University. They were divided in two groups of eight, where each group corresponds with a class

from a different year in the two-year programme, and each group was assigned a different set of four songs to mix. Each mixing engineer allocated up to 6 hours to each of their four mix assignments, and was allowed to use Avid’s Pro Tools including built-in effects (with automation) and the Lexicon PCM Native Reverb Plug-In Bundle, a set of tools they were familiar with.

Four out of eight songs are available on a new multi-track testbed including raw tracks, the rendered mixes and the complete Pro Tools project files, allowing others to reproduce or extend the research. The testbed can be found on [c4dm.eecs.qmul.ac.uk/multitrack](http://c4dm.eecs.qmul.ac.uk/multitrack). The authors welcome all appropriately licensed contributions consisting of shareable raw, multitrack audio, DAW project files, rendered mixes, or a subset thereof. Due to copyright restrictions, the other songs could not be shared.

We consider three types of instruments - drums, bass, and lead vocal - as they are featured in all test songs in this research, and as they are common elements in contemporary music in general. Furthermore, we split up the drums in the elements kick drum, snare drum, and ‘rest’. Three out of eight songs had a male lead vocalist, and half of the songs featured a double bass (in one case part bowed) while the other half had a bass guitar for the bass part.

For the purpose of this investigation, we consider a fragment of the song only, consisting of the second verse and chorus, as all considered sources (drums, bass and lead vocal) are active here.

Whereas the audio was recorded and mixed at a sampling ratio of 96 kHz, we converted all audio to 44.1 kHz to reduce computational cost and to calculate spectral features based on the mostly audible region. The processed tracks are rendered from the digital audio workstation with all other tracks inactive, but with an unchanged signal path including send effects and bus processing<sup>1</sup>.

### 3. FEATURES

The set of features we consider (Table 1) has been tailored to reflect properties relevant to the production of music in the dynamic, spatial and spectral domain. We consider the mean of the feature over all frames of a track fragment.

We use the perceptually informed measure of loudness relative to the loudness of the mix, as a simple RMS level can be strongly influenced by high energy at frequencies the human ear is not very sensitive to. To accurately measure loudness in the context of multitrack content, we use the highest performing modification in [12] (i.e. using a time constant of 280 ms and a pre filter gain of +10 dB) on the most recent ITU standard on measuring audio programme loudness [3].

<sup>1</sup> When disabling the other tracks, non-linear processes on groups of tracks (such as bus dynamic range compression) will result in a different effective effect on the rendered track since the processor may be triggered differently (such as a reduced trigger level). While for the purpose of this experiment, the difference in triggering of bus compression does not affect the considered features significantly, it should be noted that for rigorous extraction of processed tracks, in such a manner that when summed together they result in the final mix, the true, time-varying bus compression gain should be measured and applied on the single tracks.

Category	Feature	Reference
Dynamic	Loudness	[3, 12]
	Crest factor (100 ms and 1 s)	[17]
	Activity	[7]
Spatial	SPS	[16]
	$P_{[band]}$	[16]
	Side/mid ratio	
	Left/right imbalance	
Spectral	Centroid	[5]
	Brightness	
	Spread	
	Skewness	
	Kurtosis	.
	Rolloff (.95 and .85)	.
	Entropy	
	Flatness	
	Roughness	
	Irregularity	
	Zero-crossing rate	
Low energy	[5]	
Octave band energies		

**Table 1:** List of extracted features

To reflect the properties of the signal related to dynamic range on the short term, we calculate the crest factor over a window of 100 ms and over a window of 1 s [17].

To quantify gating, muting, and other effects that make the track (in)audible during processing, we measure the percentage of time the track is active, with the activity state indicated by a Schmitt trigger with thresholds at  $-25$  and  $-30$  dB LUFS [7].

To analyse the spatial processing, we use the Stereo Panning Spectrum (SPS), which shows the spatial position of a certain frequency bin in function of time, and the Panning Root Mean Square ( $P_{[band]}$ ), the RMS of the SPS over a number of frequency bins [16]. In this work, we use the absolute value of SPS, averaged over time, and the standard  $P_{total}$  (all bins),  $P_{low}$  (0-250 Hz),  $P_{mid}$  (250-2500 Hz) and  $P_{high}$  (2500-22050 Hz), also averaged over time. Furthermore, we propose a simple stereo width measure, the side/mid ratio, calculated as the power of side channel (sum of left and right channel) over the power of the mid channel (average of left channel and polarity-reversed right channel). We also define the left/right imbalance, as  $(R - L)/(R + L)$  where  $L$  is the total/average power of the left channel, and  $R$  is the total/average power of the right channel. A centred track has low imbalance and low side/mid ratio, while a hard panned track has high imbalance and high side/mid ratio. Note that while these features are related, they do not mean the same thing. A source could have uncorrelated signals with the exact same energy in the left and right channel respectively, which would lead to a low left/right imbalance and a high side/mid ratio.

Finally, we use features included in the MIR Toolbox [5] (with the default 50 ms window length) as well as octave band energies to describe the spectral characteristics of the audio.

## 4. ANALYSIS AND DISCUSSION

### 4.1 Analysis of variance

Table 2 shows the mean values of the features, as well as the standard deviation between different mixing engineers and the standard deviation between different songs. Most considered features show greater variance for the same engineer across different songs, than for the same song over different engineers. Exceptions to this are the left/right imbalance and spectral roughness, which on average appear to be more dependent on the engineer than on the source content. The change of features (difference before and after processing, where applicable) varies more for different mixing engineers than for different songs, too, for all features. However, when considering the features instrument by instrument, the source material only rarely causes the means of the feature to differ significantly (the means are only significantly different through the effect of source material for the zero-crossing rate of the snare drum track, and for the spectral entropy of the vocal track). This suggests that engineers would disagree on processing values, whereas the source material has less effect.

For each feature, we perform an analysis of variance to investigate for which feature we can reject the hypothesis that the different ‘treatments’ (different source material, mixing engineer or instrument) result in the same feature value. For those features for which there is a significant effect ( $p < 0.05$ ), we perform a multiple comparison of population means using the Bonferroni correction to establish what the mean values of the feature are as a function of the determining factor, and which instruments or songs have a significantly lower or higher mean than others. We discuss the outcome of these tests in the following paragraphs.

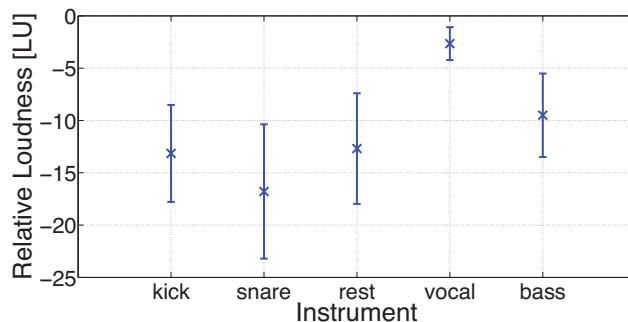
As some elements were not used by the mixing engineer, some missing values are dropped when calculating the statistics in the following sections.

### 4.2 Balance and dynamics processing

In general, the relative loudness of tracks, averaged over all instruments, is dependent on the song ( $p < 5 \cdot 10^{-11}$ ). However, when looking at each instrument individually, the relative loudness of the bass guitar ( $p < 0.01$ ), snare drum ( $p < 0.05$ ) and other drum instruments (‘rest’, i.e. not snare or kick drum,  $p < 5 \cdot 10^{-4}$ ) is dependent on mixing engineer.

In automatic mixing research, a popular assumption is that the loudness of the different tracks or sources should be equal [7]. A possible exception to this is the main element, usually the vocal, which can be set at a higher loudness [1]. From Figure 1, it is apparent that the vocal is significantly louder than the other elements considered here, whereas no significant difference of the mean relative loudness of the other elements can be shown. Furthermore, the relative loudness of the vocal shows a relative narrow range of values ( $-2.7 \pm 1.6$  LU), suggesting an agreement on a ‘target loudness’ of about  $-3$  LU relative to the overall mix loudness.

It should be noted that due to crosstalk between the drum microphones, the effective loudness of the snare drum



**Figure 1:** Average and standard deviation of loudness of sources relative to the total loudness of the mix, across songs and mixing engineers.

and kick drum will differ from the loudness measured from the snare drum and kick drum tracks. As a result, disagreement of the relative loudnesses of snare drum and other drum elements such as overhead and room microphones does not necessarily suggest a significantly different desired loudness of the snare drum, as the snare drum is present in both of these tracks. In this work, however, we are interested in the manipulations of the different tracks as they are available to the engineer.

The crest factor is affected by both the instrument ( $p < 5 \cdot 10^{-3}$ ) and song ( $p < 10^{-20}$ ), and every instrument individually shows significantly different crest factor values for different engineers ( $p < 5 \cdot 10^{-3}$ ). One exception to the latter is the kick drum for a crest factor window size of 1 s, where the hypothesis was not disproved for one group of engineers.

All instruments show an increase in crest factor compared to the raw values (ratio significantly greater than one). This means that the short-term dynamic range is effectively expanded, which can be an effect of dynamic range compression as transients are left unattenuated due to the response time of the compressor, while the rest of the signal is reduced in level.

The percentage of the time the track was active did not meaningfully change under the influence of different source material, individual mixing engineers or instruments. A drop in activity in some instances is due to gating of kick drum, but this is the decision of certain mixing engineers for certain songs, and no consistent trend.

### 4.3 Stereo panning

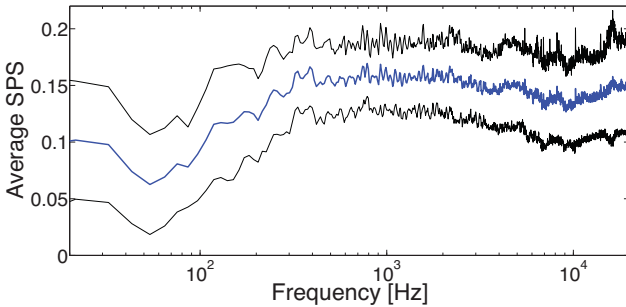
Both the average left/right imbalance and average side/mid ratio were significantly higher for the non-pop/rock songs ( $p < 10^{-6}$ ).

The Panning Root Mean Square values  $P_{[band]}$  all show a larger value for the total mix and for the ‘rest’ group. The difference is significant except for the lowest band, where only the bass is significantly more central than the total mix. This can be explained by noting that most of the low frequency sources are panned centre (see further).

In literature on automatic mixing and mixing engineering textbooks, it is stated that low-frequency sources as well as lead vocals and snare drums should be panned central [1, 2, 4, 6, 8–10, 14]. To quantify the spatialisation for different frequencies, we display the panning as a function

Feature	Kick drum	Snare drum	Rest drums	Bass	Lead vocal	Average	Mix
Loudness [LU]	-13.15 ± 4.05 3.89	-16.78 ± 6.17 4.57	-12.68 ± 5.46 2.80	-2.65 ± 1.52 1.31	-9.50 ± 3.51 2.86	-10.95 ± 4.14 3.09	N/A
Crest (100 ms)	3.599 ± 0.603 0.330	4.968 ± 0.998 0.469	4.510 ± 1.065 0.354	2.565 ± 0.443 0.166	3.315 ± 0.403 0.208	3.791 ± 0.634 0.274	3.332 ± 0.294 0.116
Crest (1 s)	9.824 ± 3.074 1.911	16.724 ± 6.458 3.135	12.472 ± 4.710 1.823	4.339 ± 1.098 0.449	5.283 ± 1.102 0.514	9.728 ± 2.907 1.398	5.315 ± 0.997 0.554
Activity	0.676 ± 0.250 0.122	0.861 ± 0.161 0.078	0.909 ± 0.115 0.029	0.958 ± 0.076 0.009	0.844 ± 0.089 0.044	0.850 ± 0.117 0.048	0.995 ± 0.009 0.004
L/R imbalance	<b>0.075 ± 0.094</b> <b>0.137</b>	<b>0.144 ± 0.153</b> <b>0.227</b>	0.361 ± 0.303 0.213	<b>0.107 ± 0.135</b> <b>0.176</b>	<b>0.045 ± 0.072</b> <b>0.085</b>	<b>0.146 ± 0.139</b> <b>0.152</b>	0.088 ± 0.075 0.074
Side/mid ratio	<b>0.036 ± 0.055</b> <b>0.076</b>	<b>0.036 ± 0.040</b> <b>0.043</b>	0.242 ± 0.183 0.154	<b>0.009 ± 0.013</b> <b>0.015</b>	<b>0.022 ± 0.018</b> <b>0.022</b>	0.069 ± 0.060 0.059	0.101 ± 0.049 0.046
$P_{total}$	0.104 ± 0.102 0.090	0.108 ± 0.082 0.059	0.307 ± 0.028 0.027	0.075 ± 0.093 0.083	<b>0.134 ± 0.022</b> <b>0.027</b>	0.145 ± 0.060 0.052	0.234 ± 0.030 0.027
$P_{low}$	<b>0.066 ± 0.078</b> <b>0.087</b>	0.122 ± 0.102 0.073	0.243 ± 0.045 0.041	0.040 ± 0.063 0.059	<b>0.147 ± 0.034</b> <b>0.042</b>	0.123 ± 0.061 0.056	0.188 ± 0.042 0.034
$P_{mid}$	<b>0.066 ± 0.074</b> <b>0.076</b>	0.114 ± 0.090 0.064	0.290 ± 0.023 0.023	0.052 ± 0.082 0.067	<b>0.177 ± 0.027</b> <b>0.035</b>	0.140 ± 0.054 0.048	0.248 ± 0.027 0.023
$P_{high}$	0.106 ± 0.104 0.091	0.105 ± 0.081 0.058	0.309 ± 0.029 0.028	0.076 ± 0.094 0.085	<b>0.124 ± 0.022</b> <b>0.028</b>	0.144 ± 0.061 0.053	0.231 ± 0.033 0.029
Centroid [Hz]	2253.8 ± 1065.6 729.8	4395.3 ± 1448.6 554.2	4130.8 ± 1228.1 483.2	1046.5 ± 520.1 232.4	2920.2 ± 452.1 264.7	2949.3 ± 872.1 418.6	2478.8 ± 517.9 247.1
Brightness	0.306 ± 0.105 0.103	0.598 ± 0.156 0.069	0.557 ± 0.115 0.058	0.135 ± 0.082 0.031	0.455 ± 0.071 0.040	0.410 ± 0.100 0.056	0.362 ± 0.070 0.034
Spread	3250.1 ± 783.2 447.5	4363.6 ± 701.9 335.9	4422.1 ± 734.6 292.3	2426.6 ± 559.2 320.4	3369.9 ± 324.6 191.3	3566.5 ± 587.5 298.0	3453.2 ± 421.7 200.6
Skewness	3.649 ± 1.068 0.886	1.492 ± 0.663 0.301	1.665 ± 0.682 0.246	6.234 ± 1.885 0.630	2.470 ± 0.573 0.243	3.102 ± 0.912 0.427	2.779 ± 0.600 0.257
Kurtosis	23.847 ± 11.997 9.164	5.965 ± 2.905 1.474	7.053 ± 3.449 1.263	58.870 ± 31.874 11.107	11.579 ± 4.267 1.784	21.463 ± 9.834 4.477	13.646 ± 4.511 2.073
Rolloff .95 [Hz]	8880.1 ± 3679.2 2151.2	13450.9 ± 3100.6 1582.2	13373.4 ± 2594.1 1007.4	4389.4 ± 2714.7 1244.5	9879.0 ± 1335.7 725.3	9994.5 ± 2498.0 1240.8	9679.0 ± 1563.8 734.3
Rolloff .85 [Hz]	4513.7 ± 2736.6 1788.8	8984.3 ± 3139.7 1348.5	8755.3 ± 2742.5 975.6	1625.5 ± 1205.0 594.3	5595.8 ± 1121.4 609.7	5894.9 ± 2047.2 986.1	5026.2 ± 1337.8 599.8
Entropy	0.655 ± 0.104 0.090	0.840 ± 0.084 0.057	0.832 ± 0.051 0.025	0.552 ± 0.073 0.026	0.735 ± 0.043 0.016	0.723 ± 0.066 0.038	0.744 ± 0.043 0.015
Flatness	0.148 ± 0.072 0.051	0.350 ± 0.142 0.056	0.337 ± 0.118 0.045	0.073 ± 0.035 0.020	0.167 ± 0.030 0.018	0.215 ± 0.074 0.035	0.174 ± 0.046 0.020
Roughness	<b>84.72 ± 84.85</b> <b>98.32</b>	<b>36.30 ± 41.16</b> <b>43.32</b>	67.57 ± 71.76 46.28	<b>236.04 ± 160.38</b> <b>176.05</b>	<b>247.00 ± 319.15</b> <b>247.36</b>	<b>134.33 ± 319.30</b> <b>338.44</b>	<b>1843.31 ± 1341.50</b> <b>1419.35</b>
Irregularity	0.158 ± 0.098 0.063	0.235 ± 0.151 0.079	0.297 ± 0.135 0.069	0.502 ± 0.176 0.065	0.540 ± 0.165 0.094	0.346 ± 0.136 0.075	0.705 ± 0.090 0.078
Zero-crossing	584.7 ± 509.5 409.4	2222.0 ± 1183.3 604.7	1988.9 ± 944.1 466.1	246.6 ± 217.8 89.6	1177.5 ± 233.7 143.6	1243.9 ± 554.3 305.4	905.2 ± 237.4 118.8
Low energy	0.752 ± 0.113 0.081	0.723 ± 0.084 0.055	0.682 ± 0.047 0.034	0.507 ± 0.096 0.033	0.544 ± 0.065 0.048	0.641 ± 0.073 0.048	<b>0.541 ± 0.035</b> <b>0.038</b>

**Table 2:** Average values of features per instrument, including average over instrument and value of total mix, with standard deviation between different songs by the same mixing engineer (top), and between different mixes of the same song (bottom). Values for which the variation across different mixes for the same song is greater than the variation across different songs for the same engineer are displayed in bold.



**Figure 2:** Mean Stereo Panning Spectrum (with standard deviation) over all mixes and songs

of frequency in Figure 2, using the average Stereo Panning Spectrum over all mixes and songs. From this figure a clear increase in SPS with increasing frequency is apparent between 50 Hz and 400 Hz. However, this trend does not extend to the very low frequencies (20-50 Hz) or higher frequencies (>400 Hz).

#### 4.4 Equalisation

To assess the spectral processing of sources, mostly equalisation in this context, we consider both the absolute values of the spectral features (showing the desired features of the processed audio) as well as the change in features (showing common manipulations of the tracks). When only tak-

ing the manipulations into account, and not the features of the source audio, similar to blindly applying a software equaliser's presets, the results would be less translatable to situations where the source material's spectral characteristics differs from that featured in this work [2]. However, considering the change in features could reveal common practices that are less dependent on the features of the source material. Therefore, we investigate both.

The spectral centroid of the whole mix varies strongly depending on the mixing engineer ( $p < 5 \cdot 10^{-6}$ ). The centroid of the snare drum track is consistently increased through processing, due to a reduction of the low energy content as well as spill of instruments like kick drum (see further regarding the reduction of low energy) and/or the emphasis of a frequency range above the original centroid.

The brightness of each track except bass guitar and kick drum (the sources with the highest amount of low energy) is increased.

For a large set of spectral features (spectral centroid, brightness, skewness, roll-off, flatness, zero-crossing, and roughness), the engineers disagree on the preferred value for all instruments except kick drum. In other words, the values describing the spectrum of a kick drum across engineers are overlapping, implying a consistent spectral target (a certain range of appropriate values). For other features

(spread, kurtosis and irregularity) the value corresponding with the kick drum track is also significantly different across engineers. The roughness shows no significantly different means for any instrument except the ‘rest’ bus.

The low energy of each track is reduced for each instrument, with significantly more reduction for snare drum than for kick drum and bass guitar. Its absolute value for bass and vocal is significantly different across engineers, whereas there is a general overlap for all other instruments including the mix. As the variation in the resulting value of low energy is higher than the variation for the unprocessed versions, no target value is apparent for any instrument, nor for the total mix.

Analysis of the octave band energies reveals definite trends across songs and mixing engineers, for a certain instrument as well as the mix. The standard deviation does not consistently decrease or increase over the octave bands for any instrument when compared to the raw audio. The suggested ‘mix target spectrum’ is in agreement with [11], which derived a ‘target spectrum’ based on average spectra of number one hits from various genres and over several decades. Figure 4 shows the measured average mix spectrum against the octave band values of the average spectrum of a number one hit after 2000 from that work, which lies within a standard deviation from our result with the exception of the highest band. The average relative change in energies is not significantly different from zero (no bands are consistently boosted or cut for certain instruments), but taking each song individually in consideration, a strong agreement of reasonably drastic boosts or cuts is shown for some songs. This confirms that the equalisation is highly dependent on the source material, and engineers largely agree on the necessary treatment for source tracks showing spectral anomalies.

## 5. CONCLUSION

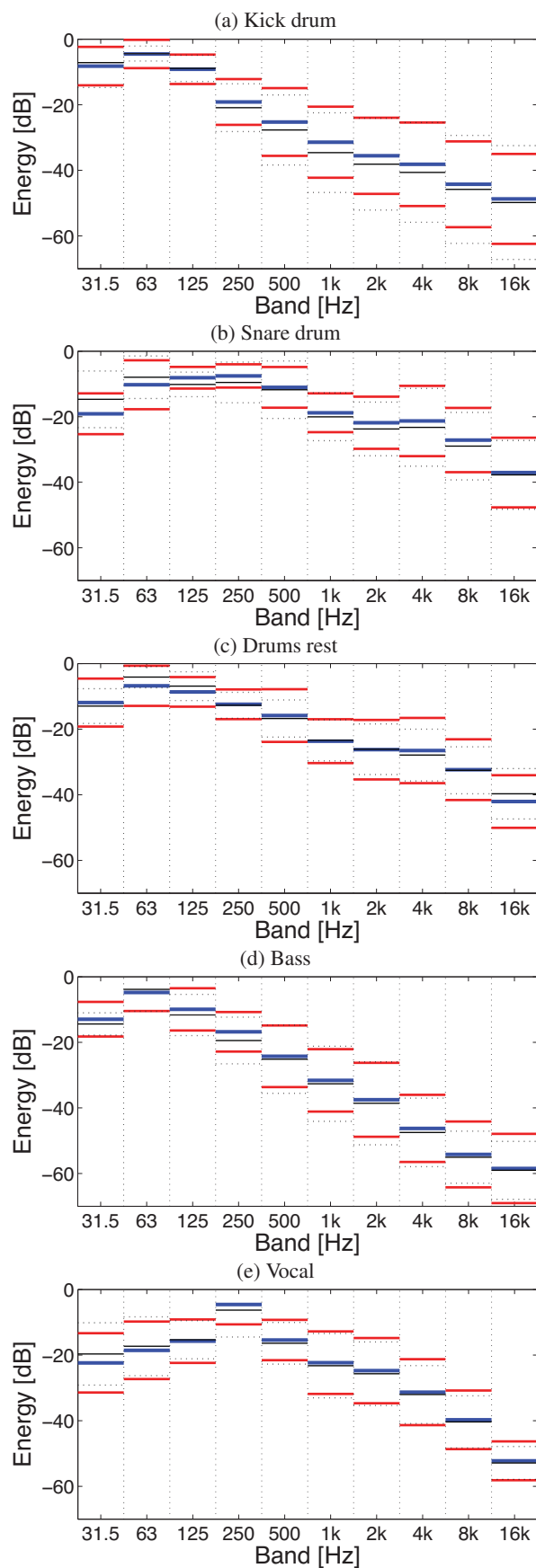
We conducted a controlled experiment where eight multitrack recordings mixed by eight mixing engineers were analysed in terms of dynamic, spatial and spectral processing of common key elements.

We measured a greater variance of features across songs than across engineers, for each considered instrument and for the total mix, whereas the mean values corresponding to the different engineers were more often statistically different from each other.

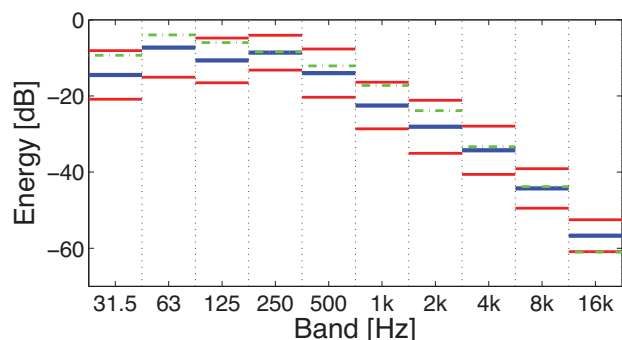
The relative loudness of the lead vocal track was found to be significantly louder than all other tracks, with an average value of  $-3$  LU relative to the total mix loudness.

The amount of panning as a function of frequency was investigated, and found to be increasing with frequency up to about 400 Hz, above which it stays more or less constant.

We measured a consistent decrease of low frequency energy and an increase of crest factor for all instruments, and an increase of the spectral centroid of the snare drum track. Spectral analysis has shown a definite target spectrum that agrees with the average spectrum of recent commercial recordings.



**Figure 3:** Average octave band energies (blue) with standard deviation (red) for different instruments after processing, compared to the raw signal (black).



**Figure 4:** Average octave band energies for total mix, compared to ‘After 2000’ curve from [11] (green dashed line)

## 6. FUTURE WORK

Future work will be concerned with perceptual evaluation of mixes and its relation to features, using both qualitative (‘which sonic descriptors correspond with which features?’) and quantitative analysis (‘which manipulation of audio is preferred?’).

Further research is needed to establish the desired loudness of sources, as opposed to loudness of tracks, and its variance throughout songs, genres, and mixing engineers.

An extrapolation of the analysis described in this paper to other instruments is needed to validate the generality of the conclusions regarding the processing of drums, bass and lead vocal at the mixing stage, and to further explore laws underpinning the processing of different instruments.

Based on the findings of this work, which showed trends and variances of different relevant features, we can inform knowledge engineered or machine learning based systems that automate certain mixing tasks (balancing, panning, equalising and compression).

This work was based on a still relatively limited set of mixes, for which the engineers came from the same institution. Through initiatives such as the public multitrack testbed presented in this paper, it will be possible to analyse larger corpora of mixes, where more parameters can be investigated with more significance.

## 7. ACKNOWLEDGEMENTS

The authors would like to thank George Fazekas for assistance with launching the multitrack testbed.

This study was funded in part by the Engineering and Physical Sciences Research Council (EPSRC) Semantic Media grant (EP/J010375/1).

## 8. REFERENCES

- [1] Alex Case. *Mix Smart: Professional Techniques for the Home Studio*. Focal Press. Taylor & Francis, 2011.
- [2] Brecht De Man and Joshua D. Reiss. A knowledge-engineered autonomous mixing system. In *135th Convention of the Audio Engineering Society*, 2013.
- [3] ITU. Recommendation ITU-R BS.1770-3 Algorithms to measure audio programme loudness and true-peak audio level. Technical report, Radiocommunication Sector of the International Telecommunication Union, 2012.
- [4] Roey Izhaki. *Mixing audio: concepts, practices and tools*. Focal Press, 2008.
- [5] Olivier Lartillot and Petri Toiviainen. MIR in Matlab (II): A toolbox for musical feature extraction from audio. In *Proceedings of the 8th International Society for Music Information Retrieval Conference*, 2007.
- [6] Stuart Mansbridge, Saoirse Finn, and Joshua D. Reiss. An autonomous system for multi-track stereo pan positioning. In *133rd Convention of the Audio Engineering Society*, 2012.
- [7] Stuart Mansbridge, Saoirse Finn, and Joshua D. Reiss. Implementation and evaluation of autonomous multi-track fader control. In *132nd Convention of the Audio Engineering Society*, 2012.
- [8] Bobby Owsinski. *The Mixing Engineer’s Handbook*. Course Technology, 2nd edition, 2006.
- [9] Enrique Perez-Gonzalez and Joshua D. Reiss. Automatic mixing: Live downmixing stereo panner. In *10th International Conference on Digital Audio Effects (DAFx-10)*, 2007.
- [10] Pedro Pestana. *Automatic mixing systems using adaptive digital audio effects*. PhD thesis, Catholic University of Portugal, 2013.
- [11] Pedro Duarte Pestana, Zheng Ma, Joshua D. Reiss, Alvaro Barbosa, and Dawn A. A. Black. Spectral characteristics of popular commercial recordings 1950-2010. In *135th Convention of the Audio Engineering Society*, 2013.
- [12] Pedro Duarte Pestana, Joshua D. Reiss, and Alvaro Barbosa. Loudness measurement of multitrack audio content using modifications of ITU-R BS.1770. In *Audio Engineering Society Convention 134*, 2013.
- [13] Jeff Scott and Youngmoo E. Kim. Analysis of acoustic features for automated multi-track mixing. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, 2011.
- [14] Jeff Scott and Youngmoo E. Kim. Instrument identification informed multi-track mixing. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, 2013.
- [15] M. Senior. *Mixing Secrets*. Taylor & Francis, 2012.
- [16] George Tzanetakis, Randy Jones, and Kirk McNally. Stereo panning features for classifying recording production style. In *Proceedings of the 8th International Society for Music Information Retrieval Conference*, 2007.
- [17] Earl Vickers. The loudness war: Background, speculation, and recommendations. *129th Convention of the Audio Engineering Society*, 2010.

# DETECTING DROPS IN ELECTRONIC DANCE MUSIC: CONTENT BASED APPROACHES TO A SOCIALLY SIGNIFICANT MUSIC EVENT

Karthik Yadati, Martha Larson, Cynthia C. S. Liem, Alan Hanjalic

Delft University of Technology

{n.k.yadati,m.a.larson,c.c.s.liem,a.hanjalic}@tudelft.nl

## ABSTRACT

Electronic dance music (EDM) is a popular genre of music. In this paper, we propose a method to automatically detect the characteristic event in an EDM recording that is referred to as a *drop*. Its importance is reflected in the number of users who leave comments in the general neighborhood of drop events in music on online audio distribution platforms like SoundCloud. The variability that characterizes realizations of drop events in EDM makes automatic drop detection challenging. We propose a two-stage approach to drop detection that first models the sound characteristics during drop events and then incorporates temporal structure by zeroing in on a watershed moment. We also explore the possibility of using the drop-related social comments on the SoundCloud platform as weak reference labels to improve drop detection. The method is evaluated using data from SoundCloud. Performance is measured as the overlap between tolerance windows centered around the hypothesized and the actual drop. Initial experimental results are promising, revealing the potential of the proposed method for combining content analysis and social activity to detect events in music recordings.

## 1. INTRODUCTION

Electronic dance music (EDM) is a popular genre of dance music which, as the name suggests, is created using electronic equipment and played in dance environments. Outside of clubs and dance festivals, EDM artists and listeners actively share music on online social platforms. Central to the enjoyment of EDM is a phenomenon referred to as “The Drop”. Within the EDM community, a *drop* is described as a moment of emotional release, where people start to dance “like crazy” [12]. There is no precise recipe for creating a drop when composing EDM. Rather, a drop occurs after a *build*, a building up of tension, and is followed by the re-introduction of the full bassline [1]. A given EDM track may contain one or more drop moments.

The designation “The Drop” is generally reserved for the overall phenomenon rather than specific drop events.

In this paper we address the challenge of automatically detecting a drop in a given EDM track. The social significance of the drop in the EDM context can be inferred, for instance, from the websites that compile a playlist of the best drops<sup>1</sup>. It is also evident from vivid social activity around drop events on online audio distribution platforms such as SoundCloud<sup>2</sup>. We also mention here a documentary, scheduled to be released in 2014, tracking the evolution of EDM as a cultural phenomenon, and titled *The Drop*<sup>3</sup>. Ultimately, the drop detection approach proposed in this paper could serve both EDM artists and listeners. For example, it would enable artists to compare drop creation techniques, and would also support listeners to better locate their favorite drop moments.

The challenge of drop detection arises from the high variability in different EDM tracks, which differ in their musical content and temporal development. Our drop detection approach uses audio content analysis and machine learning techniques to capture this variability. As an additional source of reference labels for classifier training, we explore the utility of drop-related social data in the form of *timed comments*, comments associated with specific time codes. We draw our data from SoundCloud, a music distribution platform that supports timed comments and is representative of online social sharing of EDM. The paper makes three contributions:

- We propose a two-step content-based drop detection approach.
- We verify the ability of the approach to detect *drops* in EDM tracks.
- We demonstrate utility of the social features (timed comments on SoundCloud) to reduce the amount of hand-labeled data needed to train our classifier.

The remainder of this paper is organized as follows. Section 2 discusses related work, and is followed by the presentation and evaluation of our method in sections 3 and 4. Section 5 provides a summary and an outlook towards future work.



© Karthik Yadati, Martha Larson, Cynthia C. S. Liem, Alan Hanjalic.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Karthik Yadati, Martha Larson, Cynthia C. S. Liem, Alan Hanjalic. “Detecting Drops in Electronic Dance Music: Content based approaches to a socially significant music event”, 15th International Society for Music Information Retrieval Conference, 2014.

<sup>1</sup> <http://www.beatport.com/charts/top-10-edm-drops-feb1/252641>

<sup>2</sup> <http://soundcloud.com>

<sup>3</sup> <http://www.imdb.com/title/tt2301898/>

## 2. RELATED WORK

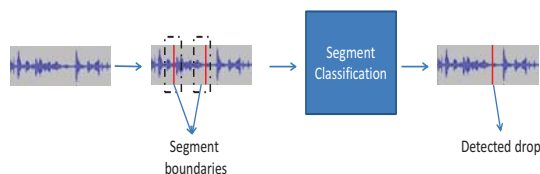
Although Electronic Dance Music is a popular music genre attracting large audiences, it has received little attention in the music information retrieval research community. Research on EDM is limited to a small number of contributions. Here, we mention the most notable. Hockman et al. [5] propose a genre-specific beat tracking system that is designed to analyze music from the following EDM sub-genres: Hardcore, Jungle, Drum and Bass. Kell et al. in [6] also apply audio content analysis to EDM in order to investigate track ordering and selection, which is usually carried out by human experts, i.e., Disc Jockeys (DJ). The work report findings on which content features influence the process of ordering and selection. A musicological perspective is offered by Collins in [3], who applies audio content analysis and machine learning techniques to empirically study the creative influence of earlier musical genres on the later ones using a date annotated database of EDM tracks, with specific focus on the sub-genres Detroit techno and Chicago house. Our work strives to redress the balance and give more attention to EDM. It draws attention to SoundCloud as an important source of music data and associated social annotations, and also to “The Drop”, a music event of key significance for the audience of EDM.

The rise of social media has also seen the rise in availability of user-contributed metadata (e.g., comments and tags). Social tags have recently grown in importance in music information retrieval research. In [11], they were used to predict perceived or induced emotional responses to music. This work reports findings on the correlation between the emotion tags associated with songs on Last.fm—“happy”, “sad”, “angry” and “relax”—and the user emotion ratings for perceived and induced emotions. Social data is generally noisy, since generating precise labels is not users’ primary motivation for tagging or commenting. However, this data can still prove useful as weak reference labels, reducing the burden of producing ground-truth labels for a large set of music tracks, which is an expensive and time consuming task. Social tags available on Last.fm have been used to automatically generating tags for songs [4]. An interesting direction of research is described in [13], where the authors use content-based analysis of the song to improve the tags provided by users. Existing work makes use of social tags that users assign to a song as a whole. In contrast, our work makes use of *timed comments* that users contribute associated with specific time points during a song.

Obtaining time-code level ground-truth labels for a large set of music tracks is an expensive and time consuming task. One way to obtain reference labels is to use crowdsourcing, where users are explicitly offered a task (e.g., label the type of emotion [9]). Our approach of using timed comments spares the expense of crowdsourcing. It has the additional advantage that users have contributed the comments spontaneously, i.e., they have not been asked to explicitly assign them, making them a more natural expression of user reactions during their listening experience.

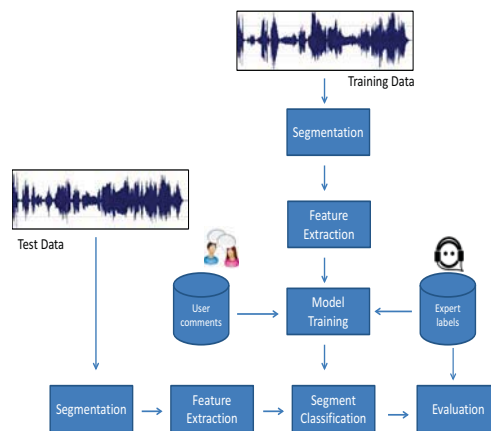
## 3. PROPOSED APPROACH

Our proposed two-step approach is based on general properties of the “The Drop”. As previously mentioned drops are characterized by a build up towards a climax followed by reintroduction of the bassline. We hypothesize that the switch will coincide with a structural segment that ends at a drop moment. For this reason, the first step in our approach is segmentation. However, not all segment boundaries are drops. For this reason, the second step in our approach is a content-based classification of segments that eliminates segments whose boundaries are not drop points. Figure 1 illustrates the two-stage approach, where we first segment to identify drop candidates and then classify in order to isolate candidates that are actually drop moments.



**Figure 1.** Two-stage approach to drop detection

The classification framework we propose to find drop events is illustrated in Figure 2. At the heart of the framework are the following modules: Segmentation, feature extraction, classification and evaluation.



**Figure 2.** The proposed classification framework

### 3.1 Segmentation

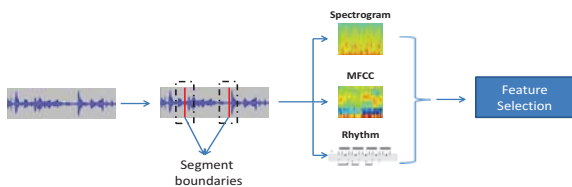
The segmentation step carries out unsupervised segment boundary detection. Exploratory experiments revealed that the segmentation method proposed in [10] gives a good first approximation of the drops in an EDM track, and we have adopted it for our experiments. The method uses the chroma features computed from the audio track to identify the segment boundaries. We use the same parameters as used in [10]: 12 pitch classes, a window length of 209 ms, and a hop size of 139 ms. We carried out an intermediate evaluation to establish the quality of the drop candidates



generated by the segmentation step alone. The average distance between the actual drop (ground-truth) and a segment boundary generated by our segmentation method is 2.5 seconds, and less than 8% of the drops are missed in our training set (described in Section 4.1).

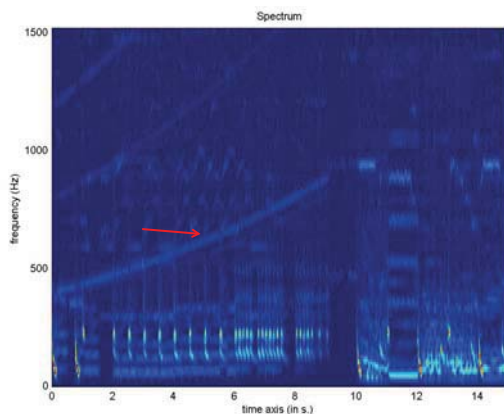
### 3.2 Feature Extraction for Classification

An overview of the feature extraction process is illustrated in Figure 3.



**Figure 3.** Feature extraction procedure

After segmentation, we extract content-based features from a fixed length window around the segment boundary. We use the following features: Spectrogram, MFCC and features related to rhythm. We adopt Mel-Frequency Cepstral Coefficients (MFCC) and features computed from the spectrogram because of their effectiveness. A unique feature of a drop is that it is preceded by a buildup or *build*. Figure 4 indicates that this buildup can be clearly observed in the spectrogram of an audio segment containing a drop. This provides additional motivation to use features computed from the spectrogram in our approach. We use the statistics computed from the spectrogram in our method (mean and standard deviations of the frequencies). For MFCC and spectrogram calculation, we use a window size of 50 msec with a 50% overlap with the subsequent windows. We use 13 coefficients for the MFCC. Due to a



**Figure 4.** Spectrogram of an audio segment indicating a build (red arrow) towards a drop at 10 seconds.

switch of rhythm at the drop moment, features related to rhythm are another important source of information. We use the rhythm related features: rhythm patterns, rhythm histogram, temporal rhythm histogram [8]. We concatenate the rhythm features, MFCC and statistics computed from the spectrogram into a single feature vector. Feature

selection, following the approach of [2], is performed on the training data in order to reduce the dimensionality of the feature vector and also to ensure that we use the most informative features in the classification step.

### 3.3 Training and Classification

To train the classifier, we assign drop (1) vs. non-drop (0) labels to time-points in the track using two sources of information: high fidelity ground-truth (manual labels provided by an expert) and user comments (weak reference labels).

Prior to training the model, we map the ground-truth labels to the nearest segment boundaries. We note that the segmentation step reduces the search space for the drop, as we no longer search for it in the entire track, but focus on features around the segment boundaries. We use a binary SVM classifier with a linear kernel as our training algorithm.

### 3.4 Evaluation

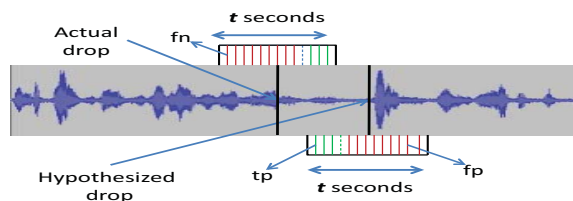
Our method predicts time points in a track at which the drop occurs. We consider each detected drop to be a distinct drop. The fact that the drop can only be hypothesized at a segment boundary keeps detections from occurring close together, given that the average length of segments generated by our segmentation algorithm is 16.5 seconds.

In order to report the performance in terms of accuracy and precision, we utilize the F1-score. Although the drop is annotated as a point in the track, it is characterized by the music around the point. This aspect of the drop motivates our choice of using a tolerance window of varying temporal resolutions around the hypothesized drop and use temporal overlap to compute the F1-score. We follow these steps to compute the F1-score:

1. Place a tolerance window of size  $t$  seconds centered around the hypothesized (from our approach) and the reference drop (ground-truth).
2. Compute the number of true positives (tp), false positives (fp) and false negatives (fn) as illustrated in Figure 5 (the unit of measurement being seconds). Note that the numbers computed here are related to the number of seconds of overlap between the windows placed over the actual drop and the predicted drop. These are computed for every detected drop in the track.
3. Compute the F1-score using the following equation: 
$$F1 = \frac{2tp}{2tp + fn + fp}.$$
4. Repeat the above steps for different sizes of  $t$ . We use window sizes of  $t = 15 \text{ sec}, 13 \text{ sec}, 11 \text{ sec}, 9 \text{ sec}, 7 \text{ sec}, 5 \text{ sec}, 3 \text{ sec}$  to compute the F1 score.
5. If there is more than one drop in the track, repeat all the above steps and compute an average F1-score for each size of the window  $t$ .

## 4. EXPERIMENTS

We have proposed a classification framework for detecting drops in an EDM track. We use MIRTtoolbox [7] to extract features related to spectrogram and MFCC, while we



**Figure 5.** Illustration to compute true positive (tp), false positive (fp) and false negative (fn) using a rectangular window of size  $t$  seconds.

use the source code provided by the authors of [8] to extract features related to rhythm. We carry out feature selection with a mechanism, adopted from [2], that uses support vector machines to identify the most informative features. For the binary classification of drop vs. non-drop, we use a support vector machine classifier provided in LibSVM. The experiments have been designed to address the two research questions of this paper:

- Can our proposed approach detect drops successfully? (Section 4.3), and
- What is the utility of the timed comments in the limited presence of explicit ground-truth data? (Section 4.4)

#### 4.1 Dataset

In order to evaluate our method, we collect music and social data from SoundCloud, which can be seen as a representative of modern online social audio distribution platforms. It allows users to upload, record and share their self-created music. One of the unique features of SoundCloud is that it allows users to comment at particular time-points in the sound. These comments are referred to as “timed comments”. Figure 6 illustrates a screenshot of the audio player on SoundCloud along with the timed comments.



**Figure 6.** Screenshot of the audio player on SoundCloud.

These comments offer a rich source of information as they are associated with a specific time-point and could indicate useful information about the sound difficult to infer from the signal. Table 1 illustrates a few example timed comments, which provide different kinds of information about the sound. These timed comments can be noisy with respect to their timestamps due to discrepancies between when users hear interesting events, and when they comment on them.

SoundCloud provides a well-documented API that can be used to build applications using SoundCloud data and information on select social features. In order to collect our dataset, we used the Python SDK to search for re-

Timestamp	Comment
01:21	Dunno what it is about this song, inspires me to make more tunes though! love it!
00:28	Love the rhythm!!
00:49	love that drop! nice bassline! nice vocals! epic!

**Table 1.** Examples of timed comments on SoundCloud.

cent sounds belonging to the following three sub-genres of EDM: *dubstep*, *electro* and *house*. Using the returned list of track identification numbers, we download the track (if its allowed by the user who uploaded the sound) and the corresponding timed comments. We then filter out the comments which do not contain the word “drop”. At the end of the data collection process, we have a set of tracks belonging to the above mentioned genres, the associated timed comments containing the word “drop”, and the corresponding timestamp of the comment. Table 2 provides some statistics of the dataset.

Genre	# files	Aver. Duration	Aver. # comments	Aver. # drop comments	Aver. # drops
Dubstep	36	4 min.	278	4	3
Electro	36	3.6 min.	220	3	3
House	28	3.9 min.	250	5	2

**Table 2.** Statistics of the dataset

As we have filtered out the non-drop comments and all the tracks in the dataset have at least one drop comment, we can assume that there is at least one drop in each track. We use a dataset of 100 tracks with a split of 60–20–20 for the training, development and testing respectively.

#### 4.2 Ground-truth annotations

As we are developing a learning framework to detect drops in an EDM track, we need reference labels for the time-points at which drops occur in our dataset, as mentioned previously. We utilize two sources of information: explicit ground-truth (high fidelity labels) and implicit ground-truth (user comments). In order to obtain high fidelity drop labels, one of the authors has listened to the 100 tracks and manually marked the drop points. The labeled points refer to the point where the buildup ends and the bassline is re-introduced. Instead of listening to the entire track, the author skips 30 seconds after he hears a drop as it is highly unlikely that a second drop would occur within 30 seconds. It took approximately 6 hours for the author to label the entire dataset. When computing F1-score in the experiments, we use the manual labels as ground-truth.

Explicit ground-truth labels are expensive as creating them requires experts to spend time and effort to listen to the tracks and mark the drop points. Relying on explicit ground-truth data also hampers the scalability of the dataset, as it would require much more time and effort from the annotators for a larger dataset. Keeping with the social nature of SoundCloud, users contribute comments, some which remark on the pretense or quality of a drop (Table 1). We investigate the possibility of using these timed comments as weak reference labels in predicting the drop. We refer to timed comments as *weak* reference labels owing to their noisy nature. For example, only 20 % of the drop comments in the training set are located at the actual drop in a track. Note that we treat each comment as a distinct

drop. We have a total of 190 drops and 225 drop comments in our dataset. As we can see, there are more comments than the actual drops. Mapping multiple drop comments, which are nearer to each other, to a single time point is a consideration for the future.

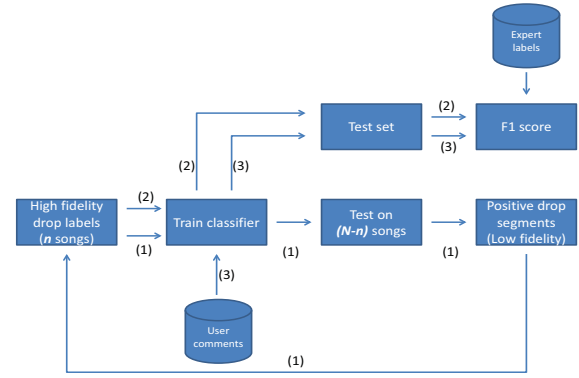
### 4.3 Detecting drop using content-based features

In this experiment, we evaluate the performance of the content based features in detecting a drop using the explicit ground-truth labels. We compute the F1-score for each track separately. The F1-score is averaged if there is more than one drop in the track. In Table 3, we report three results: (1) F1-score, averaged across the entire dataset; (2) Highest F1-score for a single track and (3) Lowest F1-score for a single track. As mentioned before, we use windows of sizes  $t = 3, 5, 7, 9, 11, 13, 15 \text{ sec}$ . The size of the window ( $t$ ) represents the temporal precision to which the F1-score is reported. Observing the results for the average performance (first row of Table 3), we achieve a maximum F1 score of 0.71 for a 15 second tolerance window. However, we already achieve an F1 score greater than 0.6 for a tolerance window as small as 3 seconds. The second row of Table 3 illustrates the F1 scores for one single track which has the best drop detection and we observe that the F1 scores are high and go up to 0.96 for a 15 second tolerance window. The third row of Table 3 illustrates the F1 scores for one single track which has the worst drop detection and we observe that the F1 scores are very low, as it has more false positives. Moreover, the structure segment boundaries do not capture the drops particularly well in this track.

### 4.4 Utility of timed comments

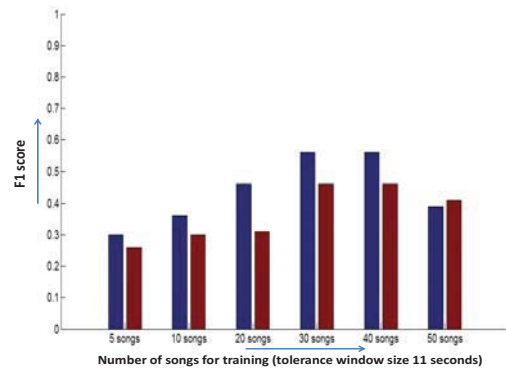
Timed comments are an important source of information as they could indicate the time point where a drop occurs. Figure 7 illustrates a pipeline for the experiment to assess the utility of timed comments as weak reference labels. It is carried out in three stages labeled as (1), (2) and (3) in the figure. The stages are explained here. We divide the complete training set of  $N$  tracks into two mutually exclusive sets of  $n$  and  $N - n$  tracks. Assuming that the  $n$  tracks have ground-truth labels, we train a model (1) and use it to classify the unlabeled segment boundaries from the  $N - n$  tracks. We segment boundaries labeled positive by the classifier, which will be of low fidelity, and add them to the training data. In the second stage (2), we use the expanded training data ( $n$  tracks + low fidelity positive segment boundaries) to predict the drop segments in the test set and compute the F1 score for evaluation. Then, the features computed from a window sampled around user drop comments are added to the training data. The data now includes features from the  $n$  tracks, and low fidelity predicted positive segment boundaries, and around sampled at user comments. We use this data to train a model (3) and use it to predict the drop segments in the test set and compute the F1 score for evaluation.

In this experiment, we use the following training data sizes which are expressed in terms of the number of tracks:



**Figure 7.** Procedure to assess the utility of timed comments in detecting drop.

$n = 5, 10, 20, 30, 40, 50$ . F1 scores over different window sizes are computed to demonstrate the drop detection performance. Figure 8 illustrates the performance of the binary classifier when we have increasing sizes of training data. Due to space constraints, we illustrate the results only for one size of the tolerance window: 11 seconds. Difference in F1 scores when we add user comments is visualized in Figure 8. Inspecting the figure, we can say that



**Figure 8.** F1 scores for combining high fidelity ground-truth labels and user comments for a tolerance window size of 11 seconds and different training set sizes: 5 tracks, 10 tracks, 20 tracks, 30 tracks, 40 tracks, 50 tracks. First bar in each group indicates the results of stage (3) of the experiment and the second bar indicates the F1 score for the stage (2) of the experiment

reasonable F1 scores are obtained when we use  $n = 30$  and  $n = 40$  tracks as training set and a tolerance window size of 11 seconds. We observe that the F1 scores are lower than with explicit ground-truth annotations, which we attribute to the noise of user comments.

## 5. CONCLUSION AND OUTLOOK

We have proposed and evaluated content-based approach that detects an important music event in EDM referred to as a drop. To this end, we have made use of music and user-contributed timed-comments from an online social audio

	3 sec	5 sec	7 sec	9 sec	11 sec	13 sec	15 sec
Average Performance	0.61	0.62	0.66	0.66	0.68	0.69	0.71
Track with Best Performance	0.83	0.9	0.92	0.94	0.95	0.96	0.96
Track with Worst Performance	0.2	0.36	0.43	0.47	0.49	0.51	0.52

**Table 3.** Experimental results indicating the average, best and worst F1 scores for increasing window sizes

distribution platform: SoundCloud. We reported performance in terms of F1, using a tolerance window of varying time resolutions around the reference drop time-points and the drop time-points hypothesized by our approach. With a tolerance window of 5 seconds, which we estimate to be an acceptable size to listeners, we obtain an F1 score greater than 0.6. “Timed-comments”, contributed by users in association with specific time-codes were demonstrated to be useful as weak labels to supplement hand-labeled reference data. We achieved a reasonable accuracy using a standard set of music related features. One of the future steps would be to come up with a set of features which can model the variability and the temporal structure during drop events, which will in turn improve the accuracy. We concentrated on a subset of genres: dubstep, electro and house in this paper as these were the more popular genres on SoundCloud (in terms of number of comments). An immediate direction would be to expand the current dataset by including various sub-genres of EDM, e.g., techno and drum & bass.

Our work demonstrates that musical events in popular electronic music can be successfully analyzed with the help of time-level social comments contributed by users in online social sharing platforms. This approach to music event detection opens up new vistas for future research. Our next step is to carry out a user study with our drop detector aimed at discovering exactly how it can be of use to EDM artists and listeners. Such a study could also reveal the source of “noise” in the timed comments, allowing us to understand why users often comment about drops in neighborhoods far from where an actual drop has occurred. This information could in-turn allow us to identify the most useful drop comments to add to our training data. Further, we wish to widen our exploration of information sources that could possibly support drop detection to also include MIDI files that are posted by users online together with the audio. Currently, the availability of these files is limited, but we anticipate that they might be helpful for bootstrapping. Another source of information is a crowdsourcing, which could be used to identify drops directly, or to filter comments directly related to the drop, from less-closely related or unrelated comments.

## 6. ACKNOWLEDGEMENT

This research is supported by funding from the European Commission’s 7th Framework Program under grant agreement no. 610594 (CrowdRec) and 601166 (PHENICX).

## 7. REFERENCES

- [1] M.J. Butler. *Unlocking the Groove: Rhythm, Meter, and Musical Design in Electronic Dance Music*. Profiles in popular music. Indiana University Press, 2006.
- [2] Yi-Wei Chen and Chih-Jen Lin. Combining SVMs with various feature selection strategies. In *Feature Extraction*, volume 207 of *Studies in Fuzziness and Soft Computing*, pages 315–324. Springer Berlin Heidelberg, 2006.
- [3] Nick Collins. Influence in early electronic dance music: An audio content analysis investigation. In *ISMIR*, pages 1–6, 2012.
- [4] Douglas Eck, Paul Lamere, Thierry Bertin-Mahieux, and Stephen Green. Automatic generation of social tags for music recommendation. In *NIPS*, 2007.
- [5] Jason Hockman, Matthew E. P. Davies, and Ichiro Fujinaga. One in the jungle: Downbeat detection in hardcore, jungle, and drum and bass. In *ISMIR*, pages 169–174, 2012.
- [6] Thor Kell and George Tzanetakis. Empirical analysis of track selection and ordering in electronic dance music using audio feature extraction. In *ISMIR*, pages 505–510, 2013.
- [7] Olivier Lartillot and Petri Toivianen. Mir in matlab (ii): A toolbox for musical feature extraction from audio. In *ISMIR*, pages 127–130, 2007.
- [8] Thomas Lidy and Andreas Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *ISMIR*, pages 34–41, 2005.
- [9] Erik M. Schmidt and Youngmoo E. Kim. Modeling musical emotion dynamics with conditional random fields. In *ISMIR*, pages 777–782, 2011.
- [10] Joan Serrà, Meinard Müller, Peter Grosche, and Josep LLuis Arcos. Unsupervised music structure annotation by time series structure features and segment similarity. *IEEE Transactions on Multimedia*, PP(99):1–1, 2014.
- [11] Yading Song, Simon Dixon, Marcus Pearce, and Andrea R. Halpern. Do online social tags predict perceived or induced emotional responses to music? In *ISMIR*, pages 89–94, 2013.
- [12] John Steventon. *DJing for Dummies*. –For dummies. Wiley, 2007.
- [13] Yi-Hsuan Yang, Dmitry Bogdanov, Perfecto Herrera, and Mohamed Sordo. Music retagging using label propagation and robust principal component analysis. In *WWW*, pages 869–876, 2012.

# TOWARDS AUTOMATIC CONTENT-BASED SEPARATION OF DJ MIXES INTO SINGLE TRACKS

**Nikolay Glazyrin**  
Ural Federal University  
nglazyrin@gmail.com

## ABSTRACT

DJ mixes and radio show recordings constitute an important and underexploited music and data source. In this paper we try to approach the problem of separation of a continuous DJ mix into single tracks or timestamping a mix. Sharing some aspects with the task of structural segmentation, this problem has a number of distinctive features that make difficulties for structural segmentation algorithms designed to work with a single track. We use the information derived from spectrum data to separate tracks from each other. We show that the metadata that usually comes with DJ mixes can be exploited to improve the separation. An iterative algorithm that can consider both content-based data and user provided metadata is proposed and evaluated on a collection of freely available timestamped DJ mix recordings of various styles.

## 1. INTRODUCTION

DJ mixes provide a great source of music data, which does not gain much attention from the MIR community yet. The work by Kell and Tzanetakis [6], which gives an analysis of track selection and ordering in DJ mixes is one of the few exceptions.

Besides playing in clubs many DJs nowadays produce weekly radio shows with latest and greatest and sometimes exclusive tracks. These shows are often freely available through the internet and are very popular among electronic music lovers. Tracklists for the shows are often provided by DJs themselves or by their fans.

For many people it is important to know which track is playing now. The cue sheet file format [2] suits perfectly to carry this kind of information. It was designed to describe how the tracks on CD are laid out, but later it was supported by many audio players and CD burning software. There are communities, such as <http://cuenation.com> or <http://themixingbowl.org>, which bring together the people who create cue sheets for DJ mixes and radio shows. But the wiki page [1] on the first site says nothing about any tools for automatical or semi-automatical generation of cue sheets.

The most time consuming part of this process is finding the moments when one track gives place to another. This may be a big problem for an untrained listener, because making smooth transitions between tracks is one of the skills every DJ should have. For a trained person it is not so hard, but to find a precise position of a transition one has to listen carefully through dozens of seconds of the audio. A tool that can propose most probable transition positions can facilitate this task. Such a tool can also be used by DJs who upload their mixes to special sharing services or online radio stations. These services will be able to timestamp the mix automatically instead of forcing the uploader to do this. The timestamps may be then used to provide fast access to particular tracks within the mix and to easily share previews of unreleased tracks played in radio shows. Timestamped recordings of DJ mixes can be used by recommendation systems to calculate content-based features and relate them to sequential tracks.

The task of DJ mix separation is essentially the task of audio segmentation, so the concepts and approaches can be shared between these tasks. But some conditions and requirements make them different. These differences will be discussed in section 2. In section 3 we describe the proposed method to separate tracks in DJ mix recordings. In section 4 we describe the experiments and the evaluation methodology. Finally, in section 5 we conclude and formulate open problems and directions for future work.

## 2. PROBLEM FORMULATION AND RELATED WORK

Music structural segmentation is a very popular and elaborated task. Paulus et al. in [9] distinguish three different classes of music segmentation methods. Repetition-based methods try to identify recurring patterns. Novelty-based methods try to find transitions between contrasting parts. Homogeneity-based methods, contrary to novelty-based ones, try to determine fragments that are consistent with respect to some characteristic. Combined methods have also been proposed. Some recent ones try to combine novelty-based and homogeneity-based approaches [4] or combine novelty-based approach with harmonical information in a joint probabilistic model [10].

A DJ mix can be viewed as a very long composition of individual tracks. These tracks constitute the segments in our task. It is important that no track can occur more than once within a typical mix. So repetition-based methods are



© Nikolay Glazyrin.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Nikolay Glazyrin. "Towards Automatic Content-Based Separation of DJ Mixes into Single Tracks", 15th International Society for Music Information Retrieval Conference, 2014.

not suitable at the level of tracks.

Novelty-based approach seems to be the most suitable for track boundaries detection. Algorithms that implement this approach generally have 2 main steps: segmentation and grouping.

Segmentation is usually done using an intermediate representation in the form of self-similarity matrix (or self-distance matrix). Since the original audio is not very informative, it needs to be transformed into a sequence of feature vectors, for which this matrix is calculated. The list of features often used for this includes MFCCs, constant-Q spectrum, various low-level spectrum features, such as spectral centroid, spectral spread and others.

The most popular method of obtaining initial segmentation from a self-similarity matrix was proposed by Foote [3]. It is based on so called checkerboard novelty kernels, which are essentially an  $M \times M$  matrix with checkerboard-like structure. Novelty estimations can be obtained by convolving this kernel along the main diagonal of the self-similarity matrix. Peaks of the resulting novelty function provide the initial segment borders.

Homogeneity-based methods come up as a direct continuation of this novelty-based segmentation. They group similar segments together. A good review of the whole variety of methods can be found in [9]. Many of them perform clustering of segments, e.g. [7], [5]. Any information about the desired result can be helpful at this stage to build the most effective grouping procedure.

In case of DJ mix separation the grouping procedure becomes especially important. It is quite common for dance compositions to have a so called “break” in the middle, where the sound can change dramatically. Such breaks should be overcome to properly detect track boundaries. At the same time, two adjacent segments that belong to different tracks should not be joined.

A typical DJ mix lasts considerably longer than a typical musical composition. So the method must be able to work with recordings that span hours of audio. On the other hand, this loosens the requirements to border detection: an error of seconds or sometimes even tens of seconds can be acceptable. Even humans can have different opinions about one exact moment when a track has transitioned to the next one. An interesting task of detecting transition periods (where two or more tracks are playing simultaneously) comes up here, but we don’t consider it in this paper. Marolt in [8] works with similar time scale and boundaries requirements, but with a limited set of possible segment types that sound quite differently.

Transitions can vary significantly for different music styles. It is more likely to find sharp cuts in drum’n’bass mixes, than in deep house mixes, which tend to have long gradual transitions. Average track length is also dependent on music style. But these are generally not the strict rules.

Radio shows often have an intro, which is played in the beginning and often becomes a part of the tracklist. Jingles, interludes or talks where the music gets faded can occur at random places within a recording. But it is not required to discriminate them, as they usually don’t get in-

cluded into tracklists.

The existence of tracklists also makes a great difference from structural segmentation task. It can be seen from the potential applications described in section 1, that the separation of a DJ mix is not much valuable per se. But it becomes really useful when it can be connected with metadata: artist name and track title. Because this metadata is often available, it can also be used in the algorithm. For example, the information about the number of segments in the separation gives a barrier to segmentation and/or grouping process. And if a large music base is available to the algorithm, parts of a mix can be matched to corresponding music recordings to provide even better estimation of track borders.

There may be the cases where matching is not possible though. Sometimes DJs play tracks that are not yet released officially, and therefore cannot appear in any catalogue or database. Some tracks never get released officially. Some tracks have been released years ago, and it’s almost impossible to obtain rights on them or find them in any database. That is why the development of the informed automatic DJ mix separation system cannot be reduced to a number of calls to track identification software.

Therefore, further we will suppose that there is a tracklist available for a mix recording, but not the timestamps, and no identification software is available. And the task will be to determine those timestamps based on the audio data and the information from the tracklist, or to align the mix tracklist to the audio. The authors are not informed about any works on this task existing at the moment.

### 3. SYSTEM DESCRIPTION

We adopt the approach based on novelty-based segmentation followed by grouping of similar segments.

#### 3.1 Features

Constant-Q log-spectrograms are calculated at first for audio recordings, which sampling frequency has been left at the default value of 44100 Hz. We used Constant Q plugin from Queen Mary vamp plugin set<sup>1</sup> with the following parameters: step size and block size are both equal to 16384 samples (0.37 s), 12 components per octave, spanning MIDI pitches from 36 to 84 (65 to 936 Hz) with tuning frequency of 440 Hz. A relatively large block size and zero overlap have been chosen because of the large time scale and to speed up computations. Low frequencies are captured, because electronic dance music often has very accented bass that changes from track to track. The upper frequency limit has been chosen rather arbitrarily, and we do not investigate its influence in this study.

A sliding 2D median filter is then applied to spectrogram with window size (31, 1) (which corresponds to 11.5 seconds and 1 spectral component) to smooth it.

<sup>1</sup> <http://www.vamp-plugins.org/plugin-doc/qm-vamp-plugins.html>

### 3.2 Segmentation

To accelerate calculations, the self-distance matrix is calculated for a spectrogram with 10 times less resolution by time axis (3.7 s per column), where each 10 sequential columns of original spectrogram are replaced with their average. We also restricted it to only include cosine distances between segments which are no more than 10 minutes apart from each other, because it is very unlikely to meet a track that lasts longer than that in a DJ mix.

Novelty score is then calculated from the self-distance matrix using the checkerboard kernels with gaussian taper proposed in [3]. We used relatively small kernels of size 16 (composed of 4 squares of size  $8 \times 8$ ). All the peaks of the resulting novelty function form the initial set of borders.

### 3.3 Clustering

Here we find a use for the information from the mix tracklist. The total number of tracks provides the desired number of clusters. This is an important advantage over the traditional segmentation task, where the number of segments is unknown. On the other hand, there is a very strong requirement to the borders between segments. If one true border is not detected or one false border is detected in the beginning of the mix, all the subsequent tracks become misaligned with the real audio, even if all the other borders are detected perfectly.

Another piece of information from the tracklist that can be used here is the presence of intro and outro. Many radioshows and regular podcasts have such an intro, fewer ones have also an outro. These segments are relatively short (shorter than 1 minute), but are often included in tracklists. A reasonable assumption is that if the name of the first track contains the string *intro* and/or the name of the last track contains the string *outro*, then an intro and/or an outro should be expected. A good clustering algorithm could be able to detect them automatically, but we add a special handling for these cases. If an intro is expected, among the novelty function peaks during the first 60 seconds of audio the highest one is selected and declared as the intro right border. The same is done at the end of the recording if an outro is expected there.

For the remainder of the recording an iterative clustering procedure is applied. Within each segment the average of all its feature vectors is calculated and normalized by dividing all its components by the maximal one. All the pairwise distances between segments whose beginnings are not more than 600 seconds away from each other are calculated as Euclidean distances between their average feature vectors. This gives a Segment Distance Matrix similar to the one introduced in [4].

All the segment pairs  $((l_i, r_i), (l_j, r_j))$ ,  $i < j$  (where  $l_i$  and  $r_i$  are correspondingly segment's left and right borders) for which the distance was calculated are sorted according to the following condition:  $D_{ij} \cdot (r_j - l_i)$ , where  $D_{ij}$  is the distance between  $i$ -th and  $j$ -th segments. Only the pair that produces the smallest value is then merged. If the segments from this pair are not contiguous, all the intermediate ones are also included. To avoid too big segments, a pair gets a

penalty when  $r_j - l_i > 1.25 \cdot \text{average\_track\_length}$ : its condition becomes  $100000 \cdot (r_j - l_i)$ .

## 4. RESULTS

The proposed method was evaluated on a collection of 103 DJ mix recordings<sup>2</sup> downloaded from free online sources. The corresponding timestamped tracklists in the form of .cue files were downloaded from <http://cuenation.com> and used without any corrections. Timestamps have only been used to validate the correctness of track separation. All recordings were taken from different radio shows and live sessions of different disk jockeys. Most of recordings are dated 2014, but there were also recordings from 2007-2013. The dominant music style within the selected recordings is trance (uplifting, progressive, big room, psychedelic), probably due to overall popularity of DJs playing this music. But house, drum'n'bass, breakbeat, techno, hardstyle, downtempo mixes are also included.

For the reasons described in section 3.3 we pay less attention to the conventional precision and recall metrics. Instead, two values have been calculated for each mix: the average and the maximum absolute distances in seconds from true track beginnings to detected ones. This way we can evaluate the usefulness of the method in real life applications: if the average absolute distance approaches the average track length within a mix, the method becomes nearly useless for this mix. The maximum absolute distance gives an estimation of the worst case. These values are then averaged across the whole collection to give an integral measure of method performance.

Frame-based pairwise precision, recall and F-measure have also been calculated to provide more traditional estimation of segmentation quality. They are defined as follows. Each recording is separated into 1 second frames. All frame pairs where both frames belong to the same track form the sets  $P_E$  (for the system result) and  $P_A$  (for the ground truth). The *pairwise precision rate* can be calculated by  $P = \frac{|P_E \cap P_A|}{|P_E|}$ , *pairwise recall rate* by  $R = \frac{|P_E \cap P_A|}{|P_A|}$ , and *pairwise F-measure* by  $F = \frac{2PR}{P+R}$ . These values are then also averaged across the collection.

As a baseline we will use the same values calculated for the naive separation, where all track borders are evenly spaced within the mix and all tracks have the same duration. In case of explicit intro/outro information the naive separation will allocate them 30 seconds in the beginning or in the end of the mix.

In the first experiment<sup>3</sup> the system was not informed about the presence of intro and outro sections in the mixes. The results are shown in Table 1. The "Good" column shows the number of mixes where the average absolute distance is less than 90 seconds (rather arbitrary limit). From the numbers in this table it seems that the proposed method performs not much better than the naive separation, which

<sup>2</sup> The list of file names is available from [https://github.com/nglazyrin/MixSplitter/blob/master/mix\\_list.txt](https://github.com/nglazyrin/MixSplitter/blob/master/mix_list.txt)

<sup>3</sup> Full log is available from [https://github.com/nglazyrin/MixSplitter/blob/master/logs/paper\\_test.log](https://github.com/nglazyrin/MixSplitter/blob/master/logs/paper_test.log)

Separation	CAvg. abs. dist.	CAvg. max dist.	Good
Proposed	143.73 s	328.99 s	42
Baseline	152.83 s	318.35 s	30

**Table 1.** Results with no information about intro and outro.

Separation	CAvg. abs. dist.	CAvg. max dist.	Good
Proposed	111.82 s	286.61 s	62
Baseline	126.87 s	284.41 s	49

**Table 2.** Results with information about intro and outro.

is confirmed by p-value of 0.096 returned by Wilcoxon test. But looking closer at the performance on particular mixes, we can see that in some cases the proposed method has real advantage. E.g. for the mix *M.PRAVDA - Best of 2013 (Part 2) (promodj.com).mp3* it gives average absolute distance of 8.59 s (which is great) versus 60.22 s obtained by the naive separation. On the other hand, for some mixes (e.g. *Trancecoda Podcast 008 - GMix Eddie Bitar.mp3*) the average absolute distance exceeds 6 minutes, which is absolutely unacceptable.

In the second experiment<sup>4</sup> the system was informed about the presence of intro and outro sections and could react appropriately. From the Table 2 we can see that this information can be really helpful. In this experiment the  $p < 0.01$  was returned by Wilcoxon test. The result has moved nearer to the “Good” limit of 90 seconds average difference, and the difference between the proposed and the baseline methods became bigger. And if the limit of “goodness” has decreased to 60 seconds, the difference gets more explicit: 54 good separations by the proposed method versus 24 good naive separations. For 30 seconds limit on average absolute difference only 25 versus 6 good separations are left.

This result shows that the proposed method can give good result for a reasonable amount of mixes (62 out of 103 here). But for some mixes the results are still too bad. We provide two case-studies that describe common errors of the method.

Table 3 shows the comparison of true and detected borders for one of the mixes – *4H\_Community\_Guest Mix\_The\_2nd\_Anniversary\_of\_Room51\_Show\_by\_Breeze\_Quadrat\_PureFM.mp3* – with average absolute difference of 177.11 seconds. First 3 tracks are aligned good, but then the system detects wrong border in the middle of 4th track. In spite of more or less properly detected other borders (the detected value in row  $i + 1$  is near the true value in row  $i$ ), they all mark beginnings of track  $i + 1$  instead of  $i$ -th track.

The same information is represented graphically on Figure 1. Vertical yellow lines on the constant Q spectrogram mark the true borders, vertical black lines correspond to detected borders.

The errors of this kind can be overcome with a better

<sup>4</sup>Full log is available from [https://github.com/nlazyrin/MixSplitter/blob/master/logs/paper\\_test\\_explicit\\_intro\\_outro.log](https://github.com/nlazyrin/MixSplitter/blob/master/logs/paper_test_explicit_intro_outro.log)

No.	Detected	True	Difference
1	0.00 s	0.00 s	0.00 s
2	308.38 s	312.08 s	3.70 s
3	628.57 s	613.00 s	-15.56 s
4	872.56 s	1029.11 s	156.55 s
5	1025.19 s	1363.48 s	338.29 s
6	1360.54 s	1757.29 s	396.75 s
7	1757.15 s	1961.62 s	204.47 s
8	1970.53 s	2292.27 s	321.74 s
9	2321.36 s	2552.34 s	230.98 s
10	2748.30 s	2979.58 s	231.28 s
11	3247.66 s	3198.78 s	-48.88 s

**Table 3.** Detailed result for the mix by 4H Community.

Separation	Precision	Recall	F-measure
Proposed (1)	0.8145	0.7761	0.7941
Baseline (1)	0.7024	0.6397	0.6688
Proposed (2)	0.8077	0.7892	0.7977
Baseline (2)	0.7069	0.6637	0.6839

**Table 4.** Framewise precision, recall and F-measure.

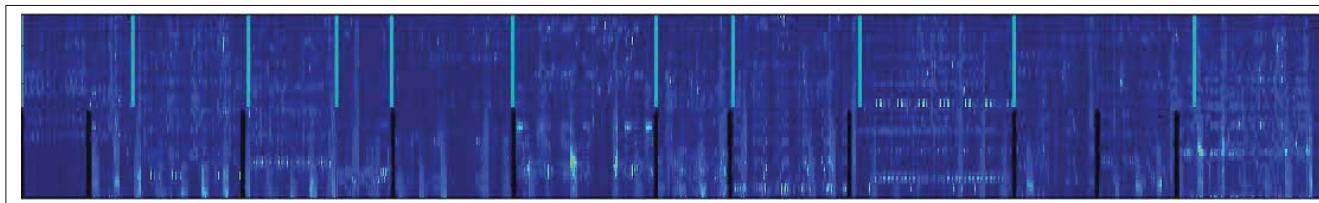
sorting function for segment pairs or with a different segment grouping strategy. As can be seen from Table 4 (the number in parentheses in the first column corresponds to the experiment number), the proposed method really locates borders much better than the baseline. But since some borders are misplaced, the final pairwise precision and recall rates are not so close to 1 as they could be.

Another source of errors are mixes that contain tracks of various durations, e.g. a pile of 1 minute long tracks followed by 4 minute long tracks, or several interludes throughout the recording. An example of such mix is *01-friction.-bbc\_radio1\_(chase\_and\_status\_special)-sat-10-13-2013-talion.mp3*, which contains 35 tracks per 2 hours, and 6 of them are grouped between 55 and 65 minutes. The separation is shown on the Figure 2. The described method tends to join short segments and to return more or less evenly spaced track borders because of the sorting condition and the penalty for long tracks. So it does not fit to these highly-variable mixes, which are characteristic for music genres such as drum’n’bass. But the separations obtained without using the penalty were worse than the ones obtained by the baseline method.

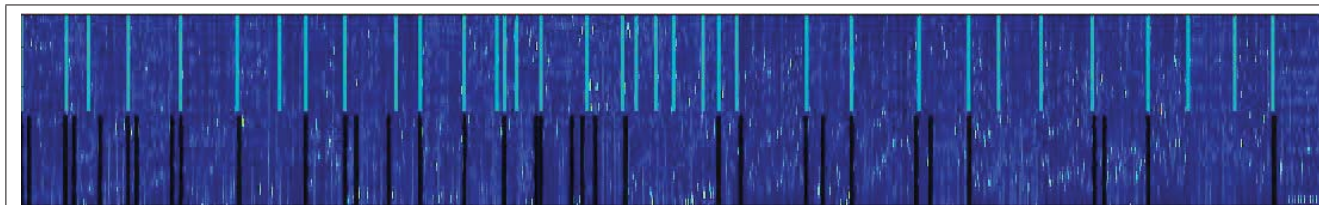
Table 5 groups the results by music genres, which were manually annotated for each mix. The mixes labeled as having *various* genre contain tracks from two or more very different genres, such as house and drum’n’bass. The Cnt column gives the total count of mixes of a given genre within our test collection.

Because the test set is very unbalanced by music genre (which is dictated by the available cue sheet files), it’s hard to make conclusions for music genres other than house and trance (which can be themselves separated into various subgenres). The proposed system outperforms the baseline method on these genres, but both methods are failing on





**Figure 1.** The separation for the mix by 4H Community.



**Figure 2.** The separation for the mix by Chase & Status.

Style	Cnt	Separation	Abs. dist.	Max dist.
trance	59	Proposed	91.43 s	244.26 s
		Baseline	114.85 s	255.38 s
house	29	Proposed	106.55 s	280.78 s
		Baseline	117.88 s	270.36 s
techno	4	Proposed	122.11 s	284.51 s
		Baseline	104.61 s	219.00 s
downtempo	3	Proposed	304.59 s	702.77 s
		Baseline	308.58 s	609.34 s
hardstyle	2	Proposed	81.91 s	232.66 s
		Baseline	93.22 s	221.95 s
drum'n'bass	2	Proposed	330.67 s	798.21 s
		Baseline	343.03 s	865.92 s
various	2	Proposed	211.28 s	429.65 s
		Baseline	203.85 s	407.48 s
breakbeat	2	Proposed	191.88 s	399.71 s
		Baseline	124.22 s	346.01 s

**Table 5.** Results by music genre.

downtempo and drum'n'bass music.

## 5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a method for informed content-based separation of DJ mixes into single tracks that outperforms a naive baseline evenly separating method. We showed that this method provides good results for a reasonable amount of mixes. The resulting separations are good enough to use them for further applications. We also showed how a simple information about the presence of intro and outro sections in the mix can improve the separation quality.

This paper establishes a basis for further work on DJ mixes separation. Another clustering methods need to be developed to prevent false border detection errors and border miss errors. It makes sense also to include higher frequencies into the initial spectrum, as they may carry some

meaningful details. On the other hand, the novelty detection method does not seem to have a major impact, because the initial border candidate set is sufficiently large to select values nearby the true borders.

More feature types need to be exploited. It also makes sense to consider the tempo information to avoid false border detections, because the tempo does not change often during transitions, but changes within a track when a break starts or ends. A deeper modification or a new method is needed to handle mixes that contain tracks with highly-varying durations. A separate method to detect interludes and talks can be helpful here.

Finally, a significant improvement may be expected from the usage of a track identification system, as it may help to align at least some of the tracks properly. But this poses a separate technical and legal task.

## 6. REFERENCES

- [1] Online: [http://wiki.themixingbowl.org/Cue\\_sheet](http://wiki.themixingbowl.org/Cue_sheet), accessed on May 5, 2014.
- [2] Online: [http://wiki.hydrogenaudio.org/index.php?title=Cue\\_sheet](http://wiki.hydrogenaudio.org/index.php?title=Cue_sheet), accessed on May 5, 2014.
- [3] J. Foote: "Automatic audio segmentation using a measure of audio novelty" *Proceedings of IEEE International Conference on Multimedia and Expo*, Vol. 1, pp. 452–455, 2000.
- [4] F. Kaiser, and G. Peeters: "A simple fusion method of state and sequence segmentation for music structure discovery" *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pp. 257–262, 2013.
- [5] F. Kaiser, and T. Sikora: "Music Structure Discovery in Popular Music using Non-negative Matrix Factorization" *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pp. 429–434, 2010.

- [6] T. Kell and G. Tzanetakis: “Empirical analysis of track selection and ordering in electronic dance music using audio feature extraction” *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pp. 505-510, 2013.
- [7] M. Levy, M. Sandler, and M. Casey: “Extraction of High-Level Musical Structure From Audio Data and Its Application to Thumbnail Generation” *Proceedings of the International Conference on Acoustics, Speech and Signal Processing 2006*, Vol. 5, 2006.
- [8] M. Marolt: “Probabilistic Segmentation and Labeling of Ethnomusicological Field Recordings” *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pp. 75–80, 2009.
- [9] J. Paulus, M. Müller, and A. Klapuri: “Audio-based Music Structure Analysis” *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, pp. 625–636, 2010.
- [10] J. Pauwels, F. Kaiser, and G. Peeters: “Combining harmony-based and novelty-based approaches for structural segmentation” *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pp. 601-606, 2013.

# ***MedleyDB*: A MULTITRACK DATASET FOR ANNOTATION-INTENSIVE MIR RESEARCH**

Rachel Bittner<sup>1</sup>, Justin Salamon<sup>1,2</sup>, Mike Tierney<sup>1</sup>, Matthias Mauch<sup>3</sup>, Chris Cannam<sup>3</sup>, Juan Bello<sup>1</sup>

<sup>1</sup>Music and Audio Research Lab, New York University

<sup>2</sup>Center for Urban Science and Progress, New York University

<sup>3</sup>Centre for Digital Music, Queen Mary University of London

{rachel.bittner, justin.salamon, mt2568, jpbello}@nyu.edu {m.mauch, chris.cannam}@eecs.qmul.ac.uk

## ABSTRACT

We introduce *MedleyDB*: a dataset of annotated, royalty-free multitrack recordings. The dataset was primarily developed to support research on melody extraction, addressing important shortcomings of existing collections. For each song we provide melody  $f_0$  annotations as well as instrument activations for evaluating automatic instrument recognition. The dataset is also useful for research on tasks that require access to the individual tracks of a song such as source separation and automatic mixing. In this paper we provide a detailed description of *MedleyDB*, including curation, annotation, and musical content. To gain insight into the new challenges presented by the dataset, we run a set of experiments using a state-of-the-art melody extraction algorithm and discuss the results. The dataset is shown to be considerably more challenging than the current test sets used in the MIREX evaluation campaign, thus opening new research avenues in melody extraction research.

## 1. INTRODUCTION

Music Information Retrieval (MIR) relies heavily on the availability of annotated datasets for training and evaluating algorithms. Despite efforts to crowd-source annotations [9], most annotated datasets available for MIR research are still the result of a manual annotation effort by a specific researcher or group. Consequently, the size of the datasets available for a particular MIR task is often directly related to the amount of effort involved in producing the annotations.

Some tasks, such as cover song identification or music recommendation, can leverage weak annotations such as basic song metadata, known relationships or listening patterns oftentimes compiled by large music services such as *last.fm*<sup>1</sup>. However, there is a subset of MIR tasks dealing

with detailed information from the music signal for which time-aligned annotations are not readily available, such as the fundamental frequency ( $f_0$ ) of the melody (needed for melody extraction [13]) or the activation times of the different instruments in the mix (needed for instrument recognition [1]). Annotating this kind of highly specific information from real world recordings is a time consuming process that requires qualified individuals, and is usually done in the context of large annotation efforts such as the Billboard [3], SALAMI [15], and Beatles [8] datasets. These sets include manual annotations of structure, chords, or notes, typically consisting of categorical labels at time intervals on the order of seconds. The annotation process is even more time-consuming for  $f_0$  values or instrument activations for example, which are numeric instead of categorical, and at a time-scale on the order of milliseconds. Unsurprisingly, the datasets available for evaluating these tasks are often limited in size (on the order of a couple dozen files) and comprised solely of short excerpts.

When multitrack audio is available, annotation tasks that would be difficult with mixed audio can often be expedited. For example, annotating the  $f_0$  curve for a particular instrument from a full audio mix is difficult and tedious, whereas with multitrack stems the process can be partly automated using monophonic pitch tracking techniques. Since no algorithm provides 100% estimation accuracy in real-world conditions, a common solution is to have experts manually correct these machine annotations, a process significantly simpler than annotating from scratch. Unfortunately, collections of royalty-free multitrack recordings that can be shared for research purposes are relatively scarce, and those that exist are homogeneous in genre. This is a problem not only for evaluating annotation-intensive tasks but also for tasks that by definition require access to the individual tracks of a song such as source separation and automatic mixing.

In this paper we introduce *MedleyDB*: a multipurpose audio dataset of annotated, royalty-free multitrack recordings. The dataset includes melody  $f_0$  annotations and was primarily developed to support research on melody extraction and to address important shortcomings of the existing collections for this task. Its applicability extends to research on other annotation-intensive MIR tasks, such as instrument recognition, for which we provide instrument activations. The dataset can also be directly used for re-

<sup>1</sup><http://www.last.fm>



© Rachel Bittner<sup>1</sup>, Justin Salamon<sup>1,2</sup>, Mike Tierney<sup>1</sup>, Matthias Mauch<sup>3</sup>, Chris Cannam<sup>3</sup>, Juan Bello<sup>1</sup>.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Rachel Bittner<sup>1</sup>, Justin Salamon<sup>1,2</sup>, Mike Tierney<sup>1</sup>, Matthias Mauch<sup>3</sup>, Chris Cannam<sup>3</sup>, Juan Bello<sup>1</sup>. “*MedleyDB*: A Multitrack Dataset for Annotation-Intensive MIR Research”, 15th International Society for Music Information Retrieval Conference, 2014.

search on source separation and automatic mixing. Further track-level annotations (e.g. multiple  $f_0$  or chords) can be easily added in the future to enable evaluation of additional MIR tasks.

The remainder of the paper is structured as follows: in Section 2 we provide a brief overview of existing datasets for melody extraction evaluation, including basic statistics and content. In Section 3 we provide a detailed description of the *MedleyDB* dataset, including compilation, annotation, and content statistics. In Section 4 we outline the types of annotations provided and the process by which they were generated. In Section 5 we provide some insight into the challenges presented by this new dataset by examining the results obtained by a state-of-the-art melody extraction algorithm. The conclusions of the paper are provided in Section 6.

## 2. PRIOR WORK

### 2.1 Datasets for melody extraction

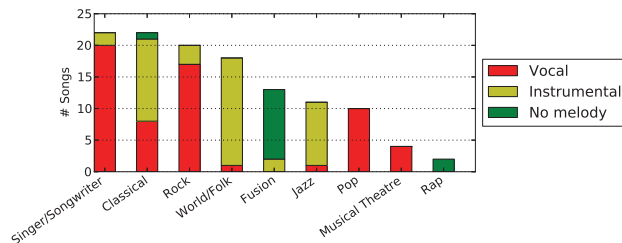
Table 1 provides a summary of the datasets commonly used for the benchmarking of melody extraction algorithms. It can be observed that datasets that are stylistically varied and contain “real” music (e.g. ADC2004 and MIREX05) are very small in size, numbering no more than two dozen files and a few hundred seconds of audio. On the other hand, large datasets such as MIREX09, MIR1K and the RWC pop dataset tend to be stylistically homogeneous and/or include music that is less realistic. Furthermore, all datasets, with the exception of RWC, are limited to relatively short excerpts. Note that the main community evaluation for melody extraction, the MIREX AME task,<sup>2</sup> has been limited to the top 4 datasets.

In [14], the authors examined how the aforementioned constraints affect the evaluation of melody extraction algorithms. Three aspects were studied – inaccuracies in the annotations, the use of short excerpts instead of full-length songs, and the limited number of excerpts used. They found that the evaluation is highly sensitive to systematic annotation errors, that performance on excerpts is not necessarily a good predictor for performance on full songs, and that the collections used for the MIREX evaluation [5] are too small for the results to be statistically stable. Furthermore, they noted that the only MIREX dataset that is sufficiently large (MIREX 2009) is highly homogeneous (Chinese pop music) and thus does not represent the variety of commercial music that algorithms are expected to generalize to. This finding extrapolates to the MIR1K and RWC sets.

To facilitate meaningful future research on melody extraction, we sought to compile a new dataset addressing the following criteria:

1. **Size:** the dataset should be at least one order of magnitude greater than previous heterogeneous datasets such as ADC2004 and MIREX05.

<sup>2</sup>[http://www.music-ir.org/mirex/wiki/Audio\\_Melody\\_Extraction](http://www.music-ir.org/mirex/wiki/Audio_Melody_Extraction)



**Figure 1.** Number of songs per genre with breakdown by melody source type.

2. **Duration:** the dataset should primarily consist of full length songs.
3. **Quality:** the audio should be of professional or near-professional quality.
4. **Content:** the dataset should consist of songs from a variety of genres.
5. **Annotation:** the annotations must be accurate and well-documented.
6. **Audio:** each song and corresponding multitrack session must be available and distributable for research purposes.

### 2.2 Multitrack datasets

Since we opted to use multitracks to facilitate the annotation process, it is relevant to survey what multitrack datasets are currently available to the community. The TRIOS [6] dataset provides 5 score-aligned multitrack recordings of musical trios for source separation, the MASS<sup>3</sup> dataset contains a small collection of raw and effects-processed multitrack stems of musical excerpts for work in source separation, and the Mixploration dataset [4] for automatic mixing contains 24 versions of four songs. These sets are too small and homogeneous to fit our criteria; the closest candidate is the Structural Segmentation Multitrack Dataset [7] which contains 103 rock and pop songs with structural segmentation annotations. While the overall size of this dataset is satisfactory, there is little variety in genre and the dataset is not uniformly formatted, making batched processing difficult or impossible.

Since no sufficient multitrack dataset currently exists, we curated *MedleyDB* which fits our needs and can be used for other MIR tasks as well, and is described in detail in the following section.

## 3. DATASET

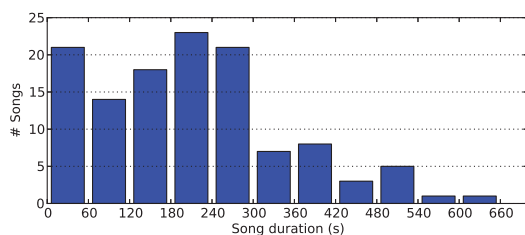
### 3.1 Overview

The dataset consists of 122 songs, 108 of which include melody annotations. The remaining 14 songs do not have a discernible melody and thus were not appropriate for melody extraction. We include these 14 songs in the dataset because of their use for other applications including instrument ID, source separation and automatic mixing.

<sup>3</sup><http://mtg.upf.edu/download/datasets/mass>

Name	# Songs	Song duration	Total duration	% Vocal Songs	Genres	Content
ADC2004	20	~20 s	369 s	60%	Pop, jazz, opera	Real recordings, synthesized voice and MIDI
MIREX05	25	~10–40 s	686 s	64%	Rock, R&B, pop, jazz, solo classical piano	Real recordings, synthesized MIDI
INDIAN08	8	~60 s	501 s	100%	North Indian classical music	Real recordings
MIREX09	374	~20–40 s	10020 s	100%	Chinese pop	Recorded singing with karaoke accompaniment
MIR1K	1000	~10 s	7980 s	100%	Chinese Pop	Recorded singing with karaoke accompaniment
RWC	100	~240 s	24403 s	100%	Japanese Pop, American Pop	Real recordings
<i>MedleyDB</i>	108	~20–600 s	26831 s	57%	Rock, pop, classical, jazz, rock, pop, fusion, world, musical theater, singer-songwriter	Real recordings

**Table 1.** Existing collections for melody extraction evaluation (ADC2004 through RWC) and the new *MedleyDB* dataset.



**Figure 2.** Distribution of song durations.

Each song in the dataset is freely available online<sup>4</sup> under a Creative Commons Attribution - NonCommercial - ShareAlike 3.0 Unported license<sup>5</sup>, which allows the release of the audio and annotations for non-commercial purposes.

We provide a stereo mix and both dry and processed multitrack stems for each song. The content was obtained from multiple sources: 30 songs were provided by various independent artists, 32 were recorded at NYU’s Dolan Recording Studio, 25 were recorded by Weathervane Music<sup>6</sup>, and 35 were created by Music Delta<sup>7</sup>. The majority of the songs were recorded in professional studios and mixed by experienced engineers.

In Figure 1 we give the distribution of genres present within the dataset, as well as the number of vocal and instrumental songs within each genre. The genres are based on nine generic genre labels. Note that some genres such as Singer/Songwriter, Rock and Pop are strongly dominated by vocal songs, while others such as Jazz and World/Folk are mostly instrumental. Note that the Rap and most of the Fusion songs do not have melody annotations. Figure 2 depicts the distribution of song durations. A total of 105 out of the 122 songs in the dataset are full length songs, and the majority of these are between 3 and 5 minutes long. Most recordings that are under 1 minute long were created by Music Delta. Finally, the most represented instruments in the dataset are shown in Figure 3. Unsurprisingly, drums, bass, piano, vocals and guitars dominate the distribution.

<sup>4</sup> <http://marl.smusic.nyu.edu/medleydb>

<sup>5</sup> [http://creativecommons.org/licenses/by-nc-sa/3.0/deed.en\\_US](http://creativecommons.org/licenses/by-nc-sa/3.0/deed.en_US)

<sup>6</sup> <http://weathervanemusic.org/>

<sup>7</sup> <http://www.musicdelta.com/>

### 3.2 Multitrack Audio Structure

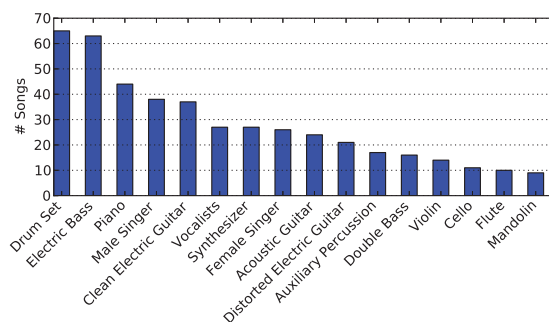
The structure of the audio content in *MedleyDB* is largely determined by the recording process, and is exemplified in Figure 4, which gives a toy example of how the data could be organized for a recording of a jazz quartet.

At the lowest level of the process, a set of microphones is used to record the audio sources, such that there may be more than one microphone recording a single source – as is the case for the piano and drum set in Figure 4. The resulting files are *raw* unprocessed mono audio tracks. Note that while they are “unprocessed”, they are edited such that there is no content present in the raw audio that is not used in the mix. The raw files are then grouped into stems, each corresponding to a specific sound source: double bass, piano, trumpet and drum set in the example. These *stems* are stereo audio components of the final mix and include all effects processing, gain control, and panning. Finally, we refer to the *mix* as the complete polyphonic audio created by mixing the stems and optionally mastering the mix.

Therefore, a song consists of the *mix*, *stems*, and *raw audio*. This hierarchy does not perfectly model every style of recording and mixing, but it works well for the majority of songs. Thus, the audio provided for this dataset is organized with this hierarchy in mind.

### 3.3 Metadata

Both song and stem-level metadata is provided for each song. The song-level metadata includes basic information about the song such as the artist, title, composer, and website. Additionally, we provide genre labels corresponding to the labels in Figure 1. Some sessions correspond to



**Figure 3.** Occurrence count of the most frequent instruments in the dataset.

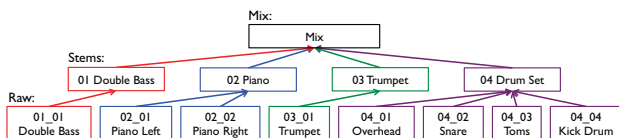


Figure 4. The hierarchy of audio files for a jazz quartet.

recordings of ensembles, where the microphones may pick up sound from sources other than the one intended, a phenomenon known as *bleeding*. Because bleed can affect automated annotation methods and other types of processing, songs that contain any stems with bleed are tagged.

Stem-level metadata includes instrument labels based on a predefined taxonomy given to annotators, and a field indicating whether the stem contains melody.

The metadata is provided as a YAML<sup>8</sup> file, which is both human-readable as a text file, and a structured format that can be easily loaded into various programming environments.

## 4. ANNOTATIONS

### 4.1 Annotation Task Definitions

When creating annotations for *MedleyDB*, we were faced with the question of what definition of melody to use. The definition of melody used in MIREX 2014 defines melody as the predominant pitch where, “pitch is expressed as the fundamental frequency of the main melodic voice, and is reported in a frame-based manner on an evenly-spaced time-grid.” Many of the songs in the dataset do not reasonably fit the definition of melody used by MIREX because of the constraint that the melody is played by a single voice, but we felt that the annotations should have consistency with the existing melody annotations.

Our resolution was to provide melody annotations based on three different definitions of melody that are in discussion within the MIR community.<sup>9</sup> In the definitions we consider, melody is defined as:

1. The  $f_0$  curve of the predominant melodic line drawn from a single source.
2. The  $f_0$  curve of the predominant melodic line drawn from multiple sources.
3. The  $f_0$  curves of all melodic lines drawn from multiple sources.

Definition 1 coincides with the definition for the melody annotations used in MIREX. This definition requires the choice of a lead instrument and gives the  $f_0$  curve for this instrument. Definition 2 expands on definition 1 by allowing multiple instruments to contribute to the melody. While a single lead instrument need not be chosen, an indication of which instrument is predominant at each point in time is required to resolve the  $f_0$  curve to a single point at each time frame. Definition 3 is the most complex, but also the most general. The key difference in this definition

<sup>8</sup> <http://www.yaml.org/>

<sup>9</sup> <http://ameannotationinitiative.wikispaces.com>

is that at a given time frame, multiple  $f_0$  values may be “correct”.

For instrument activations, we simply assume that signal energy in a given stem, above a predefined limit, is indicative of the presence of the corresponding instrument in the mix. Based on this notion, we provide two types of annotations: a list of time segments where each instrument is active; and a matrix containing the activation confidence per instrument per unit of time.

### 4.2 Automatic Annotation Process

The melody annotation process was semi-automated by using monophonic pitch tracking on selected stems to return a good initial estimate of the  $f_0$  curve, and by using a voicing detection algorithm to compute instrument activations. The monophonic pitch tracking algorithm used was pYIN [11] which is an improved, probabilistic version of the well-known YIN algorithm.

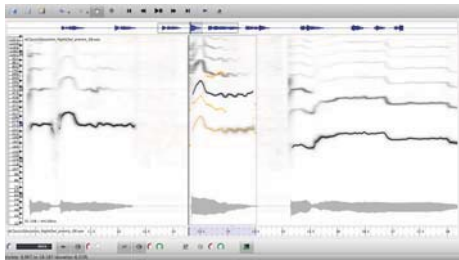
As discussed in the previous section, for each song we provide melody annotations based upon the 3 different definitions. The melody annotations based on Definition 1 were generated by choosing the single most dominant melodic stem. The Definition 2 annotations were created by sectioning the mix into regions and indicating the predominant melodic stem within each region. The melody curve was generated by choosing the  $f_0$  curve from the indicated instrument at each point in time. The Definition 3 annotations contain the  $f_0$  curves from each of the annotated stems.

The annotations of instrument activations were generated using a standard envelope following technique on each stem, consisting of half-wave rectification, compression, smoothing and down-sampling. The resulting envelopes are normalized to account for overall signal energy and total number of sources, resulting in the  $t \times m$  matrix  $H$ , where  $t$  is the number of analysis frames, and  $m$  is the number of instruments in the mix. For the  $i^{\text{th}}$  instrument, the confidence of its activations as a function of time can be approximated via a logistic function:

$$C(i, t) = 1 - \frac{1}{1 + e^{(H_{it} - \theta)\lambda}}. \quad (1)$$

where  $\lambda$  controls the slope of the function, and  $\theta$  the threshold of activation. Frames where instrument  $i$  is considered active are those for which  $C(i, t) \geq 0.5$ . No manual correction was performed on these activations.

Note that monophonic pitch tracking, and the automatic detection of voicing and instrument activations, fail when the stems contain bleed from other instruments, which is the case for 25 songs within the collection. Source separation, using a simple approach based on Wiener filters [2], was used on stems with bleed to clean up the audio before applying the algorithms. The parameters of the separation were manually and independently optimized for each track containing bleed.



**Figure 5.** Screenshot of *Tony*. An estimated pitch curve is selected and alternative candidates are shown in yellow.

### 4.3 Manual Annotation Process

The manual annotation process was facilitated by the use of a recently developed tool called *Tony* [10], which enables efficient manual corrections (see Figure 5). *Tony* provides 3 types of semi-manual correction methods: (1) deletion (2) octave shifting and (3) alternative candidates.

When annotating the  $f_0$  curves, unvoiced vocal sounds, percussive attacks, and reverb tail were removed. Sections of a stem which were active but did not contain melody were also removed. For example, a piano stem in a jazz combo may play the melody during a solo section and play background chords throughout the rest of the piece. In this case, only the solo section would be annotated, and all other frames would be marked as unvoiced.

The annotations were created by five annotators, all of which were musicians and had at least a bachelor’s degree in music. Each annotation was evaluated by one annotator and validated by another. The annotator/validator pairs were randomized to make the final annotations as unbiased as possible.

### 4.4 Annotation Formats

We provide melody annotations based on the three definitions for 108 out of the 122 songs. Note that while definition 1 is not appropriate for all of the annotated songs (i.e. there are songs where the melody is played by several sources and there is no single clear predominant source throughout the piece), we provide type 1 melody annotations for all 108 melodic tracks so that an algorithm’s performance on type 1 versus type 2 melody annotations can be compared over the full dataset. Of the 108 songs with melody annotations, 62 contain predominantly vocal melodies and the remaining 47 contain instrumental melodies.

Every melody annotation begins at time 0 and has a hop size of 5.8 ms (256 samples at  $f_s = 44.1$  kHz). Each time stamp in the annotation corresponds to the center of the analysis frame (i.e. the first frame is centered on time 0). In accordance with previous annotations, frequency values are given in Hz, where unvoiced frames (i.e. frames where there is no melody) are indicated by a value of 0 Hz.

We provide instrument activation annotations for the entire dataset. Confidence values are given as matrices where the first column corresponds to time in seconds, starting at 0 with a hop size of 46.4 ms (2048 samples at  $f_s = 44.1$

Dataset	$\nu$	VxR	VxF	RPA	RCA	OA
<i>MDB</i> – All	.2	.78 (.13)	.38 (.14)	.55 (.26)	.68 (.19)	<b>.54</b> (.17)
<i>MDB</i> – All	-1	.57 (.20)	.20 (.12)	.52 (.26)	.68 (.19)	<b>.57</b> (.18)
<i>MDB</i> – VOC	-1	.69 (.15)	.23 (.13)	.63 (.23)	.76 (.15)	<b>.66</b> (.14)
<i>MDB</i> – INS	-1	.41 (.15)	.16 (.09)	.38 (.23)	.57 (.18)	<b>.47</b> (.17)
MIREX11	.2	.86	.24	.80	.82	<b>.75</b>

**Table 2.** Performance of Melodia [12] on different subsets of *MedleyDB* (*MDB*) for type 1 melody annotations, and comparison to performance on the MIREX datasets. For each measure we provide the mean with the standard deviation in parentheses.

kHz), and each subsequent column corresponds to an instrument identifier. Confidence values are continuous in the range  $[0, 1]$ . We also provide a list of activations, each a triplet of start time, end time and instrument label.

## 5. NEW CHALLENGES

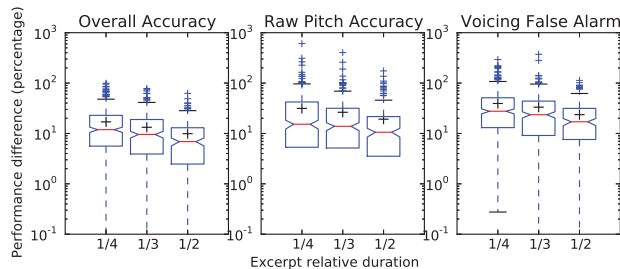
To gain insight into the challenges presented by this new dataset and its potential for supporting progress in melody extraction research, we evaluate the performance of the Melodia melody extraction algorithm [12] on the subset of *MedleyDB* containing melody annotations. In the following experiments we use the melody annotations based on Definition 1, which can be evaluated using the standard five measures used in melody extraction evaluation: voicing recall (VxR), voicing false alarm (VxF), raw pitch accuracy (RPA), raw chroma accuracy (RCA), and overall accuracy (OA). For further details about the measures see [13].

In the first row of Table 2 we give the results obtained by Melodia using the same parameters (voicing threshold  $\nu = .2$ ) employed in MIREX 2011 [12]. The first thing we note is that for all measures, the performance is considerably lower on *MedleyDB* than on MIREX11. The overall accuracy is 21 percentage points lower, a first indication that the new dataset is more challenging. We also note that the VxF rate is considerably higher compared to the MIREX results. In the second row of Table 2 we provide the results obtained when setting  $\nu$  to maximize the overall accuracy ( $\nu = -1$ ). The increase in overall accuracy is relatively small (3 points), indicating that the dataset remains challenging despite using the best possible voicing parameter. In the next two rows of Table 2, we provide a breakdown of the results by vocal vs. instrumental songs. We see that the algorithm does significantly better on vocal melodies compared to instrumental ones, consistent with the observations made in [12]. For instrumental melodies we observe a 19-point drop between raw chroma and pitch accuracy, indicating an increased number of octave errors. The bias in performance towards vocal melodies is likely the result of all previous datasets being primarily vocal.

In Table 3 we provide a breakdown of the results by genre. In accordance with the the previous table, we see that genres with primarily instrumental melodies are considerably more challenging. Finally, we repeat the experiment carried out in [14], where the authors compared performance on recordings to shorter sub-clips taken from the same recordings to see whether the results on a dataset of

Genre	VxR	VxF	RPA	RCA	OA
MUS	.73 (.16)	.14 (.04)	.74 (.18)	.87 (.08)	<b>.73</b> (.14)
POP	.74 (.12)	.22 (.09)	.65 (.20)	.73 (.15)	<b>.69</b> (.12)
S/S	.66 (.13)	.23 (.12)	.64 (.19)	.74 (.16)	<b>.66</b> (.11)
ROC	.71 (.18)	.29 (.15)	.53 (.29)	.73 (.18)	<b>.59</b> (.16)
JAZ	.44 (.14)	.12 (.06)	.55 (.17)	.68 (.15)	<b>.57</b> (.14)
CLA	.46 (.20)	.15 (.07)	.35 (.30)	.56 (.22)	<b>.51</b> (.23)
WOR	.40 (.12)	.18 (.09)	.44 (.19)	.63 (.14)	<b>.44</b> (.13)
FUS	.41 (.04)	.17 (.02)	.32 (.07)	.51 (.01)	<b>.43</b> (.04)

**Table 3.** Performance of Melodia [12] ( $\nu = -1$ ) on different genres in *MedleyDB* for type 1 melody annotations. For each measure we provide the mean with the standard deviation in parentheses.



**Figure 6.** Relative performance differences between full songs and excerpts. The large black crosses mark the means of the distributions.

excerpts would generalize to a dataset of full songs. The novelty in our experiment is that we use full length songs, as opposed to clips sliced into even shorter sub-clips. The results are presented in Figure 6, and are consistent with those reported in [14]. We see that as the relative duration of the excerpts (1/4, 1/3 or 1/2 of the full song) gets closer to 1, the relative difference in performance goes down (significant by a Mann-Whitney U test,  $\alpha = 0.01$ ). This highlights another benefit of *MedleyDB*: since the dataset primarily contains full length songs, one can expect better generalization to real-world music collections. While further error analysis is required to understand the specific challenges presented by *MedleyDB*, we identify (by inspection) some of the musical characteristics across the dataset that make *MedleyDB* more challenging – rapidly changing notes, a large melodic frequency range (43-3662 Hz), concurrent melodic lines, and complex polyphony.

## 6. CONCLUSION

Due to the scarcity of multitrack audio data for MIR research, we presented *MedleyDB* – a dataset of over 100 multitrack recordings of songs with melody  $f_0$  annotations and instrument activations. We provided a description of the dataset, including how it was curated, annotated, and its musical content. Finally, we ran a set of experiments to identify some of the new challenges presented by the dataset. We noted how the increased proportion of instrumental tracks makes the dataset significantly more challenging compared to the MIREX datasets, and confirmed that performance on excerpts will not necessarily generalize well to full-length songs, highlighting the greater generalizability of *MedleyDB* compared with most existing

datasets. Since 2011 there has been no significant improvement in performance on the MIREX AME task. If we previously attributed this to some glass ceiling, we now see that there is still much room for improvement. *MedleyDB* represents a shift towards more realistic datasets for MIR research, and we believe it will help identify future research avenues and enable further progress in melody extraction research and other annotation-intensive MIR endeavors.

## 7. REFERENCES

- [1] J.G.A. Barbedo. Instrument recognition. In T. Li, M. Ogihara, and G. Tzanetakis, editors, *Music Data Mining*. CRC Press, 2012.
- [2] L. Benaroya, F. Bimbot, and R. Gribonval. Audio source separation with a single sensor. *IEEE TASLP*, 14(1):191–199, 2006.
- [3] J. A. Burgoyne, J. Wild, and I. Fujinaga. An expert ground truth set for audio chord recognition and music analysis. In *ISMIR'11*, pages 633–638, 2011.
- [4] M. Cartwright, B. Pardo, and J. Reiss. Mixploration: Rethinking the audio mixer interface. In *19th Int. Conf. on Intelligent User Interfaces*, pages 365–370, 2014.
- [5] J. Stephen Downie. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008.
- [6] J. Fritsch and M. D. Plumbley. Score informed audio source separation using constrained nonnegative matrix factorization and score synthesis. In *IEEE ICASSP'13*, pages 888–891, 2013.
- [7] S. Hargreaves, A. Klapuri, and M. Sandler. Structural segmentation of multitrack audio. *IEEE TASLP*, 20(10):2637–2647, 2012.
- [8] C. Harte, M. B. Sandler, S. A. Abdallah, and E. Gómez. Symbolic representation of musical chords: A proposed syntax for text annotations. In *ISMIR'05*, pages 66–71, 2005.
- [9] M. I. Mandel and D. P. W. Ellis. A web-based game for collecting music metadata. *J. of New Music Research*, 37(2):151–165, 2008.
- [10] M. Mauch and G. Cannam, C. Fazekas. Efficient computer-aided pitchtrack and note estimation for scientific applications, 2014. SEMPRES'14, extended abstract.
- [11] M. Mauch and S. Dixon. pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *IEEE ICASSP'14*, 2014. In press.
- [12] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE TASLP*, 20(6):1759–1770, 2012.
- [13] J. Salamon, E. Gómez, D. P. W. Ellis, and G. Richard. Melody extraction from polyphonic music signals: Approaches, applications and challenges. *IEEE Signal Processing Magazine*, 31(2):118–134, 2014.
- [14] J. Salamon and J. Urbano. Current challenges in the evaluation of predominant melody extraction algorithms. In *ISMIR'12*, pages 289–294, 2012.
- [15] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure, and J. S. Downie. Design and creation of a large-scale database of structural annotations. In *ISMIR'11*, pages 555–560, 2011.



# MELODY EXTRACTION FROM POLYPHONIC AUDIO OF WESTERN OPERA: A METHOD BASED ON DETECTION OF THE SINGER'S FORMANT

Zheng Tang

University of Washington, Department  
of Electrical Engineering  
zhtang@uw.edu

Dawn A. A. Black

Queen Mary University of London, Electronic Engi-  
neering and Computer Science  
dawn.black@qmul.ac.uk

## ABSTRACT

Current melody extraction approaches perform poorly on the genre of opera [1, 2]. The singer's formant is defined as a prominent spectral-envelope peak around 3 kHz found in the singing of professional Western opera singers [3]. In this paper we introduce a novel melody extraction algorithm based on this feature for opera signals. At the front end, it automatically detects the singer's formant according to the Long-Term Average Spectrum (LTAS). This detection function is also applied to the short-term spectrum in each frame to determine the melody. The Fan Chirp Transform (FChT) [4] is used to compute pitch saliency as its high time-frequency resolution overcomes the difficulties introduced by vibrato. Subharmonic attenuation is adopted to handle octave errors which are common in opera vocals. We improve the FChT algorithm so that it is capable of correcting outliers in pitch detection. The performance of our method is compared to 5 state-of-the-art melody extraction algorithms on a newly created dataset and parts of the ADC2004 dataset. Our algorithm achieves an accuracy of 87.5% in singer's formant detection. In the evaluation of melody extraction, it has the best performance in voicing detection (91.6%), voicing false alarm (5.3%) and overall accuracy (82.3%).

## 1. INTRODUCTION

Singing voice can be considered to carry the main melody in Western opera. Melody extraction from a polyphonic signal including singing voice requires both of the following: estimation of the correct pitch of singing voice in each time frame and voicing detection to determine when the singing voice is present or not.

The singer's (or singing) formant was first introduced by Johan Sundberg [3] and described as a clustering of the third, fourth, and fifth formants to form a prominent spectral-envelope peak around 3 kHz. It is purportedly generated by widening the pharynx and lowering the larynx. The existence of a singer's formant has been confirmed in the singing voices of classically trained male

Western opera singers and some female singers, but it has not yet been found in soprano singers [5] or Chinese opera singers [6]. It has been proposed that singers develop the singer's formant in order to be heard above the orchestra. In Western opera, orchestral instruments typically occupy the same frequency range as the singers. Therefore singers train their vocal equipment in order to raise the amplitude of frequencies at this range.

The LTAS is the average of all short-term spectra in a signal, has been shown to be an excellent tool to observe the singer's formant [7] as can be seen in Figure 1. Characteristics of the singer's formant in the spectral domain include a peak greater than 20 dB below the overall sound pressure level, a peak-location at 2.5-3.2 kHz, and a bandwidth of around 500-800 Hz [5, 7]. However, to date, there has been no method developed to automatically detect the presence of a singer's formant or to quantify its characteristics.

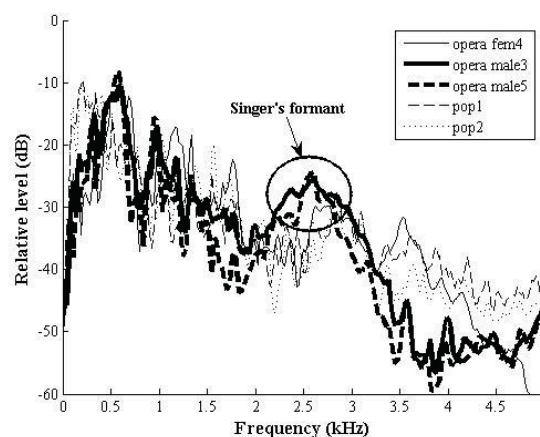


Figure 1. Normalized LTAS for 5 audio excerpts from the ADC2004 test collection [1].

### 1.1 Related Work

In 2004, the Music Technology Group of the Universitat Pompeu Fabra organized a melody extraction contest presented at the International Society for Music Information Retrieval Conference. The Music Information Retrieval Evaluation eXchange (MIREX) was set up in 2005 and audio melody extraction has been a highly competitive field ever since. Currently, over 60 algorithms have been



© Zheng Tang, Dawn A. A. Black.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Zheng Tang, Dawn A. A. Black. "Melody Extraction From Polyphonic Audio Of Western Opera: A Method Based On Detection Of The Singer's Formant", 15th International Society for Music Information Retrieval Conference, 2014.

submitted and evaluated. So far, none of the approaches consider the presence of the singer's formant.

The majority of algorithms presented at MIREX are salience based [2]. These assume that the fundamental frequency of the melody is equivalent to the most salient pitch value in each frame. The Short-Time Fourier Transform (STFT) is often chosen to compute pitch salience [7, 8]. In 2008, Pablo Cancela proposed the Fan Chirp Transform (FChT) method, combined with Constant Q Transform (CQT) in music processing. The FChT is a time-warped version of the Fourier Transform that provides better time-frequency resolution [4, 9]. Although the STFT provides adequate resolution in the majority of cases, it fails to generate a satisfying outcome when dealing with Western opera signals. This is because opera typically exhibits complex spectral characteristics due to vocal ornamentations such as vibrato [1]. Vibrato is a regular fluctuation of singing pitch produced by singers. This increases the difficulty in tracking the melody. With better resolution, the fast change of pitch salience can be better observed and tracked by using FChT.

It has been proposed that the singer's formant may cause octave errors [2]. The presence of a spectral peak (the singer's formant) at a higher frequency may cause the fundamental frequency to be confused with the frequency at the centre of the singer's formant. To address this, Cancela developed a method called 'subharmonic attenuation' that can minimize the negative effects of ghost pitch values at the multiple and submultiple peaks of a certain fundamental frequency [2, 9].

Voicing detection typically receives much less attention than pitch detection, to the extent that some previous melody extraction algorithms did not contain this procedure [10]. The most common approach is to set an energy threshold, which might be fixed or dynamic [9]. However, this technique is too simplistic since the loudness of musical accompaniment in Western opera may fluctuate considerably. It is therefore impossible to define an appropriate threshold. An alternative technique is to use a trained classifier based on a Hidden Markov Model (HMM) [11] but it is time-consuming to create a large dataset for training and there are always exceptions beyond the scope of the training set. In 2009, Regnier and Peeters proposed a voicing detection algorithm based on extraction of vocal vibrato [12], but has not been applied to melody extraction. In general, the high rate of false positives when detecting voiced frames limits the overall accuracy of melody extraction algorithms and a reduction of this is beneficial [2, 13].

This paper is organized as follows. In Section 2, we describe the design and implementation of our proposed algorithm for melody extraction. Starting with a general workflow of the system, the function and novelty of each component is explained in detail. Section 3 explains the evaluation process and presents a comparison of existing algorithms. The creation of the new dataset is also pre-

sented in this section. Finally, we draw conclusions from the results and give suggestions for future work.

## 2. DESCRIPTION OF THE ALGORITHM

### 2.1 General Workflow

Figure 2 shows an overview of our system. In order to extract the pitch of singing voice from polyphonic audio, we must first determine whether the audio contains singing voice. The presence of a singer's formant would indicate the presence of a classically trained singer. The LTAS is used to determine whether a singer's formant exists in the audio, and hence determines whether our method can be applied.

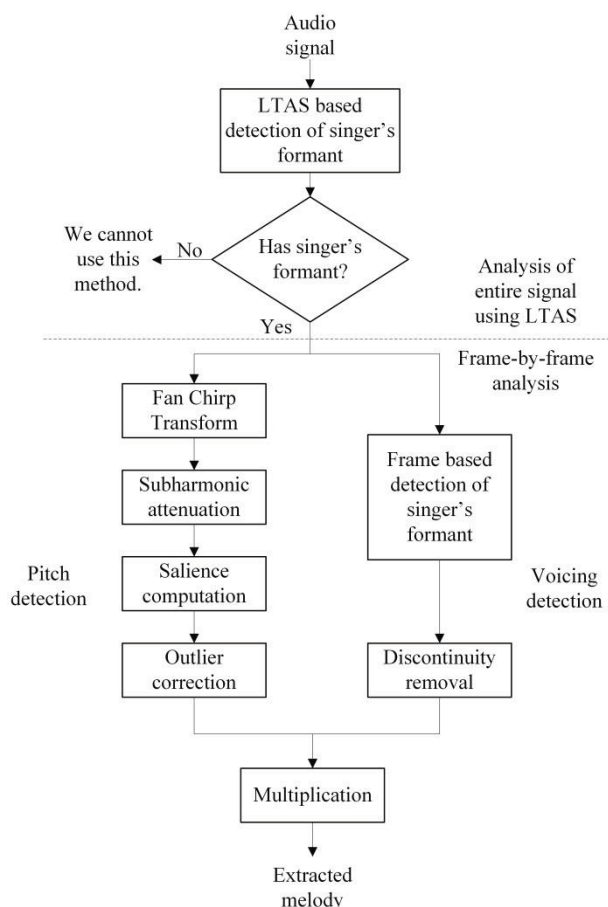


Figure 2. System overview.

Once the presence of a singer is confirmed the spectrum is analysed on a frame-by-frame basis. Two decisions are made for each frame: firstly, does the frame contain singing and hence a salient pitch? Secondly, what is the salient pitch of that frame?

We examine the spectral content of each frame to establish the presence of a singer's formant in that frame. If present, that frame is designated 'voiced' and assumed to contain melody carried by the singer's pitch.

Each frame is also transformed to the frequency domain using the FChT and further processed by subharmonic attenuation to obtain the pitch.

## 2.2 Singer's Formant Detection and Voicing Detection

Based on the characteristics of the singer's formant (see Section 1) we introduce a novel algorithm to automatically detect the presence of a singer's formant (and hence the presence of a classically trained singer). Using Monson's method to compute the LTAS of the input audio signal [14] the presence of a singer's formant would be confirmed if the LTAS exhibited the following properties:

1. There exists a spectral peak which has an amplitude greater than 20 dB less than the overall sound pressure level.
2. The peak is located between 2.5 and 3.2 kHz.
3. The peak has a bandwidth of around 500-800 Hz.

However, these properties were observed through analysis of singing voice in the absence of musical accompaniment [7]. When analysing singing with accompaniment, these criteria had to be modified in the following ways: the amplitude threshold of the spectral peak was found to be lower than the theoretical value and thus the first criteria becomes:

1. The spectral peak has an amplitude greater than 30 dB less than the overall sound pressure level.

The LTAS exhibited irregular fluctuations that made accurate identification of the singer's formant peak problematic. We therefore smoothed the LTAS (20 point average) and used polynomial fitting of degree 30. This smoothing and polynomial fitting will shift the location of the spectral peak and hence the range of the peak must be expanded. The second criteria is therefore modified to:

2. The peak is located between 2.2 and 3.4 kHz.

Similarly, we observe that the polynomial bandwidth may be slightly different from the LTAS curve. Therefore the bandwidth of the singer's formant is set to be larger than the original value:

3. The peak has a bandwidth larger than 600 Hz.

We must then add another criteria to ensure the significance of the peak. In order to measure the significance, we employed the first-order and second-order derivatives of the LTAS to measure the LTAS curvature and, from empirical evidence, designate significance to be a peak with a curvature greater than 0.01:

4. The curvature exceeds 0.01 at the location of the spectral peak.

In order to illustrate the criteria, we present the following figures. Figure 3 shows the fitting polynomials of smoothed LTAS for 5 samples from the MIREX ADC2004 test collection [1]. The singer's formant can be clearly observed for the male opera samples. Presented in Figure 4 is the second-order derivative of LTAS. This is negative when the curve is convex and hence can be used to determine the formant bandwidth. Our constraint that the bandwidth be at least 600 Hz is illustrated. In Figure 5, we show that the constraint on curvature can ensure the degree of convexity of the curve. It is clear from all plots

that the opera signals sung by male singers contain the singer's formant but others do not.

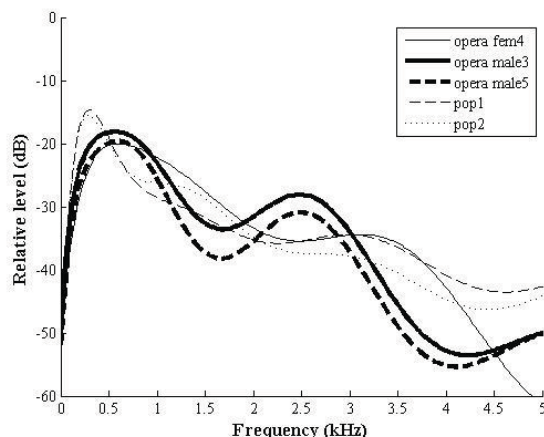


Figure 3. The fitting polynomials of smoothed LTAS for 5 audio excerpts from ADC2004 [1].

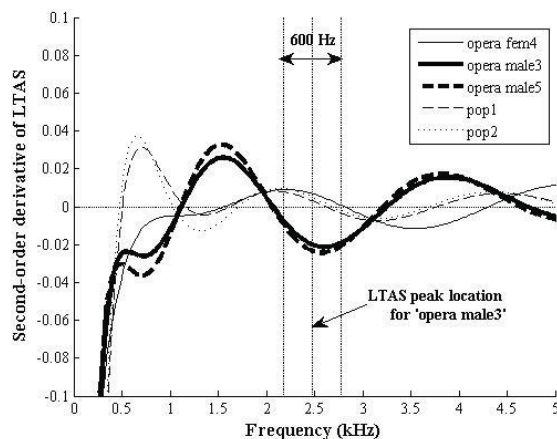


Figure 4. The second-order derivatives of LTAS for 5 audio excerpts from ADC2004 [1].

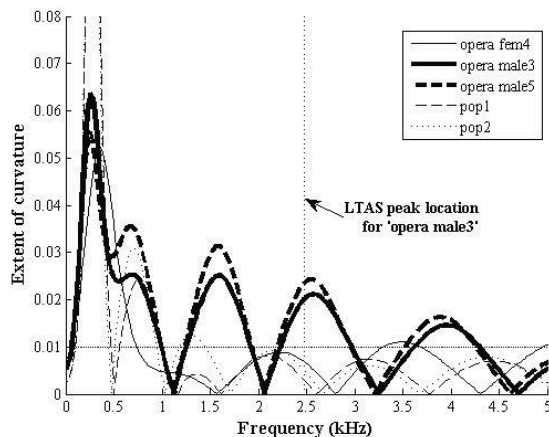


Figure 5. The curvatures of LTAS for 5 audio excerpts from ADC2004 [1].

If the LTAS satisfies all four criteria the audio is presumed to contain a trained singer. Use of the same criteria to analyse the spectrum of a single audio frame can indicate whether the frame is voiced (contains singing) or not. For a single frame, only the second and third criteria are applied, as the other two criteria are more influenced by observed amplitude variations in individual short-term spectra. The output of this stage is a two-value sequence whose length is the number of frames, with ‘1’ indicating a voiced frame and ‘-1’ unvoiced. Subsequently, when considering points of discontinuity causing false detections, single values within a sequence are removed.

### 2.3 Pitch Detection

If a frame is classified as voiced it can be expected to contain a clearly defined pitch. Vibrato in singing can cause pitch ambiguity. We therefore adopt Cancela’s method to perform FChT since it exhibits optimal time-frequency resolution. This chirp-based transform is based on an FFT performed in a warped time domain. It is combined with CQT in order to guarantee high resolution even when the fan chirp rate is not ideal. More details can be found in [4] and [9].

In Western opera the singer’s formant will cause peaks at frequencies higher than the fundamental [2]. The algorithm from Cancela provides subharmonic attenuation - an effective solution to this problem. It will suppress multiple and submultiple pitch peaks of the fundamental frequency. Then, we can perform salience computation to detect the pitch in each frame.

In the outlier correction stage, to improve Cancela’s method, we compute two additional peaks per frame as candidate substitutes for the wrong pitch. Firstly, the most salient pitch peaks are compared with those from adjacent frames. If a difference of more than 2 semitones occurs on both sides, the estimated pitch in this frame is considered as a wrong detection. In this case we substitute the pitch for this frame with the pitch among the three candidates which is closest to the average of the two adjacent estimations. Due to subharmonic attenuation, the influence of the subharmonics of the top peak is reduced when calculating the other pitch candidates.

Our method is novel to Cancela’s in the following ways: (1) The algorithm designed by Cancela extracts multiple salient peaks simultaneously and these are viewed as separate melodies. We introduce the correction block so that the less salient peaks are taken as substitutes of wrong pitch detection in a single estimation of melody. (2) We improve the voicing detection by considering the singer’s formant. (3) Cancela’s method is not specifically designed for opera items and its potential for dealing with vibrato and other spectrum characteristics has not been explored.

Finally, the estimated pitch sequence is multiplied by the two-value voicing detection sequence. The output of our algorithm follows the standard format of MIREX and

records the time-stamp and estimated frequency of each frame.

## 3. EVALUATION

### 3.1 The Dataset

The dataset we used for evaluation is a combination of the ADC2004 test collection and our own dataset<sup>1</sup>. Details of the dataset can be found in Table 1.

Among the existing test collections in MIREX, only ADC2004 contains 2 excerpts in the genre of Western opera. In order to evaluate the performance of melody extraction algorithms upon sufficient amount of opera samples for meaningful comparison, we created a new dataset. Nine students from the Central Academy of Drama in Beijing were recorded. All had received more than 5 years of classical voice training except for an amateur Western opera male singer. Their singing voices were recorded in a practice room, about 10×5×5 m with moderate reverberation. The equipment included a Sony PCM-D50 recorder and an AKG C5 microphone. The accompaniments played by orchestra were recorded separately. All the signals were digitized at a sample rate of 44.1 kHz with bit depth 16. We normalized the maximum amplitude of the singing voices to be -1.25 dB. The signal-to-accompaniment ratio is set to 0 dB. The ground truth for melody extraction was generated by a monophonic pitch tracker in SMSTools with manual adjustment [2] using the vocal track only. The frame size was 2048 samples with a step size of 256 samples.

We conducted two evaluations based on this combined dataset. The test set for melody extraction consists of 18 excerpts of 15s-25s duration sung by classically trained Western opera tenors. For the evaluation of singer’s formant detection, we will compare them with 14 excerpts sung by trained Western opera sopranos, trained Peking opera singers, pop singers, and a single unprofessional Western opera male singer.

Test set	Singing type	No. of songs	Expectation/detection of singer’s formant
ADC2004	Tenor, Western	2	Yes/ Yes
	Soprano, Western	2	No/ No
	Popular music	4	No/ No
The dataset recorded at the Central Academy of Drama	Tenor, Western	16	Yes/ Yes
	Soprano, Western	2	No/ Yes
	Amateur, Western	2	No/ Yes
	Laosheng, Peking	2	No/ No
	Qingyi, Peking	2	No/ No

**Table 1.** Test dataset for the evaluations of melody extraction and singer’s formant detection.

<sup>1</sup>This database is available for download under a creative commons license at <http://c4dm.eecs.qmul.ac.uk/rdr/> all usage should cite this paper.

### 3.2 Melody Extraction Comparison

Of the many melody extraction algorithms submitted to MIREX, few are freely available. We present five algorithms for comparison. We were limited in our choice by availability, but the methods are representative of the majority of algorithms submitted to MIREX in that they cover common approaches and best performance. Each method is briefly introduced next.

Cancela's algorithm was submitted in 2008 [9]. He used FChT combined with CQT to estimate the pitch in each time frame. Voicing detection is conducted through the calculation of an adaptive threshold, but this procedure is not included in the open-source code provided online. For the purposes of comparison, we added a common voicing detection function utilizing an adaptive energy threshold as described in [9].

Salamon's algorithm was introduced in 2011 [8]. It has been developed into a melody extraction vamp plug-in: MELODIA. This algorithm achieved the best score in MIREX 2011. It applies contour tracking to the salience function calculated by STFT to remove all the contours except for the melody. The voicing detection step is carried out by removing the contours that are not salient.

The algorithm developed by Sutton in 2006 [11] innovatively combines two pitch detectors based on the features of singing voice including pitch instability and high-frequency dominance. A modified HMM processes the estimated melodies and determines the voicing.

The final two algorithms were both proposed by Vincent in 2005 [10]. One makes use of a Bayesian harmonic model to estimate the melody, and the other is achieved via loudness-weighted YIN method. Vincent assumed that the melody was continuous throughout the audio, and voicing detection was not included in his algorithm.

### 3.3 Results

The evaluation results of singer's formant detection can be found in Table 1. Among the 32 audio files in the dataset, the assumption is that only the 18 excerpts sung by Western opera tenors possess the singer's formant, while the others do not. The results show that 28 of the files (87.5%) meet our expectation. The singer's formant is also detected in the excerpts of the Western opera amateur and sopranos in our dataset. The amateur singer is from the Acting Department at the Central Academy of Drama (Beijing) and declares that he has not received any formal training in opera. However he used to take courses in vocal music due to a requirement of the school. Thus, there is a possibility that the presence of singer's formant only requires a short period of training. Although sources state that there is no singer's formant present in soprano singing [5, 7], the mean pitch of the two excerpts in our dataset is at the low end of the range for sopranos (550.43 Hz). The presence of a singer's formant is pitch related. The higher the pitch, the less likely a singer's formant is present. A precise study of this relationship is a topic for future work.

Table 2 shows the melody extraction results of the 6 algorithms. Voicing detection measures the probability of correct detection of voiced frames, while voicing false alarm is the probability of incorrect detection of unvoiced frames. Raw pitch accuracy and raw chroma accuracy both measure the accuracy of pitch detection, with the latter ignoring octave errors. The overall accuracy is the proportion of frames labeled with correct pitch and voicing. Since Vincent's algorithms did not perform voicing detection, their voicing metrics and overall accuracy are inapplicable.

First author/ completion year	Voicing detection	Voicing false alarm	Raw pitch accuracy	Raw chroma accuracy	Overall accuracy
Vincent (Bayes)/ 2005	N/A	N/A	64.8%	68.6%	N/A
Vincent (YIN)/ 2005	N/A	N/A	69.5%	72.2%	N/A
Sutton/ 2006	89.3%	51.9%	87.0%	87.6%	76.9%
Cancela/ 2008 <sup>1</sup>	72.6%	39.3%	83.9%	84.8%	62.4%
Salamon/ 2011	62.3%	21.8%	25.4%	30.1%	31.3%
Our method	91.6%	5.3%	84.3%	85.1%	82.3%

**Table 2.** Results of the audio melody extraction evaluation.

Our algorithm ranks highest in overall accuracy. We also achieve the highest voicing detection rate as 91.6% and the lowest voicing false alarm rate as 5.3%, which proves that voicing detection based on the singer's formant is extremely effective for male Western opera. The improvement in raw pitch accuracy by outlier correction when compared to Cancela's method is not large. This allows us to hypothesise that the melody in Western opera may be so prominent that the influence of any accompaniment can be disregarded.

Sutton's method also has excellent performance on our dataset. That success might be attributed to his similar focus on the characteristics of singing voice. He also makes use of the vibrato feature to estimate the pitch of melody. Due to the application of a high-frequency correlogram, Sutton's algorithm may indirectly benefit from the presence of a singer's formant. However, the method we propose for voicing detection is much more convenient than the use of an HMM. Moreover, Sutton's algorithm exhibits a much higher voicing false alarm rate.

The poor performance of Salamon's algorithm on our dataset can be explained by the fact that it fails to estimate the pitch in detected unvoiced frames accurately.

We also evaluated the 4 audio files that contradicted our expectation in singer's formant detection (two West-

<sup>1</sup>The voicing detection part of this algorithm is implemented by us and cannot represent the original design of the author.

ern soprano singers and one amateur male Western opera singer). The performance of our algorithm declines significantly with a voicing detection rate of 53.1% and an overall accuracy of 53.7%. This may be due to the fact that the singer's formant, although present, is not as pronounced or stable as the Western opera tenor's.

#### 4. CONCLUSION AND FURTHER WORK

In this paper, we have presented a novel melody extraction algorithm based on the detection of singer's formant. This detection relies on 4 criteria modified from previously proposed characteristics of the singer's formant. The pitch detection step of our algorithm is achieved using FChT and subharmonic attenuation to overcome the known difficulties when detecting the melody in opera. We also improved the algorithm so it is capable of removing outliers in pitch detection.

From the evaluation results, it can be seen that our algorithm can detect the singer's formant accurately. Melody extraction evaluation on our dataset confirms that our algorithm provides a clear improvement in voicing detection. Furthermore, its overall accuracy is comparable to state-of-the-art methods when dealing with Western opera signals.

In the future, we plan to study the performance of this algorithm on signals in other genres and expand its scope of application. Additionally, the possible effects of performing environments and accompaniment music to the usage of singer's formant will also be explored.

#### 5. ACKNOWLEDGMENTS

This paper is based upon a research collaboration with the Department of Peking Opera at the Central Academy of Drama. Thanks to Prof. Ma Li and his students for their recording samples and professional advice on traditional opera. Besides, we would like to express our thanks to Pablo Cancela, Justin Salamon, Emilia Gómez, Christopher Sutton and Emmanuel Vincent for contributing their algorithm codes.

#### 6. REFERENCES

- [1] E. Gómez, S. Streich, B. Ong, R. P. Paiva, S. Tappert, J. M. Batke, G. Poliner, D. Ellis, and J. P. Bello: "A quantitative comparison of different approaches for melody extraction from polyphonic audio recordings," Univ. Pompeu Fabra, Barcelona, Spain, 2006, Tech. Rep. MTG-TR-2006-01.
- [2] J. Salamon, E. Gómez, D. Ellis, and G. Richard: "Melody extraction from polyphonic music signals: approaches, applications and challenges," *IEEE Signal Processing Magazine*, Vol. 31, No. 2, pp. 118-134, 2013.
- [3] J. Sundberg: "Articulatory interpretation of the 'singing formant'," *The Journal of the Acoustical Society of America*, Vol. 55, No. 4, pp. 838-844, 1974.
- [4] L. Weruaga, and M. Képesi: "The fan-chirp transform for non-stationary harmonic signals," *Signal Processing*, Vol. 87, No. 6, pp. 1504-1522, 2007.
- [5] R. Weiss, Jr, W. S. Brown, and J. Moris: "Singer's formant in sopranos: fact or fiction?" *Journal of Voice*, Vol. 15, No. 4, pp. 457-468, 2001.
- [6] J. Sundberg, L. Gu, Q. Huang, and P. Huang: "Acoustical study of classical Peking Opera singing," *Journal of Voice*, Vol. 26, No. 2, pp. 137-143, 2012.
- [7] J. Sundberg: "Level and center frequency of the singer's formant," *Journal of voice*, Vol. 15, No. 2, pp. 176-186, 2001.
- [8] J. Salamon, and E. Gómez: "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No. 6, pp. 1759-1770, 2012.
- [9] P. Cancela, E. López, and M. Rocamora: "Fan chirp transform for music representation," *Proceedings of the 13th Int Conference on Digital Audio Effects DAFx10 Graz Austria*, 2010.
- [10] E. Vincent, and M. D. Plumbley: "Predominant-F0 estimation using Bayesian harmonic waveform models," *2005 Music Information Retrieval Evaluation eXchange (MIREX)*, 2005.
- [11] C. Sutton: "Transcription of vocal melodies in popular music," Report for the degree of MSc in Digital Music Processing, Queen Mary University of London, 2006.
- [12] L. Regnier, and G. Peeters: "Singing voice detection in music tracks using direct voice vibrato detection," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1685-1688, 2009.
- [13] G. E. Poliner, D. P. Ellis, A. F. Ehmann, E. Gómez, S. Streich, and B. Ong: "Melody transcription from music audio: Approaches and evaluation," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 15, No. 4, pp. 1247-1256, 2007.
- [14] B. B. Monson: "High-frequency energy in singing and speech," Doctoral dissertation, University of Arizona, 2011.

# CODEBOOK-BASED SCALABLE MUSIC TAGGING WITH POISSON MATRIX FACTORIZATION

Dawen Liang, John Paisley, Daniel P. W. Ellis

Department of Electrical Engineering  
Columbia University

{dliang, dpwe}@ee.columbia.edu, jpaisley@columbia.edu

## ABSTRACT

Automatic music tagging is an important but challenging problem within MIR. In this paper, we treat music tagging as a matrix completion problem. We apply the Poisson matrix factorization model jointly on the vector-quantized audio features and a “bag-of-tags” representation. This approach exploits the shared latent structure between semantic tags and acoustic codewords. Leveraging the recently-developed technique of stochastic variational inference, the model can tractably analyze massive music collections. We present experimental results on the CAL500 dataset and the Million Song Dataset for both annotation and retrieval tasks, illustrating the steady improvement in performance as more data is used.

## 1. INTRODUCTION

Automatic music tagging is the task of analyzing the audio content (waveform) of a music recording and assigning to it human-relevant semantic tags [16] – which may relate to style, genre, instrumentation, or more subtle aspects of the music, such as those contributed by users on social media sites. Such “autotagging” [5] relies on labelled training examples for each tag, and performance typically improves with the number of training examples consumed, although training schemes also take longer to complete. In the era of “Big Data”, it is necessary to develop models which can rapidly handle massive amount of data; a starting point for music data is the Million Song Dataset [2], which includes user tags from Last.fm [1].

In this paper, we treat the automatic music tagging as a matrix completion problem, and use the techniques of stochastic variational inference to be able to learn from large amounts of data presented in an online fashion [9]. The “matrix completion” problem treats each track as a row in a matrix, where the elements describe both the acoustic properties (represented, for instance, as a histogram of occurrences of vector-quantized acoustic features) and the relevance of a large vocabulary of tags: We can regard the

tag information as incomplete or missing for some of the rows, and seek to “complete” these rows based on information inferred from the complete, present rows.

### 1.1 Related work

There have been a large number of papers on automatic tagging of music audio in recent years. In addition to the papers mentioned above, work particularly relevant to this paper includes the Codeword Bernoulli Average (CBA) approach of Hoffman *et al.* [7], which uses a similar VQ histogram representation of the audio to build a simple but effective probabilistic model for each tag in a discriminative fashion. Xie *et al.* [17] directly fits a regularized logistic regression model to the normalized acoustic codeword histograms to predict each tag and achieves state-of-the-art results, and Ellis *et al.* [6] further improves tagging accuracy by employing multiple generative models that capture different characteristics of a music piece, which are combined in an optimized “bag-of-systems”.

Much of the previous work has been performed on the CAL500 dataset [16] of 502 Western popular music tracks that were carefully labelled by at least three human annotators with their relevance to 149 distinct labels spanning instrumentation, genre, emotions, vocal characteristics, and use cases. This small dataset tends to reward approaches that can maximize the information extracted from the sparse data regardless of the computational cost. A relatively larger dataset in this domain is CAL10k [15] with over 10,000 tracks described by over 500 tags, mined from Pandora’s website<sup>1</sup>. However, neither of these datasets can be considered industrial scale, which implies handling millions of tracks and tens of thousands of tags.

Matrix factorization techniques, in particular, nonnegative matrix factorization (NMF), have been widely used to analyze music signals [8, 11] in the context of source separation. Paisley *et al.* [12] derived scalable Bayesian NMF for topic modeling, which we develop here. To our knowledge, this is the first application of matrix factorization to VQ acoustic features for automatic music tagging.

## 2. DATA REPRESENTATION

For our automatic tagging system, the data comes from two sources: vector-quantized audio features and a “bag-



© Dawen Liang, John Paisley, Daniel P. W. Ellis.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Dawen Liang, John Paisley, Daniel P. W. Ellis. “Codebook-based scalable music tagging with Poisson matrix factorization”, 15th International Society for Music Information Retrieval Conference, 2014.

<sup>1</sup><http://www.pandora.com/>

of-tags” representation.

- **Vector-quantized audio features** Instead of directly working with audio features, we vector quantize all the features following the standard procedure: We run the  $K$ -means algorithm on a subset of randomly selected training data to learn  $J$  cluster centroids (codewords). Then for each song, we assign each frame to the cluster with the smallest Euclidean distance to the centroid. We form the VQ feature  $y_{VQ} \in \mathbb{N}^J$  by counting the number of assignments to each cluster across the entire song.
- **Bag-of-tags** Similar to the bag-of-words representation, which is commonly used to represent documents, we represent the tagging information (whether or not the tag applies to a song) with a binary bag-of-tags vector  $y_{BoT} \in \{0, 1\}^{|V|}$ , where  $V$  is the set of all tags.

For each song, we will simply concatenate the VQ feature  $y_{VQ}$  and the bag-of-tags vector  $y_{BoT}$ , thus the dimension of the data is  $D = J + |V|$ . When we apply the matrix factorization model to the data, the latent factors we learn will exploit the shared latent structure between semantic tags and acoustic codewords. Therefore, we can utilize the shared latent structure to predict tags when only given the audio features.

### 3. POISSON MATRIX FACTORIZATION

We adopt the notational convention that bold letters (e.g.  $\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\beta}$ ) denote matrices.  $i \in \{1, \dots, I\}$  is used to index songs.  $d \in \{1, \dots, D\}$  is used to index feature dimensions.  $k \in \{1, \dots, K\}$  is used to index latent factors from the matrix factorization model. Given the data  $\mathbf{y} \in \mathbb{N}^{I \times D}$  as described in Section 2, the Poisson matrix factorization model is formulated as follows:

$$\begin{aligned} \theta_{ik} &\sim \text{Gamma}(a, ac), \\ \beta_{kd} &\sim \text{Gamma}(b, b), \\ y_{id} &\sim \text{Poisson}(\sum_{k=1}^K \theta_{ik} \beta_{kd}), \end{aligned} \quad (1)$$

where  $\beta_k \in \mathbb{R}_+^D$  denote the  $k$ th latent factors and  $\theta_i \in \mathbb{R}_+^K$  denote the weights for song  $i$ .  $a$  and  $b$  are model hyperparameters.  $c$  is a scalar on the weights that we tune to maximize the likelihood.

There are a couple of reasons to choose a Poisson model over a more traditional Gaussian model [14]. First, the Poisson distribution is a more natural choice to model count data. Secondly, real-world tagging data is extremely noisy and sparse. If a tag is not associated with a song in the data, it could be either because that tag does not apply to the song, or simply because no one has labelled the song with the tag yet. The Poisson matrix factorization model has the desirable property that it does not penalize values of 0 as strongly as the Gaussian distribution [12]. Therefore, even weakly labelled data can be used to learn the Poisson model.

## 4. VARIATIONAL INFERENCE

To learn the latent factors  $\boldsymbol{\beta}$  and the corresponding decomposition weights  $\boldsymbol{\theta}$  from the training data  $\mathbf{y}$ , we need to compute the posterior distribution  $p(\boldsymbol{\theta}, \boldsymbol{\beta} | \mathbf{y})$ . However, no closed-form expression exists for this hierarchical model. We therefore employ mean-field variational inference to approximate this posterior [10].

The basic idea behind mean-field variational inference is to choose a factorized family of variational distributions,

$$q(\boldsymbol{\theta}, \boldsymbol{\beta}) = \prod_{k=1}^K \left( \prod_{i=1}^I q(\theta_{ik}) \right) \left( \prod_{d=1}^D q(\beta_{kd}) \right), \quad (2)$$

to approximate the posterior  $p(\boldsymbol{\theta}, \boldsymbol{\beta} | \mathbf{y})$ , so that the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior is minimized. Following a further approximation discussed in the next section, the factorized distribution allows for a closed-form expression of this variational objective, and thus tractable inference. Here we choose variational distributions from the same family as the prior:

$$\begin{aligned} q(\theta_{ik}) &= \text{Gamma}(\theta_{ik}; \gamma_{ik}, \chi_{ik}), \\ q(\beta_{kd}) &= \text{Gamma}(\beta_{kd}; \nu_{kd}, \lambda_{kd}). \end{aligned} \quad (3)$$

Minimizing the KL divergence is equivalent to maximizing the following variational objective:

$$\mathcal{L} = \mathbb{E}_q[\ln p(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\beta})] + H(q), \quad (4)$$

where  $H(q)$  is the entropy of the variational distribution  $q$ . We can optimize the variational objective using coordinate ascent via two approaches: batch inference, which requires processing of the entire dataset for every iteration; or stochastic inference, which only needs a small batch of data for each iteration and can be potentially scale to much larger datasets where batch inference is no longer computationally feasible.

### 4.1 Batch inference

Although the model in Equation (1) is not conditionally conjugate by itself, as demonstrated in [4] we can introduce latent random variables  $z_{idk} \sim \text{Poisson}(\theta_{ik} \beta_{kd})$  with the variational distribution being  $q(z_{idk}) = \text{Multi}(z_{id}; \phi_{id})$ , where  $z_{id} \in \mathbb{N}^K$ ,  $\phi_{idk} \geq 0$  and  $\sum_k \phi_{idk} = 1$ . This makes the model conditionally conjugate, which means that closed-form coordinate ascent updates are available.

Following the standard results of variational inference for conditionally conjugate model (e.g. [9]), we can obtain the updates for  $\theta_{ik}$ :

$$\begin{aligned} \gamma_{ik} &= a + \sum_{d=1}^D y_{id} \phi_{idk}, \\ \chi_{ik} &= ac + \sum_{d=1}^D \mathbb{E}_q[\beta_{kd}]. \end{aligned} \quad (5)$$

The scale  $c$  is updated as  $c^{-1} = \frac{1}{IK} \sum_{i,k} \mathbb{E}_q[\theta_{ik}]$ .



Similarly, we can obtain the updates for  $\beta_{kd}$ :

$$\begin{aligned}\nu_{kd} &= b + \sum_{i=1}^I y_{id} \phi_{idk}, \\ \lambda_{kd} &= b + \sum_{i=1}^I \mathbb{E}_q[\theta_{ik}].\end{aligned}\quad (6)$$

Finally, for the latent variables  $z_{idk}$ , the following update is applied:

$$\phi_{idk} \propto \exp\{\mathbb{E}_q[\ln \theta_{ik} \beta_{kd}]\}.\quad (7)$$

The necessary expectations for  $\theta_{ik}$  are:

$$\begin{aligned}\mathbb{E}_q[\theta_{ik}] &= \gamma_{ik} / \chi_{ik}, \\ \mathbb{E}_q[\ln \theta_{ik}] &= \psi(\gamma_{ik}) - \ln \chi_{ik},\end{aligned}\quad (8)$$

where  $\psi(\cdot)$  is the digamma function. The expectations for  $\beta_{kd}$  have the same form, but use  $\nu_{kd}$  and  $\lambda_{kd}$ .

## 4.2 Stochastic inference

Batch inference will alternate between updating  $\theta$  and  $\beta$  using the entire data at each iteration until convergence to a local optimum, which could be computationally intensive for large datasets. We can instead adopt stochastic optimization by selecting a subset (mini-batch) of the data at iteration  $t$ , indexed by  $B_t \subset \{1, \dots, I\}$ , and optimizing over a noisy version of the variational objective  $\mathcal{L}$ :

$$\mathcal{L}_t = \frac{I}{|B_t|} \sum_{i \in B_t} \mathbb{E}_q[\ln p(y_i, \theta_i | \beta)] + \mathbb{E}_q[\ln p(\beta)] + H(q).\quad (9)$$

By optimizing  $\mathcal{L}_t$ , we are optimizing  $\mathcal{L}$  in expectation.

The updates for weights  $\theta_{ik}$  and latent variables  $z_{idk}$  are essentially the same as batch inference, except that now we are only inferring weights for the mini-batch of data for  $i \in B_t$ . The optimal scale  $c$  is updated accordingly:

$$c^{-1} = \frac{1}{|B_t|K} \sum_{i \in B_t, k} \mathbb{E}_q[\theta_{ik}].\quad (10)$$

After alternating between updating weights  $\theta_{ik}$  and latent variables  $z_{idk}$  until convergence, we can take a gradient step, preconditioned by the inverse Fisher information matrix of variational distribution  $q(\beta_{kd})$ , to optimize  $\beta_{kd}$  (see [9] for more technical details),

$$\begin{aligned}\nu_{kd}^{(t)} &= (1 - \rho_t) \nu_{kd}^{(t-1)} + \rho_t \left( b + \frac{I}{|B_t|} \sum_{i \in B_t} y_{id} \phi_{idk} \right), \\ \lambda_{kd}^{(t)} &= (1 - \rho_t) \lambda_{kd}^{(t-1)} + \rho_t \left( b + \frac{I}{|B_t|} \sum_{i \in B_t} \mathbb{E}_q[\theta_{ik}] \right),\end{aligned}\quad (11)$$

where  $\rho_t > 0$  is a step size at iteration  $t$ . To ensure convergence [3], the following conditions must be satisfied:

$$\sum_{t=1}^{\infty} \rho_t = \infty, \quad \sum_{t=1}^{\infty} \rho_t^2 < \infty.\quad (12)$$

One possible choice of  $\rho_t$  is  $\rho_t = (t_0 + t)^{-\kappa}$  for  $t_0 > 0$  and  $\kappa \in (0.5, 1]$ . It has been shown [9] that this update corresponds to stochastic optimization with a natural gradient step, which better fits the geometry of the parameter space for probability distributions.

## 4.3 Generalizing to new songs

Once the latent factor  $\beta \in \mathbb{R}_+^{K \times D}$  is inferred, we can naturally divide it into two blocks: the VQ part  $\beta_{VQ} \in \mathbb{R}_+^{K \times J}$ , and the bag-of-tags part  $\beta_{BoT} \in \mathbb{R}_+^{K \times |V|}$ .

Given a new song, we can first obtain the VQ feature  $y_{VQ}$  and fit it with  $\beta_{VQ}$  to compute posterior of the weights  $p(\theta | y_{VQ}, \beta_{VQ})$ . We can approximate this posterior with the variational inference algorithm in Section 4.1 with  $\beta$  fixed. Then to predict tags, we can compute the expectation of the dot product between the weights  $\theta$  and  $\beta_{BoT}$  under the variational distribution:

$$\hat{y}_{BoT} = \mathbb{E}_q[\theta^T \beta_{BoT}].\quad (13)$$

Since for different songs the weights  $\theta$  may be scaled differently, before computing the dot product we normalize  $\mathbb{E}_q[\theta]$  so that it lives on the probability simplex. To do automatic tagging, we could annotate the song with top  $M$  tags according to  $\hat{y}_{BoT}$ . To compensate for a lack of diversity in the annotations, we adopt the same heuristic used in [7] by introducing a ‘‘diversity factor’’  $d$ : For each predicted score, we subtract  $d$  times the mean score for that tag. In our system, we set  $d = 3$ .

## 5. EVALUATION

We evaluate the model’s performance on an annotation task and a retrieval task using CAL500 [16] and Million Song Dataset (MSD) [2]. Unlike the CAL500 dataset where tracks are carefully-annotated, the Last.fm dataset [1] associated with MSD comes from real-world user tagging, and thus contains only weakly labelled data with a tagging vocabulary that is much larger and more diverse. We compare our results on these tasks with two other sets of codebook-based methods: Codeword Bernoulli Average (CBA) [7] and  $\ell_2$  regularized logistic regression [17]. Like the Poisson matrix factorization model, both methods are easy to train and can scale to relatively large dataset on a single machine. However, since both methods perform optimization in a batch fashion, we will later refer to them as ‘‘batch algorithms’’, along with the Poisson model with batch inference described in Section 4.1.

For the hyperparameters of the Poisson matrix factorization model, we set  $a = b = 0.1$ , and the number of latent factors  $K = 100$ . To learn the latent factors  $\beta$ , we followed the procedure in Section 4.1 for batch inference until the relative increase on the variational objective is less than 0.05%. For stochastic inference, we used a mini-batch size  $|B_t| = 1000$  unless otherwise specified and took a full pass of the randomly permuted data. As for the learning rate, we set  $t_0 = 1$  and  $\kappa = 0.6$ . All the source code in Python is available online<sup>2</sup>.

### 5.1 Annotation task

The purpose of annotation task is to automatically tag unlabelled songs. To evaluate the model’s ability for annotation, we computed the average per-tag precision, recall,

<sup>2</sup>[http://github.com/dawenl/stochastic\\_PMF](http://github.com/dawenl/stochastic_PMF)

and F-score on a test set. Per-tag precision is defined as the average fraction of songs that the model annotates with tag  $v$  that are actually labelled  $v$ . Per-tag recall is defined as the average fraction of songs that are actually labelled  $v$  that the model also annotates with tag  $v$ . F-score is the harmonic mean of precision and recall, and is one overall metric for annotation performance.

## 5.2 Retrieval task

The purpose of the retrieval task is, when given a query tag  $v$ , to provide a list of songs which are related to tag  $v$ . To evaluate the models' retrieval performance, for each tag in the vocabulary we ranked each song in the test set by the predicted score from Equation (13). We evaluated the area under the receiver-operator curve (AROC) and mean average precision (MAP) for each ranking. AROC is defined as the area under the curve, which plots the true positive rate against the false positive rate, and MAP is defined as the mean of the average precision (AP) for each tag, which is the average of the precisions at each possible level of recall.

## 5.3 Results on CAL500

Following the procedure similar to that described in [7, 17], we performed a 5-fold cross-validation to evaluate the annotation and retrieval performance on CAL500. We selected the top 78 tags, which are annotated more than 50 times in the dataset, and learned a codebook of size  $J = 2000$ . For the annotation task, we labelled each song with the top 10 tags based on the predicted score. Since CAL500 is a relatively small dataset, we only performed batch inference for Poisson matrix factorization model.

The results are reported in Table 1, which shows that the Poisson model has comparable performance on the annotation task, and does slightly worse on the retrieval task. As mentioned in Section 3, the Poisson matrix factorization model is particularly suitable for noisy and sparse data where 0's are not necessarily interpreted as explicit observations. However, this may not be the case for CAL500, as the vocabulary was well-chosen and the data was collected from a survey where the tagging quality is understandably higher than the actual tagging data in the real world, like the one from Last.fm. Therefore, this task cannot fully exploit the advantage brought by the Poisson model. Meanwhile, the amount of data in CAL500 is fairly small – the data  $\mathbf{y}$  fit to the model is simply a 502-by-2078 matrix. This prevents us from adopting stochastic inference, which will be shown being much more effective than batch inference even on a 10,000-song dataset in Section 5.4.

## 5.4 Results on MSD

To demonstrate the scalability of the Poisson matrix factorization model, we conducted experiments using MSD and the associated Last.fm dataset. To our knowledge, there has not been any previous work where music tagging results are reported on the MSD.

Model	Prec	Recall	F-score	AROC	MAP
CBA	0.41	0.24	0.29	0.69	0.47
$\ell_2$ LogRegr	0.48	0.26	0.34	0.72	0.50
PMF-Batch	0.42	0.23	0.30	0.67	0.45

**Table 1.** Results for the top 78 popular tags on CAL500, for Codeword Bernoulli Average (CBA),  $\ell_2$  regularized logistic regression ( $\ell_2$  LogRegr), and Poisson matrix factorization with batch inference (PMF-Batch). The results for CBA and  $\ell_2$  LogRegr are directly copied from [17].

Since the Last.fm dataset contains 522,366 unique tags, it is not realistic to build the model with all of them. We first selected the tags with more than 1,000 appearances and removed those which do not carry discriminative information (e.g. “my favorite”, “awesome”, “seen live”, etc.). Then we ran the stemming algorithm implemented in *NLTK*<sup>3</sup> to further reduce the potential duplications and correct for alternate spellings (e.g. “pop-rock” v.s. “pop rock”, “love song” v.s. “love songs”), which gave us a vocabulary of 561 tags. Using the default train/test artist split from MSD, we filtered out the songs which have been labelled with tags from the selected vocabulary. This gave us 371,209 songs for training. For test set, we further selected those which have at least 20 tags (otherwise, it is likely that this song is very weakly labelled). This gave us a test set of 2,757 songs. The feature we used is the Echo Nest's timbre feature, which is very similar to MFCC.

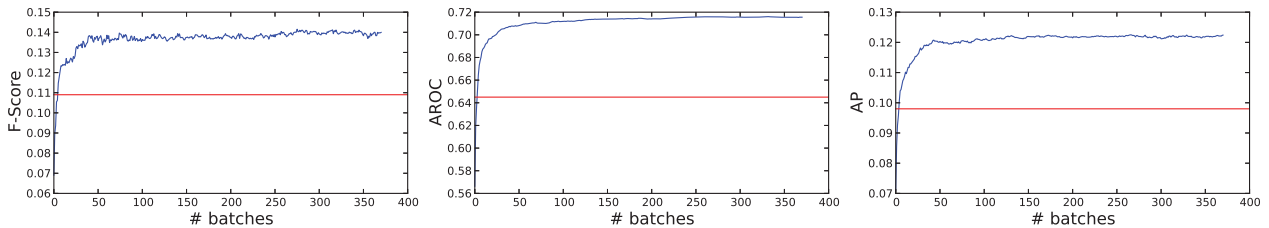
We randomly selected 10,000 songs as the data which can fit into the memory nicely for all the batch algorithms, and trained the following models with different codebook sizes  $J \in \{256, 512, 1024, 2048\}$ : Codeword Bernoulli Average (CBA),  $\ell_2$  regularized logistic regression ( $\ell_2$  LogRegr), Poisson matrix factorization with batch inference (PMF-Batch) and stochastic inference by a single pass of the data (PMF-Stoc-10K). Here we used batch size  $|B_t| = 500$  for PMF-Stoc-10K, as otherwise there will only be 10 mini-batches from the subset. However, given enough data, in general larger batch size will lead to relatively superior performance, since the variance of the noisy variational objective in Equation (9) is smaller. To demonstrate the effectiveness of the Poisson model on massive amount of data (exploiting the stochastic algorithm's ability to run without loading the entire dataset into memory), we also trained the model with the full training set with stochastic inference (PMF-Stoc-full). For the annotation task, we labelled each song with the top 20 tags based on the predicted score.

The results are reported in Table 2. In general, the performance of Poisson matrix factorization is comparably better for smaller codebook size  $J$ . Specifically, for stochastic inference, even if the amount of training data is relatively small, it is not only significantly faster than batch inference, but can also help improve the performance by quite a large margin. Finally, not surprisingly, PMF-Stoc-full dominates all the metrics, regardless of the size of the codebook, because it is able to learn from more data.

<sup>3</sup><http://www.nltk.org/>

Codebook size	Model	Precision	Recall	F-score	AROC	MAP
$J = 256$	CBA	0.112 (0.007)	0.121 (0.008)	0.116	0.695 (0.005)	0.112 (0.006)
	$\ell_2$ LogRegr	0.091 (0.008)	0.093 (0.006)	0.092	0.692 (0.005)	0.110 (0.006)
	PMF-Batch	0.113 (0.007)	0.105 (0.006)	0.109	0.647 (0.005)	0.094 (0.005)
	PMF-Stoc-10K	0.116 (0.007)	0.127 (0.007)	0.121	0.682 (0.005)	0.105 (0.006)
	PMF-Stoc-full	<b>0.127 (0.008)</b>	<b>0.143 (0.008)</b>	<b>0.134</b>	<b>0.704 (0.005)</b>	<b>0.115 (0.006)</b>
$J = 512$	CBA	0.120 (0.007)	0.127 (0.008)	0.124	0.689 (0.005)	0.117 (0.006)
	$\ell_2$ LogRegr	0.096 (0.008)	0.108 (0.007)	0.101	0.693 (0.005)	0.113 (0.006)
	PMF-Batch	0.111 (0.007)	0.108 (0.006)	0.109	0.645 (0.005)	0.098 (0.005)
	PMF-Stoc-10K	0.112 (0.007)	0.128 (0.007)	0.120	0.687 (0.005)	0.110 (0.006)
	PMF-Stoc-full	<b>0.130 (0.008)</b>	<b>0.154 (0.008)</b>	<b>0.141</b>	<b>0.715 (0.005)</b>	<b>0.122 (0.006)</b>
$J = 1024$	CBA	0.118 (0.007)	0.126 (0.007)	0.122	0.692 (0.005)	0.117 (0.006)
	$\ell_2$ LogRegr	0.113 (0.008)	0.129 (0.008)	0.120	0.698 (0.005)	0.115 (0.006)
	PMF-Batch	0.112 (0.007)	0.109 (0.006)	0.111	0.635 (0.005)	0.098 (0.006)
	PMF-Stoc-10K	0.111 (0.007)	0.127 (0.007)	0.118	0.687 (0.005)	0.111 (0.006)
	PMF-Stoc-full	<b>0.127 (0.008)</b>	<b>0.146 (0.008)</b>	<b>0.136</b>	<b>0.712 (0.005)</b>	<b>0.120 (0.006)</b>
$J = 2048$	CBA	<b>0.124 (0.007)</b>	0.129 (0.007)	0.127	0.689 (0.005)	0.117 (0.006)
	$\ell_2$ LogRegr	0.115 (0.008)	0.137 (0.008)	0.125	0.698 (0.005)	<b>0.118 (0.006)</b>
	PMF-Batch	0.109 (0.007)	0.110 (0.006)	0.110	0.637 (0.005)	0.098 (0.006)
	PMF-Stoc-10K	0.107 (0.007)	0.124 (0.007)	0.115	0.682 (0.005)	0.106 (0.006)
	PMF-Stoc-full	0.120 (0.007)	<b>0.147 (0.008)</b>	<b>0.132</b>	<b>0.712 (0.005)</b>	<b>0.118 (0.006)</b>

**Table 2.** Annotation (evaluated using precision, recall, and F-score) and retrieval (evaluated using area under the receiver-operator curve (AROC) and mean average precision (MAP)) performance on the Million Song Dataset with various codebook sizes, from Codeword Bernoulli Average (CBA),  $\ell_2$  regularized logistic regression ( $\ell_2$  LogRegr), Poisson matrix factorization with batch inference (PMF-Batch) and stochastic inference by a single pass of the subset (PMF-Stoc-10K) and full data (PMF-Stoc-full). One standard error is reported in the parenthesis.



**Figure 1.** Improvement in performance with the number of mini-batches consumed for the PMF-Stoc-full system with  $J = 512$ . Red lines indicate the performance of PMF-Batch which is trained on 10k examples; that system’s performance is exceeded after fewer than 5 mini-batches.

Figure 1 illustrates how the metrics improve as more data becomes available for the Poisson matrix factorization model, showing how the F-score, AROC, and MAP improve with the number of (1000-element) mini-batches consumed up to the entire 371k training set. We see that initial growth is rapid, thanks to the natural gradient, with much of the benefit obtained after just 50 batches. However, we see continued improvement beyond this; it is reasonable to believe that if more data becomes available, the performance can be further improved.

Table 3 contains information on the qualitative performance of our model. The tagging model works by capturing correlations between semantic tags and acoustic codewords in each latent factor  $\beta_k$ . As discussed, when a new song arrives with missing tag information, only the portion of  $\beta_k$  corresponding to acoustic codewords is used, and the semantic tag portion of  $\beta_k$  is used to make predictions of the missing tags. Similar to related topic models [9], we

can therefore look at the highly probable tags for each  $\beta_k$  to understand what portion of the acoustic codeword space is being captured by that factor, and whether it is musically coherent. We show an example of this in Table 3, where we list the top 7 tags from 9 latent factors  $\beta_k$  learned by our model with  $J = 512$ . We sort the tags according to expected relevance under the variational distribution  $\mathbb{E}_q[\beta_{kd}]$ . This shows which tags are considered to have high probability for a song that has the given factor expressed. As is evident, each factor corresponds to a particular aspect of a music genre. We note that other factors contained similarly coherent tag information.

## 6. DISCUSSION AND FUTURE WORK

We present a codebook-based scalable music tagging model with Poisson matrix factorization. The system learns the joint behavior of acoustic features and semantic tags, which

“Pop”	“Indie”	“Jazz”	“Classical”	“Metal”	“Reggae”	“Electronic”	“Experimental”	“Country”
pop	indie	chillout	piano	metal	reggae	house	instrumental	country
female vocal	rock	loungue	instrumental	death metal	funk	electro	ambient	classic country
dance	alternative	chill	ambient	thrash metal	funky	electronic	experimental	male vocal
electronic	indie rock	downtempo	classic	brutal death metal	dance	dance	electronic	blues
sexy	post punk	smooth jazz	beautiful	grindcore	hip-hop	electric house	psychedelic	folk
love	psychedelic	relax	chillout	heavy metal	party	techno	progressive	love songs
synth pop	new wave	ambient	relax	black metal	sexy	minimal	rock	americana

**Table 3.** Top 7 tags from 9 latent factors for PMF-Stoc-full with  $J = 512$ . For each factor, we assign the closest music genre on top. As is evident, each factor corresponds to a particular aspect of a music genre.

can be used to infer the most appropriate tags given the audio alone. The Poisson model is naturally less sensitive to zero values than some alternatives, making it a good match to “noisy” training examples derived from real users’ taggings, where the fact that no user has applied a tag does not necessarily imply that the term is irrelevant. By learning this model using stochastic variational inference, we are able to efficiently exploit much larger training sets than are tractable using batch approaches, making it feasible to learn from an entire set of over 370k tagged examples. Although much of the improvement comes in the earlier iterations, we see continued improvement implying this approach can benefit from much larger, effectively unlimited sources of tagged examples, as might be available on a commercial music service with millions of users.

There are a few areas where our model can be easily developed. For example, stochastic variational inference requires we set the learning rate parameters  $t_0$  and  $\kappa$ , which is application-dependent. By using adaptive learning rates for stochastic variational inference [13], model inference can converge faster and to a better local optimal solution. From a modeling perspective, currently the hyperparameters for weights  $\theta$  are fixed, indicating that the sparsity level of the weight for each song is assumed to be the same *a priori*. Alternatively we could put *song-dependent* hyper-priors on the hyperparameters of  $\theta$  to encode the intuition that some of the songs might have denser weights because more tagging information is available. This would offer more flexibility to the current model.

## 7. ACKNOWLEDGEMENTS

The authors would like to thank Matthew Hoffman for helpful discussion. This work was supported in part by NSF grant IIS-1117015.

## 8. REFERENCES

- [1] Last.fm dataset, the official song tags and song similarity collection for the Million Song Dataset. <http://labrosa.ee.columbia.edu/millionsong/lastfm>.
- [2] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere. The Million Song Dataset. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 591–596, 2011.
- [3] L. Bottou. Online learning and stochastic approximations. *On-line learning in neural networks*, 17(9), 1998.
- [4] A. T. Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009.
- [5] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic generation of social tags for music recommendation. In *Advances in Neural Information Processing Systems*, pages 385–392, 2007.
- [6] K. Ellis, E. Coviello, A. Chan, and G. Lanckriet. A bag of systems representation for music auto-tagging. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(12):2554–2569, 2013.
- [7] M. Hoffman, D. Blei, and P. Cook. Easy as CBA: A simple probabilistic model for tagging music. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 369–374, 2009.
- [8] M. Hoffman, D. Blei, and P. Cook. Bayesian nonparametric matrix factorization for recorded music. In *Proceedings of the 27th Annual International Conference on Machine Learning*, pages 439–446, 2010.
- [9] M. Hoffman, D. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.
- [10] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.
- [11] D. Liang, M. Hoffman, and D. Ellis. Beta process sparse non-negative matrix factorization for music. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 375–380, 2013.
- [12] J. Paisley, D. Blei, and M.I. Jordan. Bayesian nonnegative matrix factorization with stochastic variational inference. In E.M. Airoldi, D. Blei, E.A. Erosheva, and S.E. Fienberg, editors, *Handbook of Mixed Membership Models and Their Applications*. Chapman and Hall/CRC, 2015.
- [13] R. Ranganath, C. Wang, D. Blei, and E. Xing. An adaptive learning rate for stochastic variational inference. In *Proceedings of The 30th International Conference on Machine Learning*, pages 298–306, 2013.
- [14] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, pages 1257–1264, 2008.
- [15] D. Tingle, Y.E. Kim, and D. Turnbull. Exploring automatic music annotation with acoustically-objective tags. In *Proceedings of the international conference on Multimedia information retrieval*, pages 55–62. ACM, 2010.
- [16] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2):467–476, 2008.
- [17] B. Xie, W. Bian, D. Tao, and P. Chordia. Music tagging with regularized logistic regression. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 711–716, 2011.



Oral Session 2  
**Transcription**

This Page Intentionally Left Blank

# TEMPLATE ADAPTATION FOR IMPROVING AUTOMATIC MUSIC TRANSCRIPTION

Emmanouil Benetos<sup>†</sup>, Roland Badeau<sup>‡</sup>, Tillman Weyde<sup>†</sup> and Gaël Richard<sup>‡</sup>

<sup>†</sup> Department of Computer Science, City University London, UK  
{emmanouil.benetos.1, t.e.veyde}@city.ac.uk

<sup>‡</sup> Institut Mines-Télécom, Télécom ParisTech, CNRS LTCI, France  
{roland.badeau, gael.richard}@telecom-paristech.fr

## ABSTRACT

In this work, we propose a system for automatic music transcription which adapts dictionary templates so that they closely match the spectral shape of the instrument sources present in each recording. Current dictionary-based automatic transcription systems keep the input dictionary fixed, thus the spectral shape of the dictionary components might not match the shape of the test instrument sources. By performing a *conservative* transcription pre-processing step, the spectral shape of detected notes can be extracted and utilized in order to adapt the template dictionary. We propose two variants for adaptive transcription, namely for single-instrument transcription and for multiple-instrument transcription. Experiments are carried out using the MAPS and Bach10 databases. Results in terms of multi-pitch detection and instrument assignment show that there is a clear and consistent improvement when adapting the dictionary in contrast with keeping the dictionary fixed.

## 1. INTRODUCTION

Automatic music transcription (AMT) is defined as the process of converting an acoustic music signal into some form of music notation [3]. Subtasks of AMT include multi-pitch detection, onset/offset detection, and instrument identification. Recently, the vast majority of transcription approaches use *spectrogram factorization* methods such as non-negative matrix factorization (NMF) and probabilistic latent component analysis (PLCA), which attempt to decompose an input non-negative spectrogram into spectral templates and note activations (e.g. [2, 10, 17]). The spectral templates can either be pre-extracted and stored in a template dictionary [2, 17] or can be estimated using parametric spectral models [10]. An open problem with dictionary-based methods is that the templates might not match the spectral shape of the input instrument sources.

EB is supported by a City University London Research Fellowship. This work was supported by a Télécom ParisTech sabbatical grant.



© E. Benetos, R. Badeau, T. Weyde, and G. Richard.  
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Emmanouil Benetos, Roland Badeau, Tillman Weyde, and Gaël Richard. “Template adaptation for improving automatic music transcription”, 15th International Society for Music Information Retrieval Conference, 2014.

Also, unconstrained methods such as NMF and standard PLCA that jointly update the spectral templates and pitch activations can lead to the creation of non-informative bases, and thus, to poor transcription results. It has been shown (e.g. [3]) that the use of templates from the same instrument model or recording conditions can dramatically improve transcription performance.

Related work on automatically estimating or adapting templates for transcription includes [12], where the authors proposed a system for user-assisted (i.e. semi-automatic) music transcription in an NMF setting. The user can label a few notes in the recording; knowledge of the labelled notes can be used in order to create a dictionary that matches the input source. In addition, in [18], the authors propose a dictionary adaptation step within a sparse model that is suitable for single-instrument multi-pitch detection.

In this paper, we propose a method for template adaptation suitable for multiple-instrument polyphonic music transcription (supporting both multi-pitch detection and instrument assignment). The proposed method is based on a multiple-instrument transcription system using PLCA, and supporting tuning changes and frequency modulations. By performing a conservative transcription in a pre-processing step, notes are detected with a high degree of confidence and are used in order to expand the current template dictionary. An additional PLCA-based dictionary adaptation step can further refine the dictionary, so that it matches closely the input source(s). Two system variants are proposed, for single- and multiple-instrument transcription. Experiments using the MAPS [8] and Bach10 [7] databases show a consistent improvement in multi-pitch detection and instrument assignment performance when the proposed template adaptation method is used.

The outline of the paper is as follows. In Section 2, the proposed single-instrument transcription system is presented, with the multiple-instrument version presented in Section 3. The employed datasets, evaluation metrics, and results are detailed in Section 4. Finally, conclusions are drawn and future directions are indicated in Section 5.

## 2. SINGLE-INSTRUMENT SYSTEM

In the following, we describe a method for single-instrument polyphonic music transcription based on a dictionary of pre-extracted note templates, which is adapted in order to

match the input instrument source. The proposed system contains a “conservative” transcription pre-processing step in order to detect notes with a high degree of confidence, a dictionary adaptation step, and a final transcription step. The diagram of the proposed system can be seen in Fig. 1.

## 2.1 Pre-processing

As a pre-processing step, we perform an initial transcription which uses a fixed template dictionary (in which the templates might not be extracted from the same instrument source, model, or recording conditions). The main goal is to only detect notes for which we have a high degree of confidence; in order to achieve this, we perform a “conservative” transcription, as in [16], where the employed transcription system detects notes with high precision and low recall. In other words, the system returns few false alarms but might miss several notes present in the recording.

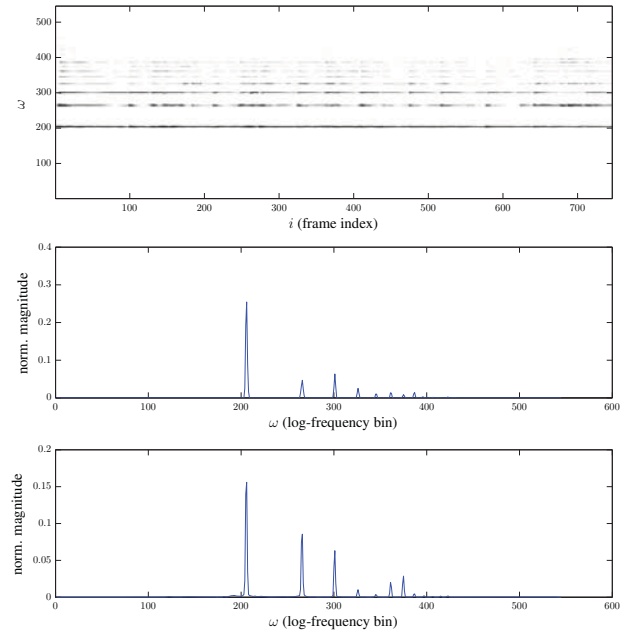
In order to perform the conservative transcription pre-processing step, we use the spectrogram factorization-based model of [2], which is based on probabilistic latent component analysis (PLCA) [14] and supports the use of a fixed template dictionary. It should be noted that the system in [2] ranked first in the MIREX transcription tasks [1]. The model of [2] takes as input a normalized log-frequency spectrogram  $V_{\omega,t} \in \mathbb{R}^{\Omega \times T}$  ( $\omega$  denotes frequency and  $t$  time) and approximates it as a bivariate probability distribution  $P(\omega, t)$ .  $P(\omega, t)$  is in turn decomposed as:

$$P(\omega, t) = P(t) \sum_{p,f,s} P(\omega|s,p,f) P_t(f|p) P_t(s|p) P_t(p) \quad (1)$$

where  $p, f, s$  denote pitch, log-frequency shifting, and instrument source (in the single-instrument case,  $s$  refers to instrument model), respectively.  $P(t)$  is the spectrogram energy (known quantity) and  $P(\omega|s,p,f)$  are pre-extracted spectral templates for pitch  $p$ , source/model  $s$ , which are also pre-shifted across log-frequency according to parameter  $f$ .  $P_t(f|p)$  is the time-varying log-frequency shifting for pitch  $p$ ,  $P_t(s|p)$  is the source contribution, and  $P_t(p)$  is the pitch activation. As a log-frequency representation we use the constant-Q transform [13] with 60 bins/octave, resulting in  $f \in [1, \dots, 5]$ , where  $f = 3$  is the ideal tuning position for the template (using equal temperament).

Using a fixed template dictionary, the parameters that need to be estimated are  $P_t(f|p)$ ,  $P_t(s|p)$ , and  $P_t(p)$ . This can be achieved using the expectation-maximization (EM) algorithm [5], with 15-20 iterations being typically sufficient. The resulting multi-pitch output is given by  $P(p, t) = P(t)P_t(p)$ .

In order to extract note events in spectrogram factorization-based AMT algorithms, typically thresholding is performed on the pitch activations ( $P(p, t)$  in this case). The value of the threshold  $\theta$  controls the levels of precision/recall. A low threshold has a high recall and low precision; the opposite occurs with a high threshold. By selecting a high value of  $\theta$ , in essence we perform a conservative transcription. The final output of the pre-processing step is a collection of pitches and time frames  $\{p_1, t_1\}, \{p_2, t_2\}, \dots, \{p_N, t_N\}$  which can be used in order to adapt the template dictionary.



**Figure 2.** Top: a collection of spectra  $\hat{V}^{(42)}$  (note D4) from piano recording ‘alb\_se2’ taken from the MAPS database (piano model: ENSTDkCl). Middle: extracted normalised template  $P(\omega|p = 42)$ . Bottom: a D4 template from piano model AkPnBcht from the MAPS database.

## 2.2 Template Adaptation

Given a collection of detected pitches, the first step regarding template adaptation is to collect the spectra that correspond to the aforementioned pitches in the recording. Thus, for each pitch  $p$  all time frames  $t_{ip}$  that contain that pitch are collected (where  $i = 1, \dots, N_p$  and  $N_p$  is the number of frames containing  $p$ ).

Subsequently, for each pitch  $p$  we create a collection of spectra where that pitch is observed:

$$\hat{V}^{(p)} = V_{\omega,t \in t_{ip}} \otimes \mathbf{h}_p \quad (2)$$

where  $\mathbf{h}_p$  is a harmonic comb that serves as an indicator function (setting to zero all frequency bins not belonging to pitch  $p$ ), and  $\otimes$  denotes elementwise multiplication. In other words,  $\hat{V}^{(p)} \in \mathbb{R}^{\Omega \times N_p}$  is a collection of the spectra corresponding to detected pitch  $p$  in the input recording.

Using information from  $\hat{V}^{(p)}$ , new spectral templates are created for each  $p$  that was detected in the conservative transcription step. In order to create the new templates, the standard PLCA algorithm is used with one component [14], with the input in each case being  $\hat{V}^{(p)}$ . The output for each  $p$  is a spectral template  $\mathbf{w}^{(p)}$  which can be used in order to expand the present dictionary.

Given that the conservative transcription step might not have detected all possible pitches present in the recording, information from the extracted templates can be used in order to estimate missing templates. As in the user-assisted case of [12], we can derive templates at missing pitches by simply shifting existing templates across log-frequency. Given a missing pitch template, we consider a neighbor-



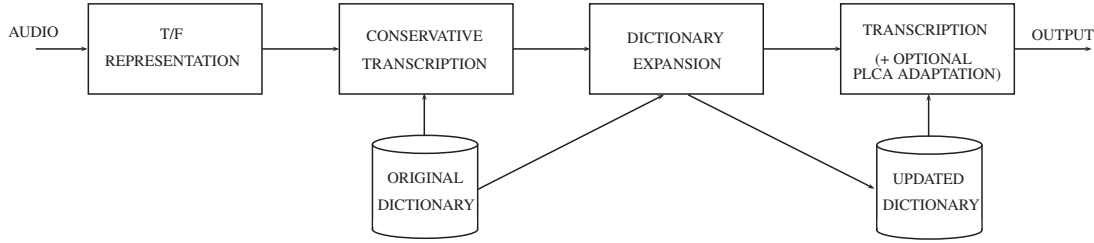


Figure 1. Proposed system diagram.

hood of up to 4 semitones; if a template exists in the neighborhood, it is shifted accordingly in order to estimate the missing template. Finally, the resulting template dictionary is pre-shifted across log-frequency over a semitone range in order to account for tuning deviations and frequency modulations. The output of the template adaptation step is normalized and denoted as  $P(\omega|s = s_{new}, p, f)$ , where  $s_{new}$  refers to the new instrument source that is added to the existing dictionary.

The template adaptation step is illustrated in Fig. 2, where a collection of extracted spectra for note D4 of a piano recording can be seen, along with the computed template, as well as with a template for the same note taken from a different piano model. By comparing the two piano spectra, the importance of adapting templates to the specific instrument source can be seen.

### 2.3 Transcription

Having created an expanded dictionary with a set of note templates taken from the instrument source present in the recording, the recording is re-transcribed using the new dictionary and the model of (1). In order to further adapt the extracted templates to the input source, an optional step is also applied on updating the new template set during the PLCA iterations. The modified iterative update rule is based on the work of [15] (which incorporated prior information on PLCA update rules) and is applied only for the new set of templates. It is formulated as:

$$\hat{P}(\omega|s_{new}, p, f) = \frac{\sum_t \alpha P_t(p, f, s_{new}|\omega) V_{\omega,t} + (1 - \alpha) P(\omega|s_{new}, p, f)}{\sum_{\omega,t} \alpha P_t(p, f, s_{new}|\omega) V_{\omega,t} + (1 - \alpha) P(\omega|s_{new}, p, f)} \quad (3)$$

where  $P_t(p, f, s|\omega)$  is the posterior of the model (defined in [2]), and  $\alpha$  is a parameter which controls the weight of the PLCA adaptation, with  $(1 - \alpha)$  giving weight to the set of extracted templates from the procedure of Section 2.2. In this work,  $\alpha$  is set to 0.05, thus the PLCA template adaptation is only slightly changing the shape of the templates (given that the model is unconstrained, giving a large weight to the PLCA adaptation step would result in non-meaningful templates).

Finally, the output of the transcription step is given by  $P(p, t) = P(t)P_t(p)$ . For converting the non-binary pitch activation into a binary piano-roll representation, as in [6] we perform thresholding on  $P(p, t)$  followed by a process removing note events with a duration less than 80ms.

### 3. MULTIPLE-INSTRUMENT SYSTEM

In dictionary-based multiple-instrument transcription, the dictionary typically consists of one or more sets of templates per instrument. Thus, in order to update dictionary templates for multiple instruments, modifications need to be made from the system presented in Section 2.

Regarding the pre-processing step, we still use the model of (1), which supports multiple-instrument transcription. In this case,  $s$  denotes instrument source. An advantage of the model of (1) is that it can produce an instrument assignment output (i.e. each detected note is assigned to a specific instrument). Thus, having estimated the unknown model parameters, the instrument assignment output for instrument  $s_{ins}$  is given by the following time-pitch representation:

$$P(s = s_{ins}, p, t) = P_t(s = s_{ins}|p)P_t(p)P(t) \quad (4)$$

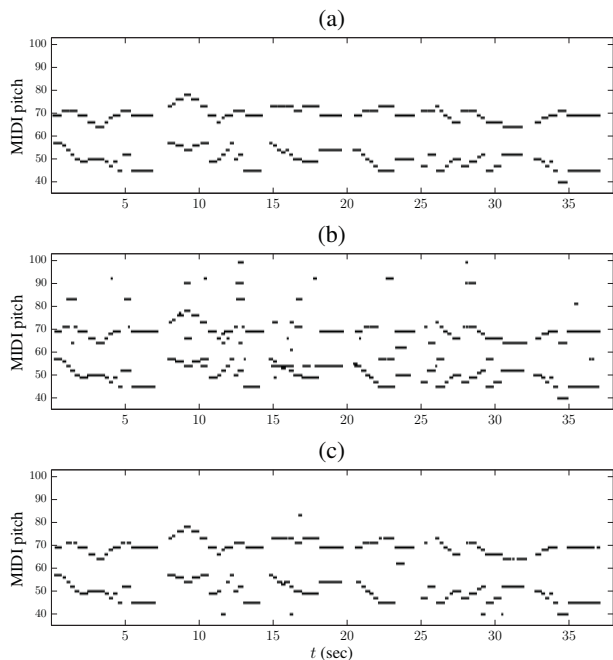
The representation  $P(s, p, t)$  can be thresholded in the same way as the pitch activation in order to derive a binary piano-roll representation of the notes produced by a specific instrument. Here, we perform “conservative” thresholding (i.e. use a high  $\theta$  value) for every instrument in  $P(s, p, t)$  in order to create a collection of detected pitches and time frames per instrument:

$$\{s_1, p_1, t_1\}, \{s_2, p_2, t_2\}, \dots, \{s_N, p_N, t_N\} \quad (5)$$

where  $s \in 1, \dots, S$ ,  $p \in 1, \dots, 88$ , and  $t \in 1, \dots, T$ .

For performing multi-instrument template adaptation, we collect all time frames  $t_{ips}$  that contain pitch  $p$  and instrument  $s$ . We create a collection of spectra  $\hat{V}^{(p,s)}$  where a pitch is observed for a specific instrument, in the same way as in (2). Using information from  $\hat{V}^{(p,s)}$ , new spectral templates are created for specific cases of  $s$  and  $p$  using the single-component PLCA algorithm. As in Section 2.2, templates at missing pitches are derived by translating existing templates across log-frequency. The output of the template adaptation step is denoted as  $P(\omega|s = \{s_{new1}, s_{new2}, \dots\}, p, f)$  where  $s_{new1}, s_{new2}, \dots$  denote the new sets of templates for the existing instruments.

Finally, the input recording is re-transcribed using the model of (1), by utilizing the expanded dictionary. We also apply the same optional PLCA-based dictionary adaptation step shown in Section 2.3. The multiple-instrument transcription system has two sets of outputs: the pitch activation  $P(p, t)$  (which is used for multi-pitch detection evaluation) and the instrument contribution  $P(s, p, t)$  (which is used for instrument assignment evaluation).



**Figure 3.** (a) The pitch ground truth for the bassoon-violin duet ‘Nun bitten’ from the Bach10 database. (b) The transcription piano-roll without template adaptation. (c) The transcription piano-roll with template adaptation.

An example of how template adaptation can improve transcription performance for a multiple-instrument recording (bassoon and violin) is given in Fig. 3, where the transcription output with template adaptation has significantly fewer false alarms compared with transcription without template adaptation (in which many extra detected notes can be seen in higher pitches).

## 4. EVALUATION

### 4.1 Datasets

For training the single-instrument system of Section 2, we used isolated note recordings from the ‘AkPnBcht’ and ‘Sp-tkBGCl’ piano models of the MAPS database [8]. We used the standard PLCA algorithm with one component [14] in order to extract a single template per note, covering the complete piano note range. For testing the single-instrument system, we used thirty piano segments of 30s duration from the MAPS database from the ‘ENSTDkCl’ piano model; the test dataset has in the past been used for multi-pitch evaluation (e.g. [2,4,19]). For comparative purposes, we also extracted training templates from the same test source (‘ENSTDkCl’).

For training the multiple-instrument system of Section 3, we used isolated note samples of bassoon and violin from the RWC database [11], covering the complete note range of the instruments. For testing the multiple-instrument system, we created ten duets of bassoon-violin, mixed from single instrument tracks from the multi-track Bach10 dataset [7]. The duration of the recordings varies from 25-41sec. For comparative purposes, we also extracted dictionary tem-

System	$Pre_n$	$Rec_n$	$F_n$
C1	66.41%	48.41%	55.33%
C2	68.07%	48.80%	56.26%
C3	67.84%	49.38%	56.56%
C4 (oracle)	70.43%	50.35%	58.17%

**Table 1.** Multi-pitch detection results for the single-instrument system using the MAPS database.

plates for bassoon and violin from the single instrument tracks of the Bach10 database, in order to demonstrate the upper performance limit of the transcription system.

### 4.2 Metrics

For evaluating the performance of the proposed systems for multi-pitch detection, we employ onset-only note-based transcription metrics, which are used in the MIREX note tracking task [1]. A detected note is considered correct if its pitch matches a ground truth pitch and its onset is within a 50ms tolerance of a ground-truth onset. The resulting note-based precision, recall, and F-measure are defined as:

$$Pre_n = \frac{N_{tp}}{N_{sys}} \quad Rec_n = \frac{N_{tp}}{N_{ref}} \quad F_n = \frac{2Rec_nPre_n}{Rec_n + Pre_n} \quad (6)$$

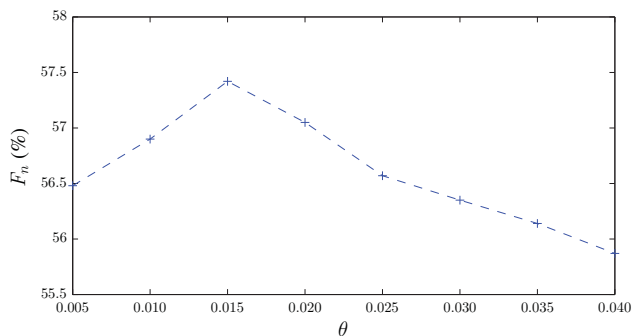
where  $N_{tp}$  is the number of correctly detected pitches,  $N_{sys}$  is the number of pitches detected by the system, and  $N_{ref}$  is the number of reference pitches.

For the instrument assignment evaluations we use the pitch ground-truth of each instrument separately (compared with the instrument-specific piano-roll output of the system), and compute the F-measure metrics for bassoon ( $F_b$ ) and violin ( $F_v$ ).

### 4.3 Results - Single Instrument Evaluation

For single-instrument transcription evaluation using the 30 MAPS recordings, results are shown in Table 1 using four different system configurations. Configuration C1 corresponds to the system without template adaptation; C2 to the system with template adaptation; C3 to the system with template adaptation using both the creation of the new dictionary plus the PLCA update of the dictionary, as shown in Section 2.2. Finally, C4 refers to comparative experiments without template adaptation, but using templates from the same instrument source (‘ENSTDkCl’ model in the single-instrument case), which is meant to demonstrate the upper performance limit of the transcription system.

From the single-instrument multi-pitch detection results, it can be seen that an improvement of +0.9% in terms of  $F_n$  is reported when using the template adaptation procedure; the improvement rises to +1.2% when also using the PLCA dictionary adaptation updates. The performance difference between the original C1 system (without knowledge of the source templates) and the ‘optimal’ system (C4) which contains templates from the same test source is 2.8%; thus, the proposed template adaptation steps can help bridge the gap, without requiring any knowledge of



**Figure 4.** Multi-pitch detection results on the MAPS-ENSTDkCl set using different values of  $\theta$ .

the test instrument source. Regarding precision and recall, in all cases it can be seen that the transcription system has fewer false alarms than missed note detections. The proposed template adaptation steps help in equally improving precision and recall.

In order to determine the value of the conservative transcription threshold  $\theta$ , we used a training subset of 10 recordings from the MAPS ‘SptkBGCl’ models; the value of  $\theta = 0.028$  was selected by maximising  $Pre_n$ . In Figure 4, transcription performance on the MAPS-ENSTDkCl set is reported by selecting various values for  $\theta$ . It can be seen that the transcription performance can reach up to  $F_n=57.4\%$  with  $\theta = 0.015$ , which enforces the argument that the proposed template adaptation method can successfully adapt dictionary templates so that they match the input instrument source.

Another comparison for the single-instrument system is made, where the dictionary derived from Section 2.2 replaces the dictionary of instrument ‘SptkBGCl’ (instead of expanding the original dictionary). The resulting  $F_n$  is 55.88%, indicating that expanding the dictionary leads to better results compared with replacing the dictionary. It should also be noted that the achieved transcription performance outperforms the system in [19] which reports a frame-based F-measure of 52.4%, whereas the template adaptation system reports a frame-based  $F$  of 59.73%. Finally, no rigorous figures for statistical significance of the results can be given since all signal frames cannot be considered as independent samples. However, the reported tests are run on several thousands of frames which leads, if the samples were independent, to a statistically significant difference of the order of 0.6% (with 95% confidence).

#### 4.4 Results - Multiple Instrument Evaluation

For multiple-instrument evaluation, we also use the four different system configurations that were used for single-instrument transcription. For system configuration C3, we perform the PLCA dictionary update using 3 variants: by updating the bassoon only, by updating the violin only, or by updating both dictionaries. Transcription results for the multiple-instrument case are shown in Table 2.

It can be seen that without any template adaptation (C1),  $F_n = 67.72\%$ ; by performing the proposed template adaptation step (C2), the multi-pitch detection F-measure im-

System	$Pre_n$	$Rec_n$	$F_n$	$F_b$	$F_v$
C1	64.79%	71.20%	67.72%	70.19%	42.10%
C2	69.71%	75.72%	72.51%	70.81%	45.98%
C3 (violin)	70.02%	75.41%	72.50%	70.54%	44.51%
C3 (bassoon)	72.49%	77.67%	74.90%	68.77%	45.87%
C3 (both)	71.30%	77.37%	74.11%	67.57%	44.08%
C4 (oracle)	74.90%	82.94%	78.64%	81.25%	62.05%

**Table 2.** Multi-pitch detection and instrument assignment results for the multiple-instrument system using the Bach10 dataset.

proves by +4.8%.

By performing template adaptation with C3 which also includes the PLCA update rule of (3), although no performance gain is obtained over the C2 configuration for the violin updates, there is a +2.4% improvement over C2 when updating the bassoon dictionary only. Finally, when updating both dictionaries, there is a performance drop for  $F_b$  and  $F_v$  over the C2 configuration (but the system still outperforms the original C1 system). The performance of the PLCA-based dictionary updates can be explained by the fact that the update rule of (3) might combine the observed spectra from both instruments and produce dictionaries that might represent a combination of the two instruments. Finally, the C4 system represents the upper performance limit, which is +11.7% higher than when using a dictionary from a different instrument models or recording conditions. It can be seen that the proposed template adaptation methods help in bridging that performance gap, resulting in a dictionary that closely matches the test instrument sources.

Regarding instrument assignment performance, in all cases the bassoon note identification reports better results compared to violin note identification. It can be seen that with the proposed template adaptation, the bassoon identification remains relatively constant (a small improvement of +0.6% is reported when comparing C1 with C2). On the other hand, violin identification improves by +3.9%; this indicates that the RWC bassoon templates closely matched the Bach10 bassoon models, whereas the violin RWC templates could greatly benefit from template adaptation.

By comparing the MAPS and Bach10 evaluations, an observation can be made that the performance improvement using the proposed template adaptation method depends on the mismatch between the original dictionary and the spectral shape of the instruments present in the recordings. Thus, the 11.7% performance gap for the Bach10 dataset led to a greater improvement for the template adaptation method compared to the 2.8% performance gap reported for the MAPS dataset (which led to a smaller, yet consistent improvement when using the proposed template adaptation method).

## 5. CONCLUSIONS

In this paper, we proposed a novel method for template adaptation for automatic music transcription, that can be used in dictionary-based systems. We utilized a multiple-

instrument transcription system based on probabilistic latent component analysis, and performed a conservative transcription pre-processing step in order to detect notes with a high confidence. Based on the initial transcription, the spectra of the detected notes are collected, processed, and are used in order to create a new dictionary that closely matches the spectral characteristics of the input instrument source(s). Both single-instrument and multi-instrument variants of the proposed method are presented and evaluated, in terms of multi-pitch detection and instrument assignment. Experimental results using the MAPS and Bach10 datasets show that there is a clear and consistent performance improvement when using the proposed template adaptation method, especially when there is a large discrepancy between the original dictionary and the spectral characteristics of the test instrument sources.

In the future, we will evaluate the proposed system using multiple-instrument recordings with more than two instruments. Parametric models (such as source-filter models) will also be investigated for updating the note templates, along with adaptive methods for deriving the conservative transcription threshold. We also plan to evaluate the proposed system in the next MIREX evaluations [1]. Finally, the proposed template adaptation steps will also be evaluated in the context of score-informed source separation using spectrogram factorization models [9].

## 6. REFERENCES

- [1] Music Information Retrieval Evaluation eXchange (MIREX). <http://music-ir.org/mirexwiki/>.
- [2] E. Benetos, S. Cherla, and T. Weyde. An efficient shift-invariant model for polyphonic music transcription. In *6th Int. Workshop on Machine Learning and Music*, Prague, Czech Republic, September 2013.
- [3] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: challenges and future directions. *J. Intelligent Information Systems*, 41(3):407–434, December 2013.
- [4] J. J. Carabias-Orti, T. Virtanen, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Cañadas-Quesada. Musical instrument sound multi-excitation model for non-negative spectrogram factorization. *IEEE J. Selected Topics in Signal Processing*, 5(6):1144–1158, 2011.
- [5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society*, 39(1):1–38, 1977.
- [6] A. Dessen, A. Cont, and G. Lemaitre. Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence. In *11th Int. Society for Music Information Retrieval Conf.*, pages 489–494, 2010.
- [7] Z. Duan, B. Pardo, and C. Zhang. Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Trans. Audio, Speech, and Language Processing*, 18(8):2121–2133, 2010.
- [8] V. Emiya, R. Badeau, and B. David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Trans. Audio, Speech, and Language Processing*, 18(6):1643–1654, 2010.
- [9] S. Ewert, B. Pardo, M. Müller, and M. D. Plumbley. Score-informed source separation for musical audio recordings. *IEEE Signal Processing Magazine*, 31(3):116–124, May 2014.
- [10] B. Fuentes, R. Badeau, and G. Richard. Harmonic adaptive latent component analysis of audio and application to music transcription. *IEEE Trans. Audio, Speech, and Language Processing*, 21(9):1854–1866, 2013.
- [11] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: music genre database and musical instrument sound database. In *International Conference on Music Information Retrieval*, October 2003.
- [12] H. Kirchhoff, S. Dixon, and Anssi Klapuri. Missing template estimation for user-assisted music transcription. In *IEEE Int. Conf. Audio, Speech and Signal Processing*, pages 26–30, 2013.
- [13] C. Schörkhuber and A. Klapuri. Constant-Q transform toolbox for music processing. In *7th Sound and Music Computing Conf.*, Barcelona, Spain, July 2010.
- [14] M. Shashanka, B. Raj, and P. Smaragdis. Probabilistic latent variable models as nonnegative factorizations. *Computational Intelligence and Neuroscience*, 2008. Article ID 947438.
- [15] P. Smaragdis and G. Mysore. Separation by “humming”: user-guided sound extraction from monophonic mixtures. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 69–72, New Paltz, USA, October 2009.
- [16] D. Tidhar, M. Mauch, and S. Dixon. High precision frequency estimation for harpsichord tuning classification. In *IEEE Int. Conf. Audio, Speech and Signal Processing*, pages 61–64, Dallas, USA, March 2010.
- [17] F. Weninger, C. Kirst, B. Schuller, and H.-J. Bungartz. A discriminative approach to polyphonic piano note transcription using supervised non-negative matrix factorization. In *IEEE Int. Conf. Audio, Speech and Signal Processing*, pages 6–10, May 2013.
- [18] T. B. Yakar, P. Sprechmann, R. Litman, A. M. Bronstein, and G. Sapiro. Bilevel sparse models for polyphonic music transcription. In *14th Int. Society for Music Information Retrieval Conf.*, pages 65–70, 2013.
- [19] K. Yoshii and M. Goto. Infinite composite autoregressive models for music signal analysis. In *13th Int. Society for Music Information Retrieval Conf.*, pages 79–84, October 2012.

# NOTE-LEVEL MUSIC TRANSCRIPTION BY MAXIMUM LIKELIHOOD SAMPLING

**Zhiyao Duan**

University of Rochester  
Dept. Electrical and Computer Engineering  
zhiyao.duan@rochester.edu

**David Temperley**

University of Rochester  
Eastman School of Music  
dtemperley@esm.rochester.edu

## ABSTRACT

Note-level music transcription, which aims to transcribe note events (often represented by pitch, onset and offset times) from music audio, is an important intermediate step towards complete music transcription. In this paper, we present a note-level music transcription system, which is built on a state-of-the-art frame-level multi-pitch estimation (MPE) system. Preliminary note-level transcription achieved by connecting pitch estimates into notes often lead to many spurious notes due to MPE errors. In this paper, we propose to address this problem by randomly sampling notes in the preliminary note-level transcription. Each sample is a subset of all notes and is viewed as a note-level transcription candidate. We evaluate the likelihood of each candidate using the MPE model, and select the one with the highest likelihood as the final transcription. The likelihood treats notes in a transcription as a whole and favors transcriptions with less spurious notes. Experiments conducted on 110 pieces of J.S. Bach chorales with polyphony from 2 to 4 show that the proposed sampling scheme significantly improves the transcription performance from the preliminary approach. The proposed system also significantly outperforms two other state-of-the-art systems in both frame-level and note-level transcriptions.

## 1. INTRODUCTION

Automatic Music Transcription (AMT) is one of the fundamental problems in music information retrieval. Generally speaking, AMT is the task of converting a piece of music audio into a musical score. A complete AMT system needs to transcribe both the pitch and rhythmic content [5]. On transcribing the pitch content, AMT can be performed at three levels from low to high: frame-level, note-level, and stream-level [7]. Frame-level transcription (also called *multi-pitch estimation*) aims to estimate concurrent pitches and instantaneous polyphony in each time frame. Note-level transcription (also called *note tracking*) transcribes notes, which are characterized not only by pitch,

but also by onset and offset. Stream-level transcription (also called *multi-pitch streaming*) organizes pitches (or notes) into streams according to their instruments. From the frame-level to the stream-level, more parameters and structures need to be estimated, and the system is closer to a complete transcription system.

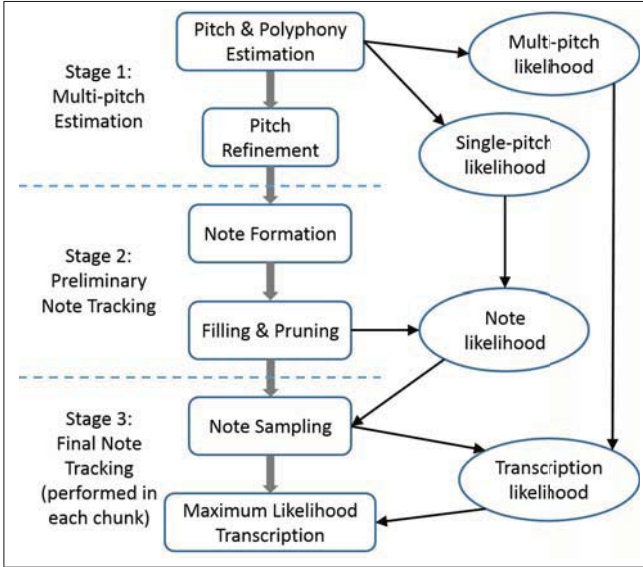
While there are many systems dealing with frame-level music transcription, only a few transcribe music at the note level [5]. Among these systems, most are built based on frame-level pitch estimates. The simplest way to convert frame-level pitch estimates to notes is to connect consecutive pitches into notes [4, 9, 15]. During this process, non-significant errors in frame-level pitch estimation can cause significant note tracking errors. False alarms in pitch estimates will cause many notes that are too short, while misses can break a long note into multiple short ones. To alleviate these errors, researchers often fill the small gaps to merge two consecutive notes with the same pitch [2, 7], and apply minimum length pruning to remove too-short notes [4, 6, 7]. This idea has also been implemented with more advanced techniques such as hidden Markov models [12]. Besides the abovementioned methods that are entirely based on frame-level pitch estimates, some methods utilize other information in note tracking, such as onset information [10, 14] and musicological information [13, 14].

In this paper, we propose a new note-level music transcription system. It is built based on an existing multi-pitch estimation method [8]. In [8], a multi-pitch likelihood function was defined and concurrent pitches were estimated in a maximum likelihood fashion. This likelihood function tells how well the set of pitches as a whole fit to the audio frame. In this paper, we modify [8] to also define a single-pitch likelihood function. It tells the likelihood (salience) that a pitch is present in the audio frame. Then preliminary note tracking is performed by connecting consecutive pitches into notes and removing too-short notes. The likelihood of each note is calculated as the product of the likelihood of all its pitches. The next step is the key step in the proposed system. We randomly sample subsets of notes according to their likelihood and lengths. Each subset is treated as a possible note-level transcription. The likelihood of such a transcription is then defined as the product of its multi-pitch likelihood in each frame. Finally, the transcription with the maximum likelihood is returned as the output of the system. We carried out experiments on the Bach10 dataset [8] containing Bach chorales



© Zhiyao Duan, David Temperley.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Zhiyao Duan, David Temperley. "Note-level Music Transcription by Maximum Likelihood Sampling", 15th International Society for Music Information Retrieval Conference, 2014.



**Figure 1.** System overview of the proposed note-level transcription system.

with different polyphony. Experiments show that the proposed system significantly improves the transcription performance from the preliminary transcription, and significantly outperforms two state-of-the-art systems at both the note level and frame level on the dataset.

## 2. PROPOSED SYSTEM

Figure 1 illustrates the overview of the system. It consists of three main stages: multi-pitch estimation, preliminary note tracking, and final note tracking. The first stage is based on [8] with some modifications. The second stage adopts the common filling/pruning strategies used in the literature to convert pitches into notes. The third stage is the main contribution of the paper. Figure 2 shows transcription results obtained at different stages of the system on a piece of J.S. Bach 4-part chorale.

### 2.1 Multi-pitch Estimation

In [8], Duan et al. proposed a maximum likelihood method to estimate pitches from the power spectrum of each time frame. In the maximum likelihood formulation, pitches (and the polyphony) are the parameters to be estimated while the power spectrum is the observation. The likelihood function  $L_{mp}(\{p_1, \dots, p_N\})$  describes how well a set of  $N$  pitches  $\{p_1, \dots, p_N\}$  as a whole fit with the observed spectrum, and hence is called a *multi-pitch likelihood* function. The power spectrum is represented as peaks and the non-peak region, and the likelihood function is defined for both parts. The peak likelihood favors pitch sets whose harmonics can explain peaks, while the non-peak region likelihood penalizes pitch sets whose harmonic positions are in the non-peak region. Parameters of the likelihood function were trained from thousands of musical chords mixed with note samples whose ground-truth pitches were pre-calculated. The maximum likelihood estimation process uses an iterative greedy search strategy.

It starts from an empty pitch set, and in each iteration the pitch candidate that results in the highest multi-pitch likelihood increase is selected. The process is terminated by thresholding on the likelihood increase, which also serves for polyphony estimation. After estimating pitches in each frame, a pitch refinement step that utilizes contextual information is performed to remove inconsistent errors.

In this paper, we use the same method to perform MPE in each frame. Differently, we change the instantaneous polyphony estimation parameter settings to achieve a high recall rate of the pitch estimates. This is because the note sampling module in Stage 3 will only remove false alarm notes but cannot add back missing notes (detailed explanation in Section 2.3). In addition, we also calculate a *single-pitch likelihood*  $L_{sp}(p)$  for each estimated pitch  $p$ . We define it as the multi-pitch likelihood plugged in with the single pitch, i.e.,  $L_{sp}(p) = L_{mp}(\{p\})$ . This likelihood describes how well the single pitch can explain the mixture spectrum, which apparently will not be very good. But from another perspective, this likelihood can be viewed as a salience of the pitch. One important property of multi-pitch likelihood is that it is not additive, i.e., the multi-pitch likelihood of a set of pitches is usually much smaller than the sum of their single-pitch likelihoods:

$$L_{mp}(\{p_1, \dots, p_N\}) < \sum_{i=1}^N L_{mp}(\{p_i\}) = \sum_{i=1}^N L_{sp}(p_i) \quad (1)$$

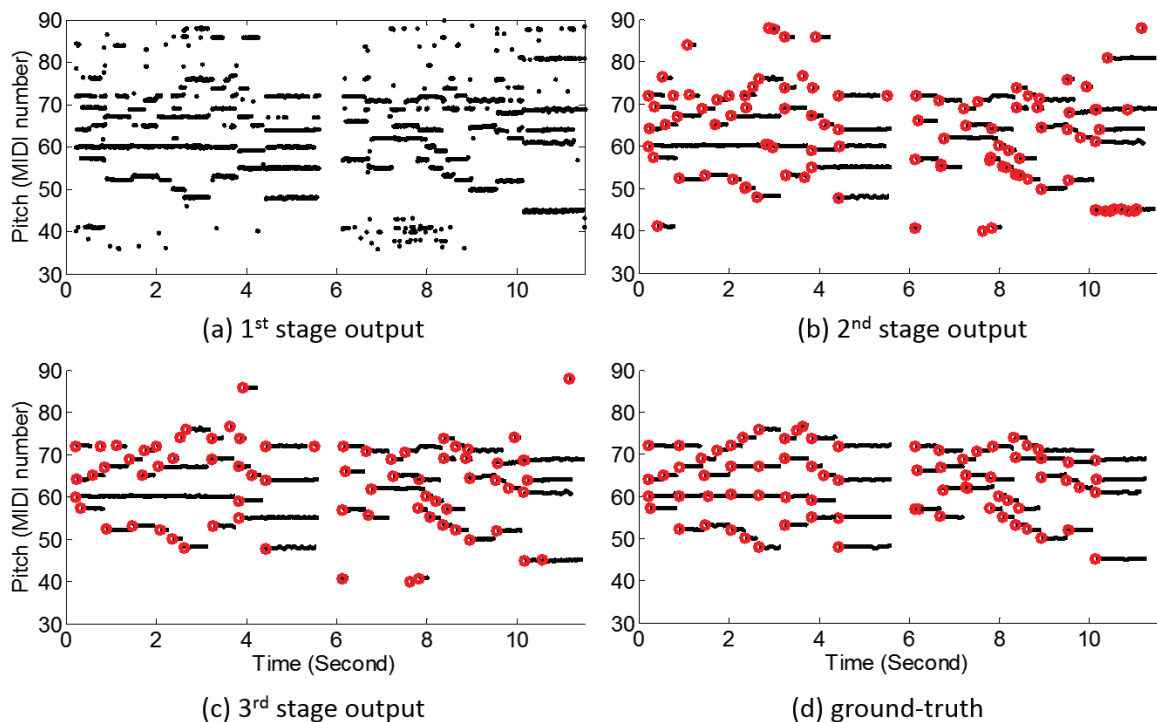
The reason is that the multi-pitch likelihood definition in [8] considers the interaction between pitches. For example, in the peak likelihood definition, a peak will be explained by only one pitch in the pitch set, the one whose corresponding harmonic gives the best fit to the frequency and amplitude of the peak, even if the peak could be explained by multiple pitches. In other words, the single-pitch likelihood considers each pitch independently while the multi-pitch likelihood considers the set as a whole.

The reason of calculating the single-pitch likelihood is because we need to calculate a likelihood (salience) for each note in the second stage, which is further because we need to sample notes using their likelihood in the third stage. Since pitches in the same frame belong to different notes, we need to figure out the likelihood (salience) of each pitch instead of the likelihood of the whole pitch set.

Figure 2(a) shows the MPE result on the example piece. Compared to the ground-truth in (d), it is quite noisy and contains many false alarm pitches, although the main notes can be inferred visually.

### 2.2 Preliminary Note Tracking

In this stage, we implement a preliminary method to connect pitches into notes, with the ideas of filling and pruning that were commonly used in the literature [2, 4, 6, 7]. We first connect pitches whose frequency difference is less than 0.3 semitones and time difference is less than 100 ms. Each connected component is then viewed as a note. Then notes shorter than 100 ms are removed. The 0.3 semitones threshold corresponds to the range within which the pitch



**Figure 2.** Transcription results on the first 11 seconds of *Ach Lieben Christen*, a piece of 4-part chorales by J.S. Bach. In (a), each pitch is plotted as a point. In (b)-(d), each note is plotted as a line whose onset is marked by a red circle.

often fluctuates within a note, while the 100 ms threshold is a reasonable length of a fast note, as it is the length of a 32nd note in music with a tempo of 75 beats per minute.

Each note is characterized by its pitch, onset, offset, and *note likelihood*. The onset and offset times are the time of the first and last pitch in the note, respectively. The pitch and likelihood are calculated by averaging the pitches and single-pitch likelihood values of all the pitches within the note. Again, this likelihood describes the saliency of the note in the audio.

Figure 2(b) shows the preliminary note tracking result. Compared to (a), many noisy isolated pitches have been removed. However, compared to (d), there are still a number of spurious notes, caused by consistent MPE errors (e.g., the long spurious note starting at 10 seconds around MIDI number 80, and a shorter note starting at 4.3 seconds around MIDI number 60). A closer look tells us that both notes and many other spurious notes are higher octave errors of some already estimated notes. This makes sense as octave errors take about half of all errors in MPE [8].

Due to the spurious notes, the instantaneous polyphony constraint is often violated. The example piece has four monophonic parts and at any time there should be no more than four pitches. However, it is often to see more than four notes going simultaneous in Figure 2(b) (e.g., 0-1 seconds, 4-6 seconds, and 10-11 seconds). On the other hand, these spurious notes are hard to remove if we consider them independently: They are long enough from being pruned by the minimum length; They also have high enough likelihood, as the note likelihood is the average likelihood of its pitches. Therefore, we need to consider the interaction be-

tween different notes to remove these spurious notes. This leads to the next stage of the system.

## 2.3 Final Note Tracking

The idea of this stage is quite simple. Thanks to the MPE algorithm in Stage 1, the transcription obtained in Stage 2 inherits the high recall and low precision property. Therefore, a subset of the notes that do not contain many spurious notes but contain almost all correct notes must be a better transcription. The only question now is how can we know which subset is a good transcription. This question can be addressed by an exploration-evaluation strategy: we first explore a number of subsets, and then we evaluate these subsets according to some criterion. But there are two problems of this strategy: 1) how can we efficiently explore the subsets? The number of all subsets is two to the power of the number of notes, hence it is inefficient to enumerate all the subsets. 2) What criterion should we use to evaluate the subsets? If our criterion considers notes independently, then it would not work well, as the spurious notes are hard to distinguish from correct notes in terms of individual note properties such as length and likelihood.

### 2.3.1 Note Sampling

Our idea to address the exploration problem is to perform note sampling. We randomly sample notes without replacement according to their weights. The weight equals to the product of the note length and the inverse of the negative logarithmic note likelihood. Essentially, longer notes with higher likelihood are more likely to be sampled into

the subset. In this way, we can explore different note subsets, and can guarantee that notes contained in each subset are mostly correct. During the sampling, we also consider the instantaneous polyphony constraint. A note will not be sampled if adding it to the subset would violate the instantaneous polyphony constraint. The sampling process stops if there is no valid note to sample any more.

We perform the sampling process  $M$  times to generate  $M$  subsets of the notes output in Stage 2. Each subset is viewed as a transcription candidate. We then evaluate the *transcription likelihood* for each candidate and select the one with the highest likelihood. The transcription likelihood is defined as the product of the multi-pitch likelihood of all time frames in the transcription. Since multi-pitch likelihood considers interactions between simultaneous pitches, the transcription likelihood also considers interactions between simultaneous notes. This can help remove spurious notes which are higher octave errors of some correctly transcribed notes. This is because all the peaks that a higher octave error pitch can explain can also be explained by the correct pitch, hence having the octave error pitch in addition to the correct pitch would not increase the multi-pitch likelihood much.

### 2.3.2 Chunking

The number of subsets (i.e., the sampling space) increases with the number of notes exponentially. If we perform sampling on an entire music piece that contains hundreds of notes, it is likely to require many times of sampling to reach a good subset (i.e., transcription candidate). In order to reduce the sampling space, we segment the preliminary note tracking transcription into one-second long non-overlapping chunks and perform sampling and evaluation in each chunk. Finally, selected transcriptions of different chunks are merged together to get the final transcription of the entire piece. Notes that span across multiple chunks can be sampled in all the chunks, and they will appear in the final transcription if they appear in the selected transcription of some chunk. Depending on the tempo and polyphony of the piece, the number of notes within a chunk can be different. For the 4-part Bach chorales tested in this paper, there are about 12 notes per chunk, and we found sampling 100 subsets gives good accuracy and efficiency.

Figure 2(c) shows the final transcription of the system. We can see that many spurious notes are removed from (b) while most correct notes remain, resulting in a much better transcription. The final transcription is very close to the ground-truth transcription.

## 3. EXPERIMENTS

### 3.1 Data Set

We use the Bach10 dataset [8] to evaluate the proposed system. This dataset consists of real musical instrumental performances of ten pieces of J.S. Bach four-part chorales. Each piece is about thirty seconds long and was performed by a quartet of instruments: violin, clarinet, tenor saxophone and bassoon. Both the frame-level and note-level

ground-truth transcriptions are provided with the dataset. In order to evaluate the system on music pieces with different polyphony, we use the dataset-provided matlab script to create music pieces with different polyphony, which are different combinations of the four parts of each piece. Six duets, four trios and one quartet for each piece was created, totaling 110 pieces of music with polyphony from 2 to 4.

### 3.2 Evaluation Measure

We evaluate the proposed transcription system with commonly used note-level transcription measures [1]. A note is said to be correctly transcribed, if it satisfies both the pitch condition and the onset condition: its pitch is within a quarter tone from the pitch of the ground-truth note, and its onset is within 50 ms from the ground-truth onset. Offset is not considered in determining correct notes. Then precision, recall, and F-measure are defined as

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN}, F = \frac{2PR}{(P + R)}, \quad (2)$$

where  $TP$  (true positives) is the number of correctly transcribed notes,  $FP$  (false positives) is the number of reported notes not in the ground-truth, and  $FN$  (false negatives) is the number of ground-truth notes not reported.

Although note offset is not used in determining correct notes, we do measure the Average Overlap Ratio (AOR) between correctly transcribed notes and their corresponding ground-truth notes. It is defined as

$$AOR = \frac{\min(offsets) - \max(onsets)}{\max(offsets) - \min(onsets)} \quad (3)$$

AOR ranges between 0 and 1, where 1 means that the transcribed note overlaps exactly with the ground-truth note.

To see the improvement of different stages of the proposed system, we also evaluate the system using frame-level measures. Again, we use precision, recall, and F-measures defined in Eq. (2), but here the counts are on the pitches instead of notes. A pitch is considered correctly transcribed if its frequency is within a quarter tone from a ground-truth pitch in the same frame.

### 3.3 Comparison Methods

#### 3.3.1 Benetos et al.'s System

We compare our system with a state-of-the-art note-level transcription system proposed by Benetos et al. [3]. This system first uses shift-invariant Probabilistic Latent Component Analysis (PLCA) to decompose the magnitude spectrogram of the music audio with a pre-learned dictionary containing spectral templates of all semitone notes of 13 kinds of instruments (including the four kinds used in the Bach10 dataset). The activation weights of the dictionary elements provide the soft version of the frame-level transcription. It is then binarized to obtain the hard version of the frame-level transcription. Note-level transcription is obtained by connecting consecutive pitches, filling short gaps between pitches, and pruning short notes.



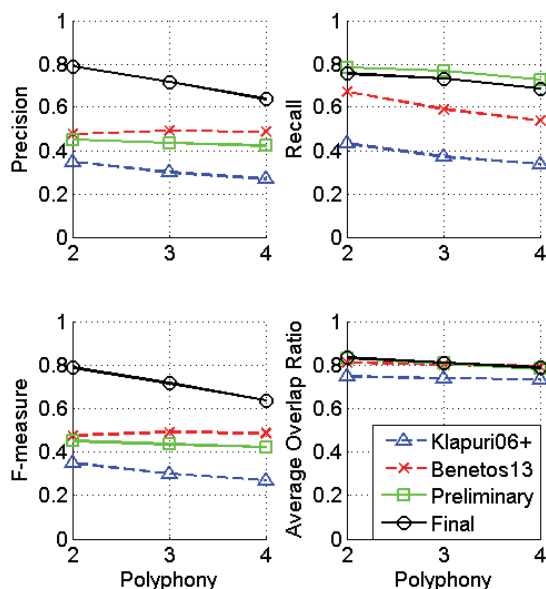


Figure 3. Note-level transcription performances.

The author’s own implementation is available online to generate the soft version frame-level transcription. We then implemented the postprocessing steps according to [3]. Since the binarization threshold is very important in obtaining good transcriptions, we performed a grid search between 1 and 20 with a step size of 1 on the trio pieces. We found 12 gave the best note-level F-measure and used it in all experiments. The time threshold for filling and pruning were set to 100 ms, same as the other comparison methods. We denote this comparison system by “Benetos13”.

### 3.3.2 Klapuri’s System

Klapuri’s system [11] is a state-of-the-art general-purposed frame-level transcription system. It employs an iterative spectral subtraction approach. At each iteration, a pitch is estimated according to a salience function and its harmonics are subtracted from the mixture spectrum. We use Klapuri’s original implementation and suggested parameters. Since Klapuri’s system does not output note-level transcriptions, we employ the preliminary note tracking stage in our system to convert Klapuri’s frame-level transcriptions into note-level transcriptions. We denote this comparison system by “Klapuri06+”.

## 3.4 Results

Figure 3 compares the note-level transcription performance of the preliminary and final results of the proposed system with Benetos13 and Klapuri06+. It can be seen that the precision of the final transcription of the proposed system is improved significantly from the preliminary transcription for all polyphony. This is accredited to the note sampling stage of the proposed system. As shown in Figure 2, note sampling removes many spurious notes and leads to higher precision. On the other hand, the recall of the final transcription is just slightly decreased (about 3%),

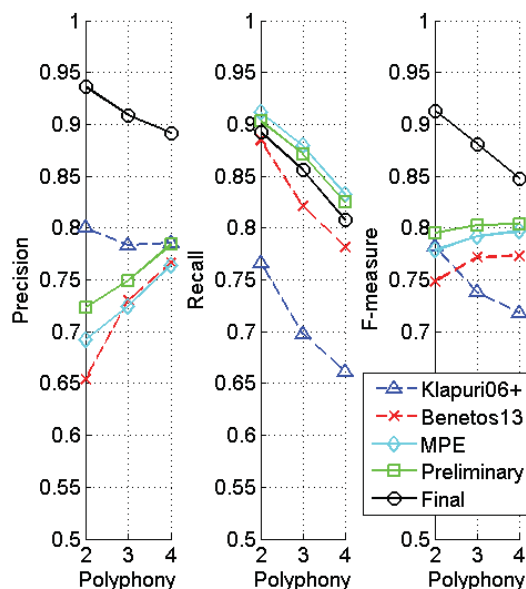


Figure 4. Frame-level transcription performances.

which means most correct notes survive during the sampling. Therefore, the F-measure of the final transcription is significantly improved from the preliminary transcription for all polyphony, leading to a very promising performance on this dataset. The average F-measure on the 60 duets is about 79%, which is about 35% higher than the preliminary result in absolute value. The average F-measure on the 10 quartets is about 64%, which is also about 22% higher than the preliminary transcription.

Compared to the two state-of-the-art methods, the final transcription of the proposed system also achieves much higher F-measure. In fact, the preliminary transcription is a little inferior to Benetos13. However, the note sampling stage makes the final transcription surpass Benetos13.

In terms of average overlap ratio (AOR) of the correctly transcribed notes with the ground-truth notes, both preliminary and the final transcription of the proposed system and Benetos13 achieve a similar performance, which is about 80% for all polyphony. This is about 5% higher than Klapuri06+. It is noted that 80% AOR indicates a very good estimation of the note lengths/offsets.

Figure 4 presents the frame-level transcription performance. In this comparison, we also include the MPE result which is the output of Stage 1. There are several interesting observations. First of all, similar to the results in Figure 3, the final transcription of the proposed system improves from the preliminary transcription significantly in both precision and F-measure, and degrades slightly in recall. This is accredited to the note sampling stage. Second, preliminary transcription of the proposed system has actually improved from the MPE result in F-measure. This validates the filling and pruning operations in the second stage, although the increase is only about 3%. Third, the final transcription of the proposed system achieves significantly higher precision and F-measure than the two com-

parison methods, leading to about 91%, 88%, and 85% F-measure for polyphony 2, 3, and 4, respectively. This performance is very promising and may be accurate enough for many other applications.

#### 4. CONCLUSIONS AND DISCUSSIONS

In this paper, we built a note-level music transcription system based on an existing frame-level transcription approach. The system first performs multi-pitch estimation in each time frame. It then employs a preliminary note tracking to connect pitch estimates into notes. The key step of the system is to perform note sampling to generate a number of subsets of the notes, where each subset is viewed as a transcription candidate. The sampling was based on the note length and note likelihood, which was calculated using the single-pitch likelihood of pitches in the note. Then the transcription candidates are evaluated using the multi-pitch likelihood of simultaneous pitches in all the frames. Finally the candidate with the highest likelihood is returned as the system output. The system is simple and effective. Transcription performance was significantly improved due to the note sampling and likelihood evaluation step. The system also significantly outperforms two other state-of-the-art systems on both note-level and frame-level measures on music pieces with polyphony from 2 to 4.

The technique proposed in this paper is very simple, but the performance improvement is unexpectedly significant. We think the main reason is twofold. First, the note sampling step lets us explore the transcription space, especially the good regions of the transcription space. The single-pitch likelihood of each estimated pitch plays an important role in sampling the notes. In fact, we think that probably any kind of single-pitch salience function that have been proposed in the literature can be used to perform note sampling. The second reason is that we use the multi-pitch likelihood, which considers interactions between simultaneous pitches, to evaluate these sampled transcriptions. This is important because notes contained in a sampled transcription must have high salience, however, when considered as a whole, they may not fit with the audio as well as another sampled transcription. One limitation of the proposed sampling technique is that it can only remove false alarm notes in the preliminary transcription but not adding back missing notes. Therefore, it is important to make the preliminary transcription have a high recall rate before sampling.

#### 5. ACKNOWLEDGEMENT

We thank Emmanouil Benetos and Anssi Klapuri for providing the source code or executable program of their transcription systems for comparison.

#### 6. REFERENCES

- [1] M. Bay, A.F. Ehmann, and J.S. Downie, "Evaluation of multiple-F0 estimation and tracking systems," in *Proc. ISMIR*, 2009, pp. 315-320.
- [2] J.P. Bello, L. Daudet, M.B., Sandler, "Automatic piano transcription using frequency and time-domain information," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 2242-2251, 2006.
- [3] E. Benetos, S. Cherla, and T. Weyde, "An efficient shift-invariant model for polyphonic music transcription," in *Proc. 6th Int. Workshop on Machine Learning and Music*, 2013.
- [4] E. Benetos and S. Dixon, "A shift-invariant latent variable model for automatic music transcription," *Computer Music J.*, vol. 36, no. 4, pp. 81-94, 2012.
- [5] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, "Automatic music transcription: challenges and future directions," *J. Intelligent Information Systems*, vol. 41, no. 3, pp. 407-434, 2013.
- [6] A. Dessein, A. Cont, G. Lemaitre, "Real-time polyphonic music transcription with nonnegative matrix factorization and beta-divergence," in *Proc. ISMIR*, 2010, pp. 489-494.
- [7] Z. Duan, J. Han, and B. Pardo, "Multi-pitch streaming of harmonic sound mixtures," *IEEE Trans. Audio Speech Language Processing*, vol. 22, no. 1, pp. 1-13, 2014.
- [8] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Trans. Audio Speech Language Processing*, vol. 18, no. 8, pp. 2121-2133, 2010.
- [9] G. Grindlay and D. Ellis, "Transcribing multi-instrument polyphonic music with hierarchical eigeninstruments," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1159-1169, 2011.
- [10] P. Grosche, B. Schuller, M. Mller, and G. Rigoll, "Automatic transcription of recorded music," *Acta Acustica United with Acustica*, vol. 98, no. 2, pp. 199-215, 2012.
- [11] A. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes," in *Proc. ISMIR*, 2006, pp. 216-221.
- [12] G. Poliner, and D. Ellis, "A discriminative model for polyphonic piano transcription," in *EURASIP J. Advances in Signal Processing*, vol. 8, pp. 154-162, 2007.
- [13] S.A. Raczynski, N. Ono, and S. Sagayama. "Note detection with dynamic bayesian networks as a post-analysis step for NMF-based multiple pitch estimation techniques," in *Proc. WASPAA*, 2009, pp. 49-52.
- [14] M. Ryyänen and A. Klapuri, "Polyphonic music transcription using note event modeling," in *Proc. WASPAA*, 2005, pp. 319-322.
- [15] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Trans. Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528-537, 2010.

# DRUM TRANSCRIPTION VIA CLASSIFICATION OF BAR-LEVEL RHYTHMIC PATTERNS

Lucas Thompson, Matthias Mauch and Simon Dixon

Centre for Digital Music, Queen Mary University of London

contact@lucas.im, {m.mauch, s.e.dixon}@qmul.ac.uk

## ABSTRACT

We propose a novel method for automatic drum transcription from audio that achieves the recognition of individual drums by classifying bar-level drum patterns. Automatic drum transcription has to date been tackled by recognising individual drums or drum combinations. In high-level tasks such as audio similarity, statistics of longer rhythmic patterns have been used, reflecting that musical rhythm emerges over time. We combine these two approaches by classifying bar-level drum patterns on sub-beat quantised timbre features using support vector machines. We train the classifier using synthesised audio and carry out a series of experiments to evaluate our approach. Using six different drum kits, we show that the classifier generalises to previously unseen drum kits when trained on the other five (80% accuracy). Measures of precision and recall show that even for incorrectly classified patterns many individual drum events are correctly transcribed. Tests on 14 acoustic performances from the ENST-Drums dataset indicate that the system generalises to real-world recordings. Limited by the set of learned patterns, performance is slightly below that of a comparable method. However, we show that for rock music, the proposed method performs as well as the other method and is substantially more robust to added polyphonic accompaniment.

## 1. INTRODUCTION

The transcription of drums from audio has direct applications in music production, metadata preparation for musical video games, transcription to musical score notation and for musicological studies. In music retrieval, robust knowledge of the drum score would allow more reliable style recognition and more subtle music search by example. Yet like related tasks such as polyphonic piano transcription [1], a versatile, highly reliable drum transcription algorithm remains elusive.

Audio drum transcription methods have been classified into two different strategies [10, 18]: *segment and classify*

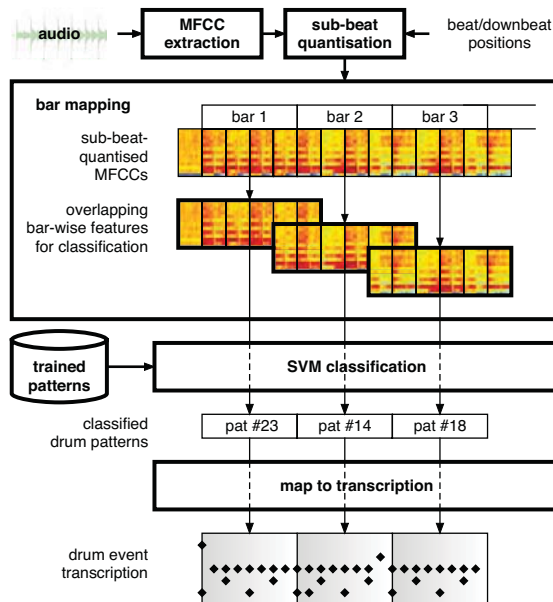


Figure 1. Overview of the system at prediction time.

and *separate and detect*. Systems in the first category detect a regular or irregular event grid in the signal, segment the signal according to the grid, extract features such as MFCCs [19] or multiple low-level features [23] and then classify the segments using Gaussian Mixture Models [19],  $k$  nearest neighbour classification [21], or Support Vector Machines [23]. Systems in the second category first detect multiple streams corresponding to drum types, usually via a signal or spectral decomposition approach, e.g. [2, 7], or simpler sub-band filtering [15], and then identify onsets in the individual streams. Other methods combine aspects of both categories, via adaptation [24] or joint detection of onsets and drums [18]. To ensure temporal consistency (smoothness) many approaches make use of high-level statistical models that encode some musical knowledge, e.g. hidden Markov models [18]. The methods greatly differ in terms of the breadth of instruments they are capable of detecting; most detect only bass drum, snare drum and hi-hat [7, 14, 18, 23] or similar variants, probably because these instruments (unlike crash and ride cymbals) can be represented in few frames due to their very fast decay.

Despite the evident diversity of strategies, all existing methods aim directly at detecting individual or simultaneous drum events. As we will see later, our approach is qualitatively different, using higher-level patterns as its classi-



fication target. One advantage of this is that the long decay of cymbals is naturally modelled at the feature level.

Since drum transcription from polyphonic audio is only partially solved, music information retrieval tasks rely on “soft” mid-level audio features to represent rhythm. Fluctuation patterns [17] summarise the frequency content of sub-band amplitude envelopes across 3-second windows and were used to evaluate song similarity and to classify pop music genres; they have also been used to describe rhythmic complexity [13]. Bar-wise rhythmic amplitude envelope patterns have been shown to characterise ballroom dance music genres [5] and bar-wise pseudo-drum patterns have been shown to correlate with popular music genres [6]. Rhythmic patterns have also formed the basis of beat tracking systems [11] and been used for downbeat detection [8]. These methods share a more musically holistic approach to rhythm, i.e. they summarise rhythmic components in a longer temporal context. Drum tutorials, too, usually focus on complete rhythms, often a bar in length, because “with the command of just a few basic rhythms you can make your way in a rock band” [22]. In fact, we have recently shown that drum patterns are distributed such that a small number of drum patterns can describe a large proportion of actual drum events [12].

Motivated by this result and by the effectiveness of more holistic approaches to rhythm description, we propose a novel drum transcription method based on drum pattern classification. Our main contribution is to show that the classification of bar-length drum patterns is a good proxy for predicting individual drum events in synthetic and real-world drum recordings.

## 2. METHOD

The proposed method is illustrated in Figure 1. It can broadly be divided into two parts: a feature extraction step, in which MFCC frame-wise features are calculated and formatted into a bar-wise, sub-beat-quantised representation, and a classification step, in which bar-wise drum patterns are predicted from the feature representation and then translated into the desired drum transcription representation. For the sake of this study, we assume that correct beat and bar annotations are given.

### 2.1 Feature extraction

Following Paulus and Klapuri [19], we choose Mel-frequency cepstral coefficients (MFCCs) as basis features for our experiments. MFCCs are extracted from audio sampled at 44.1 kHz with a frame size of 1024 samples (23ms) and a hop size of 256 samples (6ms), using an adaptation of the implementation provided in the Vampy plugin examples.<sup>1</sup> We extract 14 MFCCs (the mentioned implementation uses a bank of 40 Mel-filters) per frame, but discard the 0<sup>th</sup> coefficient to eliminate the influence of overall signal level.

In order to obtain a tempo-independent representation, we assume that we know the positions of musi-

<sup>1</sup><http://www.vamp-plugins.org/vampy.html>

cal beats and quantise the feature frames into a metrical grid. This is needed for subsequent bar-wise segmentation. Whereas beat-quantised chroma representations usually summarise chroma frames within a whole inter-beat interval [16], drum information requires finer temporal resolution. Hence, following [12] we choose 12 sub-beats per beat, which is sufficient to represent the timing of the most common drum patterns. The MFCC frames belonging to each sub-beat are summarised into a single value by taking the mean over the sub-beat duration to give 12 quantised frames per beat.

Since we assume we know which beat is the downbeat, it is now trivial to extract bar representations from sub-beat-quantised MFCC features. For example, in a  $\frac{4}{4}$  time signature, one bar corresponds to  $4 \times 12 = 48$  sub-beat-quantised MFCC frames. However, slight deviations in timing and natural decay times of cymbals and drum membranes mean that information on a bar pattern will exist even outside the bar boundaries. For this reason we also add an extra beat either side of the bar lines (further discussion in Section 3), leading to the overlapping bar representations illustrated in Figure 1, each  $6 \times 12 = 72$  frames long. The features we are going to use to classify  $\frac{4}{4}$  bars into drum patterns will therefore comprise 936 elements ( $72 \text{ frames} \times 13 \text{ MFCCs}$ ).

### 2.2 Classification and transcription mapping

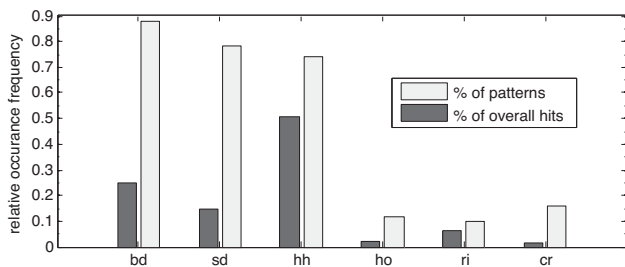
As our classifier, we use the one-vs-one multi class Support Vector Machine implementation provided in the *sklearn.svm.SVC*<sup>2</sup> package of the Python machine learning library, *scikit-learn* [20], with the default settings using a radial basis kernel,  $K(x, x') = e^{-\gamma \|x - x'\|^2}$ , where  $\gamma = \frac{1}{N}$  and  $N = 936$  is the feature dimension. Once the classifier has predicted a drum pattern for a particular bar, we perform a simple mapping step to obtain a drum transcription: using the information about the actual start and end time of the bar in the recording, each of the drum events that constitute the pattern are assigned to a time stamp within this time interval, according to their position in the pattern.

## 3. EXPERIMENTS AND RESULTS

We conducted three experiments to test the effectiveness of the proposed method, one with synthesised test data, and two with real recordings of human performances. In all experiments, the drum pattern data for training was encoded as MIDI and then synthesised using the FluidSynth software. Our drum pattern dictionary contains the top 50 most common drum patterns, including the empty pattern, in a collection of 70,000 MIDI files (containing only **bd** - kick, **sd** - snare, **hh** - closed hi-hat, **oh** - open hi-hat, **ri** - ride and **cr** - crash cymbals) [12].<sup>3</sup> Figure 2 details how each drum class is distributed. Data examples and further

<sup>2</sup><http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>

<sup>3</sup>[http://www.eecs.qmul.ac.uk/~matthiasm/ndrum/patternstats/full\\_1-2-3-4-5-6/patternvisual\\_reduced](http://www.eecs.qmul.ac.uk/~matthiasm/ndrum/patternstats/full_1-2-3-4-5-6/patternvisual_reduced)



**Figure 2.** Relative occurrence of the drum classes in terms of overall number of drum events and the number of patterns containing each class. There are 50 patterns with an average of 11 events per pattern.

information can be found on the web page that accompanies this paper.<sup>4</sup>

We evaluate both the pattern classification performance and the quality of transcription of drum events. Pattern accuracy is defined as

$$A = \frac{\text{number of correctly classified bars}}{\text{total number of bars in test set}}. \quad (1)$$

The transcription of drum events is evaluated using precision, recall and the F-measure (their harmonic mean)

$$P = \frac{N_c}{N_d}, \quad R = \frac{N_c}{N}, \quad F = \frac{2PR}{P+R}, \quad (2)$$

where  $N_d$  is the number of detected drum hits,  $N_c$  is the number of correctly detected drum hits and  $N$  the number of drum hits in the ground truth. The individual drum hits are solely based on the presence or absence of a drum hit at a particular discrete position in the pattern grid used in the dictionary. In Sections 3.2 and 3.3 the ground truth drum hits, given as onset times, are quantised to a position in the grid. Tanghe’s method [23] (Sections 3.2 and 3.3) is evaluated against the original ground truth with an acceptance window of 30ms, as in the original paper.

### 3.1 Multiple Synthesised Drum Kits

The aim of this experiment is to see how well the proposed classifier performs on synthetic data generated using multiple drum kits.

#### 3.1.1 Training and Test Data

In order to create varied training and test data, we first generate 100 unique *songs*, each of which is simply a randomly permuted list of the 50 drum patterns from our dictionary. These *songs* are encoded as MIDI files, and we introduce randomised deviations in note velocity and onset times (velocity range 67–127, timing range  $\pm 20$  ms) to humanise the performances. All MIDI files are then rendered to audio files (WAV) using 6 drum kits from a set of SoundFonts we collected from the internet. In order to avoid complete silence, which is unrealistic in real-world scenarios, we add white noise over the entirety of each

<sup>4</sup> <http://www.eecs.qmul.ac.uk/~matthiasm/drummify/>

drum-kit	overall drum events			classification accuracy
	R	P	F	
00	98.9 (98.5)	98.9 (98.7)	98.9 (98.6)	91.1 (88.8)
01	97.1 (97.1)	97.6 (97.7)	97.4 (97.4)	89.2 (87.9)
02	98.3 (97.9)	98.3 (98.0)	98.3 (98.0)	87.7 (86.5)
03	84.8 (80.3)	82.3 (85.8)	83.6 (83.0)	50.1 (47.5)
04	92.7 (92.2)	91.2 (90.8)	92.0 (91.5)	72.0 (66.4)
05	97.2 (97.1)	98.5 (98.5)	97.9 (97.8)	91.6 (88.6)
mean	94.8 (93.9)	94.5 (94.9)	94.7 (94.4)	80.3 (77.6)

**Table 1.** Mean classification accuracy (%) and overall drum event R, P and F metrics for left out drum-kit from leave-one-out cross validation on 6 different drum-kits (see Section 3.1). Results for non-overlapping bars are in brackets.

drum-type	overall drum events		
	R	P	F
bd	96.2 (96.0)	95.4 (95.0)	95.8 (95.5)
sd	96.5 (95.4)	99.3 (99.3)	97.8 (97.0)
hh	96.5 (95.3)	93.7 (94.8)	95.0 (95.0)
ho	59.9 (57.1)	77.3 (77.3)	61.1 (56.8)
ri	86.3 (86.4)	98.3 (99.5)	88.2 (89.0)
cr	84.4 (75.8)	97.0 (96.5)	89.3 (82.5)

**Table 2.** R, P and F for each drum type, taken over the whole test set and all kits from leave-one-out cross validation on 6 different drum-kits (see Section 3.1). Results for non-overlapping bars are in brackets.

song at a SNR of 55 dB. We then calculate the bar-wise beat-quantised MFCC features as described in section 2.1. This yields a dataset of  $6 \times 100 = 600$  files.

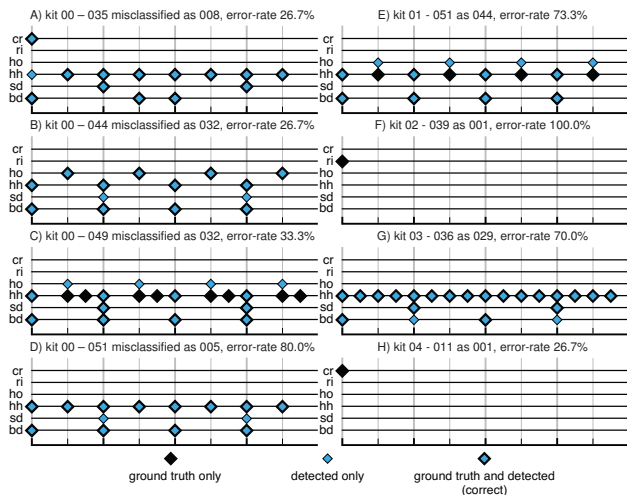
We use a random 70:30 train/test split of the 100 songs, where each of the 70 training songs appears in five variations synthesised from different drum kit SoundFonts. The remaining 30 songs, synthesised by the sixth drum kit, are used for testing. In order to assess performance on different drum kits, we cycle the use of the test drum kit in a leave-one-out fashion.

#### 3.1.2 Results

As Table 1 shows, our method achieves a high average accuracy of 80.3%, despite strong variation between drum kits. Irrespective of whether overlapping bar-wise features were used, the accuracy on drum kits 00, 01, 02 and 05 exceeds 85%. Performance is substantially worse on drum kits 03 and 04 (accuracies of 50.1% and 72.0%, respectively). Listening to a subset of the synthesised songs for drum kits 03 revealed that the recording used for the closed hi-hat sounds contains hi-hats that are slightly open, which is likely to cause confusion between the two hi-hat sounds.

To demonstrate the benefit of considering extra beats either side of the bar boundaries, Table 1 includes the results for non-overlapping bars. In this case we can see that the context given by the neighbouring beats increases classification accuracy (mean increase  $\approx 3$  percentage points). The greatest increase in accuracy ( $\approx 6$  percentage points) is observed in drum-kit 04.

To gain an insight into the types of patterns being misclassified, we consider those patterns for each drum-kit that are misclassified more than a quarter of the time. Figure 3 contains a few example cases. The single undetected



**Figure 3.** Examples of misclassified patterns (see Section 3.1.2).

ride or crash cymbals on the first beat in the ground-truth (cases F and H) are likely to be caused by the system confusing them for remainders of the previous bar. For cases A, B, D and G, the differences are subtle. In case A, the patterns differ by one hi-hat on the first beat. Cases B, D and G show that on occasions the classifier chooses a pattern where the majority of the drum events are correct, apart from a few inserted bass or snare drum events.

If we compare the individual drum events of the predicted pattern against the ground-truth and use precision and recall measures (see Table 2) we see that the system achieves high F-measures for the majority of drum classes (mean 0.88–0.97 for bd, sd, hh, ri, cr over all kits), but not for the open hi-hat class (mean F-measure 0.61).

Using audio features with overlapping bars leads to a substantial increase of over 8 percentage points in the recall of crash cymbal hits (84.4%) with respect to using no overlap (75.8%). The majority of the crash hits in our pattern dictionary occur on the first beat of the bar, and many of the patterns which were misclassified without the benefit of the overlapping neighbouring beats are such patterns, highlighting that the added context helps distinguish the pattern from those with a decaying hi-hat or other cymbal at the end of the previous bar. Note that since crash cymbals usually occur no more than once per bar, the classification accuracy in Table 1 shows larger improvement than the overall drum event precision and recall values.

## 3.2 Real drums Without Accompaniment

Having evaluated the performance of our system on synthesised data, we now test its robustness to real acoustic drum data.

### 3.2.1 Training and test data

We use the set of 100 songs described in the previous experiment (Section 3.1.1) synthesised on all 6 drum kits ( $6 \times 100 = 600$  files). Since we have shown that overlapping bar-wise features provide higher accuracy (Section 3.1.2), we use only this feature configuration to train

a re-usable model, which is used in the remainder of the experiments.

As test data we use the ENST-Drums database [9], which contains a wide range of drum recordings and ground-truth annotations of drum event onset times. We selected 13 *phrase* performances (15–25 s) which contain a number of similar patterns to ones in our dictionary, with expressional variations and fills, and one song from the *minus-one* category, a 60’s rock song, which contains extensive variations and use of drum fills for which there are no similar patterns in our dictionary. In order to convert the provided ground-truth annotations to bar length drum pattern representations of the same format as those in our pattern dictionary, we annotated the beat and downbeat times in a semi-automatic process using Sonic Visualiser [3] and a Vamp-plugin implementation<sup>5</sup> of Matthew Davies’ beat-tracker [4].

### 3.2.2 Results

The results for the ENST-Drums tracks are given in Table 3. The system’s performance strongly varies by track. Our system performs particularly well on the disco and rock genre recordings (F-measure 0.479–0.924), for which our pattern dictionary contains very similar patterns. The shuffle-blues and hard-rock patterns perform much worse (F-measure 0.037–0.525), which is largely due to the fact that they utilise patterns outside our dictionary, bringing the mean F-measure down to 0.563. In order to understand the impact of out-of-dictionary patterns, Table 3 also provides the maximum possible F-measure  $F_{max}$  calculated from our dictionary by choosing the transcription that results in the highest F-measure for each bar, and computing the overall F-measure of this transcription.

For example, ENST recording 069 only achieves an  $F$  score of 0.288, falling short of  $F_{max} = 0.583$ , as it mostly consists of a typical shuffle drum pattern utilising the ride cymbal which is outside of the dictionary. However, the pattern which the system predicts is in fact one that contains a ride cymbal, from a total of five (see Figure 2). The hard rock recordings make extensive use of the open hi-hat, which is not utilised in the same fashion in our dictionary; here, the classifier most often predicts an empty bar (hence the very low scores). Note that all scores are obtained on a very diverse set of 6 drum and cymbal types.

For comparison, we obtained an implementation of an existing drum transcription method by Tanghe [23] and ran it on the ENST recordings, using the default pre-trained model. Since Tanghe’s method only considers bass drum, snare drum and hi-hat, we constrain the evaluation to those drum types, and map the open and closed hi-hat events from our algorithm to single hi-hat events. Table 4 shows that our system has an F-measure of 0.73; Tanghe’s system performs better overall (0.82), which is largely due to excellent bass drum detection. Note however that our system obtains better performance for the snare drum (F-measure 0.74 vs 0.70) particularly with respect to precision (0.93 vs

<sup>5</sup> <http://vamp-plugins.org/plugin-doc/qm-vamp-plugins.html#qm-barbeatracker>

	genre	tempo	detected drum events			$F_{max}$
			R	P	F	
038	disco	slow	72.6	84.9	78.3	86.7
039	disco	medium	90.2	94.8	92.4	95.8
040	disco	fast	93.1	87.1	90.0	100.0
044	rock	slow	48.1	47.6	47.9	59.8
045	rock	medium	52.7	47.5	50.0	58.5
046	rock	fast	54.5	52.5	53.5	63.6
055	disco	slow	75.9	63.8	69.3	98.3
061	rock	slow	93.8	84.3	88.8	92.5
069	SB	slow	25.6	32.8	28.8	58.3
070	SB	medium	50.0	55.4	52.5	59.5
075	HR	slow	1.9	50.0	3.7	58.7
076	HR	medium	3.8	100.0	7.4	53.2
085	SB	slow	49.5	49.0	49.2	79.5
116	minus-one (60s rock)		76.8	77.1	77.0	81.7
	mean		56.3	66.2	56.3	74.7

**Table 3.** Real drums without accompaniment: results in percent for ENST-Drums dataset. SB: shuffle-blues; HR: hard rock.

method	metric	bd	sd	hh	overall
Proposed	R	70.2	62.0	73.1	69.9
	P	60.6	92.7	83.5	76.3
	F	65.1	<b>74.3</b>	77.9	73.0
Tanghe et al.	R	87.0	65.0	89.8	83.8
	P	99.3	75.8	73.9	80.6
	F	<b>92.8</b>	70.0	81.1	<b>82.1</b>

**Table 4.** Real drums without accompaniment: Results in percent for drum classes reduced to bd, sd, hh (including ho) for comparison with Tanghe et al. [23].

0.76). With a larger dictionary, our method would be able to capture more details, such as drum fills, so we expect a similar system with larger dictionary to perform better.

### 3.3 Polyphonic Music

For the *minus-one* recording, the ENST-Drums database provides additional non-percussive accompaniment, which allows us to test our system on polyphonic music.

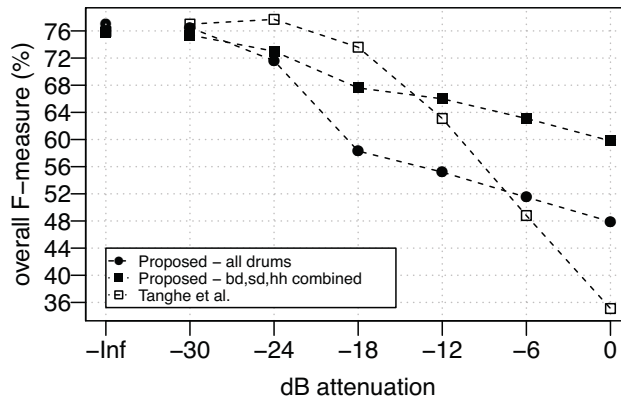
#### 3.3.1 Training and Test Data

As in the previous experiment, we use the pre-trained model from all the synthesised drum data from the experiment described in Section 3.1. The test data consists of the *minus-one* recording considered in the previous experiment. We add the polyphonic accompaniment at different levels: 0dB (fully polyphonic, no attenuation), -6dB, -12dB, -18dB, -24dB and -30dB.

#### 3.3.2 Results

The overall F-measures obtained by the system for the various levels of attenuation are detailed in Figure 4. We provide the performance of the system on the recording with no accompaniment as a baseline (overall F-measure 0.77, as in Table 3). The system’s performance on all drums decays rapidly between -24 dB and -18 dB, but then stays relatively robust for the most difficult levels considered (0dB to -18dB, overall F-measure scores of 0.48–0.58).

We compare the performance of our system to Tanghe’s method once more on the reduced drum type set (bd, sd, hh). It is interesting to observe that while the F-measure on



**Figure 4.** Overall drum events F-measure for ENST recording 116, mixed in with accompaniment at various levels of attenuation.

the pure drums is nearly the same (Tanghe: 0.76, proposed: 0.77), susceptibility to additional instruments strongly differs between the methods. The F-measure of Tanghe’s method first increases for low levels of added polyphonic music (attenuation -30, -24 dB), due to the increased recall as a result of the accompaniment being detected as correct drum hits. For increasing levels of added accompaniment, performance rapidly decreases to an overall F-measure of 0.35 for 0 dB. By direct comparison, the proposed method achieves an F-measure of 0.60 even at 0 dB, demonstrating its superior robustness against high levels of accompaniment (-12, -6, 0 dB). Even for the more difficult task of recognising all 6 drum types, the proposed method (F-measure 0.48) outperforms Tanghe’s.

## 4. DISCUSSION

Our results show not only that the proposed bar-wise drum pattern classification method is an effective, robust way to transcribe drums, but also that the first step for immediate improvement should be to increase the dictionary size in order to obtain better coverage. In addition, relaxing the strict holistic pattern approach by classifying patterns of individual instruments would allow for the recognition of combinations of patterns and hence of many new, unseen patterns. Another obvious route for improvement is to train our classifier on drum data with added polyphonic music content, which is likely to further increase robustness in polyphonic conditions.

The general approach of bar-wise drum classification is not exhausted by our particular implementation, and we expect to be able to gain further improvements by exploring different classifiers, different amounts of neighbourhood context or different basic features (e.g. non-negative matrix factorisation activations). Furthermore, to use the method in an interactive annotation system, it would be interesting to investigate bar-wise confidence scores for user guidance. Genre-specific training data could improve the performance of such systems. Finally, using more holistic features instead of single frames may also be applicable to other music informatics tasks such as chord transcription.

## 5. CONCLUSIONS

We have presented a novel approach to drum transcription from audio using drum pattern classification. Instead of detecting individual drums, our method first predicts whole drum patterns using an SVM classifier trained on a large collection of diverse synthetic data, and then maps the drums from the recognised patterns to the relative time-stamps to achieve a transcription. The method performs very well on synthetic data, even with tempo and velocity variations on previously unseen sampled drum kits (mean pattern accuracy: 80%). Even though the pattern accuracy range differs between drum kits (50.1%–91.6%) many drum events are still classified with high precision and recall (F-measure 0.836–0.989). Unlike existing techniques, our drum detection includes open hi-hat, closed hi-hat, crash and ride cymbals, which are all reliably detected in most cases. Extending the bar patterns by one beat either side and thus obtaining overlapping patterns leads to better accuracy, mainly due to improved recognition of crash cymbals. On real drum recordings performance strongly depends on genre (F-measure for rock and disco: 0.479–0.924; hard-rock and shuffle-blues: 0.037–0.525), mainly due to the limited types of drum patterns in our current dictionary. This results in a performance slightly below that of a comparable method. However, we show that for rock music, the proposed method performs as well as the other method (F-measure: 0.77) and is substantially more robust to added polyphonic accompaniment.

## 6. ACKNOWLEDGEMENTS

Matthias Mauch is supported by a Royal Academy of Engineering Research Fellowship.

## 7. REFERENCES

- [1] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: Challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013.
- [2] E. Benetos, S. Ewert, and T. Weyde. Automatic transcription of pitched and unpitched sounds from polyphonic music. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, May 2014.
- [3] C. Cannam, C. Landone, M. B. Sandler, and J. P. Bello. The Sonic Visualiser: A visualisation platform for semantic descriptors from musical signals. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, pages 324–327, 2006.
- [4] M. E. P. Davies and M. D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1009–1020, 2007.
- [5] S. Dixon, F. Gouyon, and G. Widmer. Towards characterisation of music via rhythmic patterns. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 509–516, 2004.
- [6] D. P. W. Ellis and J. Arroyo. Eigenrhythms: Drum pattern basis sets for classification and generation. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 101–106, 2004.
- [7] D. FitzGerald, R. Lawlor, and E. Coyle. Prior subspace analysis for drum transcription. In *Proceedings of the AES 114th International Convention*, 2003.
- [8] D. Gärtner. Unsupervised Learning of the Downbeat in Drum Patterns. In *Proceedings of the AES 53rd International Conference*, pages 1–10, 2014.
- [9] O. Gillet and G. Richard. ENST-Drums: An extensive audio-visual database for drum signals processing. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, pages 156–159, 2006.
- [10] O. Gillet and G. Richard. Transcription and separation of drum signals from polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):529–540, 2008.
- [11] M. Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171, 2001.
- [12] M. Mauch and S. Dixon. A corpus-based study of rhythm patterns. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR 2012)*, pages 163–168, 2012.
- [13] M. Mauch and M. Levy. Structural change on multiple time scales as a correlate of musical complexity. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, pages 489–494, 2011.
- [14] M. Miron, M. E. P. Davies, and F. Gouyon. Improving the real-time performance of a causal audio drum transcription system. In *Proceedings of the Sound and Music Computing Conference (SMC 2013)*, pages 402–407, 2013.
- [15] M. Miron, M. E. P. Davies, and Fabien Gouyon. An open-source drum transcription system for Pure Data and Max MSP. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, pages 221–225. IEEE, 2013.
- [16] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. De Bie. An end-to-end machine learning system for harmonic analysis of music. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1771–1783, 2012.
- [17] E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classification. In *Proceedings of the 6th International Conference on Music Information Retrieval*, pages 634–637, 2005.
- [18] J. Paulus and A. Klapuri. Drum sound detection in polyphonic music with hidden Markov models. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009:14, 2009.
- [19] J. K. Paulus and A. P. Klapuri. Conventional and periodic n-grams in the transcription of drum sequences. In *Proceedings of the International Conference on Multimedia and Expo (ICME 2003)*, volume 2, pages II–737. IEEE, 2003.
- [20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [21] V. Sandvold, F. Gouyon, and P. Herrera. Percussion classification in polyphonic audio recordings using localized sound models. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 537–540, 2004.
- [22] J. Strong. *Drums For Dummies*. John Wiley & Sons, 2011.
- [23] K. Tanghe, S. Degroove, and B. De Baets. An algorithm for detecting and labeling drum events in polyphonic music. In *Proceedings of the 1st Annual Music Information Retrieval Evaluation Exchange (MIREX 2005)*, pages 11–15, 2005.
- [24] K. Yoshii, M. Goto, and H. G. Okuno. Automatic drum sound description for real-world music using template adaptation and matching methods. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, pages 184–191, 2004.





Oral Session 3  
**Symbolic**

This Page Intentionally Left Blank

# DEVELOPING TONAL PERCEPTION THROUGH UNSUPERVISED LEARNING

Carlos Eduardo Cancino Chacón, Stefan Lattner, Maarten Grachten

Austrian Research Institute for Artificial Intelligence

{carlos.cancino, stefan.lattner, maarten.grachten}@ofai.at

## ABSTRACT

The perception of tonal structure in music seems to be rooted both in low-level perceptual mechanisms and in enculturation, the latter accounting for cross-cultural differences in perceived tonal structure. Unsupervised machine learning methods are a powerful tool for studying how musical concepts may emerge from exposure to music. In this paper, we investigate to what degree tonal structure can be learned from musical data by unsupervised training of a Restricted Boltzmann Machine, a generative stochastic neural network. We show that even based on a limited set of musical data, the model learns several aspects of tonal structure. Firstly, the model learns an organization of musical material from different keys that conveys the topology of the circle of fifths (CoF). Although such a topology can be learned using principal component analysis (PCA) when using pitch-only representations, we found that using a pitch-duration representation impedes the extraction of the CoF topology much more for PCA than for the RBM. Furthermore, we replicate probe-tone experiments by Krumhansl and Shepard, measuring the organization of tones within a key in human perception. We find that the responses of the RBM share qualitative characteristics with those of both trained and untrained listeners.

## 1. INTRODUCTION

Modern approaches in music theory recognize that tonality can be broadly described as the organization of pitch classes into a hierarchical structure of tensions-relaxations around a tonal axis [10, 15, 16]. This conception of tonality is not limited to western tonal classical music, but can also be applied to modal music, popular music (e.g. jazz, rock) and non-western folk music [3]. This notion of tonality is not only a music theoretic construct: perceptual processing of musical stimuli in human listeners has been found to exhibit this type of organization as well [10]. Specific types of hierarchical organization of pitch classes are partly explained by acoustic attributes of pitch, especially the consonance between pairs of pitches [10], suggesting that low-

level processing of acoustic stimuli may be relevant for the perception of tonal structure.

However, tonal structure is not only reflected in the physical attributes of pitch, it is also manifest in the statistical properties of music, such as the duration and frequency of occurrence of pitches [17], as illustrated in Figure 1. As Saffran et al. have shown [14], human listeners (including infants) are sensitive to such statistical regularities, and this leads to the view that tonal perception may be shaped by (long time) exposure to music exhibiting statistical regularities regarding frequency of occurrence of pitches, rhythmic emphasis, the position of occurrence within musical phrases, and possibly other aspects [9].

It is this process, the formation of tonal structure through exposure to musical stimuli, that we focus on in this paper. We choose a particularly straightforward approach, using a Restricted Boltzmann Machine (RBM) [6] to learn the probability distribution of melodic sequences, represented as n-grams of notes. In a first explorative experiment, we examine to what degree the feature space learned by the RBM is musically meaningful. Using the resemblance of the feature space to the circle of fifths as a quantitative criterion, we investigate the impact of the n-gram length, and compare pitch-only input representations to input representations that include both pitch and duration. In a second experiment, we use the RBM to simulate listener ratings in a probe tone test, and compare the results to ratings from human listeners of different skill levels.

The structure of the paper is as follows: In Section 2, we discuss prior work on the induction of tonal structure using computational models. Section 3 relates the different aspects of the unsupervised learning task to various perceptual mechanisms that are assumed to be at play in the perception of tonal structure. Section 4 briefly describes the RBM model, the data used for training the model, and representation of the data. The experiments on tonal organization and the organization of pitches are described in Sections 5 and 6, respectively. Conclusions and future directions are presented in Section 7.

## 2. RELATED WORK

The idea of studying the perception of tonal structure by using computational models to simulate the enculturation process is not new. For example, Tillmann et al. [18] use a hierarchical self-organizing map (SOM) [8] to learn representations of tonal structure from pitch-class representations of chord sequences. They find that their model is able



© Carlos Eduardo Cancino Chacón, Stefan Lattner, Maarten Grachten.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Carlos Eduardo Cancino Chacón, Stefan Lattner, Maarten Grachten. “Developing tonal perception through unsupervised learning”, 15th International Society for Music Information Retrieval Conference, 2014.

to develop an organization comparable to that of empirical data gathered from various studies on human perception of tonality. Leman [12] presents an alternative approach to modeling the perception of tonality. He employs a psychoacoustic model in combination with a SOM to learn tonal representations starting from acoustic data. Furthermore, Toiviainen & Krumhansl examined the perception of musical scales by projecting human ratings to the feature space of a SOM, which was trained on scale profiles of Krumhansl [19].

A commonality among the mentioned works is the choice of the self-organizing map as a model for accommodating the learning process. The reason for this preference may be that both the spatial mapping of the data, and the competitive learning algorithm employed by the SOM, are biologically plausible characteristics of the human sensory cortex [7]. The RBM model used in the work presented here, is not explicitly presented as (nor was it designed to be) a biologically plausible model of learning in the brain. Nevertheless RBMs and deep belief nets based on RBMs, in combination with sparseness constraints on the activation of hidden units, are able to learn features from visual data that strongly resemble receptive fields of neurons in the visual cortex [11]. As such, RBMs prove to be a valid computational modeling approach for learning biologically plausible representations from musical data.

A fundamental difference between SOMs and RBMs is that in the former, the hidden units represent points in an explicitly defined low-dimensional feature space. In RBMs, the feature space is defined by the set of all possible combinations of hidden unit activations, such that each hidden unit represents a dimension of the feature space. This allows for representations of data instances as a (non-linear) combination of features. The topology of this high-dimensional feature space can be visualized in a 2-D space using PCA.

### 3. PERCEPTUAL MECHANISMS

As argued by Smith and Schmuckler [17], perceptual processes like *discrimination*, *differentiation* and *organization* play an important role in the perception of musical tonality. In this Section, we will briefly describe these processes, and show how they can be related to formal aspects of the computational modeling methods, such as the choice of input data representation, and the topology of the feature space being learned.

*Perceptual discrimination* refers to the sensitivity of a system to differences along some perceivable stimulus dimension. In computational learning models, this relates to the form of input data representation. In general, the type of relevant input features depends heavily on the respective learning task [1]. Musical data comprises much context-dependent information that can not be trivially inferred from low-level representations. To decide on an appropriate representation is thus not always an easy task. For instance, pitch content can be represented in several ways, such as frequency spectra, MIDI note numbers, or pitch classes. In our current experiments, we use MIDI

note numbers as well as pitch class representations. Duration is encoded separately from pitch. An advantage of this over combined pitch-duration representations (e.g. piano-roll notation) is that the n-gram size is specified in the number of notes, rather than an absolute time interval. This allows for comparing pitch-only to pitch-duration representations. The input data will be referred to as Input Space (IS), and will be described in more detail in Section 4.3.

*Differentiation* is a higher order ability that refers to the segregation of the perceived stimuli into elements on the basis of its discriminable differences [17]. In an unsupervised model we can identify this ability as the capacity of the system to segregate the data in the IS into clusters in the Feature Space (FS). In the context of tonality, an example of differentiation would be the capacity of an unsupervised model to cluster the data in the FS in such a way that each cluster represents a musical key. A measure of quality of this clustering would then be the variance of each cluster, as smaller variances imply a better differentiation of the data with respect to each class.

*Organization* builds on the concept of differentiation, as it establishes relations between the differentiated elements, as well as the nature of the relations themselves. In an unsupervised model, this can be understood as the topology of the FS. In this way, geometric features such as the distance between clusters, as well as the relative position between them can express similarity.

Bharucha [10, cited by Krumhansl] recognizes two types of hierarchies regarding musical tonality. *Event hierarchies* refer to the functional significance of single note events in a specific musical context, while *tonal hierarchies* account for the abstract musical structure in a particular culture or genre, e.g. the functional significance of all elements of a pitch class relative to all other pitch classes.

In our case, we compare the organization of the data in the FS to the circle of fifths, a well known music theoretical construct that explains the relations and the neighborhood of keys [15]. As a measure of quality we use the Procrustes Distance (PD) [4] of the centroids of the clustered data in the feature space with respect to the CoF.

## 4. METHODS

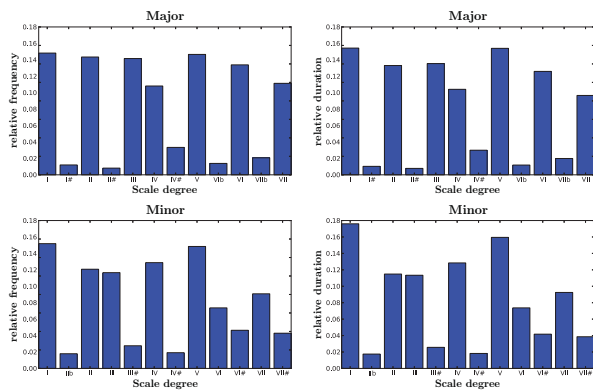
### 4.1 Restricted Boltzmann Machine

A Restricted Boltzmann Machine is a stochastic Neural Network (NN) with two layers, a visible layer with units  $\mathbf{v} \in \{0, 1\}^r$  and a hidden layer with units  $\mathbf{h} \in \{0, 1\}^q$  [6]. The units of both layers are fully interconnected with weights  $\mathbf{W} \in \mathbb{R}^{r \times q}$ , while there are no connections between the units within a layer. Given a visible vector  $\mathbf{v}$ , the free energy of the model can be calculated as:

$$\mathcal{F}(\mathbf{v}) = -\mathbf{a}^\top \mathbf{v} - \sum_i \log \left( 1 + e^{(b_i + \mathbf{W}_i \mathbf{v})} \right), \quad (1)$$

where  $\mathbf{a} \in \mathbb{R}^r$  and  $\mathbf{b} \in \mathbb{R}^q$  are bias vectors, and  $\mathbf{W}_i$  is the  $i$ -th row of the weight matrix.

Given  $\mathbf{v}$ , a sample of  $\mathbf{h}$  can be obtained from its conditional activation probability, given by:



**Figure 1.** Occurrence and duration distributions of the fugues from Bach’s Well Tempered Clavier.

$$p(\mathbf{h} = \mathbf{1} \mid \mathbf{v}) = \sigma(\mathbf{b} + (\mathbf{v}^T \mathbf{W})^T), \quad (2)$$

where  $\sigma(x) = 1/(1+e^{-x})$  is the logistic sigmoid function.

In experiment 1, we consider the conditional activation probability of vector  $\mathbf{h}$  as the result of the projection of  $\mathbf{v}$  into the FS. In the second experiment, we calculate the energy using Eq. (1).

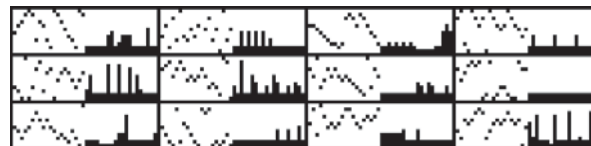
#### 4.1.1 Training

We train the model with 200 hidden units for 1000 epochs with Contrastive Divergence (CD) [6], using 3 Gibbs sampling steps and a mini-batch size of 500 for the weight updates. The learning rate is set to 0.01 and the momentum to 0.3. These parameters were empirically selected according to the rules of thumb suggested by Hinton in [5]. In addition, we use the well-known L2 weight-decay regularization which penalizes large weight coefficients.

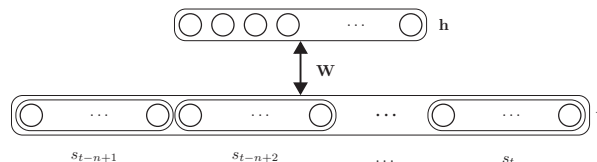
Based on properties of neural coding, sparsity and selectivity can be used as constraints for the optimization of the training algorithm [2]. Sparsity encourages competition between hidden units, and selectivity prevents over-dominance by any individual unit. These constraints are used in our training, with a linear falloff of its influence over the first 200 epochs from 50% to 30%.

## 4.2 Training Corpus

J. S. Bach’s Well Tempered Clavier (WTC), composed between 1722 and 1742, is widely recognized as one of the most influential works in music history [15]. It is also one of the most important works that systematically spans the whole range of major and minor keys, and is therefore well-suited for experiments on tonality. In this paper, we use MIDI versions of the 48 fugues of the WTC as corpus, encoded by David Huron and taken from the KernScores website (<http://kern.ccarh.org>). Each fugue is decomposed into its voices (two to five), and we consider each voice as a single monophonic melody in its respective key. In Figure 1, the distributions of the occurrence and duration of the notes of the WTC are shown. These distributions are similar to the key profiles by Krumhansl & Kessler [19].



**Figure 2.** Twelve random *pitch-duration* training instances of the WTC corpus as 20-grams before linearization. Notes are ordered horizontally, the vertical dimension accounts for pitch and duration values, respectively. The left part of each instance shows the one-out-of- $m$  pitch representation of 20 consecutive notes, the right part shows the corresponding duration representation.



**Figure 3.** The RBM architecture used. An input vector  $\mathbf{v}$  is constituted by a linearized  $n$ -gram, where  $s_j$  is a binary representation of note  $j$ .

## 4.3 Input Representation

From the monophonic melodies, we construct a set of  $n$ -grams by using a sliding window of size  $n$  and a step size of 1. Depending on the experiment, we either use only pitch information, or we use both the pitch and duration of the notes. In the first case, an  $n$ -gram is a concatenation of  $n$  bit vectors of size  $m$ , where the  $i$ -th bit vector is a one-out-of- $m$  representation of the pitch of note  $i$ .

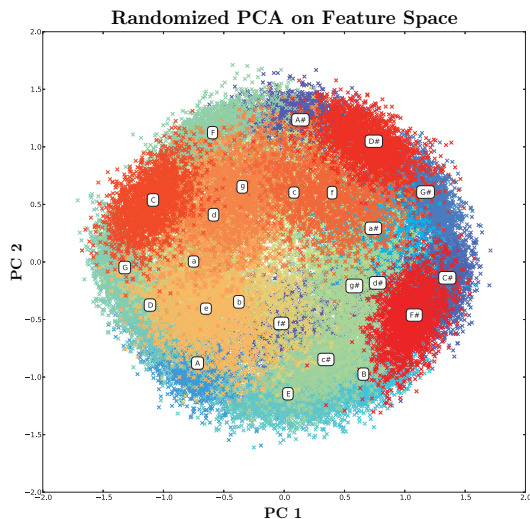
In the second case,  $n$  additional vectors are added to the  $n$ -gram, where the  $i$ -th vector now represents the duration of the  $i$ -th note (see the right half of the instances shown in Figure 2). Such a duration vector is constructed by quantizing all durations of a melody into 12 bins and by relating each of those to one of 12 units. A duration that falls into bin  $k$  is represented by activating units 1 to  $k$ . After linearization, the resulting  $n$ -gram constitutes the visible vector  $\mathbf{v}$ , as illustrated in Figure 3.

## 5. TONAL ORGANIZATION

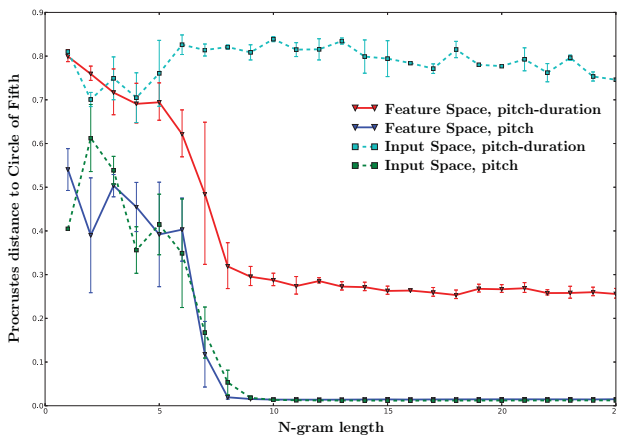
In this experiment, we examine the ability of an RBM to learn tonal relationships between  $n$ -grams. To that end, we project the FS learned by the RBM into a two-dimensional space using Randomized Principal Component Analysis (rPCA) [13]. As the CoF is the underlying music theoretical construct for the relationships between keys, we are interested to what degree we can approximate the CoF topology. As a baseline, we compare this projection to a direct projection of the IS, again using rPCA.

### 5.1 Training

We encode the WTC corpus as described in 4.3. As keys are characterized by distributions of pitch classes, the pitch range is set to  $m = 12$ . In order to examine the organization ability of the RBM under different settings, we use



**Figure 4.** 2-D visualization of n-grams in the FS using rPCA. N-grams belonging to a key have the same color, each centroid is marked with the corresponding cluster’s key label. (Best viewed in color)



**Figure 5.** The average Procrustes Distances from major key centroids to the major CoF of 5 runs for different n-gram lengths after rPCA on the IS and on the FS. *pitch* and *pitch-duration* representations are used as input.

n-grams of various lengths, and also compare *pitch* and *pitch-duration* representations.

## 5.2 Evaluation

We use rPCA to project all n-grams in both the IS and the FS into a two-dimensional space. In this space, for each key we determine the mean of all n-grams created from pieces in that key. The organization of those centroids is then compared to the organization of keys in the CoF by computing the PD of both shapes, separately for major and minor keys. To make different expansions of data points in space comparable, the PD is finally divided by the perimeter of the target CoF.

## 5.3 Results and Discussion

Figure 4 shows the organization of n-grams in the FS. Cluster centers are organized similarly to how keys are orga-

nized in the CoF, which is consistent with the representations of the probe tone ratings obtained by Krumhansl and Kessler [9, 10]. Note that relative minors tend to be shifted counterclockwise with respect to their major counterparts. This occurs in Krumhansl’s results as well [10, pp. 43], and can be explained by two factors, namely the alteration of the sixth degree in the melodic minor scale, which is identical to the seventh degree of the dominant of the relative major counterpart (e.g. the melodic Am scale shares the F# with the G major scale, the dominant of C major), and due to the tonal modulations concerning the form of the piece (e.g. fugues in minor keys tend to have certain passages in the relative major, while fugues in major keys tend to have passages in both the relative minor and the dominant).

Figure 5 shows, that the Procrustes Distance to the CoF tends to stabilize at a minimum with an n-gram length of about nine. This can be explained by the fact that n-grams of that length contain enough information to obtain the respective distribution of a key well enough. Adding duration information clearly impedes the organization of clusters in a CoF topology. As the occurrence of notes in the WTC is strongly correlated to their absolute duration (see Figure 1), and rhythmic information is not directly linked to the CoF organization, this is not unexpected. Interestingly, for larger n-gram sizes the FS of the RBM is not disrupted as much by the inclusion of distractive information as the rPCA on the IS.

## 6. ORGANIZATION OF PITCHES

A probe-tone test, proposed by Krumhansl et al. [9, 10], consists of a set of *musical stimuli* (such as scales, chord cadences, or musical pieces) that unambiguously instantiate a specific key, and a set of *probe tones*, typically the set of 12 pitch classes. Listeners are then required to rate on a numerical scale, from 1 (“very bad”) to 7 (“very good”), how well the probe tones fit the musical stimulus. In order to explore the hierarchical event organization of pitches induced by the RBM, we compare our model with a particular probe tone test conducted by Krumhansl and Shepard [10, cited by Krumhansl]. In this specific experiment, the musical stimulus consisted of an incomplete C major scale (in both ascending and descending contexts), and listeners were asked to give a numerical rating of the degree to which each probe tone fits the scale. The stimuli of this particular setup are illustrated in part Figure 6 a), while the probe tones are shown in Figure 6 c). The participants of the experiment were divided in three groups according to their number of years of formal musical training.

### 6.1 Training

As we are only interested in the ability of the model to learn tonal hierarchies in major and minor mode, we transpose all melodies to C major and C minor, respectively. In order to remain consistent with the aforementioned experiment of Krumhansl & Shepard, rather than using pitch-classes, we allow the training data to be in a range of three octaves, ranging from MIDI pitch numbers 48 to 74 (such that both the stimuli and the probe tones can be represented



**Figure 6.** Stimuli/probe tones used in the probe-tone test.

without wrapping). Most of the n-grams of the transposed WTC data fall in that range, or can be transposed octave-wise to fall in that range. N-grams for which this is not possible are ignored.

Since the fugues from the WTC contain certain tonal modulations, in order to train the RBM with prototypical examples of major and minor scales, all n-grams are classified using the Krumhansl & Kessler key-finding algorithm [10, cited by Krumhansl] and those whose annotated key is not the same as that identified by the classifier (ca. 53% of the corpus) are removed. The training is executed as described in 4.1.1.

## 6.2 Evaluation

Two different probe tone tests are conducted. The first test aims to reproduce the setup by Krumhansl and Shepard, and thus, the stimuli consist of the major ascending (starting from C3) and descending scales (starting from C5) shown in Figure 6 a). For the second experiment, the stimuli consist of ascending and descending major and melodic minor scales, but this time both are generated in the middle C octave, as shown in Figure 6 b). For both tests, the set of probe tones consist of all notes of the chromatic scale (starting from C4) as shown in Figure 6 c). We construct n-grams of length 8, consisting of the 7 notes of the target stimulus and a probe tone as the last note. This results in visible vectors  $\mathbf{v}_{pt}$  of length  $36 \times 8$ . The free energy corresponding to each combination of stimulus and probe tone is calculated using Eq. (1). In order to compare our results to those of human listeners, these energies are scaled using an affine transformation as follows:

$$\text{Judgment}(\mathbf{v}_{pt}) = \alpha \mathcal{F}(\mathbf{v}_{pt}) - \beta, \quad (3)$$

where the constants  $\alpha, \beta$  are selected such that the mean and the variance of the scaled energy are equal to those of the judgments reported in [10].

## 6.3 Results and Discussion

Figure 7 shows the results of the probe tone test, and in Table 1 the correlations of the RBM judgments with respect to those of expert and untrained listeners are presented. These results suggest that the model can learn some event hierarchy structures, such as the prevalence of diatonic over chromatic notes, similar to the judgment of

Group	$r$	$p$ -value
Expert ascending	0.7213	0.0054
Untrained ascending	0.7942	0.0012
Expert descending	0.7985	0.0011
Untrained descending	0.8344	0.0004

**Table 1.** Pearson correlations and  $p$ -values for the judgments of the probe tone tests.

trained listeners. In addition, the model develops a sense for melodic direction, preferring probe tones close to the final notes of the stimulus, which is consistent with the ratings of untrained listeners. Stimulated in the middle octave, the model is able to distinguish major and minor modes, especially the major and minor thirds reflect the characteristics of the respective diatonic triads. The model responses do not show explicit octave equivalence, since C and C' are not equally emphasized. Still it is interesting to note that a stimulus in the lower octave has implications on the pitch expectations in the middle octave, and that these implications are in correspondence with the tonal hierarchy of the key implied by the stimulus.

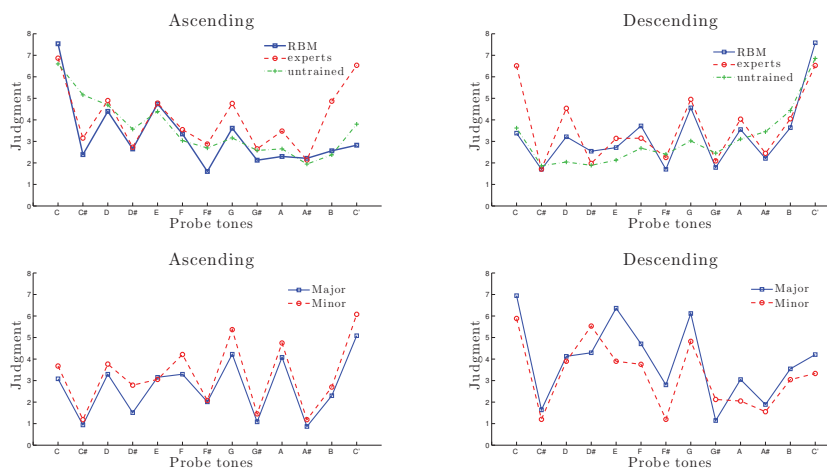
## 7. CONCLUSION

In this paper we show that tonal structure can be learned from musical data with an RBM using unsupervised training with a limited set of monophonic melodies. The model is able to reproduce the topology of the CoF using *pitch* n-gram representations of the input data. We found that for successful inference of the CoF, a minimal n-gram length of nine notes is needed, and that longer n-grams do not lead to better representations. Furthermore, although duration information profoundly disturbs the learning of tonal structure through the baseline rPCA method, the RBM model is less affected by distracting duration information.

By way of a probe tone test, we explored the organization of pitches in the context of major and minor modes. Our results show the model was able to learn several aspects of tonal structure, in particular the hierarchical prevalence of diatonic over chromatic tones. Comparing results with Krumhansl's probe tone experiments on human subjects with different levels of musical training do not yield a conclusive classification of the model: the model displays aspects of both untrained and trained subjects.

An important feature of tonal perception in trained subjects is octave equivalence. This feature was not well-reproduced by the model. It is possible that a pre-condition for octave-equivalence is the harmonic overlap of octaves. In our current setup, the overtone structure of tones is not represented. To test this hypothesis, we intend to investigate whether using harmonic tone representations leads to stronger octave-equivalence in the the model.

Furthermore we wish to investigate which factors induce more expert-like perception of tonal structure. Possible factors include the size of the training data, and the depth of the model (in terms of hidden layers).



**Figure 7.** (Top) Comparison of the judgments for the probe tones between the RBM and human listeners for both ascending (left) and descending (right) major stimulus in the lower and upper octave, respectively. (Bottom) Comparison of the judgments for the probe tones of the RBM for both major and melodic minor stimulus in the middle octave. In all cases, responses are measured in the middle octave.

## 8. ACKNOWLEDGMENTS

This work is supported by the European Union Seventh Framework Programme, through the Lrn2Cre8 project (FET grant agreement no. 610859). We thank Geraint Wiggins, Kat R. Agres and Jamie Forth for valuable suggestions and commentaries on this work.

## 9. REFERENCES

- [1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Verlag, 2009.
- [2] H. Goh, N. Thome, and M. Cord. Biasing restricted Boltzmann machines to manipulate latent selectivity and sparsity. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, pages 1–8, 2010.
- [3] E. Gómez. *Tonal description of music audio signals*. PhD thesis, Universitat Pompeu Fabra, Barcelona, 2006.
- [4] C. Goodall. Procrustes Methods in the Statistical Analysis of Shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(2):285–339, 1991.
- [5] G. E. Hinton. A practical guide to training restricted Boltzmann machines. Tech. Report UTML TR 2010-003, Department of Computer Science, University of Toronto, 2010.
- [6] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.
- [7] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.
- [8] T. Kohonen. Self-organized formation of topologically correct feature maps. *Biol. Cybernetics*, 43:59–69, 1982.
- [9] C. L. Krumhansl and L. L. Cuddy. A theory of tonal hierarchies in music. In *Handbook of Auditory Research*. Springer, New York, 2010.
- [10] C. L. Krumhansl. *Cognitive foundations of musical pitch*. Cognitive foundations of musical pitch. Oxford University Press, New York, 1990.
- [11] H. Lee, C. Ekanadham, and A. Y. Ng. Sparse deep belief net model for visual area V2. In *Advances in Neural Information Processing Systems 20*, pages 873–880. 2008.
- [12] M. Leman. A model of retroactive tone center perception. *Music Perception*, 12(4):439–471, 1995.
- [13] V. Rokhlin, A. Szlam, and M. Tygert. A randomized algorithm for principal component analysis. *arXiv.org*, page 2274, 2008.
- [14] J. R. Saffran, E. K. Johnson, R. N. Aslin, and E. L. Newport. Statistical learning of tone sequences by human infants and adults. *Cognition*, 70(1):27–52, 1999.
- [15] F. Salzer. *Structural hearing; tonal coherence in music*. New York, Dover Publications, 1962.
- [16] H. Schenker. *Harmony*. University of Chicago Press, 1980.
- [17] N. A. Smith and M. A. Schmuckler. The Perception of Tonal Structure Through the Differentiation and Organization of Pitches. *Journal of Exp. Psych.: Human Perception and Performance*, 30(2):268–286, 2004.
- [18] B. Tillmann, J. J. Bharucha, and E. Bigand. Implicit learning of tonality: a self-organizing approach. *Psychological review*, 107(4):885–913, 2000.
- [19] P. Toiviainen and C. L. Krumhansl. Measuring and modeling real-time responses to music: The dynamics of tonality induction. *Perception*, 32(6):741–766, 2003.



# EXPLOITING INSTRUMENT-WISE PLAYING/NON-PLAYING LABELS FOR SCORE SYNCHRONIZATION OF SYMPHONIC MUSIC

**Alessio Bazzica**      **Cynthia C. S. Liem**      **Alan Hanjalic**  
 Delft University of Technology   Delft University of Technology   Delft University of Technology  
 a.bazzica@tudelft.nl    c.c.s.liem@tudelft.nl    a.hanjalic@tudelft.nl

## ABSTRACT

Synchronization of a score to an audio-visual music performance recording is usually done by solving an audio-to-MIDI alignment problem. In this paper, we focus on the possibility to represent both the score and the performance using information about which instrument is active at a given time stamp. More specifically, we investigate to what extent instrument-wise “playing” (P) and “non-playing” (NP) labels are informative in the synchronization process and what role the visual channel can have for the extraction of P/NP labels. After introducing the P/NP-based representation of the music piece, both at the score and performance level, we define an efficient way of computing the distance between the two representations, which serves as input for the synchronization step based on dynamic time warping. In parallel with assessing the effectiveness of the proposed representation, we also study its robustness when missing and/or erroneous labels occur. Our experimental results show that P/NP-based music piece representation is informative for performance-to-score synchronization and may benefit the existing audio-only approaches.

## 1. INTRODUCTION AND RELATED WORK

Synchronizing an audio recording to a symbolic representation of the performed musical score is beneficial to many tasks and applications in the domains of music analysis, indexing and retrieval, like audio source separation [4, 9], automatic accompaniment [2], sheet music-audio identification [6] and music transcription [13]. As stated in [7], “sheet music and audio recordings represent and describe music on different semantic levels” thus making them complementary for the functionalities they serve.

The need for effective and efficient solutions for audio-score synchronization is especially present for genres like symphonic classical music, for which the task remains challenging due to the typically long duration of the pieces and a high number of instruments involved [1]. The existing solutions usually turn this synchronization problem

instrument 1	NP	NP	P	P	P	...	NP
instrument 2	P	P	P	NP	NP	...	NP
...							
instrument N	NP	NP	P	P	NP	...	P
	1	2	3	4	5	...	T
	→ time						

**Figure 1:** An illustration of the representation of a symphonic music piece using the matrix of playing/non-playing labels.

into an audio-to-audio alignment one [11], where the score is rendered in audio form using its MIDI representation.

In this paper, we investigate whether sequences of playing (P) and non-playing (NP) labels, extracted per instrument continuously over time, can alternatively be used to synchronize a recording of a music performance to a MIDI file. At a given time stamp, the P (NP) label is assigned to an instrument if it is (not) being played. If such labels are available, a representation of the music piece as illustrated in Figure 1 can be obtained: a matrix encoding the P/NP “state” for different instruments occurring in the piece at subsequent time stamps. Investigating the potential of this representation for synchronization purposes, we will address the following research questions:

- RQ1: How robust is P/NP-based synchronization in case of erroneous or missing labels?
- RQ2: How does synchronizing P/NP labels behave at different time resolutions?

We are particularly interested in this representation, as P/NP information for orchestra musicians will also be present in the signal information of a recording. While such information will be hard to obtain from the audio channel, it can be obtained from the visual channel. Thus, in case an audio-visual performance is available, using P/NP information opens up possibilities for video-to-score synchronization as a means to solve a score-to-performance synchronization problem.

The rest of the paper is structured as follows. In Section 2, we formulate the performance-to-score synchronization problem in terms of features based on P/NP labels. Then, we explain how the P/NP matrix is constructed to represent the score (Section 3) and we elaborate on the possibilities for extracting the P/NP matrix to represent the analyzed performance (Section 4). In Section 5 we propose an efficient method for solving the synchronization problem. The experimental setup is described in Section 6 and in Section 7 we report the results of our



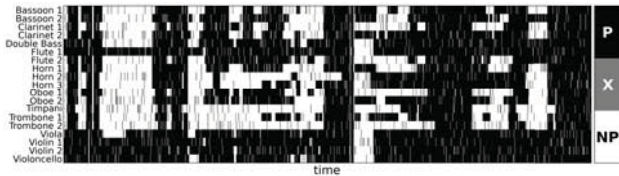


Figure 2: Example of a  $M_{\text{PNP}}$  matrix with missing labels.

experimental assessment of the proposed synchronization methodology and provide answers to our research questions. The discussion in Section 8 concludes the paper.

## 2. PROBLEM DEFINITION

Given an audio-visual recording of a performance and a symbolic representation of the performed scores, we address the problem of synchronizing these two resources by exploiting information about the instruments which are active over time.

Let  $L = \{-1, 0, 1\}$  be a set encoding the three labels non-playing (NP), missing (X) and playing (P). Let  $M_{\text{PNP}} = \{m_{ij}\}$  be a matrix of  $N_I \times N_T$  elements where  $N_I$  is the number of instruments and  $N_T$  is the number of time points at which the P/NP state is observed. The value of  $m_{ij} \in L$  represents the state of the  $i$ -th instrument observed at the  $j$ -th time point ( $1 \leq i \leq N_I$  and  $1 \leq j \leq N_T$ ). An example of  $M_{\text{PNP}}$  is given in Figure 2.

We now assume that the matrices  $M_{\text{PNP}}^{\text{AV}}$  and  $M_{\text{PNP}}^{\text{S}}$  are given and represent the P/NP information respectively extracted by the audio-visual recording and the sheet music. The two matrices have the same number of rows and each row is associated to each instrumental part. The number of columns, i.e. observations over time, is in general different. The synchronization problem can be then formulated as the problem of finding a time map  $f_{\text{sync}} : \{1 \dots N_T^{\text{AV}}\} \rightarrow \{1 \dots N_T^{\text{S}}\}$  linking the observation time points of the two resources.

## 3. SCORE P/NP REPRESENTATION

For a given piece, we generate one P/NP matrix  $M_{\text{PNP}}^{\text{S}}$  for the score relying on the corresponding MIDI file as the information source.

We start generating the representation of the score by parsing the data of each available track in the given MIDI file. Typically, one track per instrument is added and is used as a symbolic representation of the instrumental part’s score. More precisely, when there is more than one track for the same instrument (e.g. Violin 1, Violin 2 - which are two different instrumental parts), we keep both tracks as separate. In the second step, we use a sliding window that moves along the MIDI file and derive a P/NP label per track and window position. A track receives a P label if there is at least one note played within the window. We work with the window in order to comply with the fact that a played note has a beginning and end and therefore lasts for an interval of time. In this sense, a played note is registered when there is an overlap between the sliding window and the play interval of that note.

The length of the window is selected such that short rests within a musical phrase do not lead to misleading P-NP-P switches. We namely consider a musician in the “play” mode if she is within the “active” sequence of the piece with respect to her instrumental part’s score, independently whether at some time stamps no notes are played. In our experiments, we use a window length of 4 seconds which has been determined by empirical evaluation, and a step-size of 1 second. This process generates one label per track every second.

In order to generalize the parameter setting for window length and offset, we also related them to the internal MIDI file time unit. For this purpose, we set a reference value for the tempo. Once the value is assigned, the sliding window parameters are converted from seconds to beats. The easiest choice is adopting a fixed value of tempo for every performance. Alternatively, when an audio-visual recording is available, the reference tempo can be estimated as the number of beats in the MIDI file divided by the length of the recording expressed in minutes. A detailed investigation of different choices of the tempo is reported in [6].

## 4. PERFORMANCE P/NP REPRESENTATION

While an automated method could be thought of to extract the P/NP matrix  $M_{\text{PNP}}^{\text{AV}}$  from a given audio-visual recording, developing such a method is beyond the scope of this paper. Instead, our core focus is assessing the potential of such a matrix for synchronization purposes, taking into account the fact that labels obtained from real-world data can be noisy or even missing. We therefore deploy two strategies which mimic the automated extraction of the  $M_{\text{PNP}}^{\text{AV}}$  matrices. We generate them: (i) artificially, by producing (noisy) variations of the P/NP matrices derived from MIDI files (Section 4.1), and (ii) more realistically, by deriving the labels directly from the visual channel of a recording in a semi-automatic way (Section 4.2).

### 4.1 Generating synthetic P/NP matrices

The first strategy produces synthetic P/NP matrices by analyzing MIDI files as follows. Similarly to the process of generating a P/NP matrix for the score, we apply a sliding window to the MIDI file and extract labels per instrumental track at each window position. This time, however, time is randomly warped, i.e. the sliding window moves over time with non-constant velocity. More specifically, we generate random time-warping functions by randomly changing slope every 3 minutes and by adding a certain amount of random noise in order to avoid perfect piecewise linear functions. In a real audio-visual recording analysis pipeline, we expect that erroneous and missing P/NP labels will occur. Missing labels may occur if musicians cannot be detected, e.g. because of occlusion or leaving the camera’s angle of view in case of camera movement. In order to simulate such sources of noise, we modify the generated P/NP tracks by randomly flipping and/or deleting predetermined amounts of labels at random positions of the P/NP matrices.

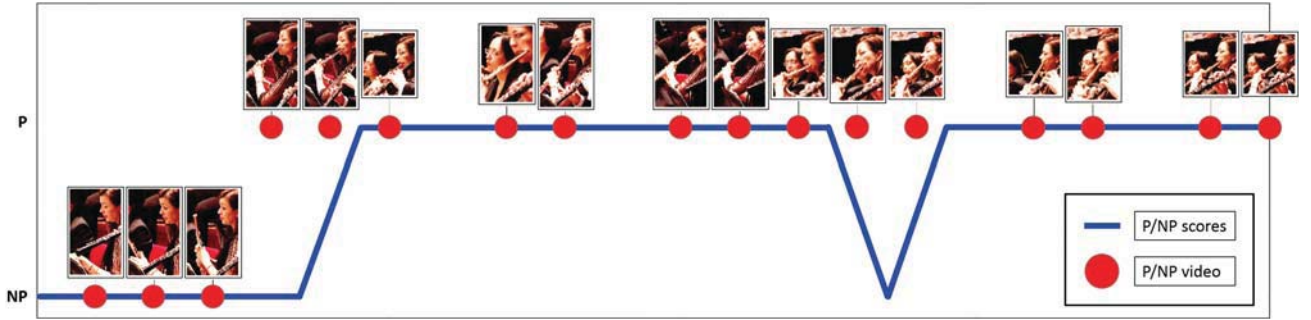


Figure 3: Example of P/NP labels extracted from the visual channel (red dots) and compared to labels extracted by the score (blue line).

## 4.2 Obtaining P/NP matrices from a video recording

The second strategy more closely mimics the actual video analysis process and involves a simple, but effective method that we introduce for this purpose. In this method, we build on the fact that video recordings of a symphonic music piece are typically characterized by regular close-up shots of different musicians. From the key frames representing these shots, as illustrated by the examples in Figure 4, it can be inferred whether they are using their instrument at that time stamp or not, for instance by investigating their body pose [14].



Figure 4: Examples of body poses indicating playing/non-playing state of a musician.

In the first step, a key frame is extracted every second in order to produce one label per second, as in the case of the scores. Faces are detected via off-the-shelf face detectors and upper-body images are extracted by extending the bounding box's areas of face detector outputs. We cluster the obtained images using low-level global features encoding color, shape and texture information. Clustering is done using  $k$ -means with the goal to isolate images of different musicians. In order to obtain high precision, we choose a large value for  $k$ . As a result, we obtain clusters mostly containing images of the same musician, but also multiple clusters for the same musician. Noisy clusters (those not dominated by a single musician) are discarded, while the remaining are labeled by linking them to the correspondent track of the MIDI file (according to the musician's instrument and position in the orchestra, i.e. the instrumental part). In order to label the upper-body images as P/NP, we generate sub-clusters using the same features as those extracted in the previous (clustering) step. Using once again  $k$ -means, but now with  $k$  equal to 3 (one cluster meant for P labels, one for NP and one extra label for possible outliers), we build sub-clusters which we label as either playing (P), non-playing (NP) or undefined (X). Once the labels for every musician are obtained, they are aggregated by instrumental part (e.g. the labels from all the Violin 2 players are combined by majority voting). An example of a P/NP subsequence extracted by visual analysis is given in Figure 3.

## 5. SYNCHRONIZATION METHODOLOGY

In this section, we describe the synchronization strategy used in our experiments. The general idea is to compare configurations of P/NP labels for every pair of performance-score time points and produce a distance matrix. The latter can then serve as input into a synchronization algorithm, for which we adopt the well-known dynamic time warping (DTW) principle. This implies we will not be able to handle undefined amounts of repeats of parts of the score. However, this is a general issue for DTW also holding for existing synchronization approaches, which we consider out of the scope of this paper.

In order to find the time map between performance and score, we need to solve the problem of finding time links between the given  $M_{PNP}^{AV}$  and  $M_{PNP}^S$  matrices. To this end, we use a state-of-the-art DTW algorithm [12].

### 5.1 Computing the distance matrix

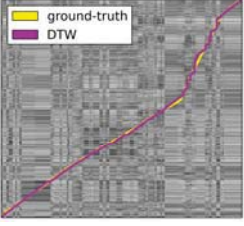
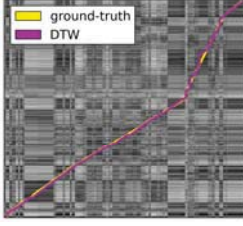
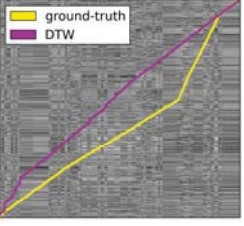
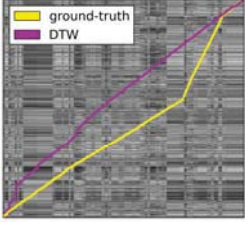
Ten Holt et. al. [12] compute the distance matrix through the following steps: (i) both dimensions of the matrices are normalized to have zero mean and unit variance, (ii) optionally a Gaussian filter is applied, and (iii) pairs of vectors are compared using the city block distance. In our case, we take advantage of the fact that our matrices contain values belonging to the finite set of 3 different integers, namely the set  $L$  introduced in Section 2. This enables us to propose an alternative, just as effective, but more efficient method to compute the distance matrix.

Let  $\mathbf{m}_j^{AV}$  and  $\mathbf{m}_k^S$  be two column vectors respectively belonging to  $M_{PNP}^{AV}$  and  $M_{PNP}^S$ . To measure how (dis-)similar those two vectors are, we define a *correlation score*  $s_{jk}$  as follows:

$$s_{jk} = \text{corr}(\mathbf{m}_j^{AV}, \mathbf{m}_k^S) = \sum_{i=1}^{N_I} m_{ij}^{AV} \cdot m_{ik}^S$$

From such definition, it follows that a pair of observed matching labels add a positive unitary contribution. If the observed labels do not match, the added contribution is unitary and negative. Finally, if one or both labels are not observed (i.e. at least one of them is X), the contribution is 0. Hence, it also holds  $-N_I \leq s_{jk} \leq +N_I$ . The maximum is reached only if the two vectors are equal. Correlation scores can be efficiently computed as dot-product of the given P/NP matrices, namely as  $(M_{PNP}^{AV})^\top M_{PNP}^S$ .

The distance matrix  $D = \{d_{jk}\}$ , whose values are zero when the compared vectors are equal, can now be computed as  $d_{jk} = N_I - s_{jk}$ . As a result,  $D$  will have  $N_I^{AV}$

noisy $M_{\text{PNP}}$		very noisy $M_{\text{PNP}}$	
			
Ten Holt et. al.	our method	Ten Holt et. al.	our method

**Table 1:** Comparing our distance matrix definition to Ten Holt et. al. [12]. By visual inspection, we observe comparable alignment performances. However, the computation of our distance matrix is much faster.

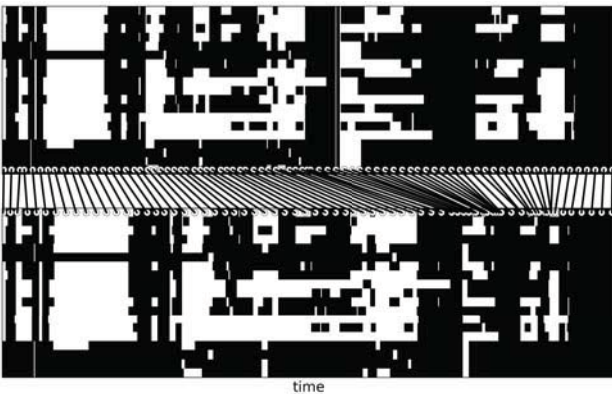
rows and  $N_T^S$  columns. When the correlation is the highest, namely equal to  $N_I$ , the distance will be zero.

Our approach has two properties that make the computation of  $D$  fast:  $D$  is computed via the dot product and it contains integer values only (as opposed to standard methods based on real-valued distances). As shown in Table 1, both the distance matrix proposed in [12] and using our definition produce comparable results. Since our method allows significantly faster computation (up to 40 times faster), we adopt it in our experiments.

## 5.2 Dynamic Time Warping

Once the distance matrix  $D$  is computed, the time map between  $M_{\text{PNP}}^{\text{AV}}$  and  $M_{\text{PNP}}^{\text{S}}$  is determined by solving the optimization problem:  $P^* = \arg \min_P \text{cost}(D, P)$  where  $P = \{(p_\ell \rightsquigarrow p_{\ell+1})\}$  is a path through the items of  $D$  having a cost defined by the function  $\text{cost}(D, P)$ . More specifically,  $p_\ell = (i_\ell^{\text{AV}}, i_\ell^{\text{S}})$  is a coordinate of an element in  $D$ . The cost function is defined as  $\text{cost}(D, P) = \sum_{\ell=1}^{|P|} d_{i_\ell^{\text{AV}}, i_\ell^{\text{S}}}$ . The aforementioned problem is efficiently solved via dynamic programming using the well-known dynamic time warping (DTW) algorithm. Examples of  $P^*$  paths computed via DTW are shown in the figures of Table 1.

Once  $P^*$  is found, the time map  $f_{\text{sync}}$  is computed through the linear interpolation of the correspondences in  $P^*$ , i.e. the set of coordinates  $\{p_\ell^* = (i_\ell^{\text{AV}}, i_\ell^{\text{S}})\}$ . This map allows to define correspondences between the two matrices, as shown in the example of Figure 5.

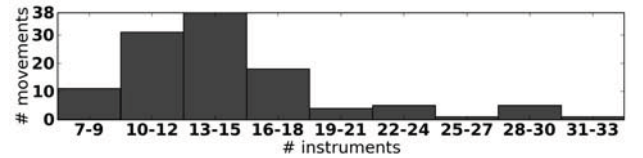


**Figure 5:** Example of produced alignment between two fully-observed  $M_{\text{PNP}}$  matrices.

## 6. EXPERIMENTAL SETUP

In this section, we describe our experimental setup including details about the dataset. In order to ensure the reproducibility of the experiments, we release the code and share the URLs of the analyzed freely available MIDI files<sup>1</sup>.

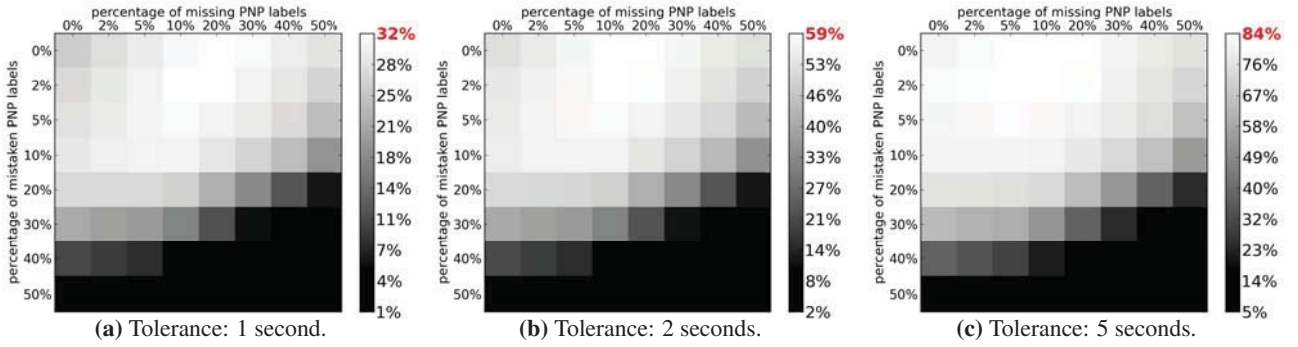
We evaluate the performances of our method on a set of 29 symphonic pieces composed by Beethoven, Mahler, Mozart and Schubert. The dataset consists of 114 MIDI files. Each MIDI file contains a number of tracks corresponding to different parts performed in a symphonic piece. For instance, first and second violins are typically encoded in two different parts (e.g. “Violin 1” and “Violin 2”). In such a case, we keep both tracks separate since musicians in the visual channel can be labeled according to the score which they perform (and not just by their instrument). We ensured that the MIDI files contain tracks which are mutually synchronized (i.e. MIDI files of type 1). The number of instrumental parts, or MIDI tracks, ranges between 7 and 31 and is distributed as shown in Figure 7.



**Figure 7:** Distribution of the number of instrumental parts across performances in the data set.

For each MIDI file, we perform the following steps. First, we generate one  $M_{\text{PNP}}^{\text{S}}$  matrix using a fixed reference tempo of 100 BPM. The reason why we use the same value for every piece is that we evaluate our method on artificial warping paths, hence we do not need to adapt the sliding window parameters to any actual performance. Then we generate one random time-warping function which has two functions: (i) it is used as ground-truth when evaluating the alignment performance, and (ii) it is used to make one time-warped P/NP matrix  $M_{\text{PNP}}^{\text{AV}}$ . The latter is used as template to build noisy copies of  $M_{\text{PNP}}^{\text{AV}}$  and evaluate the robustness of our method. Each template P/NP matrix is used to generate a set of noisy P/NP matrices which are affected by different pre-determined amounts of noise. We consider two sources of noise: mistaken and missing labels. For both sources, we generate

<sup>1</sup> <http://homepage.tudelft.nl/f8j6a/ISMIR2014baz.zip>



**Figure 6:** Average matching rates as a function of the percentage of mistaken and/or missing labels at different tolerance thresholds.

the following percentages of noisy labels: 0% (noiseless), 2%, 5%, 10%, 20%, 30%, 40% and 50%. For every pair of noise percentages, e.g. 5% mistaken + 10% missing, we create 5 different noisy versions of the original P/NP matrix<sup>2</sup>. Therefore, for each MIDI file, the final set of matrices has the size  $1 + (8 \times 8 - 1) \times 5 = 316$ . Overall, we evaluate the temporal alignment of  $316 \times 114 = 36024$  P/NP sequences.

For each pair of  $M_{PNP}$  matrices to be aligned, we compute the matching rate by sampling  $f_{sync}$  and measuring the distance from the true alignment. A match occurs when the distance between linked time points is below a threshold. In our experiments, we evaluate the matching rate using three different threshold values: 1, 2 and 5 seconds.

Finally, we apply the video-based P/NP label extraction strategy described in Section 4.2 to a multiple camera video recording of the 4th movement of Symphony no. 3 op. 55 of Beethoven performed by the Royal Concertgebouw Orchestra (The Netherlands). For this performance, in which 54 musicians play 19 instrumental parts, we use the MIDI file and the correspondent performance-score temporal alignment file which are shared by the authors of [8]. The latter is used as ground truth when evaluating the synchronization performance.

## 7. RESULTS

In this section, we present the obtained results and provide answers to the research questions posed in Section 1. We start by presenting in Figure 6 the computed matching rates in 3 distinct matrices, one for each threshold value. Given a threshold, the overall matching rates are reported in an  $8 \times 8$  matrix since we separately compute the average matching rate for each pair of mistaken-missing noise rates. Overall, we see two expected effects: (i) the average matching rate decreases for larger amounts of noise, and (ii) the performance increases with the increasing threshold. What was not expected is the fact that the best performance is not obtained in the noiseless case. For instance, when the threshold is 5 seconds, we obtained an average matching rate of 81.7% in the noiseless case and 85.0% in the case of 0% mistaken and 10% missing labels. One possible explanation is that 10% missing labels could give more “freedom” to the DTW algorithm than the noiseless

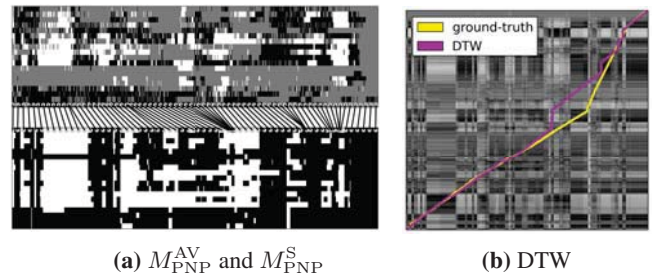
<sup>2</sup> We do not add extra copies for the pair (0%,0%), i.e. the template matrix.

case. Such freedom may lead to a better global optimization. In order to fully understand the reported outcome, however, further investigation is needed, which we leave for future work.

As for our first research question, we conclude that the alignment through P/NP sequences is more robust to missing labels than to mistaken ones. We show this by the fact that the performance for 0% mistaken and 50% missing labels are higher than in the opposite case, namely for 50% mistaken and 0% missing labels. In general the best performance is obtained for up to 10% mistaken and 30% missing labels.

In the second research question we address the behavior at different time resolutions. Since labels are sampled every second, it is clear why acceptable matching rates are only obtained at coarse resolution (namely for a threshold of 5 seconds).

Finally, we comment on the results obtained when synchronizing through the P/NP labels assigned via visual analysis. The P/NP matrix, shown in Figure 8a, is affected by noise as follows: there are 53.95% missing and 8.65% mistaken labels.



**Figure 8:** Real data example: P/NP labels by analysis of video

We immediately notice the large amount of missing labels. This is mainly caused by the inability to infer a P/NP label at those time points when all the musicians of a certain instrumental part are not recorded. Additionally, some of the image clusters generated as described in Section 4.2 are not pure and hence labeled as X.

The obtained synchronization performance at 1, 2 and 5 seconds of tolerance are respectively 18.74%, 34.49% and 60.70%. This is in line with the results obtained with synthetic data whose performance at 10% of mistaken labels and 50% of missing for the three different tolerances are 24.3%, 44.2% and 65.9%. Carrying out the second exper-

iment was also useful to get insight about the distribution of missing labels. By inspecting Figure 8a, we notice that such a type of noise is not randomly distributed. Some musicians are sparsely observed over time hence leading to missing labels patterns which differ from uniform distributed random noise.

## 8. DISCUSSION

In this paper, we presented a novel method to synchronize score information of a symphonic piece to a performance of this piece. In doing this, we used a simple feature (the act of playing or not) which trivially is encoded in the score, and feasibly can be obtained from the visual channel of an audio-visual recording of the performance. Unique about our approach is that both for the score and the performance, we start from measuring individual musician contributions, and only then aggregate up to the full ensemble level to perform synchronization. This makes a case for using the visual channel of an audio-visual recording. In the audio channel, which so far has predominantly been considered for score-to-performance synchronization, even if separate microphones are used per instrument, different instruments will never be fully isolated from each other in a realistic playing setting. Furthermore, audio source separation for polyphonic orchestral music is far from being solved. However, in the visual channel, different players are separated by default, up to the point that a first clarinet player can be distinguished from a second clarinet player, and individual contributions can be measured for both.

Our method still works at a rough time resolution, and lacks the temporal sub-second precision of typical audio-score synchronization methods. However, it is computationally inexpensive, and thus can quickly provide a rough synchronization, in which individual instrumental part contributions are automatically marked over time. Consequently, interesting follow-up approaches could be devised, in which cross- or multi-modal approaches might lead to stronger solutions, as already argued in [3, 10].

For the problem of score synchronization, a logical next step is to combine our analysis with typical audio-score synchronization approaches, or approaches generally relying on multiple synchronization methods, such as [5], to investigate whether a combination of methods improves the precision and efficiency of the synchronization procedure. Our added visual information layer can further be useful for e.g. devising structural performance characteristics, e.g. the occurrence of repeats. Our general synchronization results will also be useful for source separation procedures, since the obtained P/NP annotations indicate active sound-producing sources over time. Furthermore, results of our method can serve applications focusing on studying and learning about musical performances. We can easily output an activity map or multidimensional time-scrolling bar, visualizing which orchestra parts are active over time in a performance. Information about expected musical activity across sections can also help directing the focus of an audience member towards dedicated players or the full ensemble.

Finally, it will be interesting to investigate points where P/NP information in the visual and score channel clearly disagree. For example, in Figure 3, some time after the flutist starts playing, there is a moment where the score indicates a non-playing interval, while the flutist keeps a playing pose. We hypothesize that this indicates that, while a (long) rest is notated, the musical discourse actually still continues. While this also will need further investigation, this opens up new possibilities for research in performance analysis and musical phrasing, broadening the potential impact of this work even further.

**Acknowledgements** The research leading to these results has received funding from the European Union Seventh Framework Programme FP7 / 2007–2013 through the PHENICX project under Grant Agreement no. 601166.

## 9. REFERENCES

- [1] A. D'Aguanno and G. Vercellesi. Automatic Music Synchronization Using Partial Score Representation Based on IEEE 1599. *Journal of Multimedia*, 4(1), 2009.
- [2] R.B. Dannenberg and C. Raphael. Music Score Alignment and Computer Accompaniment. *Communications of the ACM*, 49(8):38–43, 2006.
- [3] S. Essid and G. Richard. Fusion of Multimodal Information in Music Content Analysis. *Multimodal Music Processing*, 3:37–52, 2012.
- [4] S. Ewert and M. Müller. Using Score-informed Constraints for NMF-based Source Separation. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 129–132. IEEE, 2012.
- [5] S. Ewert, M. Müller, and R.B. Dannenberg. Towards Reliable Partial Music Alignments Using Multiple Synchronization Strategies. In *Adaptive Multimedia Retrieval. Understanding Media and Adapting to the User*, pages 35–48. Springer, 2011.
- [6] C. Fremerey, M. Clausen, S. Ewert, and M. Müller. Sheet Music-Audio Identification. In *ISMIR*, pages 645–650, 2009.
- [7] C. Fremerey, M. Müller, and M. Clausen. Towards Bridging the Gap between Sheet Music and Audio. *Knowledge Representation for Intelligent Music Processing*, (09051), 2009.
- [8] M. Grachten, M. Gasser, A. Arzt, and G. Widmer. Automatic Alignment of Music Performances with Structural Differences. In *ISMIR*, pages 607–612, 2013.
- [9] Y. Han and C. Raphael. Informed Source Separation of Orchestra and Soloist Using Masking and Unmasking. In *ISCA-SAPA Tutorial and Research Workshop, Makuhari, Japan*, 2010.
- [10] C.C.S. Liem, M. Müller, D. Eck, G. Tzanetakis, and A. Hanjalic. The Need for Music Information Retrieval with User-centered and Multimodal Strategies. In *Proceedings of the 1st international ACM workshop MIRUM*, pages 1–6. ACM, 2011.
- [11] M. Müller, F. Kurth, and T. Röder. Towards an Efficient Algorithm for Automatic Score-to-Audio Synchronization. In *ISMIR*, 2004.
- [12] G.A. Ten Holt, M.J.T. Reinders, and E.A. Hendriks. Multi-dimensional Dynamic Time Warping for Gesture Recognition. In *13th annual conference of the Advanced School for Computing and Imaging*, volume 119, 2007.
- [13] R.J. Turetsky and D.P.W. Ellis. Ground-truth Transcriptions of Real Music from Force-aligned MIDI Syntheses. *ISMIR 2003*, pages 135–141, 2003.
- [14] B. Yao, J. Ma, and L. Fei-Fei. Discovering Object Functionality. In *ICCV*, pages 2512–2519, 2013.

# MULTI-STRATEGY SEGMENTATION OF MELODIES

**Marcelo Rodríguez-López**

Utrecht University  
m.e.rodriquezlopez@uu.nl

**Anja Volk**

Utrecht University  
a.volk@uu.nl

**Dimitrios Bountouridis**

Utrecht University  
d.bountouridis@uu.nl

## ABSTRACT

Melodic segmentation is a fundamental yet unsolved problem in automatic music processing. At present most melody segmentation models rely on a ‘single strategy’ (i.e. they model a single perceptual segmentation cue). However, cognitive studies suggest that multiple cues need to be considered. In this paper we thus propose and evaluate a ‘multi-strategy’ system to automatically segment symbolically encoded melodies. Our system combines the contribution of different single strategy boundary detection models. First, it assesses the perceptual relevance of a given boundary detection model for a given input melody; then it uses the boundaries predicted by relevant detection models to search for the most plausible segmentation of the melody. We use our system to automatically segment a corpus of instrumental and vocal folk melodies. We compare the predictions to human annotated segments, and to state of the art segmentation methods. Our results show that our system outperforms the state-of-the-art in the instrumental set.

## 1. INTRODUCTION

In Music Information Retrieval (MIR), segmentation refers to the task of dividing a musical fragment or a complete piece into smaller cognitively-relevant units (such as notes, motifs, phrases, or sections). Identifying musical segments aids (and in some cases enables) many tasks in MIR, such as searching and browsing large music collections, or visualising and summarising music. In MIR there are three main tasks associated with music segmentation: (1) the segmentation of musical audio recordings into notes, as part of transcription systems, (2) the segmentation of symbolic encodings of music into phrases, and (3) the segmentation of both musical audio recordings and symbolic encodings into sections. In this paper we focus on the second task, i.e. identifying segments resembling the musicological concept of *phrase*. Currently automatic segmentation of music into phrases deals mainly with monophony. Thus, this area is commonly referred to as *melody segmentation*.

When targeting melodies, segmentation is usually re-

duced to identifying *segment boundaries*, i.e. locate the points in time where one segment transitions into another.<sup>1</sup> Computer models of melody segmentation often focus on modelling *boundary cues*, i.e. the musical factors that have been observed or hypothesised to trigger human perception of boundaries. Two common examples of boundary cues are: (a) the perception of ‘gaps’ in a melody (e.g. the sensation of a ‘temporal gap’ due to long note durations or rests) and (b) the perception of repetitions (e.g. recognising a melodic figure as a modified instance of a previously heard figure). The first cue mentioned is thought to signal the *end* of phrases, and conversely the second one is thought to signal the *start* of phrases.

Findings in melodic segment perception studies suggest that, even in short melodic excerpts, listeners are able to identify multiple cues, and what is more, that the role and relative importance of these cues seems to be contextual [3, 6]. Yet, most computer models of melody segmentation rely on a *single strategy*, meaning that they often focus on modelling a single type of cue. For instance, [4] focuses on modelling cues related only to melodic gaps, while [1, 5] aim to modelling cues related only to melodic repetitions.

In this paper we propose and evaluate a *multi-strategy system* that combines single strategy models of melodic segmentation. In brief, our system first estimates the cues (and hence the single strategy models) that might be more ‘relevant’ for the segmentation of a particular input melody, combines the boundaries predicted by the models estimated relevant, and then selects which boundaries result in the ‘most plausible’ segmentation of the input melody.

**Contribution:** first, we bring together single strategy models that have not been previously tested in combination; second, our evaluation results show that our system outperforms the state-of-the-art of melody segmentation in instrumental folk songs.

The remainder of this paper is organised as follows: §2 reviews music segmentation related work using multi-strategy approaches, §3 presents a theoretical overview of the proposed system, §4 describes implementation details of the system, §5 describes and discusses our evaluation of the system, and finally, §6 provides conclusions and outlines possibilities of future work.

<sup>1</sup> Other subtasks associated to segmentation such as boundary pairing, as well as labelling of segments, are not considered.



© Marcelo Rodríguez-López, Anja Volk, Dimitrios Bountouridis.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Marcelo Rodríguez-López, Anja Volk, Dimitrios Bountouridis. “MULTI-STRATEGY SEGMENTATION OF MELODIES”, 15th International Society for Music Information Retrieval Conference, 2014.

## 2. RELATED WORK

Melody segmentation models often focus on modelling a single cue (e.g. [1, 4, 5]), leaving only a handful of models that have proposed ways to combine different cues. Perhaps the best known multi-strategy model is Grouper [11], which relies on three cues: temporal gaps, metrical parallelism, and segment length. Grouper employs temporal gap detection heuristics to infer a set of candidate boundaries, and uses dynamic programming to find an ‘optimal’ segmentation given the candidate boundaries and two regularisation constraints (metrical parallelism and segment length). Grouper constitutes the current state-of-the-art in melodic segmentation. However, Grouper relies entirely on temporal information, and as such might have difficulties segmenting melodies with low rhythmic contrast or no discernible metric.

Another multi-strategy model is ATTA [7], which merges gap, metrical, and self-similarity related cues. In ATTA the relative importance of each cue is assigned manually, requiring the tuning of over 25 parameters. Parameter tuning in ATTA is time consuming (estimated to be  $\sim 10$  mins per melody in [7]). Moreover, the parameters are non-adaptive (set at initialization), and thus make the model potentially insensitive to changes in the relative importance of a given cue during the course of a melody.

The main differences between the research discussed and ours are: (a) our system integrates single strategy models that have not been previously used (and systematically tested) in combination, and (b) our system provides ways to select which single strategy models to use for a particular melody. In §5.3.2 we compare our system to the two models that have consistently performed best in comparative studies, namely Grouper [11] and LBDM [4].<sup>2</sup>

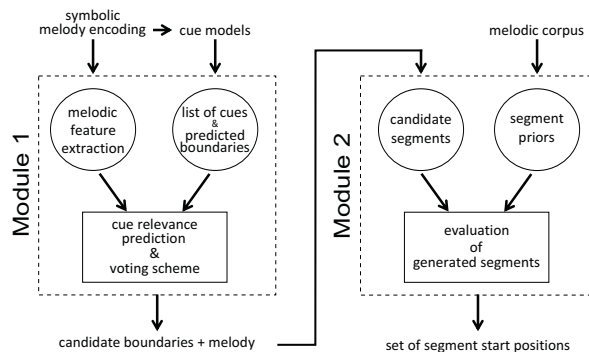
## 3. THEORETICAL OVERVIEW OF OUR SYSTEM

In this section we describe our system, depicted in Figure 1. In module 1, our system takes a group of single strategy segmentation models (henceforth ‘cue models’), selects which might be more relevant to segment the current input melody, and combines the estimated boundary locations into a single list. In module 2, the system assesses the segmentation produced by combinations of the selected boundary candidates in respect to corpus-learned priors on segment contour and segment length. Below we describe in more detail the input/output characteristics of our system, as well as each processing module.

### 3.1 Input/Output

The input to our system consists of a melody and a set of boundaries predicted by cue models. The melody is encoded as a sequence of temporally ordered note events  $e = e_1, \dots, e_i, \dots, e_n$ . In  $e$  each note event is represented by its chromatic pitch and quantized duration (onset, offset) values. The output of our system is a set of ‘optimum’ boundary locations  $b_{opt}$  of length  $m$ , constituting

<sup>2</sup> The manual tuning feature of ATTA made it impossible to include it in our evaluation.



**Figure 1.** General diagram of our system. Within the modules  $\circ$  = input elements, and  $\square$  = processing stages.

a set of segments  $S_{opt} = \{s_i\}_{1 \leq i < m}$ , where each segment  $s_i = [b_i, b_{i+1})$ .

### 3.1.1 Cue Models Characteristics

Each cue model transforms  $e$  into a set of sequences, each representing a melodic attribute (e.g. pitch class, inter-onset-interval, etc.). The specific set of attribute sequences produced by each cue model used within our system is discussed in §4.2. Each cue model processes the attribute sequences linearly, moving in steps of one event, producing a *boundary strength profile*. A boundary strength profile is simply a normalized vector of length  $n$ , where each element value encodes the strength with which a cue model ‘perceives’ a boundary at the temporal location of the element. In these profiles segment boundaries correspond to local maxima, and thus candidate boundary locations are obtained via peak selection. The method used to select peaks is discussed in §4.2.

### 3.2 Module 1: Multiple-Cue Boundary Detection

Module 1 takes as input a set of features describing the melody, and a set of boundary locations predicted by cue models. Module 1 is comprised of two processing stages, namely ‘cue relevance prediction’ and ‘voting scheme’. The first uses the input melodic features to estimate the ‘relevance’ of a given cue for the perception of boundaries in the input melody, and the second merges and filters the predicted boundary locations.

#### 3.2.1 Cue Relevance Prediction

For a given set of  $k$  cue models  $C = \{c_i\}_{1 \leq i \leq k}$ , and a set of  $h$  features describing the melodies  $F = \{f_j\}_{1 \leq j \leq h}$ , we need to estimate how well a given cue model might perform under a given performance measure  $M$  as  $P(M|C_i, F_j)$ . In this paper we use the common F1, *precision*, and *recall* measures to evaluate performance (see §5.2). In module 1 we focus on predicting a cue model’s *precision* (assuming high *recall* can be achieved by the combined set of candidate boundaries).

#### 3.2.2 Voting Scheme

Once we have estimated the relevance value of each cue model for the input melody  $P(M|C_i, F_j)$ , we combine the



candidate boundaries by simply adding the relevance values of candidate boundaries in close proximity (i.e.  $\pm 1$  note event apart). We assume that if boundaries from different cues are located  $\pm 1$  note event apart, one of them might be identifying a beginning and the other an end of segment, and thus for the processing in module 2 is beneficial to keep both.

The final output of this module is a single list of boundary locations  $b$ , each boundary with its own relevance value.

### 3.3 Module 2: Optimality-Based Segment Formation

Module 2 takes as input  $b$ ,  $e$ , and length/contour<sup>3</sup> priors computed from a melodic corpus. The task of this module is to find the ‘most plausible’ set of segments  $S_{opt}$  from the space of all possible candidate segmentations. The idea is to evaluate segmentations according to two empirical constraints: one, melodic segments tend to show small deviations from a ‘typical’ segment length, and two, melodic segments tend to show a reduced set of prototypical melodic contour shapes. We address the task of finding the most plausible set of segments given these two constraints as an optimisation problem. Thus, for a given candidate segmentation  $S_c = \{s_i\}_{1 \leq i < t}$ , derived from a subset of  $t$  candidate boundaries  $c \in b$ , where  $s_i = [c_i, c_{i+1})$ , our cost function is defined as:

$$C(S_c) = \sum_{i=1}^{t-1} T(s_i) \quad (1)$$

with

$$T(s_i) = \Phi(s_i) + \alpha(\Upsilon(s_i) + \Psi(s_i)) \quad (2)$$

Where,

- $\Phi(s_i)$  is the cost associated to each candidate boundary demarcating  $s_i$  (i.e. the inverse of the relevance value of each candidate boundary).
- $\Upsilon(s_i)$  is a cost associated to the deviation of  $s_k$  from an expected phrase contour. The cost of  $\Upsilon(s_i)$  is computed as  $-\log(\cdot)$  of the probability of the contour of the candidate phrase segment  $s_i$ .
- $\Psi(s_i)$  is a cost of the deviation from the length of  $s_i$  from an expected length. The cost of  $\Psi(s_i)$  is computed as  $-\log(\cdot)$  of the probability of the length of the candidate phrase segment  $s_i$ .
- $\alpha$  is a user defined parameter that balances the boundary related costs against the segment related costs.

Details for the computation of  $S_{opt}$  and priors on segment length/contour are given in §4.4.

<sup>3</sup> Melodic contour can be seen as an overall temporal development of pitch height

## 4. SYSTEM IMPLEMENTATION

In this section we first describe the selection and tuning of the cue models used within our system, then provide some details on the implementation of modules 1 and 2.

### 4.1 Cue Models: Selection

We selected and implemented four cue models based on two conditions: (a) the models have shown relatively high performance in previous studies, (b) the cues modelled have been identified as being important for melody segmentation within music cognition studies. All implemented models follow the same processing chain, described in §3.1, i.e. each model derives a set of melodic attribute sequences, processes each sequence linearly, and outputs a boundary strength profile  $bsp$ . Below we list and briefly describe the cue models used within our system.

**CM1 - gap detection:** Melodic gap cues are assumed to correspond to points of significant local change, e.g. a pitch or duration interval that is perceived as ‘overly large’ in respect to its immediate vicinity. We implemented a model of melodic gap detection based on [4]. The model uses a distance metric to measure local change,<sup>4</sup> and generates a  $bsp$  where peaks correspond to large distances between contiguous melodic events. Large local distances are taken as boundary candidates.

**CM2 - contrast detection:** Melodic contrast cues are assumed to correspond to points of significant change (which require a mid-to-large temporal scale to be perceptually discernible), e.g. a change in melodic pace, or a change of mode. We implemented a contrast detection model based on [9]. The model employs a probabilistic representation of melodic attributes and uses an information-theoretic divergence measure to determine contrast. The model generates a  $bsp$  where peaks correspond to large divergences between attribute distributions representing contiguous sections of the melody. The model identifies boundaries by recursively locating points of maximal divergence.

**CM3 - repetition detection:** Melodic repetition cues are assumed to correspond to salient (exact or approximate) repetitions of melodic material. We implemented a model to locate salient repeated fragments of a melody based on [5]. The model uses an exact-match string pattern search algorithm to extract repeated melodic fragments, and includes a method to score the salience of repetitions based on the length, frequency, and temporal overlap of the extracted fragments. The model generates a  $bsp$  where peaks correspond to the starting points of salient repetitions.

**CM4 - closure detection:** Tonal closure cues are assumed to correspond to points where an ongoing cognitive process of melodic expectation is disrupted. One way in which expectation of continuation might be disrupted is when a melodic event following a given context is unexpected. We implemented an unexpected-event detection model based on [8].<sup>5</sup> The model employs unsupervised probabilistic

<sup>4</sup> The model employs both pitch and temporal information, but in our tests only temporal information is used

<sup>5</sup> Our implementation is however less sophisticated than that of [8], as it requires the user to provide an upper limit for context length (specified

learning and prediction to measure the degree of unexpectedness of each note event in the input melody, given a finite preceding context. The model generates a *bsp* where peaks correspond to significant increases in (information-theoretic) surprise. Candidate boundaries are placed before surprising note events.

## 4.2 Cue Models: Tuning

We tuned the cue models used within our system to achieve maximal precision. This involved a selection of melody representation (choice of melodic attribute sequences to be processed),<sup>6</sup> tuning of parameters exclusive to the cue model, and choice and tuning of a peak selection mechanism.

The choice of attribute sequence selection and parameter tuning per cue model is listed in Table 1. The abbreviations of melodic attributes correspond to: *cp*: chromatic pitch, *ioi*: inter onset interval, *ooi*: onset to offset interval, *cpiv*: chromatic pitch interval, *pcls*: pitch class. To select peaks as boundary candidates, we experimented with several peak selection algorithms, settling for the algorithm proposed in [8].<sup>7</sup> This peak selection algorithm has only one parameter  $k$ . The optimal values of  $k$  for each cue model are given in the rightmost column of Table 1. We also provide details on the choice of parameters exclusive to each cue model, for an elaboration on their interpretation we refer the reader to the original publications.

Cue model	attribute sequence set	parameters
CM1	{ <i>cpiv</i> , <i>ioi</i> , <i>ooi</i> }	$k = 2$
CM2	{ <i>pcls</i> , <i>ioi</i> }	-
CM3	{ <i>cp</i> , <i>ioi</i> }	$F = 3$ $L = 3$ $O = 1$ $k = 3$
CM4	{ <i>cp</i> , <i>pcls</i> , <i>cpiv</i> }	PPM-C, with exclusion STM: order 5 LTM: order 2 LTM: 400 EFSC melodies $k = 2.5$

**Table 1.** Attributes and parameter settings of cue models.

## 4.3 Module 1: Predictors and Feature Selection

To evaluate cue relevance prediction, we first select a subset of 200 boundary annotated melodies from the melodic corpora used in this paper (see §5.1), and then run the cue models to obtain precision performance values for each melody. To allow an estimation of precision we partition its range into a discrete set of categories.<sup>8</sup>

as the Markov order in Table 1).

<sup>6</sup> While some cue models, e.g. [4, 11] have already a preferred choice of melodic attribute representation, the other cue models used within our system allow for many choices, and were thus selected through experimental exploration.

<sup>7</sup> This algorithm proved to work better than the alternatives for all models but CM3, for which its own peak selection heuristic worked best.

<sup>8</sup> In our experiments we used a set dividing a model's precision into two categories (1:poor, 2:good). The exact mapping *precision* :  $[0, 1] \rightarrow \{1, 2\}$  was selected manually for each cue model, to ensure a sufficient number of melodies representing each performance category is available for training.

To determine cue relevance prediction, we experimented with several off-the-shelf classifiers available as part of *Weka*<sup>9</sup>. We selected features using the common *BestFirst* with a 10-fold cross validation. The selected features were those used in all folds.

The melodic features used to predict precision by the classifiers were taken from the *Fantastic*<sup>10</sup> and *jSymbolic*<sup>11</sup> feature extractor libraries, which add up to over 200. After selection, 17 features are kept: 'melody length', 'pitch standard deviation, skewness, kurtosis, and entropy', 'pitch interval standard deviation, skewness, kurtosis, and entropy', 'duration standard deviation, skewness, kurtosis, and entropy', 'tonal clarity', 'm-type mean entropy', 'm-type Simpson's D', 'm-type productivity' (please refer to the *Fantastic* library documentation for definitions).

The classifiers we experimented with are Sequential Minimal Optimization (*SMO*, with the radial basis function kernel), K-Nearest Neighbours (*K\**) and Bayesian Network (*BNet*). To evaluate each classifier we use 10-fold cross validation. The classifier with the best performance-to-efficiency ratio is *SMO* for models CM2-CM4, with an average accuracy of 72.21%, and the simple *K\** for CM1 with an average accuracy of 66.37%.

## 4.4 Module 2: Computation of Priors and Choice of $\alpha$

To compute the optimal sequence of segments  $S_{opt}$  we minimise the cost function in Eq. 1 using a formulation of the Viterbi algorithm based on [10]. The minimisation of Eq. 1 is subject to constraints on segment contour and segment length, and to a choice for parameter  $\alpha$ . We tuned  $\alpha$  manually (a value of 0.6 worked best in our experiments). To model constraints in segment contour and segment length we use probability priors. Below we provide details on their computation.

A prior  $P(contour(s_k))$  is computed employing a Gaussian Mixture Model (GMM). Phrase contours are computed using the polynomial contour feature of the *Fantastic* library. A contour model with four nodes was selected. The GMM (one Gaussian per node) is fitted to contour distributions obtained from a subset of 1000 phrases selected randomly from the boundary annotated corpora used in this paper (see §5.1).

A prior of segment length  $P(l_k)$  is computed employing a Gaussian fitted to a distribution of lengths obtained from the same 1000 phrase subset used to derive contours.

## 5. EVALUATION

In this section we describe our test database and evaluation metrics, and subsequently describe experiments and results obtained by our system. A prototype of our system was implemented using a combination of Matlab, R, and Python. Source files and test data are available upon request.

<sup>9</sup> <http://www.cs.waikato.ac.nz/ml/weka/>

<sup>10</sup> <http://www.doc.gold.ac.uk/~mas03dm/>

<sup>11</sup> <http://jmir.sourceforge.net/jSymbolic.html>

## 5.1 Melodic Corpora

To evaluate our system we employed a set of 100 instrumental folk songs randomly sampled from the Liederbank collection<sup>12</sup> (LC) and 100 vocal folk songs randomly sampled from the German subset of the Essen Folk Song Collection<sup>13</sup> (EFSC). We chose to use the EFSC due to its benchmark status in the field of melodic segmentation. Additionally, we chose to use the LC to compare the performance of segmentation models in vocal and non-vocal melodies.<sup>14</sup>

The EFSC consists of ~6000 songs, mostly of German origin. The EFSC data was compiled and encoded from notated sources. The songs are available in EsAC and \*\*kern formats. The origin of phrase boundary markings in the EFSC has not been explicitly documented (yet it is commonly assumed markings coincide with breath marks or phrase boundaries in the lyrics of the songs).

The instrumental (mainly fiddle) subset of the LC consists of ~2500 songs. The songs were compiled and encoded from notated sources. The songs are available in MIDI and \*\*kern formats. Segment boundary markings for this subset comprise two levels: ‘hard’ and ‘soft’. Hard (section) boundary markings correspond with structural marks found in the notated sources. Soft (phrase) boundary markings correspond to the musical intuition of two experts annotators.<sup>15</sup>

## 5.2 Evaluation Measures

To evaluate segmentation results, we encode both predicted and human-annotated phrase boundary markings as binary vectors. Using these vectors we compute the number of true positives  $tp$  (hits), false positives  $fp$  (insertions), and false negatives  $fn$  (misses).<sup>16</sup> We then quantify the similarity between predictions and human annotations using the well known  $F1 = \frac{2 \cdot p \cdot r}{p+r}$ , where precision  $p = \frac{tp}{tp+fp}$  and recall  $r = \frac{tp}{tp+fn}$ . While the  $F1$  has its downsides (it assumes independence between boundaries),<sup>17</sup> it has been used extensively in the field and thus allows us to establish a comparison to previous research.

## 5.3 Experiments & Results

In our experiments we compare our system to the melody segmentation models that have consistently scored best in comparative studies: GROUPER [11] and LBDM [4]. The first is a multi-strategy model, and the second a single stra-

<sup>12</sup> <http://www.liederbank.nl/>

<sup>13</sup> <http://www.esac-data.org>

<sup>14</sup> Vocal music has dominated previous evaluations of melodic segmentation (especially large-scale evaluations), which might give an incomplete picture of the overall performance and generalisation capacity of segmentation models

<sup>15</sup> Instructions to annotate boundaries were related to performance practice (e.g. “where would you change movement of bow”).

<sup>16</sup> The first and last boundaries are treated as trivial cases which correspond, respectively, to the beginning and ending notes of a melodic phrase. These trivial cases are excluded from the evaluation. Also, we allow a tolerance of  $\pm 1$  note event for the computation of  $tp$ .

<sup>17</sup> By assuming independence between boundaries aspects such as segment length and position are discarded from the evaluation

tegy (gap detection) model.<sup>18</sup> We also compare our system to its performance when only one module is active. Additionally we compare to two naïve baselines: *always*, which predicts a segment boundary at every melodic event position, and *never* which does not make predictions.

Table 2 shows the performance results of all models over the instrumental and vocal melodic sets. We refer to our model as COMPLETE, and to the configurations when either module 1 or 2 are active as MOD1ON and MOD2ON, respectively.

We tested the statistical significance of the paired F1 differences between the three configurations of our system, the two state-of-the-art models, and the baselines. For the statistical testing we used a non-parametric Friedman test ( $\alpha = 0.05$ ). Furthermore, to determine which pairs of measurements significantly differ, we conducted a post-hoc Tukey HSD test. All pair-wise differences among configurations were found to be statistically significant, except those between MOD1ON and MOD2ON in the vocal set and between LBDM and MOD2ON in the instrumental set. In Table 2 the highest performances are highlighted in bold.

Database	Instrumental			Vocal		
	$\bar{R}$	$\bar{P}$	$\bar{F1}$	$\bar{R}$	$\bar{P}$	$\bar{F1}$
COMPLETE	0.56	0.62	<b>0.54</b>	0.49	0.67	0.56
GROUPER	0.81	0.31	0.44	0.60	0.62	<b>0.61</b>
LBDM	0.57	0.49	0.45	0.56	0.55	0.52
MOD2ON	0.51	0.49	0.44	0.48	0.45	0.47
MOD1ON	0.52	0.47	0.42	0.63	0.42	0.46
<i>always</i>	0.06	1.00	0.09	0.08	1.00	0.12
<i>never</i>	0.00	0.00	0.00	0.00	0.00	0.00

**Table 2.** Performance of models and baselines sorted in order of mean recall  $\bar{R}$ , precision  $\bar{P}$ , and  $\bar{F1}$  for instrumental and vocal melodies. The results presented in this table were obtained comparing predictions to the ‘soft’ boundary markings of the LC.

### 5.3.1 Summary of Main Results

In general, F1 performances obtained by the segmentation models in the vocal set are consistently higher than in the instrumental set. This might be simply an indication that in the instrumental set melodies constitute a more challenging evaluation scenario. However, the F1 differences might also be an indication that relevant perceptual boundary cues are not covered by the evaluated models.

In the instrumental set, COMPLETE outperforms both LBDM and GROUPER by a relatively large margin ( $\geq 10\%$ ). In the vocal set, on the other hand, GROUPER obtains the best performance. Below we discuss the three configurations of our system (COMPLETE, MOD1ON, MOD2ON).

### 5.3.2 Discussion

In both melodic sets MOD1ON shows considerably higher recall than precision. These recall/precision differences agree with intuition, since the output of MOD1ON consists of the combination of all boundaries predicted by

<sup>18</sup> For our tests we ran GROUPER and LBDM with their default settings.

the cue models, and can hence be expected to contain a relatively large number of false positives. On the other hand, MOD2ON shows smaller differences between precision and recall values, and shows higher F1 performances than MOD1ON in both melodic sets (although the difference between performances is significant only for the instrumental set). This last result highlights the robustness of the optimisation procedure driving MOD2ON.<sup>19</sup>

The large F1 differences between MOD1ON and MOD2ON in respect to COMPLETE suggest that segmentation at the phrase level is a perceptual process which, despite happening in ‘real time’ (i.e. as music unfolds itself, represented more closely by module 1), might still require repeated exposure and retrospective listening (represented more closely by module 2).

Manual examination COMPLETE reveals that, when segmenting the vocal melody set, the prediction stage of module 1 tends to overestimate the importance of cue models (i.e. it often misclassifies models as relevant when they are not). However, when altering the settings of COMPLETE so that the prediction stage of model 1 is more conservative (i.e. so that it predicts fewer boundaries), there is no significant improvement in performance. Closer analysis of these results points to a trade-off in performance, i.e. while a conservative setting increases precision (predictions have fewer ‘false positives’), it also decreases recall (predictions have fewer ‘correct positives’). This suggests that the prediction stage of module 1 might require estimation of cue relevance at a local level, i.e. on subsections of the melody rather than on the whole melody.

## 6. CONCLUSION

In this paper we introduce a multi-strategy system for the segmentation of symbolically encoded melodies. Our system combines the contribution of single strategy models of melody segmentation. The system works in two stages. First, it estimates how relevant the boundaries computed by each selected single strategy model are to the melody being analysed, and then combines boundary predictions using heuristics. Second, it assesses the segmentation produced by combinations of the selected boundary candidates in respect to corpus-learned priors on segment contour and segment length.

We tested our system on 100 vocal and 100 instrumental folk song melodies. The performance of our system showed a considerable (10% *F1*) improvement upon the state-of-the-art in melody segmentation for instrumental folk music, and showed to perform second best in the case of vocal folk songs.

In future work we will test if the relevance of cue models can be accurately estimated for sections of the melody (and not the whole melody as it is done in this paper). This

<sup>19</sup> If we consider that (with MOD1ON bypassed) the number of candidate boundaries taken as input to MOD2ON often exceeds ‘correct’ (human annotated) boundaries by a factor 2 or 3, then the number of possible segmentations of the melody shows an exponential increase, leading to local minima issues, and so it would be reasonable to expect a performance equal or worse than that of MOD1ON.

‘local’ account of relevance might play a major role in improving the system’s precision. Also, we will incorporate a more advanced model of prior segment knowledge of segment structure in our system. We hypothesise that a model of the characteristics of [2] could constitute a good alternative to model not only segment length and contour, but also to incorporate knowledge of ‘template’ phrase structure forms. Lastly, we will continue testing our model’s generalisation capacity by evaluating on larger sample sizes and genres other than folk (for the latter the authors are currently in the process of annotating a corpus of Jazz melodies).

**Acknowledgments:** We thank Frans Wiering, Remco Veltkamp, and the anonymous reviewers for the useful comments on earlier drafts of this document. Marcelo Rodríguez-López and Anja Volk (NWO-VIDI grant 276-35-001) and Dimitrios Bountouridis (NWO-CATCH project 640.005.004) are supported by the Netherlands Organization for Scientific Research.

## 7. REFERENCES

- [1] S. Ahlbäck. Melodic similarity as a determinant of melody structure. *Musicae Scientiae*, 11(1):235–280, 2007.
- [2] R. Bod. Probabilistic grammars for music. In *Belgian-Dutch Conference on Artificial Intelligence (BNAIC)*, 2001.
- [3] M. Bruderer, M. McKinney, and A. Kohlrausch. The perception of structural boundaries in melody lines of western popular music. *Musicae Scientiae*, 13(2):273–313, 2009.
- [4] E. Cambouropoulos. The local boundary detection model (LBDM) and its application in the study of expressive timing. In *Proceedings of the International Computer Music Conference (ICMC01)*, pages 232–235, 2001.
- [5] E. Cambouropoulos. Musical parallelism and melodic segmentation. *Music Perception*, 23(3):249–268, 2006.
- [6] E. Clarke and C. Krumhansl. Perceiving musical time. *Music Perception*, pages 213–251, 1990.
- [7] M. Hamanaka, K. Hirata, and S. Tojo. ATTA: Automatic time-span tree analyzer based on extended GTTM. In *ISMIR Proceedings*, pages 358–365, 2005.
- [8] M. Pearce, D. Müllensiefen, and G. Wiggins. The role of expectation and probabilistic learning in auditory boundary perception: A model comparison. *Perception*, 39(10):1365, 2010.
- [9] M. Rodríguez-López and A. Volk. Melodic segmentation using the jensen-shannon divergence. In *International Conference on Machine Learning and Applications (ICMLA12)*, volume 2, pages 351–356, 2012.
- [10] G. Sargent, F. Bimbot, E. Vincent, et al. A regularity-constrained Viterbi algorithm and its application to the structural segmentation of songs. In *ISMIR Proceedings*, 2011.
- [11] D. Temperley. *The cognition of basic musical structures*. MIT Press, 2004.

# A DATA SET FOR COMPUTATIONAL STUDIES OF SCHENKERIAN ANALYSIS

Phillip B. Kirlin

Department of Mathematics and Computer Science, Rhodes College  
kirlinp@rhodes.edu

## ABSTRACT

Schenkerian analysis, a kind of hierarchical music analysis, is widely used by music theorists. Though it is part of the standard repertoire of analytical techniques, computational studies of Schenkerian analysis have been hindered by the lack of available data sets containing both musical compositions and ground-truth analyses of those compositions. Without such data sets, it is difficult to empirically study the patterns that arise in analyses or rigorously evaluate the performance of intelligent systems for this kind of analysis. To combat this, we introduce the first publicly available large-scale data set of computer-processable Schenkerian analyses. We discuss the choice of musical selections in the data set, the encoding of the music and the corresponding ground-truth analyses, and the possible uses of these data. As an example of the utility of the data set, we present an algorithm that transforms the Schenkerian analyses into hierarchically-organized data structures that are easily manipulated in software.

## 1. CORPUS-DRIVEN RESEARCH

Corpus-driven research is now commonplace in the music informatics community. With the wealth of raw musical information now available in digital form, in many cases, it is straightforward to construct and use data sets containing numerous musical compositions. However, the problem of collecting ground-truth metadata about the content of the music still exists, especially where high-level features are concerned. This is a problem that effects researchers working with music in audio or symbolic formats.

Ground-truth data sets that include features specifically relating to music theory or music analysis are particularly labor-intensive to construct. Information about the high-level harmonic or melodic structure of compositions is often only found scattered throughout textbooks or individual research publications, and so there are few publicly-available corpora containing such information in a computer-processable format. Some data sets are created only for specific research projects and then discarded,

are not in an easy-to-use format, or are simply never made widely available.

The lack of varied ground-truth musical metadata relating to theory and analysis — especially data sets specifically designed to align with symbolic music data — hinders corpus-driven research studies because time must be spent collecting data. Sometimes the researchers must perform the music analysis themselves, possibly inadvertently introducing biases into the data. Without widely available comprehensive data sets, it is extremely difficult to conduct large-scale experiments on the structure of musical compositions in symbolic form, or quantitatively evaluate the performance of computational systems that emulate a music analysis process.

There is a particular dearth of empirical data available in the realm of *Schenkerian analysis*, a widely used analytical system that illustrates a hierarchical structure among the notes of a composition. Though Schenkerian analysis is one of the most comprehensive methods for music analysis that we have available today [1], there are no large-scale digital repositories of analyses available to researchers. In addition to the reasons stated above for the lack of corpora, Schenkerian analysis presents a number of unique challenges to creating a useful data set. First, a Schenkerian analysis for a composition is illustrated using the musical score of the composition itself, and commonly requires multiple staves to show the hierarchical structure uncovered. This requires substantial space on the printed page and thus is a deterrent to retaining large sets of analyses. Second, there is no established computer-interpretable format for Schenkerian analysis storage, and third, even if there were a format, it would take a great deal of effort to encode a number of analyses into processable computer files.

The lack of data has kept the number of computational studies of Schenkerian analysis requiring ground-truth data to a bare minimum; some examples include studies using corpora with six [7] or eight [6] pieces. Though these studies are useful, the results would likely carry more weight if the data sets used were larger.

With all of these ideas in mind, in this paper we introduce the first large-scale data set of musical compositions along with corresponding ground-truth Schenkerian analyses, called SCHENKER41<sup>1</sup>. The 41 musical selections included constitute the largest-known corpus of Schenkerian analyses in a machine-readable format. The musical selec-

<sup>1</sup> Available at [www.cs.rhodes.edu/~kirlinp/schenker41](http://www.cs.rhodes.edu/~kirlinp/schenker41).



© Phillip B. Kirlin.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Phillip B. Kirlin. “A Data Set For Computational Studies of Schenkerian Analysis”, 15th International Society for Music Information Retrieval Conference, 2014.

tions are standardized in mode, length, and instrumentation, and the analyses are stored in a novel text-based representation designed to be easily processed by a computer. We created these data with the hope that they would be useful to researchers (a) studying the Schenkerian analysis process itself from a quantitative standpoint (for instance, detecting patterns in the way analysis is done), (b) needing a data set of analyses for use with supervised machine learning techniques, and (c) performing any sort of quantitative evaluation requiring ground-truth hierarchical music analyses.

## 2. THE DATA SET

### 2.1 Creation and Content

In order to create a data set of musical compositions and corresponding ground-truth Schenkerian analyses that would be useful to researchers with a wide variety of goals, we restricted ourselves to music from the common practice period of European art music, and selected 41 excerpts from works by J. S. Bach, G. F. Handel, Joseph Haydn, M. Clementi, W. A. Mozart, L. van Beethoven, F. Schubert, and F. Chopin. All of the compositions were either for a solo keyboard instrument (or arranged for such an instrument) or for voice with keyboard accompaniment. All were in major keys and did not modulate.

The musical excerpts were also selected for the ease of locating a Schenkerian analysis for each excerpt done by an outside expert. Analyses for the 41 excerpts chosen came from four places: Forte and Gilbert's textbook *Introduction to Schenkerian Analysis* [4] and the corresponding instructor's manual [3], Cadwallader and Gagné's textbook *Analysis of Tonal Music* [2], Pankhurst's handbook *SchenkerGUIDE* [9], and a professor of music theory who teaches a Schenkerian analysis class. These four sources are denoted by the labels F&G, C&G, SG, and Expert in Table 1, which lists the excerpts in the corpus.

From a Schenkerian standpoint, we also chose excerpts such that the analyses of the excerpts would all share some commonalities. All the analyses contained a single linear progression as the fundamental background structure: either an instance of the *Ursatz* or a rising linear progression. Some excerpts contained an *Ursatz* with an *interruption*: a Schenkerian construct that occurs when a musical phrase ends with an incomplete instance of the *Ursatz*, then repeats with a complete version.

We put these restrictions on the musical content in place because we expected that if SCHENKER41 were to be used for supervised machine learning, such algorithms would be able to better model a corpus with less variability among the pieces.

Overall, SCHENKER41 contains 253 measures of music and 907 notes. The lengths of individual excerpts ranged from 6 to 53 notes.

### 2.2 Encoding

With our selected musical excerpts and our corresponding analyses in hand, we needed to translate the musical in-

formation into machine-readable form. Musical data has many established encoding schemes; we used MusicXML, a format that preserves more information from the original score than say, MIDI.

Translating the Schenkerian analyses proved harder because there is no current standard for storing such analyses in a format that a computer could easily process and manipulate. Therefore, we devised a text-based encoding scheme to represent the various notations found in a Schenkerian analysis. Each analysis is stored in a single text file that is linked to a specific MusicXML file containing the musical excerpt being analyzed.

Schenkerian analyses are primarily based on the concept of a *prolongation*, a situation where an analyst determines that a group of notes is elaborating a group of more structurally fundamental notes. Consider the descending melodic pattern D–C–B–F $\sharp$ –G all occurring over G major harmony, as is shown in Figure 1. We could imagine that an analyst would determine that this passage outlines a descending G-major triad (D–B–G), with the second note C (a passing tone) serving to melodically connect the preceding D to the following B. We would say the note C *prolongs* the motion from D to B. Similarly, the F $\sharp$  prolongs the motion from B to G. Schenkerian analysis hypothesizes that any tonal composition is structured as a nested collection of prolongations; identifying them is an important component of the analysis procedure.

Every prolongation identified in an analysis is encoded in the analysis text file using the syntax  $X(Y)Z$ , where  $X$  and  $Z$  are individual notes in the score and  $Y$  is a non-empty list of notes. Such a statement means that the notes in  $Y$  prolong the motion from note  $X$  to note  $Z$ . Additionally, we permit incomplete prolongations in the text file representation: one of  $X$  or  $Z$  may be omitted. The notes of  $X$ ,  $Y$  and  $Z$  are transcribed in the text file as is shown in Figure 1, with a measure number, followed by a pitch and octave, followed by an integer to distinguish between repeated notes in the same measure. Figure 2 shows how the prolongations of Figure 1 would be encoded. Note that the prolongation involving the F $\sharp$  is encoded with no  $X$  component; this tells us that there is no strong melodic connection from the B to the F $\sharp$ , only from the F $\sharp$  to the G.

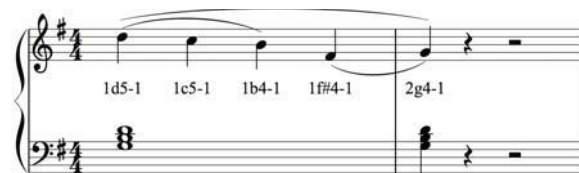


Figure 1. A melodic sequence with note names.

```
1d5-1 (1c5-1) 1b4-1
(1f#4-1) 2g4-1
1d5-1 (1b4-1) 2g4-1
```

Figure 2. An encoding of the prolongations present.

This text format easily supports encoding prolongations at differing hierarchical levels in the music. We can see

how Figure 2 encodes both the “surface-level” prolongations D–C–B and F $\sharp$ –G, but also the deeper prolongation D–B–G which outlines the fifth relationship in the G-major chord.

Aside from prolongations, the encoding system supports describing repetitions of notes that may be omitted in the analysis on the printed page; any linear progressions, including instances of the *Ursatz*; and the harmonic context present at any point in the analysis.

### 2.3 Compromises

An additional challenge not previously mentioned in creating the SCHENKER41 analyses is choosing an appropriate level of detail of the material to encode. Because the main objects in analyses are prolongations, it is natural to attempt to group them into categories like “neighbor tone” and “passing tone.” However, not all prolongations identified in analyses are easily categorized, and so category labels are often omitted in analyses not found in an educational context. This raises the question of whether or not to attempt to encode the category of prolongations in this corpus. To avoid the risk of incorrectly interpreting analyses, we have chosen to encode only what is *directly observable on the printed page* — the hierarchical relationship between groups of notes — and not categorize the prolongations found in the analyses. We recognize that this is a compromise between staying true to the data and encoding all potentially useful information.

## 3. USAGE OF THE DATA

The SCHENKER41 data set enables the undertaking of a wide variety of tasks and studies. In addition to the already-discussed endeavors of using the corpus for supervised machine learning or for quantitative evaluation, we theorize that with these data it could be possible to address the following questions:

- Do analysts identify certain types of prolongations more often than others under certain circumstances? These circumstances may involve the composer, musical genre, or even the analysis source.
- Does Schenkerian analysis align well with other forms of music analysis, such as Narmour’s implication-realization model of melodic expectation [8]?
- How well do Schenkerian analyses align with expressive performances of the music [10]? Do features of a performance such as phrasing, volume, or other quantifiable measures of musicality correspond to various Schenkerian annotations in an analysis?

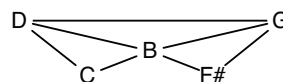
Besides answering questions about Schenkerian analysis itself, we hope that the availability of SCHENKER41 will spur others to study the utility of Schenkerian analysis in other areas of music informatics. For instance, we suspect hierarchical analyses could prove useful in constructing musical similarity metrics, because Schenkerian

analyses may highlight a common melodic pattern residing under the surface in two different musical excerpts.

Though the SCHENKER41 analyses can be directly processed by software, the nature of the flat text file format in which the data are encoded makes it difficult to see hierarchical relationships between notes not directly related by a single prolongation. Therefore, in this section we describe an algorithm to translate the analysis text files into hierarchical graph structures known as MOPs. It is possible to use the SCHENKER41 data in MOP form to automatically learn characteristics of Schenkerian analysis [5].

### 3.1 Maximal Outerplanar Graphs

Maximal outerplanar graphs, or *MOPs*, were first proposed by Yust [11] as elegant structures for representing a set of musical prolongations in a Schenkerian-style hierarchy. A MOP represents a hierarchy of melodic intervals located in a monophonic sequence of notes, though Yust proposed some extensions for polyphony. For example, the prolongations mentioned in Figures 1 and 2 are represented by the MOP shown in Figure 3.



**Figure 3.** A MOP representation of the music in Figure 1.

Formally, a MOP is a complete triangulation of a polygon, where the vertices of the polygon are notes and the outer perimeter of the polygon consists of the melodic intervals between consecutive notes of the original music, except for the edge connecting the first note to the last, which is called the *root edge*. Each triangle in the polygon specifies a prolongation. For instance, in Figure 3, the presence of triangle D–C–B means that the melodic motion from D to B is prolonged by the C. By expressing the hierarchy in this fashion, each edge  $(x, y)$  carries the interpretation that notes  $x$  and  $y$  are “consecutive” at some level of abstraction of the music. Edges closer to the root edge express more abstract relationships than edges farther away.

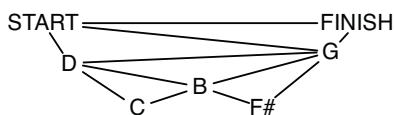
*Outerplanarity* is a property of a graph that can be drawn such that all the vertices are on the perimeter of the graph. Such a condition is necessary for us to enforce the strict hierarchy among the prolongations. A *maximal* outerplanar graph cannot have any additional edges added to it without destroying the outerplanarity; such graphs are necessarily polygon triangulations, and under this interpretation, all prolongations must occur over triples of notes.

There are three representational issues with MOPs we must address before discussing the algorithm to convert an analysis text file into MOPs. First, Schenkerian analyses as commonly encountered often include prolongations involving more than three notes. The analysis sources used in SCHENKER41 are no exception. For this reason, we relax the “maximal” qualifier for MOPs and permit prolongations involving any number of notes in our MOP representation. A prolongation involving more than three notes

will be translated into a polygon with more than three edges in the MOP representation.

Second, MOPs do not have a direct way to represent a prolongation with only a single “parent” note. Because MOPs model prolongations as a way of moving *from* one musical event *to* another event, every prolongation must have two parent notes. Music sometimes presents situations, however, that an analyst would model with a one-parent prolongation, such as an incomplete neighbor tone (we encountered this situation in Figure 2). Yust interprets such prolongations as having a “missing” origin or goal note that has been elided with a nearby structural note, which substitutes in the MOP for the missing note.

The third representational issue stems from trying to represent prolongations involving the first or last note in the music. Prolongations necessarily take place over time, and in a MOP, we interpret the temporally middle notes as prolonging the motion from the earliest note (the left parent) to the latest (the right parent). Following this temporal logic, we can infer that the root edge of a MOP must therefore necessarily be between the first note of the music and the last, implying these are the two most structurally important notes of a composition. As this is not always true in compositions, we add two pseudo-events to every MOP: an initiation event that is located temporally before the first note of the music, and a termination event, which is temporally positioned after the last note. The root edge of a MOP is fixed to always connect the initiation event and the termination event. These extra events allow for any melodic interval — and therefore any pair of notes in the music — to be represented as the most structural event in the composition. For instance, in Figure 4, which shows the D–C–B–F#–G pattern with initiation and termination events (labeled START and FINISH), the analyst has indicated that the G is the most structurally significant note in the passage, as this note prolongs the motion along the root edge.



**Figure 4.** A MOP containing initiation and termination events.

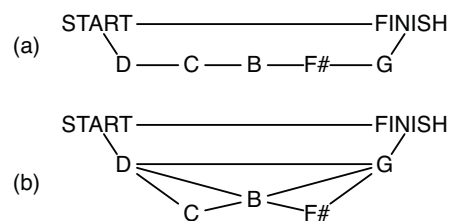
### 3.2 Converting the Corpus to MOPs

We now present an algorithm to convert a text file analysis like those in SCHENKER41 to a collection of MOPs. Because a single MOP only represents a monophonic sequence of notes, we may need multiple MOPs to store all of the prolongations in a single text file analysis. Most of the analyses in SCHENKER41 contain at least two MOPs, one representing the structure of the main melody, and one representing structure of the bass line.

The algorithm operates in three phases. In the first phase, we make a pass through the analysis text file to identify which notes will belong to which MOPs. We do this by creating a temporary graph structure consisting of all the

notes present in the analysis and initially no edges. For each prolongation in the analysis file  $X (Y) Z$ , we add the edges  $(X, Z)$  and  $(i, Z)$  for each note  $i$  in the set of notes  $Y$ . After processing every prolongation, every connected component in the graph will correspond to a single MOP.

Phase two adds edges to the MOP for all two-parent prolongations. For each MOP graph identified in phase one, we first remove all the edges, then create a “skeleton” MOP structure consisting of edges connecting only consecutive notes in the music, plus the additional edges involving the START and FINISH vertices. Figure 5(a) illustrates this skeletal structure for the prolongations described in Figure 2. We then create edges in the MOP corresponding to all prolongations in the analysis text file that have two parent notes. Adding appropriate edges is straightforward: for a prolongation  $X (Y) Z$ , we add an edge from note  $X$  to the first note of the set of notes  $Y$ , an edge from the last note of  $Y$  to note  $Z$ , and an edge from  $X$  to  $Z$ . If the consecutive notes of  $Y$  are not already connected to each other by edges, we also add such edges. At the end of phase two, we would have a structure like in Figure 5(b).



**Figure 5.** The (a) beginning and (b) end of phase two of creating a MOP.

Phase three involves adding edges in the MOP for one-parent prolongations, i.e., prolongations in the analysis text file of the form  $X (Y)$  or  $(Y) Z$ . We begin by adding edges between consecutive notes of  $Y$  as in phase two. The next step is identifying any additional edges necessary to enforce that the notes of  $Y$  should be lower in the hierarchy than  $X$  or  $Z$ , whichever parent note is present. Fortunately, it is guaranteed that every one-parent prolongation will fall into one of the six categories described below, each of which we handle separately. We briefly describe the six categories here, and their processing steps are fully described in the pseudocode of Algorithm 1. The code refers to the “smallest interior polygon” for a one-parent prolongation  $p$ , which is the smallest polygon in the MOP containing all the notes of  $p$  (the parent note and all of the child notes). This interior polygon will always exist in a MOP because MOPs express a strict hierarchy among the notes, and therefore all the notes of a prolongation will be found within a single polygon.

Category 1 corresponds to a one-parent prolongation missing a right parent, where the MOP already contains an edge connecting the left parent  $X$  to the first note of  $Y$ , and the edge in question already implies a hierarchical relationship between  $X$  and  $Y$ . In this situation, there are no extra edges to add because the necessary hierarchical relationship already exists. Category 2 corresponds to the same situation as Category 1, but reversed for a missing



**Algorithm 1**


---

```

1: procedure PROCESS-ONE-PARENT-PROLONGATIONS
2:   Let  $S$  be the set of one-parent prolongations.
3:   while  $S \neq \emptyset$  do
4:      $p \leftarrow$  shortest length prolongation in  $S$ 
5:      $I \leftarrow$  identify smallest interior polygon containing all notes of  $p$ 
6:     Assume vertices of  $I$  are numbered  $0 \dots m - 1$ 
7:     if  $\text{leftParent}(p) = I[0]$  and  $\text{firstChildNote}(p) = I[1]$  then ▷ Category 1
8:        $S \leftarrow S - \{p\}$  ▷ No additional edges needed;  $p$ 's children are already lower in the hierarchy
9:     else if  $\text{rightParent}(p) = I[m - 1]$  and  $\text{lastChildNote}(p) = I[m - 2]$  then ▷ Category 2
10:       $S \leftarrow S - \{p\}$  ▷ No additional edges needed;  $p$ 's children are already lower in the hierarchy
11:     else if  $\text{leftParent}(p) = I[0]$  then ▷ Category 3
12:       Add edge ( $\text{leftParent}(p)$ ,  $\text{firstChildNote}(p)$ ) to MOP;  $S \leftarrow S - \{p\}$ 
13:     else if  $\text{rightParent}(p) = I[m - 1]$  then ▷ Category 4
14:       Add edge ( $\text{rightParent}(p)$ ,  $\text{lastChildNote}(p)$ ) to MOP;  $S \leftarrow S - \{p\}$ 
15:     else if  $\text{rightParent}(p)$  is missing then ▷ Category 5
16:        $\text{newRight} \leftarrow$  earliest  $I[x]$  such that  $I[x]$  is later than all of  $p$ 's children
17:       if choice of  $\text{newRight}$  increases length of prolongation  $p$  then
18:         Update  $p$ 's length in  $S$ ; defer processing
19:       else
20:         Add edge ( $\text{leftParent}(p)$ ,  $\text{newRight}$ ) to MOP;  $S \leftarrow S - \{p\}$ 
21:     else if  $\text{leftParent}(p)$  is missing then ▷ Category 6
22:        $\text{newLeft} \leftarrow$  latest  $I[x]$  such that  $I[x]$  is earlier than all of  $p$ 's children
23:       if choice of  $\text{newLeft}$  increases length of prolongation then
24:         Update  $p$ 's length in  $S$ ; defer processing
25:       else
26:         Add edge ( $\text{newLeft}$ ,  $\text{rightParent}(p)$ ) to MOP;  $S \leftarrow S - \{p\}$ 

```

---

left parent note.

Category 3 corresponds to a one-parent prolongation missing a right parent, where the the MOP does *not* contain an edge connecting the left parent  $X$  to the first note of  $Y$ , but other nearby edges already imply a hierarchical relationship between  $X$  and  $Y$ . Here, we only need to add an edge from  $X$  to the first child note of  $Y$ . Category 4 corresponds to the same situation as Category 3, but reversed for a missing left parent.

Category 5 corresponds to a one-parent prolongation missing a right parent, where the the MOP does *not* contain an edge connecting the left parent  $X$  to the first note of  $Y$ , and *no other edges* in the MOP already imply a hierarchical relationship between  $X$  and  $Y$ . In this situation we must explicitly find a suitable right parent note, which we choose to be the temporally earliest note on the interior polygon that is later than all the notes of  $Y$ . Category 6 corresponds to the same situation as Category 5, but reversed for a missing left parent.

#### 4. CONCLUSIONS

In this paper, we presented SCHENKER41, the first large-scale data set of musical compositions and corresponding Schenkerian analyses in a computer-processable format. We anticipate that with the rise of corpus-driven research in music informatics, this data set will be of value to researchers investigating various characteristics of Schenkerian analysis, using machine learning techniques to study the analytical procedure, or harnessing the analyses for use in other music informatics tasks. We also presented an algorithm for translating the analyses into MOPs, which serve as useful data structures for representing the hierarchical organization of the analyses.

#### 5. REFERENCES

- [1] Matthew Brown. *Explaining Tonality*. University of Rochester Press, 2005.
- [2] Allen Cadwallader and David Gagné. *Analysis of Tonal Music: A Schenkerian Approach*. Oxford University Press, Oxford, 1998.
- [3] Allen Forte and Steven E. Gilbert. *Instructor's Manual for Introduction to Schenkerian Analysis*. W. W. Norton and Company, New York, 1982.
- [4] Allen Forte and Steven E. Gilbert. *Introduction to Schenkerian Analysis*. W. W. Norton and Company, New York, 1982.
- [5] Phillip B. Kirlin. *A Probabilistic Model of Hierarchical Music Analysis*. PhD thesis, University of Massachusetts Amherst, 2014.
- [6] Phillip B. Kirlin and David D. Jensen. Probabilistic modeling of hierarchical music analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 393–398, 2011.
- [7] Alan Marsden. Schenkerian analysis by computer: A proof of concept. *Journal of New Music Research*, 39(3):269–289, 2010.
- [8] Eugene Narmour. *Beyond Schenkerism: The Need for Alternatives in Music Analysis*. University of Chicago Press, 1977.
- [9] Tom Pankhurst. *SchenkerGUIDE: A Brief Handbook and Website for Schenkerian Analysis*. Routledge, New York, 2008.
- [10] Christopher Raphael. Symbolic and structural representation of melodic expression. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pages 555–560, 2009.
- [11] Jason Yust. *Formal Models of Prolongation*. PhD thesis, University of Washington, 2006.

Composer	Excerpt name	Analysis source
Bach	Minuet in G major, BWV Anh. 114, mm. 1–16	Expert
Bach	Chorale 233, Werde munter, mein Gemute, mm. 1–4	Expert
Bach	Chorale 317 (BWV 156), Herr, wie du willst, so schicks mit mir, mm. 1–5	F&G manual
Beethoven	Seven Variations on a Theme by P. Winter, WoO 75, Variation 1, mm.1–8	C&G
Beethoven	Seven Variations on a Theme by P. Winter, WoO 75, Theme, mm. 1–8	C&G
Beethoven	Ninth Symphony, Ode to Joy theme from finale (8 measures)	SG
Beethoven	Piano Sonata in F minor, Op. 2, No. 1, Trio, mm. 1–4	SG
Beethoven	Seven Variations on God Save the King, Theme, mm. 1–6	SG
Chopin	Mazurka, Op. 17, No. 1, mm. 1–4	SG
Chopin	Grande Valse Brilliante, Op. 18, mm. 5–12	SG
Clementi	Sonatina for Piano, Op. 38, No. 1, mm. 1–2	SG
Handel	Trio Sonata in B-flat major, Gavotte, mm. 1–4	Expert
Haydn	Divertimento in B-flat major, Hob. 11/46, II, mm. 1–8	F&G
Haydn	Piano Sonata in C major, Hob. XVI/35, I, mm. 1–8	F&G
Haydn	Twelve Minuets, Hob. IX/11, Minuet No. 3, mm. 1–8	SG
Haydn	Piano Sonata in G major, Hob. XVI/39, I, mm. 1–2	SG
Haydn	Hob. XVII/3, Variation I, mm. 19–20	SG
Haydn	Hob. I/85, Trio, mm. 39–42	SG
Haydn	Hob. I/85, Menuetto, mm. 1–8	SG
Mozart	Piano Sonata 11 in A major, K. 331, I, mm. 1–8	F&G
Mozart	Piano Sonata 13 in B-flat major, K. 333, III, mm. 1–8	F&G manual
Mozart	Piano Sonata 16 in C major, K. 545, III, mm. 1–8	F&G manual
Mozart	Six Variations on an Allegretto, K. Anh. 137, mm. 1–8	F&G manual
Mozart	Piano Sonata 7 in C major, K. 309, I, mm. 1–8	C&G
Mozart	Piano Sonata 13 in B-flat major, K. 333, I, mm. 1–4	F&G
Mozart	7 Variations in D major on “Willem van Nassau,” K. 25, mm. 1–6	SG
Mozart	Twelve Variations on “Ah vous dirai-je, Maman,” K. 265, Var. 1, mm. 23–32	SG, C&G
Mozart	12 Variations in E-flat major on “La belle Française,” K. 353, Theme, mm. 1–3	SG
Mozart	Minuet in F for Keyboard, K. 5, mm. 1–4	SG
Mozart	8 Minuets, K. 315, No. 1, Trio, mm. 1–8	SG
Mozart	12 Minuets, K. 103, No. 4, Trio, mm. 15–16	SG
Mozart	12 Minuets, K. 103, No. 3, Trio mm. 7–8,	SG
Mozart	Untitled from the London Sketchbook, K. 15a, No. 1, mm. 12–14	SG
Mozart	9 Variations in C major on “Lison dort,” K. 264, Theme, mm. 5–8	SG
Mozart	12 Minuets, K. 103, No. 12, Trio, mm. 13–16	SG
Mozart	12 Minuets, K. 103, No. 1, Trio, mm. 1–8	SG
Mozart	Piece in F for Keyboard, K. 33B, mm. 7–12	SG
Schubert	Impromptu in B-flat major, Op. 142, No. 3, mm. 1–8	F&G manual
Schubert	Impromptu in G-flat major, Op. 90, No. 3, mm. 1–8	F&G manual
Schubert	Impromptu in A-flat major, Op. 142, No. 2, mm. 1–8	C&G
Schubert	Wanderer’s Nachtlied, Op. 4, No. 3, mm. 1–3	SG

**Table 1.** The musical excerpts contained in SCHENKER41.

# SYSTEMATIC MULTI-SCALE SET-CLASS ANALYSIS

Agustín Martorell

Universitat Pompeu Fabra  
agustin.martorell@upf.edu

Emilia Gómez

Universitat Pompeu Fabra  
emilia.gomez@upf.edu

## ABSTRACT

This work reviews and elaborates a methodology for hierarchical multi-scale set-class analysis of music pieces. The method extends the systematic segmentation and representation of Sapp's 'keyscapes' to the description stage, by introducing a set-class level of description. This provides a systematic, mid-level, and standard analytical lexicon, which allows the description of any notated music based on fixed temperaments. The method benefits from the representation completeness, the compromise between generalisation and discrimination of the set-class spaces, and the access to hierarchical inclusion relations over time. The proposed *class-matrices* are multidimensional time series encoding the pitch content of every possible music segment over time, regardless the involved time-scales, in terms of a given set-class space. They provide the simplest information mining methods with the ability of capturing sophisticated tonal relations. The proposed *class-vectors*, quantifying the presence of every possible set-class in a piece, are discussed for advanced explorations of corpora. The compromise between dimensionality and informativeness provided by the class-matrices and class-vectors, is discussed in relation with standard content-based tonal descriptors, and music information retrieval applications.

## 1. INTRODUCTION

Pitch-class set theory has been used in music analysis practice since decades. However, its general applicability to post-tonal music has contributed, and still contributes, to be perceived as for specialists only. This apparent difficulty is far from real, and just a matter of the application context. The systematic and objective nature of the theory, together with the compactness of the basic representations, constitutes a powerful and flexible descriptive framework suited for any kind of pitch-based music.<sup>1</sup> This description level is purposeful for several music information

<sup>1</sup> In which the concepts of octave equivalence and fixed temperaments are applicable. Although the pitch relations of interest may be quite different, depending on the temperament and the applied context, any discrete pitch organization of the octave can be handled by the general mathematical framework. In this work, we bound to the twelve-tone equal temperament.

retrieval (MIR) applications, such as structural analysis, similarity, pattern finding, classification, and generation of content-based metadata. More interestingly, it provides a means for approaching complex topics, such as similarity, in alternative and insightful musically-grounded scenarios. In addition, the basic descriptors are trivial to compute, and they can be readily exploited by standard information mining techniques.

The remaining of this work is organised as follows. Section 2 introduces the basic set-theoretical concepts, and contextualise them in terms of our systematic analysis endeavour. Section 3 describes the computational approach. Sections 4 and 5 discuss the method in several application contexts. Section 6 summarises the proposed method, and points to future extensions.

## 2. BACKGROUND

### 2.1 Set-class description

*Pitch class* [1] is defined, in the twelve-tone equal tempered system (TET), as an integer representing the residue class modulo 12 of a pitch, that is, any pitch is mapped to a pitch class by removing its octave information. A *pitch-class set* (henceforth *pc-set*) is a set of pitch classes without repetitions in which the order of succession of the elements in the set is not of interest. In the TET system, there exist  $2^{12} = 4096$  distinct pc-sets, so a vocabulary of 4096 symbols is required for describing any possible segment of music. Any pc-set can also be represented by its intervallic content [5]. Intervals considered regardless of their direction are referred to as *interval classes*. The total count of interval classes in a pc-set can be arranged as a six-dimensional data structure called an *interval vector* [4].

Relevant relational concepts for analysis are the *set-class equivalences*, whereby two pc-sets are considered equivalent if and only if they belong to the same *class*. As pointed out by Straus, equivalence is not the same thing as identity, rather it is a link between musical entities that have something in common. This commonality underlying the surface may eventually lend unity and/or coherence to musical works [12]. In this respect, the class equivalences can be conceived as *all or nothing* similarity measures between two pc-sets. In the context of pc-sets, the number of pitch classes in a set is referred to as its *cardinality*. This is perhaps the coarsest measure of similarity. Despite its theoretical relevance, cardinality is too general a notion of similarity to be of use in many analytical situations. Among the many equivalence systems in the set-theoretical



© Agustín Martorell, Emilia Gómez.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Agustín Martorell, Emilia Gómez. "Systematic multi-scale set-class analysis", 15th International Society for Music Information Retrieval Conference, 2014.

literature, three of them are particularly useful:

1. *Interval vector equivalence* (iv-equivalence), which groups all the pc-sets sharing the same interval vector. There exist 197 different iv-types.
2. *Transpositional equivalence* ( $T_n$ -equivalence), which groups all the pc-sets related to each other by transposition. There exist 348 distinct  $T_n$ -types.
3. *Inversional and transpositional equivalence* ( $T_nI$ -equivalence), which groups all the pc-sets related by transposition and/or inversion. There exist 220 different  $T_nI$ -types (also referred to as  $T_n/T_nI$ -types).

Aside the comprehensive coverage of every possible pc-set, the compromise between discrimination and generalisation of these class-equivalence systems fits a wide range of descriptive needs, thus their extensive usage in general-purpose music analysis. From them, iv-equivalence is the most general (197 classes). It shares most of its classes with  $T_nI$ -equivalence (220 classes), with some exceptions, named *Z-relations* [4], for which the same interval vector groups pc-sets which are not  $T_nI$ -equivalent [7]. The most specific from the three systems is  $T_n$ -equivalence.

## 2.2 Systematic approaches to set-class analysis

To date, one of the most systematic approaches to set-class surface analysis is proposed in [6], under the concept of ‘tail-segment array’, whereby every note in a composition is associated with all the possible segments of a given cardinality that contains it. This segmentation is combined with certain set-class-based ‘detector functions’, in order to obtain summarized information from music pieces and collections. The usefulness of the method is comprehensively discussed in the context of style characterization. Some limitations of this technique are addressed in [8], by first identifying the segmentation, description and representation stages of the method, and extending systematization to all of them simultaneously. This is done by combining the exhaustive segmentation and representation of Sapp’s ‘keyscapes’ [11], with a systematic description of the segments in terms of set-classes. The multidimensional, massive and overlapping information resulting from this method, is managed by summarising features and interfacing design, targeting specific analytical tasks.

## 3. MULTI-SCALE SET-CLASS ANALYSIS

This work elaborates directly upon [8], in which detailed and extended discussions can be consulted. A description of our general method follows.

### 3.1 Segmentation

The input to the system is a sequence of MIDI events, which can be of any rhythmic or polyphonic complexity. This signal is processed by the segmentation stage, for which two different algorithms are used: a) an approximate technique, non comprehensive but practical for interacting

with the data; b) a fully systematic method, which exhausts all the segmentation possibilities.

The approximate method applies many overlapping sliding windows, each of them scanning the music at a different time-scale. The minimum window size and the number of time-scales are user parameters, and can be fine tuned as a trade-off between resolution and computational cost. The same hop size is applied for all the time-scales, in order to provide a regular grid for visualisation and interfacing purposes. Each segment is thus indexed by its centre location (time) and its duration (time-scale).

The fully systematic method is required for the quantitative descriptors in which completeness of representation is necessary. It is computed by finding every change in the pc-set content, whether the product of onsets or offsets, and segmenting the piece by considering all the pairwise combinations among these boundaries.

### 3.2 Description

Denoting pitch-classes by the ordinal convention ( $C=0, \dots, B=11$ ), each segment is analysed as follows. Let  $b_i = 1$  if the pitch-class  $i$  is contained (totally or partially) in the segment, or 0 otherwise. The pc-set in the segment is encoded as an integer  $p = \sum_{i=0}^{11} b_i \cdot 2^{11-i} \in [0,4095]$ .

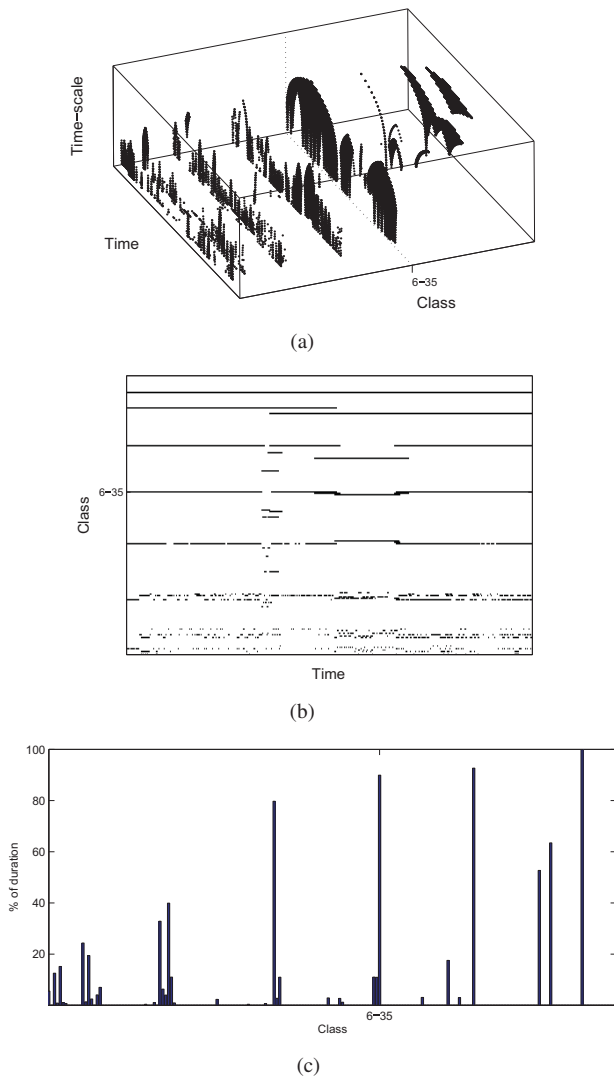
This integer serves as an index for a precomputed table of set classes,<sup>2</sup> including the iv-,  $T_nI$ - and  $T_n$ -equivalences (discussed in Section 2.1). For systematisation completeness, the three class spaces are extended to include the so-called *trivial forms*.<sup>3</sup> With this, the total number of interval vectors rises to 200, while the  $T_nI$ - and  $T_n$ -equivalence classes sum to 223 and 351 categories respectively. In this work, we use Forte’s cardinality-ordinal convention [4] to name the classes, as well as the usual A/B suffix for referring to the prime/inverted forms under  $T_n$ -equivalence. We also follow the conventional notation to name the Z-related classes, by inserting a ‘Z’ between the hyphen and the ordinal. As an example, a segment containing the pitches {G5,C3,E4,C4} is mapped to the pc-set {0,4,7} and coded as  $p = 2192$  (100010010000 in binary). The precomputed table is indexed by  $p$ , resulting in the interval vector ⟨001110⟩ (iv-equivalence, grouping all the sets containing exactly 1 minor third, 1 major third, and 1 fourth), the class 3-11 ( $T_nI$ -equivalence, grouping all the major and minor trichords), and the class 3-11B ( $T_n$ -equivalence, grouping all the major trichords). The discrimination between major and minor trichords is thus possible under  $T_n$ -equivalence (3-11A for minor, 3-11B for major), but not under iv- or  $T_nI$ -equivalences.

### 3.3 Representation

The main data structure, named *class-scape*, is the set-class equivalent of Sapp’s ‘keyscapes’ [11]. It represents the class content of every possible segment, indexed by

<sup>2</sup> As formalised in [4]. See *Supplemental material* (Section 7).

<sup>3</sup> The null set and single pitch classes (cardinalities 0 and 1, containing no intervals), the undecachords (cardinality 11) and the universal pc-set (cardinality 12, also referred to as the *aggregate*).



**Figure 1:** Debussy's *Voiles*. a) class-scape; b) class-matrix; c) class-vector.

their time position and duration. The dimensionality of the class-scapes (time, time-scale and class) is then reduced to more manageable, yet informative, data structures. The first reduction consists on projecting the class-scape to the time-class plane, which results in the concept of *class-matrix*. This is done by realising in time each point in the class-scape, thus retaining a substantial information from the lost dimension (time-scale). A further reduction summarizes the class-matrix in a single vector, named *class-vector*, by quantifying the presence of every possible class in the piece as a percentage of the piece's duration. The class-scape, class-matrix and class-vector, computed from Debussy's *Voiles* are depicted in Figure 1, with the prominent whole-tone scale (class 6-35) labelled as a reference.

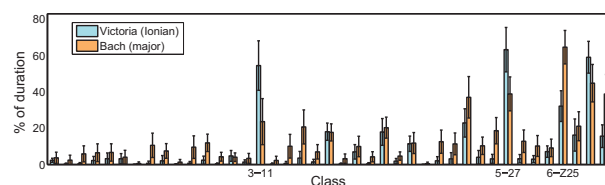
#### 4. MINING CLASS-MATRICES

In this section, we will review and elaborate upon the information conveyed by the class-matrices. Even with the loss of information, the reduction process from the class-scape to the class-matrix guarantees that every instantiation of

every class is represented in the class-matrix, regardless the involved time-scales. The class-matrix represents the temporal *activation* of every possible class over time. A time *point* activated for a given class in the matrix means that it exist at least one segment containing this time point which belongs to such class. As the representation guarantees a strict class-wise separation, the class matrix constitutes a time-series of a special kind. It does not only capture evidence from every class instantiation over time, but it also informs about their set-class *inclusion relations*. The class-matrix, thus, embeds a considerable hierarchical information, allowing the analysis of the specific *constructions* of the class instantiations.

#### 4.1 Case study: subclass analysis

An example of this analytical potential is depicted in Figure 2. It shows the comparison between the *pure diatonicisms* in Victoria's parody masses in Ionian mode<sup>4</sup> and Bach's preludes and fugues in major mode from the Well Tempered Clavier. This is done by first isolating the diatonic segments (activation of 7-35 in the class-matrix) of each movement, and constructing a *subclass-matrix* with the subset content of these segments. The differences can be quantified by computing the corresponding *subclass-vectors* out of the subclass-matrices, and averaging them across pieces in the corpora. This tells about what the particular diatonicisms (and only the diatonicisms) are made of. Some relevant differences stand out from the comparison. Victoria's larger usage of major and minor triads (3-11) and cadential chord sequences (5-27) stands out. On the other hand, Bach makes more prominent usage of the scalar formation 6-Z25: aside its instantiations as perfect cadences, it is recurrent in many motivic progressions, which are not idiomatic in Victoria's contrapuntal writing.



**Figure 2:** Diatonicism in Victoria and Bach. Mean subclass-vectors under 7-35.

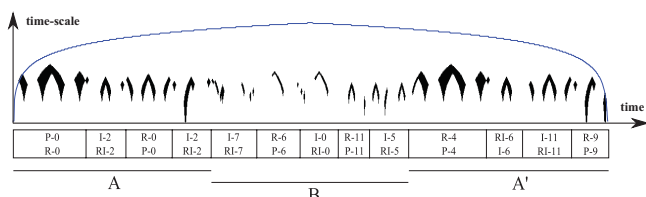
#### 4.2 Case study: structural analysis

Self-similarity matrices (SSM) are a simple standard tool used for structural analysis [3]. Classical inputs to the SSM are spectral or chroma feature time series. Some of the SSM-based methods can handle different time-scales, and some of the chroma methods allows transpositional invariance [9]. These functionalities are usually implemented at the SSM computation stage, or as a post processing. In the class-matrices, both the equivalence mappings (including their inherent hierarchies) and the multi-scale na-

<sup>4</sup> Including *Alma Redemptoris Mater*, *Ave Regina Caelorum*, *Laetatus Sum*, *Pro Victoria*, *Quam Pulchri Sunt*, and *Trahe Me Post Te*. See (Rive, 1969) for a modal classification.

ture of the information are *already* embedded in the feature time-series, so a plain SSM can be used for finding sophisticated recurrences. For instance, a passage comprised of a chord sequence can be recognized as similar than a restated passage with different arpeggiations and/or inversions of the chord intervals (e.g. from major to minor triads). A *vertical* chord and its arpeggiated version may not be recognized as very similar at the lowest cardinalities, but their common  $T_n I$ -sonority will certainly do at their corresponding time-scales. Moreover, any sonority containing the chords (*supersets*) will also be captured at their proper time-scales, climbing up the hierarchy until reaching the whole-piece segment, everything indexed by a common temporal axis. A quantification of similarity between variations may thus be possible at the level of embedded sonorities.

This is discussed next for large-scale recurrence finding in Webern's *Variations for piano*, op.27/I. This serial piece presents an  $A-B-A'$  structure, built upon several instantiations of the main twelve-tone row, at different transpositional and/or inversions levels. Figure 3 (top) depicts the class-scape of the piece, filtered by the prominent hexachordal iv-sonority  $\langle 332232 \rangle$ , and Figure 3 (bottom) shows the well-known (extensively analysed in literature) structure of the row instantiations, annotated according to [2]. Figure 4 depicts the output of a plain SSM, computed from three different inputs: a) the pc-set time series;<sup>5</sup> b) the class-matrix under  $T_n$ ; c) the class-matrix under  $T_n I$ . The pc-equivalence does not capture any large-scale recurrence. The restatement of the first two phrases in  $A$  is captured by the  $T_n$ -equivalence, as these phrases are mainly related by transposition in  $A'$ . Finally, the  $T_n I$ -equivalence reveals the complete recapitulation, including the last two phrases of  $A$ , which are restated in  $A'$  in both transposed and inverted transformations. It is worth noting that the method does not limit to compare the general sonority, the ubiquitous  $\langle 332232 \rangle$ , but its specific construction down the subclass hierarchy. This allows the discrimination of the  $B$  section, built upon the same kind of row instantiations than  $A$  and  $A'$ , but presented in distinct harmonisations.

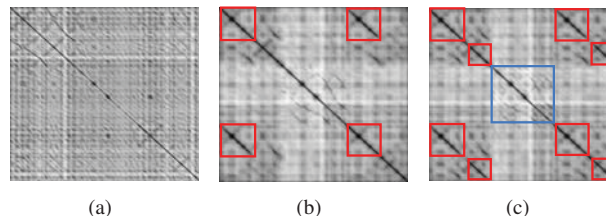


**Figure 3:** Webern's op.27/I. Top: class-scape filtered by  $\langle 332232 \rangle$ ; Bottom: structure.

A relevant advantage of the pc-set-based spaces, with respect to *continuous* ones,<sup>6</sup> is that music can be analysed in terms of different class systems at no extra computational cost. Being *finite and discrete* spaces (4096 classes at most for the TET system), the whole equivalence systems, including their inner metrics, can be precomputed.

<sup>5</sup> In some respect, the discrete *equivalent* of the chroma features.

<sup>6</sup> Such as chroma features, a *finite, but continuous* space.



**Figure 4:** SSM from Webern's op.27/I. a) pc-equivalence; b)  $T_n$ -equivalence; c)  $T_n I$ -equivalence).

The mapping from pc-sets to set-classes, as well as the distances between any pair of music segments, can thus be implemented by table indexing. Once the pc-set of each possible segment has been computed (which constitutes the actual bottleneck of the method), the rest of the process is inexpensive, and multiple *set-class lenses* can be changed in real time, allowing fast interactive explorations of the massive data. This feature, alongside with a variety of filtering options for visual exploration, can be tested with our proof-of-concept set-class analysis tool.<sup>7</sup>

## 5. MINING CLASS-VECTORS

In this section, we will review and elaborate upon the information conveyed by the class-vectors. For each class, the corresponding value in the vector accounts for the relative duration of the piece which is *interpretable* in terms of the specific class, that means, the proportion of time points which are contained in some (at least one) instance of the class. A dataset of class-vectors, thus, can be exploited in a variety of ways. Finding specific sonorities in large datasets can be combined with the extraction of the actual segments from the MIDI files. This can be exploited in varied applications, ranging from corpora analysis to music education.

A dataset of class-vectors was computed from 13480 MIDI tracks, including works by Albéniz, Albinoni, Alkan, Bach, Beethoven, Brahms, Bruckner, Busoni, Buxtehude, Byrd, Chopin, Clementi, Corelli, Couperin, Debussy, Dowland, Frescobaldi, Gesualdo, Guerrero, Haydn, Josquin, Lasso, Liszt, Lully, Mahler, Morales, Mozart, Pachelbel, Palestrina, Satie, Scarlatti, Shostakovich, Schumann, Scriabin, Soler, Stravinsky, Tchaikovsky, Telemann, Victoria and Vivaldi. It also includes anonymous medieval pieces, church hymns, and the Essen folksong collection.

### 5.1 Case study: query by set-class

A simple but useful application is querying the dataset for a given set-class sonority. It can be used, for instance, to find pieces with a relevant presence of *exotic* scales. Table 1 shows 10 retrieved pieces with a notable presence (relative duration) of the sonority 7-22, usually referred to as the Hungarian minor scale.<sup>8</sup> Both monophonic and polyphonic pieces are retrieved, ranging different styles

<sup>7</sup> See *Supplemental material* (Section 7).

<sup>8</sup> Sometimes also called Persian, major gypsy, or double harmonic scale, among other denominations.

and historic periods, as the unique requisite for capturing a given sonority it its existence as a temporal segment.

retrieved piece	7-22 (%)
Scriabin - Prelude op.33 n.3	68.61
Busoni - 6 etudes op.16 n.4	63.22
Essen - 6478	62.50
Liszt - Nuages gris	42.41
Essen - 531	36.67
Scriabin - Prelude op.51 n.2	31.74
Lully - Persee act-iv-scene-iv-28	29.73
Alkan - Esquisses op.63 n.19	28.87
Satie - Gnossienne n.1	28.15
Scriabin - Mazurka op.3 n.9	24.61

**Table 1:** Retrieved pieces: 7-22

### 5.1.1 Query by combined set-classes

The strict separation of classes in the class-vectors, allows the exploration of any class combination, whether common or unusual. For instance, the first movement of Stravinsky's *Symphony of psalms* is retrieved by querying for music containing substantial diatonic (7-35) and octatonic (8-28) material, certainly an uncommon musical combination. The class-vector also reveals the balance between both sonorities, as 30.18 % and 29.25 % of the piece duration, respectively. As discussed in Section 4, the class-matrices allow the hierarchical analysis of specific sonorities. The class-vectors, on the other hand, summarise the information in a way in which it is not possible, in general, to elucidate the subclass content under a given class. However, if the queried sonorities have a substantial presence (or absence) in the piece, the class-vectors alone can often account for some hierarchical evidence. Table 2 shows 10 retrieved pieces, characterised by a notable presence of the so-called *suspended trichord* (3-9),<sup>9</sup> constrained to cases of mostly diatonic contexts (7-35). This situation, as reflected in the results, is likely to be found in medieval melodies, early counterpoint, or works composed as reminiscent of them. It is worth noting that the 3-9 instantiations appear in quite different settings, whether in monophonic voices, as a combination of melody and tonic-dominant drones, and as actual suspended (voiced) chords.

retrieved piece	3-9 (%)	7-35 (%)
Anonym - Angelus ad virginem 1	56.79	100
Anonym - Instrumental dances 7	50.41	100
Lully - Persee prologue-3c	47.11	100
Lully - Phaeton acte-i-scene-v	45.95	90.81
Lully - Persee prologue-3b	44.36	100
Anonym - Ductia	43.82	100
Anonym - Danse royale	35.58	100
Anonym - Cantigas de Santa Maria 2	32.06	100
Anonym - Instrumental dances 9	27.43	100
Frescobaldi - Canzoni da sonare-11	26.69	81.34

**Table 2:** Retrieved pieces: mostly 7-35 with 3-9.

As non-existing sonorities may also reveal important characteristics of music, the dataset can be queried for combinations of present and absent classes. For instance, the

<sup>9</sup> A major trichord with the third degree substituted by the fourth.

sonority of fully diatonic (7-35) pieces depends on whether they contain major or minor trichords (3-11) or not. Retrieved pieces in the latter case (diatonic, not triadic) are mostly medieval melodies or early polyphonic pieces, prior to the establishment of the triad as a common sonority.

These results point to interesting applications related with music similarity, such as music recommendation and music education. We find of particular interest the potential of retrieving pieces sharing relevant tonal-related properties, but pertaining to different styles, composers, or historical periods. Music similarity is, to a great extent, a human construct, as it depends on cultural factors and musical background. It would thus be possible to *learn* to appreciate non familiar similarity criteria, which could be suggested by music discovery or recommendation systems.

## 5.2 On dimensionality and informativeness

In feature design, the ratio between the size of the feature space and the informativeness of description is a relevant factor. The class content of a piece, as described by its class-vector, have 200, 223 or 351 dimensions, depending on the chosen equivalence (*iv*,  $T_n I$  or  $T_n$ ). Compared with other tonal feature spaces, these dimensions may seem quite large. However, the benefits of class vectors are the systematicity, specificity and precision of the description. Several relevant differences with respect to other tonal-related features are to be noticed. A single class-vector, computed after a fully systematic segmentation, accounts for:

1. Every different segment in the piece, regardless of their time position or duration. No sampling artefacts of any kind are introduced.
2. Every possible sonority among the set-class space, which is *complete*. Every instantiation of every class is captured and represented.
3. An objective and precise description of the *set-class sonority*. No probabilities or estimations are involved.
4. A description in (high level) music theoretical terms, readable and interpretable by humans. Set-classes constitute a standard lexicon in music analysis.
5. An objective quantification of every possible sonority in terms of relative duration in the piece. No probabilities or estimations are involved.
6. A content-based, model-free, description of the piece. Neither statistics nor properties learned from datasets are involved.
7. In cases of large presence or notable absences of some sonorities, an approximation to the hierarchical inclusion relations (fully available through the class-matrices only).

In contrast, the most common tonal piecewise and labelwise feature (global key estimation) conveys:

1. A single label for the whole piece, often misleading for music which modulates.
2. 24 different labels, but actually two different sonorities (major and minor), non representative of a vast amount of music.
3. An *estimation* of the key: not only because of the inherent ambiguity of tonality, but also because the (most often) limited tonal *knowledge* of the algorithms.
4. A description in (high level) music theoretical terms, but conveying very little musical information (e.g. at compositional level).
5. No quantification, just a global label. At most, including an indicator of *confidence* (in the descriptor terms), usually the key strength.
6. A description based on specific models (e.g. profiling methods or rule-based), which do not generalize. Some models are trained from specific datasets, biasing the actual *meaning* of the descriptor.
7. No access to the (very rich) hierarchical relations of the piece's tonality.

With this in mind, it seems to us that a piecewise description in 200 dimensions is a reasonable trade-off between size and informativeness. Considering the somewhat sophisticated tonal information conveyed by the class-vectors, they may constitute a useful complementary feature for existing content-based metadata.

## 6. CONCLUSIONS

The proposed systematic methodology for multi-scale set-class analysis is purposeful for common music information retrieval applications. An appropriate mining of the class-matrices can bring insights about the hierarchical relations among the sets, informing about the specific construction of the class sonorities. In combination with simple recurrence finding methods, the class-matrices can be used for music structure analysis of complex music, beyond the scope of mainstream tonal features. The proposed class-vectors, as piecewise tonal summaries, convey a rich information in terms of every possible class sonority. They can be mined for querying tasks of some sophistication. Their compromise between dimensionality and informativeness, point to potential advances in music similarity and recommendation applications. The examples in this work show that set-classes can inform about very different music compositions, ranging simple folk tunes, early polyphony, common-practice period, *exotic* or uncommon scales, and atonal music. Besides our ongoing musicological analyses, and current research with chroma-based transcriptions from audio, future work may explore the potential of these methods in actual classification and recommendation systems.

## 7. SUPPLEMENTAL MATERIAL

The interactive potential of the methods discussed in this work can be tested by our multi-scale set-class analysis prototype for Matlab, freely available from <http://agustinmartorell.weebly.com/set-class-analysis.html>. A comprehensive table of set-classes, and a growing dataset of class-vectors, are also available at this site.

## 8. ACKNOWLEDGEMENTS

This work was supported by the EU 7th Framework Programme FP7/2007-2013 through PHENICX project [grant no. 601166].

## 9. REFERENCES

- [1] M. Babbit: "Some Aspects of Twelve-Tone Composition," *The Score and I.M.A. Magazine*, Vol. 12, pp. 53–61, 1955.
- [2] N. Cook: *A Guide to Music Analysis*, J. M. Dent and Sons, London, 1987.
- [3] J. Foote: "Visualizing Music and Audio Using Self-Similarity," *Proceedings of the ACM Multimedia*, pp. 77–80, 1999.
- [4] A. Forte: "A Theory of Set-Complexes for Music," *Journal of Music Theory*, Vol. 8, No. 2 pp. 136–183, 1964.
- [5] H. Hanson: *The Harmonic Materials of Modern Music: Resources of the Tempered Scale*, Appleton-Century-Crofts, New York, 1960.
- [6] E. Huovinen and A. Tenkanen: "Bird's-Eye Views of the Musical Surface: Methods for Systematic Pitch-Class Set Analysis," *Music Analysis*, Vol. 26, No. 1–2 pp. 159–214, 2007.
- [7] D. Lewin: "Re: Intervallic Relations between Two Collections of Notes," *Journal of Music Theory*, Vol. 3, No. 2 pp. 298–301, 1959.
- [8] A. Martorell and E. Gómez: "Hierarchical Multi-Scale Set-Class Analysis," *Journal of Mathematics and Music*, Online pp. 1-14, 2014.
- [9] M. Müller: *Information Retrieval for Music and Motion*, Springer, Berlin, 2007.
- [10] T. N. Rive: "An Examination of Victoria's Technique of Adaptation and Reworking in his Parody Masses - with Particular Attention to Harmonic and Cadential Procedure," *Anuario Musical*, Vol. 24, pp. 133–152, 1969.
- [11] C. S. Sapp: "Visual Hierarchical Key Analysis," *Computers in Entertainment*, Vol. 4, No. 4 pp. 1–19, 2005.
- [12] J. N. Straus: *Introduction to Post-Tonal Theory*, Prentice-Hall, Upper Saddle River, NJ, 1990.





Oral Session 4  
**Retrieval**

This Page Intentionally Left Blank

# SPOTTING A QUERY PHRASE FROM POLYPHONIC MUSIC AUDIO SIGNALS BASED ON SEMI-SUPERVISED NONNEGATIVE MATRIX FACTORIZATION

Taro Masuda<sup>1</sup> Kazuyoshi Yoshii<sup>2</sup> Masataka Goto<sup>3</sup> Shigeo Morishima<sup>1</sup>

<sup>1</sup>Waseda University <sup>2</sup>Kyoto University

<sup>3</sup>National Institute of Advanced Industrial Science and Technology (AIST)

masutaro@suou.waseda.jp yoshii@i.kyoto-u.ac.jp m.goto@aist.go.jp shigeo@waseda.jp

## ABSTRACT

This paper proposes a query-by-audio system that aims to detect temporal locations where a musical phrase given as a query is played in musical pieces. The “phrase” in this paper means a short audio excerpt that is not limited to a main melody (singing part) and is usually played by a single musical instrument. A main problem of this task is that the query is often buried in mixture signals consisting of various instruments. To solve this problem, we propose a method that can appropriately calculate the distance between a query and partial components of a musical piece. More specifically, gamma process nonnegative matrix factorization (GaP-NMF) is used for decomposing the spectrogram of the query into an appropriate number of basis spectra and their activation patterns. Semi-supervised GaP-NMF is then used for estimating activation patterns of the learned basis spectra in the musical piece by presuming the piece to partially consist of those spectra. This enables distance calculation based on activation patterns. The experimental results showed that our method outperformed conventional matching methods.

## 1. INTRODUCTION

Over a decade, a lot of effort has been devoted to developing music information retrieval (MIR) systems that aim to find musical pieces of interest by using audio signals as the query. For example, there are many similarity-based retrieval systems that can find musical pieces having similar acoustic features to those of the query [5, 13, 21, 22]. Audio fingerprinting systems, on the other hand, try to find a musical piece that exactly matches the query by using acoustic features robust to audio-format conversion and noise contamination [6, 12, 27]. Query-by-humming (QBH) systems try to find a musical piece that includes the melody specified by users’ singing or humming [19]. Note that in gen-

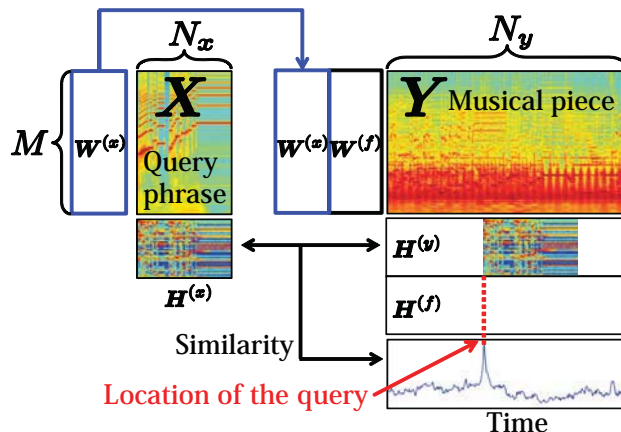


Figure 1. An overview of the proposed method.

eral information of musical scores [9, 16, 23, 31, 39] (such as MIDI files) or some speech corpus [36] should be prepared for a music database in advance of QBH. To overcome this limitation, some studies tried to automatically extract main melodies from music audio signals included in a database [25, 34, 35]. Other studies employ chroma vectors to characterize a query and targeted pieces without the need of symbolic representation or transcription [2].

We propose a task that aims to detect temporal locations at which phrases similar to the query phrase appear in different polyphonic musical pieces. The term “phrase” means a several-second musical performance (audio clip) usually played by a single musical instrument. Unlike QBH, our method needs no musical scores beforehand. A key feature of our method is that we aim to find short segments within musical pieces, not musical pieces themselves. There are several possible application scenarios in which both non-experts and music professionals enjoy the benefits of our system. For example, ordinary users could intuitively find a musical piece by playing just a characteristic phrase used in the piece even if the title of the piece is unknown or forgotten. In addition, composers could learn what kinds of arrangements are used in existing musical pieces that include a phrase specified as a query.

The major problem of our task lies in distance calculation between a query and short segments of a musical piece. One approach would be to calculate the symbolic distance between musical scores. However, this approach is impractical because even the state-of-the-art methods of



© Taro Masuda, Kazuyoshi Yoshii, Masataka Goto, Shigeo Morishima.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Taro Masuda, Kazuyoshi Yoshii, Masataka Goto, Shigeo Morishima. “Spotting a Query Phrase from Polyphonic Music Audio Signals Based on Semi-supervised Nonnegative Matrix Factorization”, 15th International Society for Music Information Retrieval Conference, 2014.

automatic music transcription [4, 11, 17, 29, 38] work poorly for standard popular music. Conventional distance calculation based on acoustic features [5] is also inappropriate because acoustic features of a phrase are drastically distorted if other sounds are superimposed in a musical piece. In addition, since it would be more useful to find locations in which the same phrase is played by different instruments, we cannot heavily rely on acoustic features.

In this paper we propose a novel method that can perform phrase spotting by calculating the distance between a query and *partial* components of a musical piece. Our conjecture is that we could judge whether a phrase is included or not in a musical piece without perfect transcription, like the human ear can. More specifically, gamma process nonnegative matrix factorization (GaP-NMF) [14] is used for decomposing the spectrogram of a query into an appropriate number of basis spectra and their activation patterns. Semi-supervised GaP-NMF is then used for estimating activation patterns of the fixed basis spectra in a target musical piece by presuming the piece to *partially* consist of those spectra. This enables appropriate matching based on activation patterns of the basis spectra forming the query.

## 2. PHRASE SPOTTING METHOD

This section describes the proposed phrase-spotting method based on nonparametric Bayesian NMF.

### 2.1 Overview

Our goal is to detect the start times of a phrase in the polyphonic audio signal of a musical piece. An overview of the proposed method is shown in Figure 1. Let  $\mathbf{X} \in \mathbb{R}^{M \times N_x}$  and  $\mathbf{Y} \in \mathbb{R}^{M \times N_y}$  be the nonnegative power spectrogram of a query and that of a target musical piece, respectively. Our method consists of three steps. First, we perform NMF for decomposing the query  $\mathbf{X}$  into a set of basis spectra  $\mathbf{W}^{(x)}$  and a set of their corresponding activations  $\mathbf{H}^{(x)}$ . Second, in order to obtain temporal activations of  $\mathbf{W}^{(x)}$  in the musical piece  $\mathbf{Y}$ , we perform another NMF whose basis spectra consist of a set of fixed basis spectra  $\mathbf{W}^{(x)}$  and a set of unconstrained basis spectra  $\mathbf{W}^{(f)}$  that are required for representing musical instrument sounds except for the phrase. Let  $\mathbf{H}^{(y)}$  and  $\mathbf{H}^{(f)}$  be sets of activations of  $\mathbf{Y}$  corresponding to  $\mathbf{W}^{(x)}$  and  $\mathbf{W}^{(f)}$ , respectively. Third, the similarity between the activation patterns  $\mathbf{H}^{(x)}$  in the query and the activation patterns  $\mathbf{H}^{(y)}$  in the musical piece is calculated. Finally, we detect locations of a phrase where the similarity takes large values.

There are two important reasons that “nonparametric” “Bayesian” NMF is needed. 1) It is better to automatically determine the optimal number of basis spectra according to the complexity of the query  $\mathbf{X}$  and that of the musical piece  $\mathbf{Y}$ . 2) We need to put different prior distributions on  $\mathbf{H}^{(y)}$  and  $\mathbf{H}^{(f)}$  to put more emphasis on fixed basis spectra  $\mathbf{W}^{(x)}$  than unconstrained basis spectra  $\mathbf{W}^{(f)}$ . If no priors are placed, the musical piece  $\mathbf{Y}$  is often represented by using only unconstrained basis spectra  $\mathbf{W}^{(f)}$ . A key feature of our method is that we *presume* that the

phrase is included in the musical piece when decomposing  $\mathbf{Y}$ . This means that we need to make use of  $\mathbf{W}^{(x)}$  as much as possible for representing  $\mathbf{Y}$ . The Bayesian framework is a natural choice for reflecting such a prior belief.

### 2.2 NMF for Decomposing a Query

We use the gamma process NMF (GaP-NMF) [14] for approximating  $\mathbf{X}$  as the product of a nonnegative vector  $\boldsymbol{\theta} \in \mathbb{R}^{K_x}$  and two nonnegative matrices  $\mathbf{W}^{(x)} \in \mathbb{R}^{M \times K_x}$  and  $\mathbf{H}^{(x)} \in \mathbb{R}^{K_x \times N_x}$ . More specifically, the original matrix  $\mathbf{X}$  is factorized as follows:

$$\mathbf{X}_{mn} \approx \sum_{k=1}^{K_x} \theta_k W_{mk}^{(x)} H_{kn}^{(x)}, \quad (1)$$

where  $\theta_k$  is the overall gain of basis  $k$ ,  $W_{mk}^{(x)}$  is the power of basis  $k$  at frequency  $m$ , and  $H_{kn}^{(x)}$  is the activation of basis  $k$  at time  $n$ . Each column of  $\mathbf{W}^{(x)}$  represents a basis spectrum and each row of  $\mathbf{H}^{(x)}$  represents an activation pattern of the basis over time.

### 2.3 Semi-supervised NMF for Decomposing a Musical Piece

We then perform semi-supervised NMF for decomposing the spectrogram of the musical piece  $\mathbf{Y}$  by fixing a part of basis spectra with  $\mathbf{W}^{(x)}$ . The idea of giving  $\mathbf{W}$  as a dictionary during inference has been widely adopted [3, 7, 15, 18, 24, 26, 28, 30, 33, 38].

We formulate Bayesian NMF for representing the spectrogram of the musical piece  $\mathbf{Y}$  by extensively using the fixed bases  $\mathbf{W}^{(x)}$ . To do this, we put different gamma priors on  $\mathbf{H}^{(y)}$  and  $\mathbf{H}^{(f)}$ . The shape parameter of the gamma prior on  $\mathbf{H}^{(y)}$  is much larger than that of the gamma prior on  $\mathbf{H}^{(f)}$ . Note that the expectation of the gamma distribution is proportional to its shape parameter.

### 2.4 Correlation Calculation between Activation Patterns

After the semi-supervised NMF is performed, we calculate the similarity between the activation patterns  $\mathbf{H}^{(x)}$  in the query and the activation patterns  $\mathbf{H}^{(y)}$  in a musical piece to find locations of the phrase. We expect that similar patterns appear in  $\mathbf{H}^{(y)}$  when almost the same phrases are played in the musical piece even if those phrases are played by different instruments. More specifically, we calculate the sum of the correlation coefficients  $r$  at time  $n$  between  $\mathbf{H}^{(x)}$  and  $\mathbf{H}^{(y)}$  as follows:

$$r(n) = \frac{1}{K_x N_x} \sum_{k=1}^{K_x} \frac{\left( \mathbf{h}_{k1}^{(x)} - \bar{\mathbf{h}}_{k1}^{(x)} \right)^T \left( \mathbf{h}_{kn}^{(y)} - \bar{\mathbf{h}}_{kn}^{(y)} \right)}{\left\| \mathbf{h}_{k1}^{(x)} - \bar{\mathbf{h}}_{k1}^{(x)} \right\| \left\| \mathbf{h}_{kn}^{(y)} - \bar{\mathbf{h}}_{kn}^{(y)} \right\|}, \quad (2)$$

where

$$\mathbf{h}_{ki}^{(\cdot)} = \left[ H_{ki}^{(\cdot)} \cdots H_{k(i+N_x-1)}^{(\cdot)} \right]^T, \quad (3)$$

$$\bar{\mathbf{h}}_{kn}^{(\cdot)} = \frac{1}{N_x} \sum_{j=1}^{N_x} H_{k(n+j-1)}^{(\cdot)} \times [1 \cdots 1]^T. \quad (4)$$

Finally, we detect a start frame  $n$  of the phrase by finding peaks of the correlation coefficients over time. This peak picking is performed based on the following thresholding process:

$$r(n) > \mu + 4\sigma, \quad (5)$$

where  $\mu$  and  $\sigma$  denote the overall mean and standard deviation of  $r(n)$ , respectively, which were derived from all the musical pieces.

## 2.5 Variational Inference of GaP-NMF

This section briefly explains how to infer nonparametric Bayesian NMF [14], given a spectrogram  $\mathbf{V} \in \mathbb{R}^{M \times N}$ . We assume that  $\boldsymbol{\theta} \in \mathbb{R}^K$ ,  $\mathbf{W} \in \mathbb{R}^{M \times K}$ , and  $\mathbf{H} \in \mathbb{R}^{K \times N}$  are stochastically sampled according to a generative process. We choose a gamma distribution as a prior distribution on each parameter as follows:

$$\begin{aligned} p(W_{mk}) &= \text{Gamma}(a^{(W)}, b^{(W)}), \\ p(H_{kn}) &= \text{Gamma}(a^{(H)}, b^{(H)}), \\ p(\theta_k) &= \text{Gamma}\left(\frac{\alpha}{K}, \alpha c\right), \end{aligned} \quad (6)$$

where  $\alpha$  is a concentration parameter,  $K$  is a sufficiently large integer (ideally an infinite number) compared with the number of components in the mixed sound, and  $c$  is the inverse of the mean value of  $\mathbf{V}$ :

$$c = \left( \frac{1}{MN} \sum_m \sum_n V_{mn} \right)^{-1}. \quad (7)$$

We then use the generalized inverse-Gaussian (GIG) distribution as a posterior distribution as follows:

$$\begin{aligned} q(W_{mk}) &= \text{GIG}\left(\gamma_{mk}^{(W)}, \rho_{mk}^{(W)}, \tau_{mk}^{(W)}\right), \\ q(H_{kn}) &= \text{GIG}\left(\gamma_{kn}^{(H)}, \rho_{kn}^{(H)}, \tau_{kn}^{(H)}\right), \\ q(\theta_k) &= \text{GIG}\left(\gamma_k^{(\theta)}, \rho_k^{(\theta)}, \tau_k^{(\theta)}\right). \end{aligned} \quad (8)$$

To estimate the parameters of these distributions, we first update other parameters,  $\phi_{kmn}$ ,  $\omega_{mn}$ , using the following equations.

$$\phi_{kmn} = \mathbb{E}_q \left[ \frac{1}{\theta_k W_{mk} H_{kn}} \right]^{-1}, \quad (9)$$

$$\omega_{mn} = \sum_k \mathbb{E}_q [\theta_k W_{mk} H_{kn}]. \quad (10)$$

After obtaining  $\phi_{kmn}$  and  $\omega_{mn}$ , we update the parameters of the GIG distributions as follows:

$$\begin{aligned} \gamma_{mk}^{(W)} &= a^{(W)}, \quad \rho_{mk}^{(W)} = b^{(W)} + \mathbb{E}_q[\theta_k] \sum_n \frac{\mathbb{E}_q[H_{kn}]}{\omega_{mn}}, \\ \tau_{mk}^{(W)} &= \mathbb{E}_q \left[ \frac{1}{\theta_k} \right] \sum_n V_{mn} \phi_{kmn}^2 \mathbb{E}_q \left[ \frac{1}{H_{kn}} \right], \end{aligned} \quad (11)$$

$$\begin{aligned} \gamma_{kn}^{(H)} &= a^{(H)}, \quad \rho_{kn}^{(H)} = b^{(H)} + \mathbb{E}_q[\theta_k] \sum_m \frac{\mathbb{E}_q[W_{mk}]}{\omega_{mn}}, \\ \tau_{kn}^{(H)} &= \mathbb{E}_q \left[ \frac{1}{\theta_k} \right] \sum_m V_{mn} \phi_{kmn}^2 \mathbb{E}_q \left[ \frac{1}{W_{mk}} \right], \end{aligned} \quad (12)$$

$$\begin{aligned} \gamma_k^{(\theta)} &= \frac{\alpha}{K}, \quad \rho_k^{(\theta)} = \alpha c + \sum_m \sum_n \frac{\mathbb{E}_q[W_{mk} H_{kn}]}{\omega_{mn}}, \\ \tau_k^{(\theta)} &= \sum_m \sum_n V_{mn} \phi_{kmn}^2 \mathbb{E}_q \left[ \frac{1}{W_{mk} H_{kn}} \right]. \end{aligned} \quad (13)$$

The expectations of  $\mathbf{W}$ ,  $\mathbf{H}$  and  $\boldsymbol{\theta}$  are required in Eqs. (9) and (10). We randomly initialize the expectations of  $\mathbf{W}$ ,  $\mathbf{H}$ , and  $\boldsymbol{\theta}$  and iteratively update each parameter by using those formula. As the number of iterations increases, the value of  $\mathbb{E}_q[\theta_k]$  over a certain level  $K^+$  decreases. Therefore, if the value is 60 dB lower than  $\sum_k \mathbb{E}_q[\theta_k]$ , we remove the related parameters from consideration, which makes the calculation faster. Eventually, the number of effective bases,  $K^+$ , gradually reduces during iterations, suggesting that the appropriate number is automatically determined.

## 3. CONVENTIONAL MATCHING METHODS

We describe three kinds of conventional matching methods used for evaluation. The first and the second methods calculate the Euclidean distance between acoustic features (Section 3.1) and that between chroma vectors (Section 3.2), respectively. The third method calculates the Itakura-Saito (IS) divergence between spectrograms (Section 3.3).

### 3.1 MFCC Matching Based on Euclidean Distance

Temporal locations in which a phrase appears are detected by focusing on the acoustic distance between the query and a short segment extracted from a musical piece. In this study we use Mel-frequency cepstrum coefficients (MFCCs) as an acoustic feature, which have commonly been used in various research fields [1, 5]. More specifically, we calculate a 12-dimensional feature vector from each frame by using the Auditory Toolbox Version 2 [32]. The distance between two sequences of the feature vector extracted from the query and the short segment is obtained by accumulating the frame-wise Euclidean distance over the length of the query.

The above-mentioned distance is iteratively calculated by shifting the query frame by frame. Using a simple peak-picking method, we detect locations of the phrase in which the obtained distance is lower than  $m - s$ , where  $m$  and  $s$  denote the mean and standard deviation of the distance over all frames, respectively.

### 3.2 Chromagram Matching Based on Euclidean Distance

In this section, temporal locations in which a phrase appears are detected in the same manner as explained in Section 3.1. A difference is that we extracted a 12-dimensional chroma vector from each frame by using the MIRtoolbox [20]. In addition, we empirically defined the threshold of the peak-picking method as  $m - 3s$ .

### 3.3 DP Matching Based on Itakura-Saito Divergence

In this section, temporal locations in which a phrase appears are detected by directly calculating the Itakura-Saito (IS) divergence [8, 37] between the query  $\mathbf{X}$  and the musical piece  $\mathbf{Y}$ . The use of the IS divergence is theoretically justified because the IS divergence poses a smaller penalty than standard distance measures such as the Euclidean distance and the Kullback-Leibler (KL) divergence when the power spectrogram of the query is included in that of the musical piece.

To efficiently find phrase locations, we use a dynamic programming (DP) matching method based on the IS divergence. First, we make a distance matrix  $\mathbf{D} \in \mathbb{R}^{N_x \times N_y}$  in which each cell  $D(i, j)$  is the IS divergence between the  $i$ -th frame of  $\mathbf{X}$  and the  $j$ -th frame of  $\mathbf{Y}$  ( $1 \leq i \leq N_x$  and  $1 \leq j \leq N_y$ ).  $D(i, j)$  is given by

$$D(i, j) = \mathcal{D}_{\text{IS}}(\mathbf{X}_i | \mathbf{Y}_j) = \sum_m \left( -\log \frac{X_{mi}}{Y_{mj}} + \frac{X_{mi}}{Y_{mj}} - 1 \right), \quad (14)$$

where  $m$  indicates a frequency-bin index. We then let  $\mathbf{E} \in \mathbb{R}^{N_x \times N_y}$  be a cumulative distance matrix. First,  $\mathbf{E}$  is initialized as  $E(1, j) = 0$  for any  $j$  and  $E(i, 1) = \infty$  for any  $i$ .  $E(i, j)$  can be sequentially calculated as follows:

$$E(i, j) = \min \left\{ \begin{array}{l} 1) E(i-1, j-2) + 2D(i, j-1) \\ 2) E(i-1, j-1) + D(i, j) \\ 3) E(i-2, j-1) + 2D(i-1, j) \end{array} \right\} + D(i, j). \quad (15)$$

Finally, we can obtain  $E(N_x, j)$  that represents the distance between the query and a phrase ending at the  $j$ -th frame in the musical piece. We let  $\mathbf{C} \in \mathbb{R}^{N_x \times N_y}$  be a cumulative cost matrix. According to the three cases 1), 2), and 3),  $C(i, j)$  is obtained as follows:

$$C(i, j) = \left\{ \begin{array}{l} 1) C(i-1, j-2) + 3 \\ 2) C(i-1, j-1) + 2 \\ 3) C(i-2, j-1) + 3. \end{array} \right. \quad (16)$$

This means that the length of a phrase is allowed to range from one half to two times of the query length.

Phrase locations are determined by finding the local minima of the regularized distance given by  $\frac{E(N_x, j)}{C(N_x, j)}$ . More specifically, we detect locations in which values of the obtained distance are lower than  $M - S/10$ , where  $M$  and  $S$  denote the median and standard deviation of the distance over all frames, respectively. A reason that we use the median for thresholding is that the distance sometimes takes

an extremely large value (outlier). The mean of the distance tends to be excessively biased by such an outlier. In addition, we ignore values of the distance which are more than  $10^6$  when calculating  $S$  for practical reasons (almost all values of  $\frac{E(N_x, j)}{C(N_x, j)}$  range from  $10^3$  to  $10^4$ ). Once the end point is detected, we can also obtain the start point of the phrase by simply tracing back along the path from the end point.

## 4. EXPERIMENTS

This section reports comparative experiments that were conducted for evaluating the phrase-spotting performances of the proposed method described in Section 2 and the three conventional methods described in Section 3.

### 4.1 Experimental Conditions

The proposed method and the three conventional methods were tested under three different conditions: 1) Exactly the same phrase specified as a query was included in a musical piece (exact match). 2) A query was played by a different kind of musical instruments (timbre change). 3) A query was played in a faster tempo (tempo change).

We chose four musical pieces (RWC-MDB-P-2001 No.1, 19, 42, and 77) from the RWC Music Database: Popular Music [10]. We then prepared 50 queries: 1) 10 were short segments excerpted from original multi-track recordings of the four pieces. 2) 30 queries were played by three kinds of musical instruments (nylon guitar, classic piano, and strings) that were different from those originally used in the four pieces. 3) The remaining 10 queries were played by the same instruments as original ones, but their tempi were 20% faster. Each query was a short performance played by a single instrument and had a duration ranging from 4 s to 9 s. Note that those phrases were not necessarily salient (not limited to main melodies) in musical pieces. We dealt with monaural audio signals sampled at 16 kHz and applied the wavelet transform by shifting short-time frames with an interval of 10 ms. The reason that we did not use short-time Fourier transform (STFT) was to attain a high resolution in a low frequency band. We determined the standard deviation of a Gabor wavelet function to 3.75 ms (60 samples). The frequency interval was 10 cents and the frequency ranged from 27.5 (A1) to 8000 (much higher than C8) Hz.

When a query was decomposed by NMF, the hyperparameters were set as  $\alpha = 1$ ,  $K = 100$ ,  $a^{(W)} = b^{(W)} = a^{(H)} = 0.1$ , and  $b^{(H^{(x)})} = c$ . When a musical piece was decomposed by semi-supervised NMF, the hyperparameters were set as  $a^{(W)} = b^{(W)} = 0.1$ ,  $a^{(H^{(y)})} = 10$ ,  $a^{(H^{(f)})} = 0.01$ , and  $b^{(H)} = c$ . The inverse-scale parameter  $b^{(H)}$  was adjusted to the empirical scale of the spectrogram of a target audio signal. Also note that using smaller values of  $a^{(\cdot)}$  makes parameters sparser in an infinite space.

To evaluate the performance of each method, we calculated the average F-measure, which has widely been used in the field of information retrieval. The precision rate was defined as a proportion of the number of correctly-found

	Precision (%)	Recall (%)	F-measure (%)
MFCC	24.8	35.0	29.0
Chroma	33.4	61.0	43.1
DP	1.9	55.0	3.6
Proposed	53.6	63.0	57.9

**Table 1.** Experimental results in a case that exactly the same phrase specified as a query was included in a musical piece.

	Precision (%)	Recall (%)	F-measure (%)
MFCC	0	0	0
Chroma	18.1	31.7	23.0
DP	1.1	66.3	6.2
Proposed	26.9	56.7	36.5

**Table 2.** Experimental results in a case that a query was played by a different kind of instruments.

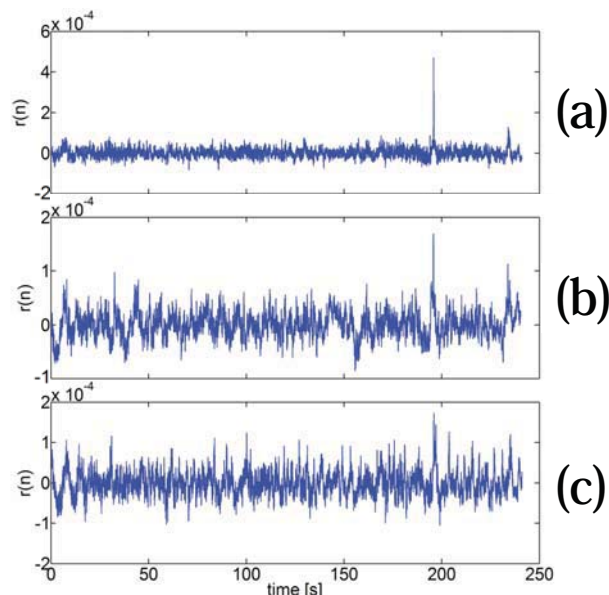
	Precision (%)	Recall (%)	F-measure (%)
MFCC	0	0	0
Chroma	12.0	19.0	14.7
DP	0.5	20.0	2.7
Proposed	15.8	45.0	23.4

**Table 3.** Experimental results in a case that the query phrases was played in a faster tempo.

phrases to that of all the retrieved phrases. The recall rate was defined as a proportion of the number of correctly-found phrases to that of all phrases included in the database (each query phrase was included only in one piece of music). Subsequently, we calculated the F-measure  $F$  by  $F = \frac{2PR}{P+R}$ , where  $P$  and  $R$  denote the precision and recall rates, respectively. We regarded a detected point as a correct one when its error is within 50 frames (500 ms).

## 4.2 Experimental Results

Tables 1–3 show the accuracies obtained by the four methods under each condition. We confirmed that our method performed much better than the conventional methods in terms of accuracy. Figure 2 shows the value of  $r(n)$  obtained from a musical piece in which a query phrase (originally played by the saxophone) is included. We found that the points at which the query phrase starts were correctly spotted by using our method. Although the MFCC-based method could retrieve some of the query phrases in the exact-match condition, it was not robust to timbre change and tempo change. The DP matching method, on the other hand, could retrieve very few correct points because the IS divergence was more sensitive to volume change than the similarity based on spectrograms. Although local minima of the cost function often existed at correct points, those minima were not sufficiently clear because it was difficult to detect the end point of the query from the spectrogram of a mixture audio signal. The chroma-based method worked better than the other conventional methods. However, it did not outperform the proposed method since the chroma-



**Figure 2.** Sum of the correlation coefficients  $r(n)$ . The target piece was RWC-MDB-P-2001 No.42. (a) The query was exactly the same as the target saxophone phrase. (b) The query was played by strings. (c) The query was played 20% faster than the target.

based method often detected false locations including a similar chord progression.

Although our method worked best of the four, the accuracy of the proposed method should be improved for practical use. A major problem is that the precision rate was relatively lower than the recall rate. Wrong locations were detected when queries were played in *staccato* manner because many false peaks appeared at the onset of *staccato* notes.

As for computational cost, it took 29746 seconds to complete the retrieval of a single query by using our method. This was implemented in C++ on a 2.93 GHz Intel Xeon Windows 7 with 12 GB RAM.

## 5. CONCLUSION AND FUTURE WORK

This paper presented a novel query-by-audio method that can detect temporal locations where a phrase given as a query appears in musical pieces. Instead of pursuing perfect transcription of music audio signals, our method used nonnegative matrix factorization (NMF) for calculating the distance between the query and partial components of each musical piece. The experimental results showed that our method performed better than conventional matching methods. We found that our method has a potential to find correct locations in which a query phrase is played by different instruments (timbre change) or in a faster tempo (tempo change).

Future work includes improvement of our method, especially under the timbre-change and tempo-change conditions. One promising solution would be to classify basis spectra of a query into instrument-dependent bases (*e.g.*,

noise from the guitar) and common ones (e.g., harmonic spectra corresponding to musical notes) or to create an universal set of basis spectra. In addition, we plan to reduce the computational cost of our method based on nonparametric Bayesian NMF.

**Acknowledgment:** This study was supported in part by the JST OngaCREST project.

## 6. REFERENCES

- [1] J. J. Aucouturier and F. Pachet. Music Similarity Measures: What's the Use?, *ISMIR*, pp. 157–163, 2002.
- [2] C. de la Bandera, A. M. Barbancho, L. J. Tardón, S. Sammartino, and I. Barbancho. Humming Method for Content-Based Music Information Retrieval, *ISMIR*, pp. 49–54, 2011.
- [3] L. Benaroya, F. Bimbot, and R. Gribonval. Audio Source Separation with a Single Sensor, *IEEE Trans. on ASLP*, 14(1):191–199, 2006.
- [4] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic Music Transcription: Breaking the Glass Ceiling, *ISMIR*, pp. 379–384, 2012.
- [5] A. Berenzweig, B. Logan, D. P. Ellis, and B. Whitman. A Large-Scale Evaluation of Acoustic and Subjective Music-Similarity Measures, *Computer Music Journal*, 28(2):63–76, 2004.
- [6] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A Review of Audio Fingerprinting, *Journal of VLSI Signal Processing Systems for Signal, Image and Video Technology*, 41(3):271–284, 2005.
- [7] Z. Duan, G. J. Mysore, and P. Smaragdis. Online PLCA for Real-Time Semi-supervised Source Separation, *Latent Variable Analysis and Signal Separation*, Springer Berlin Heidelberg, pp. 34–41, 2012.
- [8] A. El-Jaroudi and J. Makhoul. Discrete All-Pole Modeling, *IEEE Trans. on Signal Processing*, 39(2):411–423, 1991.
- [9] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith. Query by Humming: Musical Information Retrieval in an Audio Database, *ACM Multimedia*, pp. 231–236, 1995.
- [10] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC Music Database: Popular, Classical, and Jazz Music Databases, *ISMIR*, pp. 287–288, 2002.
- [11] G. Grindlay and D. P. W. Ellis. A Probabilistic Subspace Model for Multi-instrument Polyphonic Transcription, *ISMIR*, pp. 21–26, 2010.
- [12] J. Haitsma and T. Kalker. A Highly Robust Audio Fingerprinting System, *ISMIR*, pp. 107–115, 2002.
- [13] M. Helén and T. Virtanen. Audio Query by Example Using Similarity Measures between Probability Density Functions of Features, *EURASIP Journal on Audio, Speech, and Music Processing*, 2010.
- [14] M. D. Hoffman, D. M. Blei, and P. R. Cook. Bayesian Non-parametric Matrix Factorization for Recorded Music, *ICML*, pp. 439–446, 2010.
- [15] X. Jaureguiberry, P. Leveau, S. Maller, and J. J. Burred. Adaptation of source-specific dictionaries in Non-Negative Matrix Factorization for source separation, *ICASSP*, pp. 5–8, 2011.
- [16] T. Kageyama, K. Mochizuki, and Y. Takashima. Melody Retrieval with Humming, *ICMC*, pp.349–351, 1993.
- [17] H. Kameoka, K. Ochiai, M. Nakano, M. Tsuchiya, and S. Sagayama. Context-free 2D Tree Structure Model of Musical Notes for Bayesian Modeling of Polyphonic Spectrograms, *ISMIR*, pp. 307–312, 2012.
- [18] H. Kirchhoff, S. Dixon, and A. Klapuri. Multi-Template Shift-variant Non-negative Matrix Deconvolution for Semi-automatic Music Transcription, *ISMIR*, pp. 415–420, 2012.
- [19] A. Kotsifakos, P. Papapetrou, J. Hollmén, D. Gunopulos, and V. Athitsos. A Survey of Query-By-Humming Similarity Methods, *International Conference on PETRA*, 2012.
- [20] O. Lartillot and P. Toivainen. A Matlab Toolbox for Musical Feature Extraction from Audio, *DAFx*, pp. 237–244, 2007.
- [21] T. Li and M. Ogiwara. Content-based Music Similarity Search and Emotion Detection, *ICASSP*, Vol. 5, pp. 705–708, 2004.
- [22] B. Logan and A. Salomon. A Music Similarity Function Based on Signal Analysis, *International Conference on Multimedia and Expo (ICME)*, pp. 745–748, 2001.
- [23] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Cunningham. Towards the Digital Music Library: Tune Retrieval from Acoustic Input, *ACM international conference on Digital libraries*, pp. 11–18, 1996.
- [24] G. J. Mysore and P. Smaragdis. A Non-negative Approach to Semi-supervised Separation of Speech from Noise with the Use of Temporal Dynamics, *ICASSP*, pp. 17–20, 2011.
- [25] T. Nishimura, H. Hashiguchi, J. Takita, J. X. Zhang, M. Goto, and R. Oka. Music Signal Spotting Retrieval by a Humming Query Using Start Frame Feature Dependent Continuous Dynamic Programming, *ISMIR*, pp. 211–218, 2001.
- [26] A. Ozerov, P. Philippe, R. Gribonval, and F. Bimbot. One Microphone Singing Voice Separation Using Source-adapted Models, *WASPAA*, pp. 90–93, 2005.
- [27] M. Ramona and G. Peeters. AudioPrint: An efficient audio fingerprint system based on a novel cost-less synchronization scheme, *ICASSP*, pp. 818–822, 2013.
- [28] S. T. Roweis. One Microphone Source Separation, *Advances in Neural Information Processing Systems*, Vol. 13, MIT Press, pp. 793–799, 2001.
- [29] M. Ryyänen and A. Klapuri. Automatic Bass Line Transcription from Streaming Polyphonic Audio, *ICASSP*, pp. IV–1437–1440, 2007.
- [30] M. N. Schmidt and R. K. Olsson. Single-Channel Speech Separation using Sparse Non-Negative Matrix Factorization, *Interspeech*, pp. 1652–1655, 2006.
- [31] J. Shifrin, B. Pardo, C. Meek, and W. Birmingham. HMM-Based Musical Query Retrieval, *ACM/IEEE-CS Joint Conference on Digital Libraries*, pp. 295–300, 2002.
- [32] M. Slaney. Auditory Toolbox Version 2, *Technical Report #1998-010*, Interval Research Corporation, 1998.
- [33] P. Smaragdis, B. Raj, and M. Shashanka. Supervised and Semi-supervised Separation of Sounds from Single-Channel Mixtures, *Independent Component Analysis and Signal Separation*, Springer Berlin Heidelberg, pp. 414–421, 2007.
- [34] C. J. Song, H. Park, C. M. Yang, S. J. Jang, and S. P. Lee. Implementation of a Practical Query-by-Singing/Humming (QbSH) System and Its Commercial Applications, *IEEE Trans. on Consumer Electronics*, 59(2):407–414, 2013.
- [35] J. Song, S. Y. Bae, and K. Yoon. Mid-Level Music Melody Representation of Polyphonic Audio for Query-by-Humming System, *ISMIR*, pp. 133–139, 2002.
- [36] C. C. Wang, J. S. R. Jang, and W. Wang. An Improved Query by Singing/Humming System Using Melody and Lyrics Information, *ISMIR*, pp. 45–50, 2010.
- [37] B. Wei and J. D. Gibson. Comparison of Distance Measures in Discrete Spectral Modeling, *IEEE DSP Workshop*, 2000.
- [38] F. Weninger, C. Kirst, B. Schuller, and H. J. Bungartz. A Discriminative Approach to Polyphonic Piano Note Transcription Using Supervised Non-negative Matrix Factorization, *ICASSP*, pp. 26–31, 2013.
- [39] Y. Zhu and D. Shasha. Query by Humming: a Time Series Database Approach, *SIGMOD*, 2003.



# BAYESIAN AUDIO ALIGNMENT BASED ON A UNIFIED GENERATIVE MODEL OF MUSIC COMPOSITION AND PERFORMANCE

Akira Maezawa<sup>1,2</sup> Katsutoshi Itoyama<sup>2</sup> Kazuyoshi Yoshii<sup>2</sup> Hiroshi G. Okuno<sup>3</sup>

<sup>1</sup>Yamaha Corporation <sup>2</sup>Kyoto University <sup>3</sup>Waseda University

{amaezaw1, itoyama, yoshii}@kuis.kyoto-u.ac.jp, okuno@aoni.waseda.jp

## ABSTRACT

This paper presents a new probabilistic model that can align multiple performances of a particular piece of music. Conventionally, dynamic time warping (DTW) and left-to-right hidden Markov models (HMMs) have often been used for audio-to-audio alignment based on a shallow acoustic similarity between performances. Those methods, however, cannot distinguish latent musical structures common to all performances and temporal dynamics unique to each performance. To solve this problem, our model explicitly represents two state sequences: a top-level sequence that determines the common structure inherent in the music itself and a bottom-level sequence that determines the actual temporal fluctuation of each performance. These two sequences are fused into a hierarchical Bayesian HMM and can be learned at the same time from the given performances. Since the top-level sequence assigns the same state for note combinations that repeatedly appear within a piece of music, we can unveil the latent structure of the piece. Moreover, we can easily compare different performances of the same piece by analyzing the bottom-level sequences. Experimental evaluation showed that our method outperformed the conventional methods.

## 1. INTRODUCTION

Multiple audio alignment is one of the most important tasks in the field of music information retrieval (MIR). A piece of music played by different people produces different expressive performances, each embedding the unique interpretation of the player. To help a listener better understand the variety of interpretation or discover a performance that matches his/her taste, it is effective to clarify how multiple performances differ by using visualization or playback interfaces [1–3]. Given multiple musical audio signals that play a same piece of music from the beginning to the end, our goal is to find a temporal mapping among different signals while considering the underlying music score.

This paper presents a statistical method of offline multiple audio alignment based on a probabilistic generative

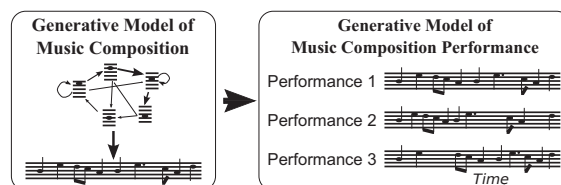


Figure 1. An overview of generative audio alignment.

model that can integrate various sources of uncertainties in music, such as spectral shapes, temporal fluctuations and structural deviations. Our model expresses how a *musical composition* gets *performed*, so it must model how they are generated.<sup>1</sup> Such a requirement leads to a conceptual model illustrated in Figure 1, described using a combination of two complementary models.

To represent the generative process of a musical composition, we focus on the general fact that small fragments consisting of multiple musical notes form the basic building blocks of music and are organized into a larger work. For example, the sonata form is based on developing two contrasting fragments known as the “subject groups,” and a song form essentially repeats the same melody. Our model is suitable for modeling the observation that basic melodic patterns are reused to form the sonata or the song.

To represent the generative process of each performance, we focus on temporal fluctuations from a common music composition. Since each performance plays the same musical composition, the small fragments should appear in the same order. On the other hand, each performance can be played by a different set of musical instruments with a unique tempo trajectory.

Since both generative processes are mutually dependent, we integrate a generative model of music composition with that of performance in a hierarchical Bayesian manner. In other words, we separate the characteristics of a given music audio signal into those originating from the underlying music score and those from the unique performance. Inspired by a typical preprocessing step in music structure segmentation [6, 7], we represent a music composition as a sequence generated from a compact, ergodic Markov model (“latent composition”). Each music performance is represented as a left-to-right Markov chain that traverses the latent composition with the state durations unique to each performance.<sup>2</sup>

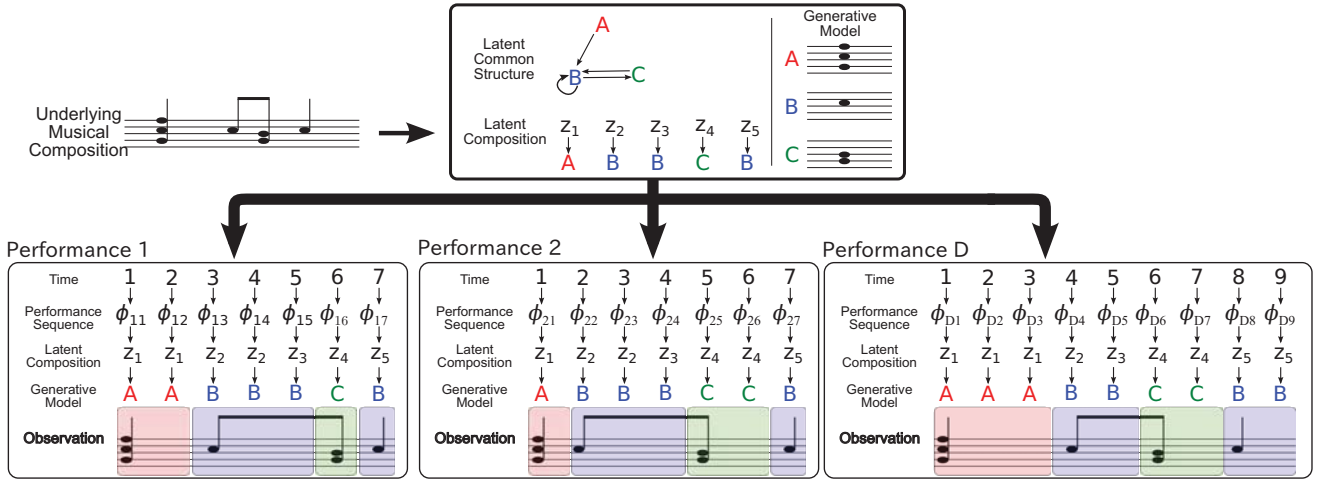
<sup>1</sup> A generative audio alignment model depends heavily on the model of both how the music is *composed* and how the composition is *performed*. This is unlike generative audio-to-score alignment [4, 5], which does not need a music composition model because a music score is already given.

<sup>2</sup> Audio samples are available on the website of the first author.



© Akira Maezawa, Katsutoshi Itoyama, Kazuyoshi Yoshii, Hiroshi G. Okuno.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Akira Maezawa, Katsutoshi Itoyama, Kazuyoshi Yoshii, Hiroshi G. Okuno. “Bayesian Audio Alignment Based on a Unified Generative Model of Music Composition and Performance”, 15th International Society for Music Information Retrieval Conference, 2014.



**Figure 2.** The concept of our method. Music composition is modeled as a sequence (composition sequence) from an ergodic Markov model, and each performance plays the composition sequence, traversing the composition sequence in the order it appears, but staying in each state with different duration.

## 2. RELATED WORK

Audio alignment is typically formulated as a problem of maximizing the similarity or minimizing the cost between a performance and another performance whose time-axis has been “stretched” by a time-dependent factor, using dynamic time warping (DTW) and its variants [8, 9] or other model of temporal dynamics [10]. To permit the use of a simple similarity measure, it is important to design robust acoustic features [11, 12].

Alternatively, tackling alignment by a probabilistic generative model has gathered attention, especially in the context of audio-to-music score alignment [4, 5]. In general, a probabilistic model is formulated to describe how each note in a music score translates to an audio signal. It is useful when one wishes to incorporate, in a unified framework, various sources of uncertainties present in music, such as inclusion of parts [13], mistakes [14], or timbral variations [15–17].

Previous studies in generative audio alignment [13, 18] ignores the organization present in musical composition, by assuming that a piece of music is generated from a left-to-right Markov chain, *i.e.*, a Markov chain whose state appears in the same order for all performances.

## 3. FORMULATION

We formulate a generative model of alignment that aligns  $D$  performances. We provide a conceptual overview, and then mathematically formalize the concept.

### 3.1 Conceptual Overview

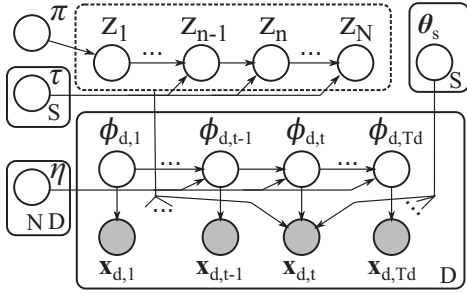
We first extract short-time audio features from each of  $D$  performances. Let us denote the feature sequence for the  $d$ th performance at frame  $t \in [1, T_d]$  as  $\mathbf{x}_{d,t}$ , where  $T_d$  is the total number of frames for the  $d$ th audio signal. Here, the kind of feature is arbitrary, and depends on the generative model of the short-time audio. Then, we model  $\mathbf{x}_{d,t}$  as a set of  $D$  state sequences. Each state is associated with

a unique generative process of short-time audio feature. In other words, each state represents a distinct audio feature, *e.g.*, distinct chord,  $f_0$  and so on, depending on how the generative model of the feature is designed.

For audio alignment, the state sequence must abide by two rules. First, the order in which each state appears is the same for all  $D$  feature sequences. In other words, every performance is described by one sequence of distinct audio features, *i.e.*, the musical piece that the performances play in common. We call such a sequence the *latent composition*. Second, the duration that each performance resides in a given state in the latent composition can be unique to the performance. In other words, each performance traverses the latent composition with a unique “tempo curve.” We call the sequence that each performance traverses over the latent composition sequence as the *performance sequence*.

The latent composition is a sequence of length  $N$  drawn from an ergodic Markov model, which we call the *latent common structure*. We describe the latent composition as  $z_n$ , a sequence of length  $N$  and  $S$  states, where each state describes a distinct audio feature. In other words, we assume that the musical piece is described by at most  $N$  distinct audio events, using at most  $S$  distinct sounds. The latent common structure encodes the structure inherent to the music. The transition probabilities of each state sheds light on a “typical” performance, *e.g.*, melody line or harmonic progression. Therefore, the latent common structure provides a generative model of music composition.

The performance sequence provides a generative model of performance. Each audio signal is modeled as an emission from a  $N$ -state left-to-right Markov model, where the  $n$ th state refers to the generative model associated with the  $n$ th position in the latent composition. Specifically, let us denote the performance sequence for audio  $d$  as  $\phi_{d,t}$ , which is a state sequence of length  $T_d$  and  $N$  states, such that state  $n$  refers to the  $n$ th element of the latent composition. Each performance sequence is constrained such that (1) it begins in state 1 and ends at state  $N$ , and (2) state  $n$  may traverse only to itself or state  $n+1$ . In other words, we



**Figure 3.** Graphical model of our method. Dotted box indicates that the arrow depends on all variables inside the dotted box. Hyperparameters are omitted.

constrain each performance sequence to traverse the latent composition in the same order but with a unique duration. Such a model conveys the idea that each performance can independently play a piece in any tempo trajectory.

### 3.1.1 An Example

Let us illustrate our method in Figure 2. In the example,  $S = 3$  and  $N = 5$ , where state “A” corresponds to a combination of notes G, C and F, “B” corresponds to the note C, and so on; moreover,  $z_n$  encodes the state sequence “AB-BCB,” as to reflect the underlying common music composition that the performances play. Note that a single note may be expressed using more than one state in the latent composition, *e.g.*, both  $z_2$  and  $z_3$  describe the note “C.” Next, each performance aligns to the latent composition, through the performance sequence. Each state of the performance sequence is associated to a position in the latent composition. For example,  $\phi_{1,3}$  is associated to position 2 of  $z$ ,  $z_2$ . Then, at each time, the observation is generated by emitting from the state in latent common structure referred by the current frame of the current audio. This is determined hierarchically by looking up the state  $n$  of the performance sequence of audio  $d$  at time  $t$ , and referring to the state  $s$  of the  $n$ th element of the latent composition. In the example,  $\phi_{1,3}$  refers to state  $n = 2$ , so the generative model corresponding to  $z_{n=2}$ , or “B,” is referred.

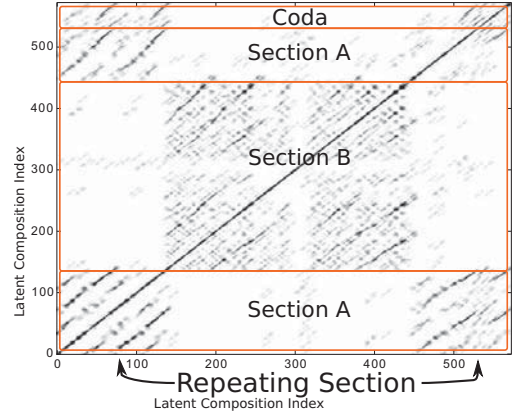
## 3.2 Formulation of the Generative Model

Let us mathematically formalize the above concept using a probabilistic generative model, summarized as a graphical model shown in Fig. 3.

### 3.2.1 Latent Composition and Common Structure

The latent composition is described as  $z_{n=\{1\dots N\}}$ , a  $S$ -state state sequence of length  $N$ , generated from the latent common structure. We shall express the latent composition  $z_n$  using one-of- $S$  representation;  $z_n$  is a  $S$ -dimensional binary variable where, when the state of  $z_n$  is  $s$ ,  $z_{n,s} = 1$  and all other elements are 0. Then, we model  $z$  as a sequence from the latent common structure, an ergodic Markov chain with initial state probability  $\pi$  and transition probability  $\tau$ :

$$p(z|\pi, \tau) = \prod_{s=1}^S \pi_s^{z_{1,s}} \prod_{n=2, s'=1, s=1}^{N, S, S} \tau_{s, s'}^{z_{n-1, s'} z_{n, s}} \quad (1)$$



**Figure 4.** Structural annotation on Chopin Op. 41-2 and the similarity matrix computed from its latent composition.

Each state  $s$  is associated with an arbitrary set of parameters  $\theta_s$  that describes the generative process of the audio feature. We assume that  $\tau_s$  is generated from a conjugate Dirichlet distribution, *i.e.*,  $\tau_s \sim \text{Dir}(\tau_{0,s})$ . The same goes for the initial state probability  $\pi$ , *i.e.*,  $\pi \sim \text{Dir}(\pi_0)$ . The hyperparameters  $\tau_{0,s}$  and  $\pi_0$  are set to a positive value less than 1, which induces sparsity of  $\tau$  and  $\pi$ , and hence leads to a compact latent common structure.

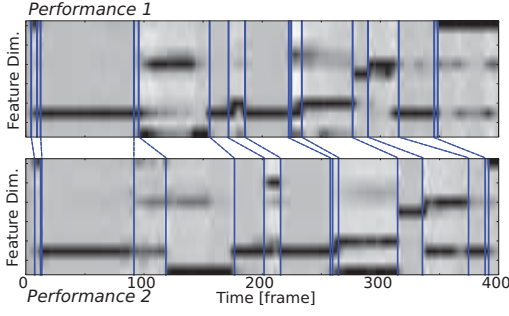
The latent composition and structure implicitly convey the information about how the music is structured and what its building blocks are. Figure 4 shows a similarity matrix derived from the estimated latent composition of Op. 41-2 by F. Chopin<sup>3</sup> having the ternary form (*a.k.a.* ABA form). The first “A” section repeats a theme of form “DED $\bar{F}$ ” repeated twice. The second section is in a modulated key. Finally, the last section repeats the first theme, and ends with a short coda, borrowing from “ $\bar{F}$ ” motive from the first theme. Noting that the diagonal lines of a similarity matrix represent strong similarity, we may unveil such a trend by analyzing the matrix. The bottom-left diagonal lines in the first section, for example, shows that a theme repeats, and the top-left diagonal suggests that the first theme is repeated at the end. This suggests that the latent composition reflects the organization of music.

Notice that this kind of structure arises because we explicitly model the organization of music, conveyed through an ergodic Markov model; simply aligning multiple performances to a single left-to-right HMM [13, 18] is insufficient because it cannot revisit a previously visited state.

### 3.2.2 Performance Sequence

Recall that we require the performance sequence such that (1) it traverses in the order of latent composition, and (2) the duration that each performance stays in a particular state in the latent composition is conditionally independent given the latent composition. To satisfy these requirements, we model the performance sequence as a  $N$ -state left-to-right Markov chain of length  $T_d$ ,  $\phi_{d,t}$ , where the first state of the chain is fixed to the beginning of the latent

<sup>3</sup> The similarity matrix  $R_{i,j}$  was determined by removing self-transitions from  $z_n$  and assigning it to  $z'_n$ , and setting  $R_{i,j} = 1$  if  $z'_i = z'_j$ , and 0 otherwise. Next, we convolved  $R$  by a two-dimensional filter that emphasizes diagonal lines.



**Figure 5.** Feature sequences (chroma vector) of two performances, overlaid by points where the state of the latent composition changes.

composition and the last state to be the end. This assumes that there are no cuts or repeats unique to a performance. Let us define  $\eta_{d,n}$  to be the probability for performance  $d$  to traverse from position  $n$  of the latent composition to  $n+1$ . Then, we model the performance sequence as follows:

$$p(\phi_{d,t=\{1..T_d\}}) = \delta(n, 1)^{\phi_{d,1,n}} \delta(n, S)^{\phi_{d,T_d,n}} \times \prod_{t=1, n=1}^{T_d, N} \left[ \eta_{d,n}^{\phi_{d,t-1,n} \phi_{d,t,n+1}} \times (1 - \eta_{d,n})^{\phi_{d,t-1,n} \phi_{d,t,n}} \right] \quad (2)$$

where  $\delta(x, y)$  indicates the Kronecker Delta, *i.e.*, its value is 1 when  $x = y$  and 0 otherwise. We assume  $\eta_{d,n}$  is drawn from a conjugate Beta distribution, *i.e.*,  $\eta_{d,n} \sim \text{Beta}(a_0, b_0)$ . The ratio  $a_0/b_0$  controls the likelihood of traversing to next states, and their magnitudes control the influence of the observation on the posterior distribution.

Figure 5 shows excerpts of the feature sequences obtained from two performances, and blue lines indicating the change of the state of the latent composition has changed. The figure suggests that the state changes with a notable change in the feature, such as when new notes are played. Since, by the definition of a left-to-right Markov model, the number of vertical lines is identical for all performances, we can align audio signals by mapping the occurrences of the  $i$ th vertical line for all performances, for each  $i$ .

### 3.2.3 Generating Audio Features

Based on the previous expositions, we can see that at time  $t$  of performance  $d$ , the audio feature is generated by choosing the state in the latent common structure that is referred at time  $t$  for performance  $d$ . This state is extracted by referring to the performance sequence to recover the position of the latent composition. Therefore, the observation likelihood is given as follows:

$$p(\mathbf{x}_{d,t} | \mathbf{z}, \phi, \theta) = \prod_{s,n} p(\mathbf{x}_{d,t} | \theta_s)^{z_{n,s} \phi_{d,t,n}} \quad (3)$$

Here,  $p(\mathbf{x} | \theta_s)$  is the likelihood of observation feature  $\mathbf{x}$  at state  $s$  of the latent common structure, and its parameter  $\theta_s$  is generated from a prior distribution  $p(\theta_s | \theta_0)$ .

For the sake of simplicity, we let  $p(\mathbf{x}_{d,t} | \theta_s)$  be a  $\text{dim}(\mathbf{x})$ -dimensional Gaussian distribution with its parameters  $\theta_s$  generated from its conjugate distribution, the Gaussian-Gamma distribution. Specifically we let  $\theta_s = \{\boldsymbol{\mu}_s, \boldsymbol{\lambda}_s\}$ ,  $\theta_0 = \{\mathbf{m}_0, \nu_0, u_0, \mathbf{k}_0\}$ , and let  $\mathbf{x}_{d,t} | \boldsymbol{\mu}_s, \boldsymbol{\lambda}_s \sim \mathcal{N}(\boldsymbol{\mu}_s, \boldsymbol{\lambda}_s^{-1})$ ,

with  $p(\boldsymbol{\mu}_{s,i}, \boldsymbol{\lambda}_{s,i}) \propto \lambda_s^{u_0 - \frac{1}{2}} e^{-(\boldsymbol{\mu}_{s,i} - \mathbf{m}_0, i)^2 \boldsymbol{\lambda}_{s,i} \nu_0 - k_0, i \boldsymbol{\lambda}_{s,i}}$ . One may incorporate a more elaborate model that better expresses the observation.

### 3.3 Inferring the Posterior Distribution

We derive the posterior distribution to the model described above. Since direct application of Bayes' rule to arrive at the posterior is difficult, we employ the variational Bayes method [19] and find an approximate posterior of form  $q(\phi, \mathbf{z}, \boldsymbol{\theta}, \boldsymbol{\eta}, \boldsymbol{\pi}, \boldsymbol{\tau}) = \prod_d q(\phi_{d,\cdot}) q(\mathbf{z}) q(\boldsymbol{\pi}) \prod_{d,n} q(\eta_{d,n}) \prod_s q(\boldsymbol{\theta}_s) q(\boldsymbol{\tau}_s)$  that minimizes the Kullback-Leibler (KL) divergence to the true posterior distribution.

$q(\phi)$  and  $q(\mathbf{z})$  can be updated in a manner analogous to a HMM. For  $q(\mathbf{z})$ , we perform the forward-backward algorithm, with the state emission probability  $\mathbf{g}_n$  at position  $n$  of the latent composition and the transition probability  $\mathbf{v}_s$  from state  $s$  given as follows:

$$\log \mathbf{g}_{n,s} = \sum_{d,t} \langle \phi_{d,t,n} \rangle \langle \log p(\mathbf{x}_{d,t} | \boldsymbol{\theta}_s) \rangle \quad (4)$$

$$\log \mathbf{v}_{s,s'} = \langle \log \tau_{s,s'} \rangle \quad (5)$$

Here,  $\langle f(x) \rangle$  denotes the expectation of  $f(x)$  w.r.t.  $q$ . Likewise, for  $q(\phi_{d,t})$ , we perform the forward-backward algorithm, with the state emission probability  $\mathbf{h}_{d,n}$  and transition probability  $\mathbf{w}_{d,s}$  given as follows:

$$\log \mathbf{h}_{d,t,n} = \sum \langle z_{n,s} \rangle \langle \log p(\mathbf{x}_{d,t} | \boldsymbol{\theta}_s) \rangle \quad (6)$$

$$\log \mathbf{w}_{d,n,n'} = \begin{cases} \langle \log \eta_{d,n} \rangle & n = n' \\ \langle \log(1 - \eta_{d,n}) \rangle & n + 1 = n' \end{cases} \quad (7)$$

We can update  $\boldsymbol{\pi}$  as  $q(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}_0 + \langle \mathbf{z}_1 \rangle)$ ,  $\boldsymbol{\eta}$  as  $q(\eta_{d,n}) = \text{Beta}(a_0 + \sum_t \langle \phi_{d,t-1,n} \phi_{d,t,n} \rangle, b_0 + \sum_t \langle \phi_{d,t-1,n-1} \phi_{d,t,n} \rangle)$ , and  $\boldsymbol{\tau}$  as  $q(\boldsymbol{\tau}_s) = \text{Dir}(\boldsymbol{\tau}_{0,s} + \sum_{n>1} \langle z_{n-1,s} \mathbf{z}_n \rangle)$ .

Based on these parameters, the generative model of audio features can be updated. Some commonly-used statistics for state  $s$  include the count  $\bar{N}_s$ , the mean  $\bar{\boldsymbol{\mu}}_s$  and the variance  $\bar{\boldsymbol{\Sigma}}_s$ , which are given as follows:

$$\bar{N}_s = \sum_{d,n,t} \langle z_{n,s} \rangle \langle \phi_{d,t,n} \rangle \quad (8)$$

$$\bar{\boldsymbol{\mu}}_s = \frac{1}{\bar{N}_s} \sum_{d,n,t} \langle z_{n,s} \rangle \langle \phi_{d,t,n} \rangle \mathbf{x}_{d,t} \quad (9)$$

$$\bar{\boldsymbol{\Sigma}}_s = \frac{1}{\bar{N}_s} \sum_{d,n,t} \langle z_{n,s} \rangle \langle \phi_{d,t,n} \rangle (\mathbf{x}_{d,t} - \bar{\boldsymbol{\mu}}_s)^2 \quad (10)$$

For example, the Gaussian/Gaussian-Gamma model described earlier can be updated as follows:

$$q(\boldsymbol{\mu}_s, \boldsymbol{\lambda}_s) = \mathcal{NG} \left( \nu_0 + \bar{N}_s, \frac{\nu_0 \mathbf{m}_0 + \bar{N}_s \bar{\boldsymbol{\mu}}_s}{\nu_0 + \bar{N}_s}, u_0 + \frac{\bar{N}_s}{2}, \mathbf{k}_0 + \frac{1}{2} \left( \bar{N}_s \bar{\boldsymbol{\Sigma}}_s + \frac{\nu_0 \bar{N}_s}{\nu_0 + \bar{N}_s} (\bar{\boldsymbol{\mu}}_s - \mathbf{m}_0)^2 \right) \right) \quad (11)$$

Hyperparameters may be set manually, or optimized by minimizing the KL divergence from  $q$  to the posterior.

### 3.4 Semi-Markov Performance Sequence

The model presented previously implicitly assumes that the state duration of the performance sequence follows the

geometric distribution. In such a model, it is noted, especially in the context of audio-to-score alignment [4], that further improvement is possible by incorporating a more explicit duration probability using an extension of the HMM known as the hidden semi-Markov models [5, 20].

In this paper, we assume that every performance plays a *particular position* in the music composition with more-or-less the same tempo. Hence, we incorporate an explicit duration probability to the performance sequence, such that the duration of each state is concentrated about some average state duration common to each performance. To this end, we assume that for each state  $n$  of the performance sequence, the state duration  $l$  follows a Gaussian distribution concentrated about a common mean:

$$p(l|\gamma_n, c) = \mathcal{N}(\gamma_n, c\gamma_n^2) \quad (12)$$

We chose the Gaussian distribution due to convenience of inference. By setting  $c$  appropriately, we can provide a trade-off between the tendency for every piece to play in a same tempo sequence, and variation of tempo among different performances.

To incorporate such a duration probability in the performance sequence model, we augment the state space of the left-to-right Markov model of the performance sequence by a “count-down” variable  $l$  that indicates the number of frames remaining in the current state. Then, we assume that the maximum duration of each state is  $L$ , and represent each state of the performance  $\phi_{d,t}$  as a tuple  $(n, l) \in [1 \cdots N] \times [1 \cdots L]$ , *i.e.*,  $\phi_{d,t,n,l}$ . In this model, state  $(n, 1)$  transitions to  $(n+1, l)$  with probability  $p(l|\mu_{n+1}, c)$ , and state  $(n, l)$  for  $l > 1$  transitions to  $(n, l-1)$  with probability one. Finally, we constrain the terminal state to be  $(N, 1)$ . Note that  $\eta$  is no longer used because state duration is now described explicitly. The parameter  $\gamma_n$  can be optimized by maximum likelihood estimation of the second kind, to yield the following:

$$\gamma_n = \frac{\sum_{d,t,l} l \langle \phi_{d,t-1,n-1,1} \phi_{d,t,n,l} \rangle}{\sum_{d,t,l} \langle \phi_{d,t-1,n-1,1} \phi_{d,t,n,l} \rangle} \quad (13)$$

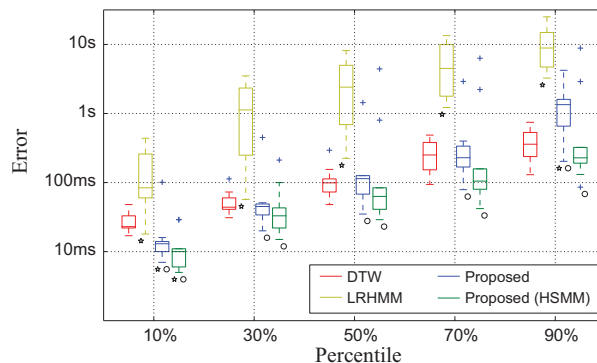
$c$  may be optimized in a similar manner, but we found that the method performs better when  $c$  is fixed to a constant.

## 4. EVALUATION

We conducted two experiments to assess our method. First, we tested the effectiveness of our method against existing methods that ignore the organization of music [13, 18]. Second, we tested the robustness of our method to the length of the latent composition, which we need to fix in advance.

### 4.1 Experimental Conditions

We prepared two to five recordings to nine pieces of Chopin’s *Mazurka* (Op. 6-4, 17-4, 24-2, 30-2, 33-2, 41-2, 63-3, 67-1, 68-3), totaling in 38 audio recordings. For each of the nine pieces, we evaluated the alignment using (1) DTW using path constraints in [21] that minimizes the net squared distance (denoted “DTW”), (2) left-to-right HMM to model musical audio as done in existing methods [13, 18] (denoted “LRHMM”), (3) proposed method (denoted “Pro-



**Figure 6.** Percentile of absolute alignment error. Asterisks indicate statistically significant difference over DTW ( $p=0.05$ ) and circles indicate statistically significant difference over LRHMM ( $p=0.05$ ), using Kruskal-Wallis H-test.

posed”), and (4) proposed method with semi-Markov performance sequence (denoted “Proposed (HSMM)”). For the feature sequence  $x_{d,t}$ , we employed the chroma vector [11] and half-wave rectified difference of the chroma ( $\Delta$  chroma), evaluated using a frame length of 8192 samples and a 20% overlap with a sampling frequency of 44.1kHz.

For the proposed method, the hyperparameters related to the latent common structure were set to  $\pi_0 = 0.1$  and  $\tau_{0,s,s'} = 0.9 + 10\delta(s, s')$ ; these parameters encourages sparsity of the initial state probability and the state transitions, while encouraging self-transitions. The parameters related to the observation were set to  $u_0 = \mathbf{k}_0 = 1$ ,  $\nu_0 = 0.1$  and  $m_0 = 0$ ; such a set of parameters encourages a sparse variance, and assumes that the mean is highly dispersed. Moreover, we used  $S = 100$  and  $N = 0.3 \min_d T_d$ . For the semi-Markov performance sequence model, we set  $c = 0.1$ . This corresponds to having a standard deviation of  $\gamma_n \sqrt{0.1}$ , or allowing the notes to deviate by a standard deviation of about 30%.

### 4.2 Experimental Results

We present below the evaluation of the alignment accuracy and the robustness to the length of the latent composition. On a workstation with Intel Xeon CPU (3.2GHz), our method takes about 3 minutes to process a minute of single musical audio.

#### 4.2.1 Alignment Accuracy

We compared the aligned data to that given by reverse conducting data of the Mazurka Project [1]. Figure 6 shows the absolute error percentile. The figure shows that our method (“Proposed”) performs significantly better than the existing method based on a LRHMM. This suggests that, for a generative model approach to alignment, not only is model of performance *difference* critical but also that of the *common* music that the performances play. We also note an improved performance of the semi-Markov model performance sequence (“Proposed (HSMM)”) over the Markovian model (“Proposed”).

Note that when using the same features and squared-error model, the semi-Markovian model performs better

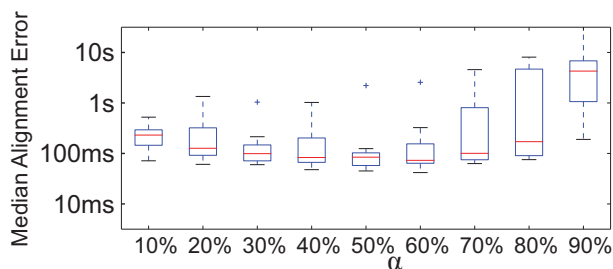


Figure 7. Median alignment error against  $\alpha$ .

than DTW. This result suggests that with appropriate structural and temporal models, a generative model approach is a viable alternative to audio alignment. The performance gain from Markov to semi-Markov model illuminates the forte of the generative model approach: temporal, spectral and structural constraints are mixed seamlessly to attain a trade-off among the trichotomy.

We note that our model is weak to compositional deviations, such as added ornaments and repeats because we assume every performance plays an identical composition. We observed that our method deals with an added note as a noise or a note that gets played very shortly by most of the audio signals, but neither captures the nature of added notes as structural deviations. Moreover, our method sometimes gets “trapped” in local optima, most likely due to the strong mutual dependency between the latent variables.

#### 4.2.2 Robustness to the Length of the Latent Composition

Since our method requires the user to set the length of latent composition  $N$ , we evaluated the quality of alignment as  $N$  is varied. To evaluate the performance of our method with different values of  $N$ , we evaluated the alignment of the proposed method when  $N$  is set to  $N = \alpha|T_{d=1}|$ , with  $\alpha$  ranging from  $\alpha = 0.1$  to  $\alpha = 0.9$  with an increment of 0.1. Figure 7 shows the median alignment error. We find that when  $\alpha$  is too small, when there is an insufficient number of states to describe a composition, the error increases. The error also increases when  $\alpha$  is too large, since the maximum total allowed deviation decreases (*i.e.*, to about  $(1 - \alpha)T_{d=1}$ ). However, outside such extremities, the performance is relatively stable for moderate values of  $\alpha$  around 0.5. This suggests that our method is relatively insensitive to a reasonable choice of  $N$ .

## 5. CONCLUSION

This paper presented an audio alignment method based on a probabilistic generative model. Based on the insight that a generative model of musical audio alignment should represent both the underlying musical composition and how it is performed by each audio signal, we formulated a unified generative model of musical composition and performance. The proposed generative model contributed to a significantly better alignment performance than existing methods. We believe that our contribution brings generative alignment on par with DTW-based alignment, opening door to alignment problem settings that require integration of various sources of uncertainties.

Future study includes incorporating better models of composition, performance and observation in our unified framework. In addition, inference over highly coupled hierarchical discrete state models is another future work.

**Acknowledgment:** This study was supported in part by JSPS KAKENHI 24220006 and 26700020.

## 6. REFERENCES

- [1] C. S. Sapp. Comparative analysis of multiple musical performances. In *ISMIR*, pages 2–5, 2007.
- [2] S. Miki, T. Baba, and H. Katayose. PEVI: Interface for retrieving and analyzing expressive musical performances with scape plots. In *SMC*, pages 748–753, 2013.
- [3] C. Fremerey, F. Kurth, M. Müller, and M. Clausen. A demonstration of the SyncPlayer system. In *ISMIR*, pages 131–132, 2007.
- [4] C. Raphael. A hybrid graphical model for aligning polyphonic audio with musical scores. In *ISMIR*, pages 387–394, 2004.
- [5] A. Cont. A coupled duration-focused architecture for real-time music-to-score alignment. *IEEE PAMI*, 32(6):974–987, 2010.
- [6] J. Paulus, M. Muller, and A. Klapuri. State of the art report: Audio-based music structure analysis. In *ISMIR*, pages 625–636, Aug. 2010.
- [7] S. A. Abdallah et al. Theory and evaluation of a Bayesian music structure extractor. In *ISMIR*, pages 420–425, 2005.
- [8] R. B. Dannenberg and N. Hu. Polyphonic audio matching for score following and intelligent audio editors. In *ICMC*, September 2003.
- [9] M. Grachten et al. Automatic alignment of music performances with structural differences. In *ISMIR*, pages 607–612, 2013.
- [10] N. Montecchio and A. Cont. A unified approach to real time audio-to-score and audio-to-audio alignment using sequential Monte-Carlo inference techniques. In *ICASSP*, pages 193–196, 2011.
- [11] T. Fujishima. Realtime chord recognition of musical sound: A system using Common Lisp Music. In *ICMC*, pages 464–467, 1999.
- [12] S. Ewert, M. Müller, and P. Grosche. High resolution audio synchronization using chroma onset features. In *ICASSP*, pages 1869–1872, 2009.
- [13] A. Maezawa and H. G. Okuno. Audio part mixture alignment based on hierarchical nonparametric Bayesian model of musical audio sequence collection. In *ICASSP*, pages 5232–5236, 2014.
- [14] T. Nakamura, E. Nakamura, and S. Sagayama. Acoustic score following to musical performance with errors and arbitrary repeats and skips for automatic accompaniment. In *SMC*, pages 200–304, 2013.
- [15] A. Maezawa et al. Polyphonic audio-to-score alignment based on Bayesian latent harmonic allocation hidden Markov model. In *ICASSP*, pages 185–188, 2011.
- [16] T. Otsuka et al. Incremental Bayesian audio-to-score alignment with flexible harmonic structure models. In *ISMIR*, pages 525–530, 2011.
- [17] C. Joder, S. Essid, and G. Richard. Learning optimal features for polyphonic audio-to-score alignment. *IEEE TASLP*, 21(10):2118–2128, 2013.
- [18] R. Miotto, N. Montecchio, and N. Orio. Statistical music modeling aimed at identification and alignment. In *AdMiRE*, pages 187–212, 2010.
- [19] M. J. Beal. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University College London, 2003.
- [20] S. Yu and H. Kobayashi. An efficient forward-backward algorithm for an explicit-duration hidden Markov model. *IEEE SPL*, 10(1):11–14, Jan 2003.
- [21] N. Hu, R. B. Dannenberg, and G. Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *WASPAA*, pages 185–188, 2003.

# AUTOMATIC SET LIST IDENTIFICATION AND SONG SEGMENTATION FOR FULL-LENGTH CONCERT VIDEOS

Ju-Chiang Wang<sup>1,2</sup>, Ming-Chi Yen<sup>1</sup>, Yi-Hsuan Yang<sup>1</sup>, and Hsin-Min Wang<sup>1</sup>

<sup>1</sup>Academia Sinica, Taipei, Taiwan

<sup>2</sup>University of California, San Diego, CA, USA

asriver.wang@gmail.com; {ymchiqq, yang, whm}@iis.sinica.edu.tw

## ABSTRACT

Recently, plenty of full-length concert videos have become available on video-sharing websites such as YouTube. As each video generally contains multiple songs, natural questions that arise include “what is the set list?” and “when does each song begin and end?” Indeed, many full concert videos on YouTube contain song lists and timecodes contributed by uploaders and viewers. However, newly uploaded content and videos of lesser-known artists typically lack this metadata. Manually labeling such metadata would be labor-intensive, and thus an automated solution is desirable. In this paper, we define a novel research problem, *automatic set list segmentation of full concert videos*, which calls for techniques in music information retrieval (MIR) such as audio fingerprinting, cover song identification, musical event detection, music alignment, and structural segmentation. Moreover, we propose a greedy approach that sequentially identifies a song from a database of studio versions and simultaneously estimates its probable boundaries in the concert. We conduct preliminary evaluations on a collection of 20 full concerts and 1,152 studio tracks. Our result demonstrates the effectiveness of the proposed greedy algorithm.

## 1. INTRODUCTION

In recent years, the practice of sharing and watching concert/performance footage on video sharing websites such as YouTube has grown significantly [12]. In particular, we have noticed that many concert videos consist of full-length, unabridged footage, featuring multiple songs. For example, the query “full concert” on YouTube returns a list of more than 2 million relevant videos. Before watching a full concert video, a viewer might like to know if the artist has performed the viewer’s favorite songs, and when are those song played in the video. Additionally, after watching a concert video, a viewer may want to know the song titles in order to locate the studio version.

To satisfy such a demand, the uploader or some viewers often post the “set list” with the timecode for each song,<sup>1</sup> so that other viewers can easily fast-forward to the desired song. This metadata can help viewers to navigate a long concert. From a technical point of view, it also helps to extract the live version of a song to enrich a music database. Such a database could be used to analyze performance style, to discover song transition [17], to train classifiers for visual event detection [28], or to generate multi-camera mashups and summaries of concert videos [22,27].

However, newly uploaded videos and those performed by less known artists typically lack this metadata, because manually identifying songs and song segmentation can be time consuming even for an expert. One reason for this is because live performances can differ substantially from the studio recordings. Another reason is that live performances often contain covers of songs by other artists. Even if the annotator can readily identify all songs, it is still necessary to go through the entire video to locate the precise times that each song begins and ends. Therefore, an automated method is desirable to annotate the rapidly growing volume of full-length concert videos available online.

The aim of this paper is threefold. First, we define a novel research problem, i.e. automatic set list segmentation of full concert videos, and discuss its challenges. Second, we propose a greedy approach to tackle the problem. Third, we construct a novel dataset designed for this task and suggest several evaluation methods.

### 1.1 Task Definition and Challenges

There are two sub-tasks for this research problem: *set list identification* and *song segmentation*. Given a full concert video, the former is to identify the sequence of song titles played in the concert based on a large collection of studio version tracks, assuming that no prior knowledge on the live performance of the artist(s) of the concert is available. The latter task is to estimate the boundaries of each identified song in the set list. This problem poses some interesting challenges as follows:

- A live song can be played in many different ways, e.g., by changing its timbre, tempo, pitch and structure, comparing to the corresponding studio version.

<sup>1</sup> A set list refers to a list of songs that a band/artist has played in a concert, and the timecode corresponds to the starting time of a song. Here is an example of full concert video with set list and timecodes on YouTube: <https://www.youtube.com/watch?v=qTOjini1ltQ>



© Ju-Chiang Wang, Ming-Chi Yen, Yi-Hsuan Yang, and Hsin-Min Wang.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Ju-Chiang Wang, Ming-Chi Yen, Yi-Hsuan Yang, and Hsin-Min Wang. “Automatic Set List Identification and Song Segmentation for Full-Length Concert Videos”, 15th International Society for Music Information Retrieval Conference, 2014.

Therefore, certain robustness should be considered.

- Live performances often feature transitions between consecutive songs, or even repeated oscillations between the sections of different songs, suggesting that one should identify songs on a small temporal scale.
- Concerts often feature sections with no reference in the collection of studio versions, such as intros, outros, solos, banter, transitions between songs, big rock endings, and applause, amongst others. Unexpected events such as broken instruments, sound system malfunctions, and interrupted songs can also be found. An ideal system should identify them or mark them as unknown songs/events, avoiding including them in a segmented song when appropriate.
- The artist may play cover songs from other artists partially or entirely throughout the concert, resulting in a much larger search space in the music database.
- The audio quality of user-contributed concert videos can vary significantly due to recording factors such as acoustic environment, position, device and user expertise [14]. The quality degradation can amplify the difficulty of the problem.

To tackle the above challenges, one may consider techniques for several fundamental problems in music information retrieval (MIR), such as audio fingerprinting/matching [3, 7], cover song identification [5, 24], audio quality assessment [14], musical event detection/tracking [32, 33], and music signal alignment and segmentation [18]. Therefore, automatic set list segmentation of full concert videos may present a new opportunity for MIR researchers to link music/audio technology to real-world applications.

## 1.2 Technical Contribution

Our technical contribution lies in the development of a greedy approach that incorporates three components: segmentation, song identification, and alignment (see Section 3). This approach provides a basic view as a baseline towards future advance. Starting from the beginning of the concert, our approach first identifies the candidate songs for a “probe excerpt” of the concert based on segmented music signals. Then, it estimates the most likely song title and boundaries of the probe excerpt based on dynamic time warping (DTW) [18]. This sequential process is repeated until the entire concert video has been processed. To evaluate the proposed algorithm, we collect 20 full concerts and 1,152 studio tracks from 10 artists (see Section 4). Moreover, we suggest three performance metrics for this task (see Section 5). Finally, we demonstrate the effectiveness of the proposed approach and observe that cover song identification works much better than audio fingerprinting for identifying the songs in a live performance (see Section 5).

## 2. RELATED WORK

According to a recent user study, YouTube was the second most preferred online music streaming service by users in 2012, just behind Pandora [12]. These community-contributed concert videos have been extensively studied in the

multimedia community. Most existing works focus on handling the visual content of the concert videos [1, 10, 22, 27, 28]. Relatively little attention, however, has been paid in the MIR community to study the audio content of this type of data. Related work mainly focused on low-level audio signal processing for tasks such as audio fingerprint-based synchronization and alignment for concert video organization [9, 11, 29], and audio quality ranking for online concert videos [14]. More recently, Rafii *et al.* proposed a robust audio fingerprinting system to identify a live music fragment [23], without exploring full-length concert videos and song segmentation. To gain deeper understanding of the content and context of live performance, our work represents an early attempt to use the full concert video data.

We note that our work is also related to PHENICX [6], an ongoing project which aims at enriching the user experience of watching classical music concerts via state-of-the-art multimedia and Internet technologies. With a system for automatic set list segmentation of full concert videos, one could index a large amount of online musical content, extracting information that helps link live performance to the associated video content.

Aside from potential applications, the technical development of our work is highly motivated by several signal matching-based music retrieval problems, which can be categorized into audio fingerprinting (AF) [3, 30], audio matching [21], and cover song identification (CSID) [5, 24], according to their *specificities* and *granularity* [4, 7]. An AF system retrieves the exact audio piece that is the source of a query audio fragment. Audio matching is defined as the task of retrieving from a database all the audio fragments that are musically relevant to a query fragment. In contrast, CSID aims at identifying different renditions of a music piece in the track level (instead of fragment-level). Unlike AF which usually holds robustness to any noises that may apply on the same rendition of a song, audio matching and CSID should handle the musically motivated variations occurring in different performances or arrangements of a music piece [7].

## 3. PROPOSED GREEDY APPROACH

The proposed approach is outlined in Algorithm 1. It employs an intuitive greedy strategy that recursively probes an excerpt  $X$  from the beginning of the unprocessed concert  $Z$ , identifies  $K$  song candidates ( $K = 5$ ) from the studio database  $\mathcal{D}$ , selects the most probable song title  $s^*$ , estimates the boundaries  $(i, j)$  of  $s^*$  in  $X$ , and finally removes  $s^*$  from  $\mathcal{D}$  and  $X(1 : j)$  from  $Z$ . The process stops when the unprocessed portion of the input concert is shorter than a pre-defined threshold  $\tau$ . We make the following assumptions while developing Algorithm 1: 1) the performer plays nearly the entire part of a song rather than a certain small portion of the song, 2) a song in the studio database is performed at most once in a concert, and 3) the concert contains only songs from the same artist without covers. In practice, the artist of a concert can be easily known from the video title. Therefore, we only take the studio tracks of the artist to construct  $\mathcal{D}$ . More details are given below.



**Algorithm 1:** Set list identification & segmentation

---

**Input:** A concert  $Z$ ; studio track database  $\mathcal{D}$ ; probe length  $l$ ; end length  $\tau$ ; candidate number  $K$ ;  
**Output:** Song list  $\mathcal{S}$ ; boundary set  $\mathcal{B}$ ;

- 1  $\mathcal{S} \leftarrow \emptyset$ ;  $\mathcal{B} \leftarrow \emptyset$ ;
- 2 **while**  $\text{length}(Z) > \tau$  **do**
- 3      $X \leftarrow Z(1 : l)$ , if  $l > \text{length}(Z)$ ,  $l = \text{length}(Z)$ ;
- 4      $\{s_k\}_{k=1}^K \leftarrow$  identify the  $K$  most probable songs that match  $X$ , based on the thumbnails of  $\mathcal{D}$ ;
- 5      $\{s^*, (i, j)\} \leftarrow$  select the best song from  $\{s_k\}_{k=1}^K$  and estimate its boundaries on  $X$ , based on the complete track of  $\mathcal{D}$ ;
- 6      $\mathcal{S} \leftarrow \mathcal{S} + s^*$ ;  $\mathcal{B} \leftarrow \mathcal{B} + (i, j)$ ;
- 7      $\mathcal{D} \leftarrow \mathcal{D} - s^*$ ;  $Z \leftarrow Z - X(1 : j)$ ;
- 8 **end**

---

### 3.1 Segmentation

In our original design, we adopt music segmentation techniques to pre-process both the concert and every studio track in the database. This enhances the robustness to variation of song structure for the music matching and identification processes. However, operating on fine-grained segments of the concert significantly increases the computational time of the algorithm. Therefore, we make heuristic modifications to gain more efficiency as follows.

First, we segment a sufficiently long probe excerpt from the beginning of an unprocessed concert that could include the first entire song played in the unprocessed concert, without involving any musically motivated segmentation. Ideally, we hope the probe length  $l$  is longer than the exact song  $s^*$  plus the events prior to  $s^*$  (e.g., banter, applause). In the experiment, we will compare different settings of  $l = \alpha \times \mu$ , where  $\alpha$  is the parameter and  $\mu$  the mean length of all studio tracks in the database.

Second, each studio track in the database is represented by its thumbnail for better efficiency in the later song candidate identification stage. Similar idea has been introduced by Grosche *et al.* [8]. We develop a simple method analogous to [15] based on structural segmentation. Segmentino [2, 16] is utilized to discover the musically homogeneous sections marked by structure labels such as ‘A,’ ‘B,’ and ‘N’ for each studio track. We compute a weighted factor  $\gamma$  that jointly considers the repetition count and average segment length for each label. The longest segment of the label that has the largest  $\gamma$  is selected as the thumbnail.

### 3.2 Song Candidate Identification

Song candidate identification uses the probe excerpt as a query and ranks the studio thumbnails in the database. We employ two strategies for the identifier: audio fingerprinting (AF) and cover song identification (CSID). For simplicity, we employ existing AF and CSID methods in this work. For future work, it might be more interesting to integrate the identifier with the subsequent boundary estimator.

For AF, we implement the identifier using the widely-known landmark-based approach proposed in [31]. It ex-

tracts prominent peaks (a.k.a. *landmarks*) from the magnitude spectrogram of a reference track (e.g. a studio version) and characterizes each pair of landmarks by the frequencies of the landmarks and the time in between them, which provide indices to a hash table that allows fast retrieval of similarity information [30]. For a query (e.g. a probe excerpt), we see whether there are sufficient number of matched landmarks between the query and a reference track by looking up the hash table. If the query track is a noisy version of the reference track, this approach is likely to perform fairly well, because the landmarks are most likely to be preserved in noise and distortion.

For CSID, we implement the identifier mainly based on the *chroma DCT-reduced log pitch* (CRP) features [19] and the *cross recurrence quantification* (CRQ) approach [25], which correspond to two major components in a state-of-the-art CSID system [26]. Specifically, we first extract the frame-based CRP features for the probe excerpt and each studio track by the Chroma Toolbox [20]. Then, we determine the key transposition using the optimal transposition index (OTI) [25]. To apply CRQ, we follow the standard procedures [25], including constructing the delay coordinate state space vectors, computing the cross recurrence plot, deriving the  $Q_{\max}$  score, and performing normalization on the scores across the database. This CSID system (cf. CYWW1) has led to performance comparable to the state-of-the-art systems in the MIREX audio cover song identification task (e.g., on Sapp’s Mazurka Collection).<sup>2</sup>

### 3.3 Song Selection and Boundary Estimation

The next step is to select the most probable song  $k^*$  from the top  $K$  studio song candidates,  $\{Y_k\}_{k=1}^K$ , and at the same time estimate its boundaries on the probe excerpt  $X$ . Accordingly, our goal is to find a  $Y_k$  and the corresponding subsequence  $X^* = X(i^* : j^*)$  that results in the best matching between  $Y_k$  and  $X^*$ , where  $1 \leq i^* < j^* \leq N$ . Such process is based on the DTW alignment between  $X$  and each  $Y_k$ , as presented in Algorithm 2.

Let  $X = \{x_1, \dots, x_N\}$  and denote the complete track of  $Y_k$  as  $Y^k = \{y_1, \dots, y_M\}$ , where  $x_i$  and  $y_i$  represent the frame-based CRP vectors and  $N > M$ . We compute the cost by the negative cosine similarity of CRP between two frames after the OTI key transposition. One can observe that Algorithm 2 includes two sub-procedures of one-side boundary estimation (cf. Algorithm 3). It first executes Algorithm 3 to search for the end boundary  $j'$  on  $X$  and then *reverses* the search from  $j'$  for the start boundary  $i'$  using Algorithm 3 with the cost matrix rotated by 180 degrees. We follow the standard procedure to compute the accumulated cost matrix  $D$  in [18]. Then, Algorithm 3 searches from  $D(\frac{N}{2} + 1, M)$  to  $D(N, M)$  for the minimum *average cost* of DTW alignments, denoted by  $\delta_k^*$ , where the average cost is defined as the accumulated cost divided by the length of its optimal warping path (OWP). The frame index of  $\delta_k^*$  is set as the boundary.

After the  $K$  candidates are processed, we pick the one

<sup>2</sup> [http://www.music-ir.org/mirex/wiki/2013:Audio\\_Cover\\_Song\\_Identification\\_Results](http://www.music-ir.org/mirex/wiki/2013:Audio_Cover_Song_Identification_Results)

**Algorithm 2:** Boundaries & average cost estimation

---

**Input:** Concert excerpt  $X$ ; a studio track  $Y'$ ;  
**Output:** Boundary pair  $(i', j')$ ; average cost  $\delta$ ;

- 1  $C \leftarrow N$ -by- $M$  cost matrix between  $X$  and  $Y'$ ;
- 2  $(j', \emptyset) \leftarrow$  one-side boundary estimation on  $C$ ;
- 3  $C \leftarrow$  rotate  $C(1 : j', 1 : M)$  by 180 degrees;
- 4  $(index, \delta) \leftarrow$  one-side boundary estimation on  $C$ ;
- 5  $i' \leftarrow j' - index + 1$ ;

---

**Algorithm 3:** One-side boundary estimation

---

**Input:** Cost matrix  $C$ ;  
**Output:** Boundary  $\beta$ ; average cost  $\delta$ ;

- 1  $D \leftarrow$  accumulated cost matrix from  $C(1, 1)$ ;
- 2 **for**  $1 \leftarrow i$  **to**  $\frac{N}{2}$  **do**
- 3      $p^* \leftarrow$  compute the OWP of  $D(1 : \frac{N}{2} + i, 1 : M)$ ;
- 4      $\Delta(i) \leftarrow D(\frac{N}{2} + i, M) / \text{length}(p^*)$ ;
- 5 **end**
- 6  $(\delta, index) \leftarrow$  the minimum value and its index of  $\Delta$ ;
- 7  $\beta \leftarrow index + \frac{N}{2}$ ;

---

with the lowest average cost,  $k^* = \arg \min_k \{\delta_k\}_{k=1}^K$ , and set the boundary pair as  $(i'_{k^*}, j'_{k^*})$ . In other words, we re-rank the top  $K$  candidates according to the results of Algorithm 2, based on the content of the complete studio tracks.

## 4. DATA COLLECTION

We collect 20 popular full concert videos (from the first few responses to the query “full concert” to Youtube) and the associated set lists and timecodes from YouTube. Therefore, the music genre is dominated by pop/rock. We manually label the start and end boundaries of each song based on the timecodes, as a timecode typically corresponds to the start time of a song and may not be always accurate. There are 10 artists. For each artist, we collect as many studio tracks as possible including the songs performed in the collected concerts to form the studio database. On average, we have 115.2 studio version tracks for each artist, and each full concert video contains 16.2 live version tracks. Table 1 summarizes the dataset.

## 5. EVALUATION

### 5.1 Pilot Study on Set List Identification

We conduct a pilot study to investigate which strategy (i.e., AF or CSID) performs better for set list identification, assuming that the song segmentation is perfect. For simplicity, we extract all the songs from the concert videos according to the manually labeled boundaries and treat each entire live song as a query (instead of thumbnail). We use mean average precision (MAP) and precision@1 with respect to the studio database as the performance metrics. We also perform random permutation ten times for each query to generate a lower bound performance, denoted by ‘Random.’ One can observe from Table 2 that CSID performs significantly better than AF in our evaluation, show-

ID	Artist Name	Concerts	Studio Tracks
1	Coldplay	2	96
2	Maroon 5	3	62
3	Linkin’ Park	4	68
4	Muse	2	100
5	Green Day	2	184
6	Guns N’ Roses	2	75
7	Metallica	1	136
8	Bon Jovi	1	205
9	The Cranberries	2	100
10	Placebo	1	126

**Table 1.** The full concert dataset.

Method	MAP	Precision@1
AF	0.060	0.048
CSID	<b>0.915</b>	<b>0.904</b>
Random	0.046	0.009

**Table 2.** Result for live song identification.

ing that the landmark-based AF approach does not work well for live version identification. This confirms our intuition as live rendition can be thought of as a cover version of the studio version [5]. In consequence, we use CSID as the song candidate identifier in the following experiments.

### 5.2 Performance Metrics

We use the following performance metrics for set list identification and song segmentation: *edit distance* (ED), *boundary deviation* (BD), and *frame accuracy* (FA). The first metric ED is originally used to estimate the dissimilarity of two strings and has been adopted in numerous MIR tasks [13]. We compute the ED between an output song sequence (a list of song indices) and the ground truth counterpart via dynamic programming. The weights for insertion, deletion, and substitution are all set to 1. ED can only evaluate the accuracy of set list identification.

The second metric BD directly measures the absolute deviation in second between the estimated boundary and that of the ground truth for only each correctly identified song, ignoring those wrongly inserted songs in the output set list, as they are not presented in the ground truth. Therefore, the average BD of a concert reflects the accuracy of song segmentation but not set list identification.

The last metric, FA, which has been used in tasks such as melody extraction, represents the accuracy at the frame-level (using non-overlapped frame with length 200 ms). Throughout the concert, we mark the frames between the start and end boundaries of each song by its song index and otherwise by ‘x’ (belonging to no song). Then, we calculate the percentage of correct frames (the intersection rate) by comparing the output frame sequence with the ground truth counterpart. Therefore, FA can reflect the accuracy of both set list identification and song segmentation.

### 5.3 Baseline Approach

To study the effectiveness of the song selection and boundary estimation algorithms (see Section 3.3), we construct a baseline approach using Algorithm 1 without Algorithms 2 and 3. Specifically, we select the song  $s^*$  with the largest

ID	A	SG	SO	ED <sup>b</sup>	sBD <sup>b</sup>	eBD <sup>b</sup>	FA
1	7	20	15	17	6.5	89.1	0.317
2	3	17	17	4	3.3	12.3	0.786
3	1	15	15	3	27.2	33.2	0.744
4	8	23	25	14	8.8	66.8	0.441
5	10	19	18	5	11.5	27.8	0.641
6	6	10	11	1	19.1	22.8	0.875
7	2	10	10	6	28.2	39.1	0.428
8	3	22	22	9	28.2	39.6	0.610
9	6	20	21	7	30.7	35.9	0.653
10	9	17	15	4	5.3	9.8	0.758
11	9	22	21	3	6	8.7	0.860
12	4	17	19	7	32.0	21.9	0.681
13	2	9	12	5	110	155	0.509
14	1	17	17	2	20.1	18.4	0.777
15	2	11	11	7	50.9	72.9	0.393
16	3	17	20	9	36.9	24.7	0.544
17	4	13	11	4	48.1	94.3	0.626
18	3	23	22	10	10	34.8	0.636
19	5	7	7	3	42.4	13.6	0.584
20	5	15	13	9	42.4	36.6	0.465
AVG( $\alpha=1.5$ )	16.2	16.1	6.5	23.4	42.9	0.616	
AVG( $\alpha=1.2$ )	16.2	18	7.3	25.7	57.3	0.582	
AVG( $\alpha=1.8$ )	16.2	14.6	8.4	29.3	45.3	0.526	
Baseline	16.2	19.7	8.9	229	241	0.434	

**Table 3.** Result of the greedy approach with  $\alpha=1.5$  for the 20 full concerts and their average (AVG) performance. While ‘AVG ( $\alpha=1.2$  or  $\alpha=1.8$ )’ only shows the average performance with different  $l$  settings. ‘Baseline’ represents the average performance of the approach in Section 5.3. Additional abbreviations: A (Artist ID), SG (number of Songs in the Ground truth set list), SO (number of Songs in the Output set list), sBD (start BD), and eBD (end BD). Symbol <sup>b</sup> marks the metrics that are the smaller the better.

CSID score on a probe excerpt. The start boundary is the start point of the probe excerpt, and the end boundary is the length( $s^*$ ). Then, we begin the next probe excerpt on a hop of  $0.1 \times \text{length}(s^*)$ .

#### 5.4 Result and Discussion

Table 3 shows the quantitative result of each concert, the average performance (AVG) with different values of  $l$ , and the average performance of Baseline. Figure 1 depicts the qualitative results of three concerts, including the best, medium, and the worst cases according to FA in Table 3.

The following observations can be made. First, the AVG performances of the complete approach are significantly better than those of Baseline in all metrics, demonstrating the effectiveness of Algorithms 2 and 3. Second, further comparison among AVG performances with different  $l$  settings shows that  $\alpha=1.5$  performs the best, revealing that live versions are likely longer than studio ones, but overly large  $l$  could yield more deletions, as observed by the smaller SO of  $\alpha=1.8$ . Third, on average our approach gives similar number of songs of a concert as that of ground truth (16.1 versus 16.2). Fourth, we find an interesting linkage between the result and the style of the live performance. For example, we find that our approach performed poorly for ‘Maroon 5’ ( $A=2$ ) and ‘Metallica’ ( $A=7$ ). As can be observed from the last two rows of Figure 1, Ma-

roon 5 tends to introduce several non-song sections such as jam and banter, which cannot be accurately modeled by our approach. They also like to make the live renditions different from their studio versions. On the other hand, we conjecture that the riffs in the heavy metal music such as Metallica may be the main reason degrading the performance of matching thumbnails by CSID, because such riffs lack long-term harmonic progressions. Fifth, the performance for ‘Bon Jovi’ ( $A=8$ ) is poor, possibly because of the relatively large quantity of studio tracks in the search space. Finally, owing to possible big rock endings or repetitive chorus in the live performance, our approach relatively cannot estimate accurate end boundary of the songs in a concert, as reflected by larger eBD than sBD. Our approach sometimes insert songs that are relatively short in length, as can be observed in Figure 1. The above two observations suggest that advanced methods (over Algorithm 3) for boundary estimation and regularizing the song length might be needed.

In short, while there is still much room for improvement, we find that the result of the proposed greedy approach is quite satisfactory in some cases (e.g., Concert 6 in Figure 1). The greedy approach is preliminary in nature. We believe that better result can be obtained by explicitly addressing the challenges described in Section 1.1.

## 6. CONCLUSION AND FUTURE DIRECTION

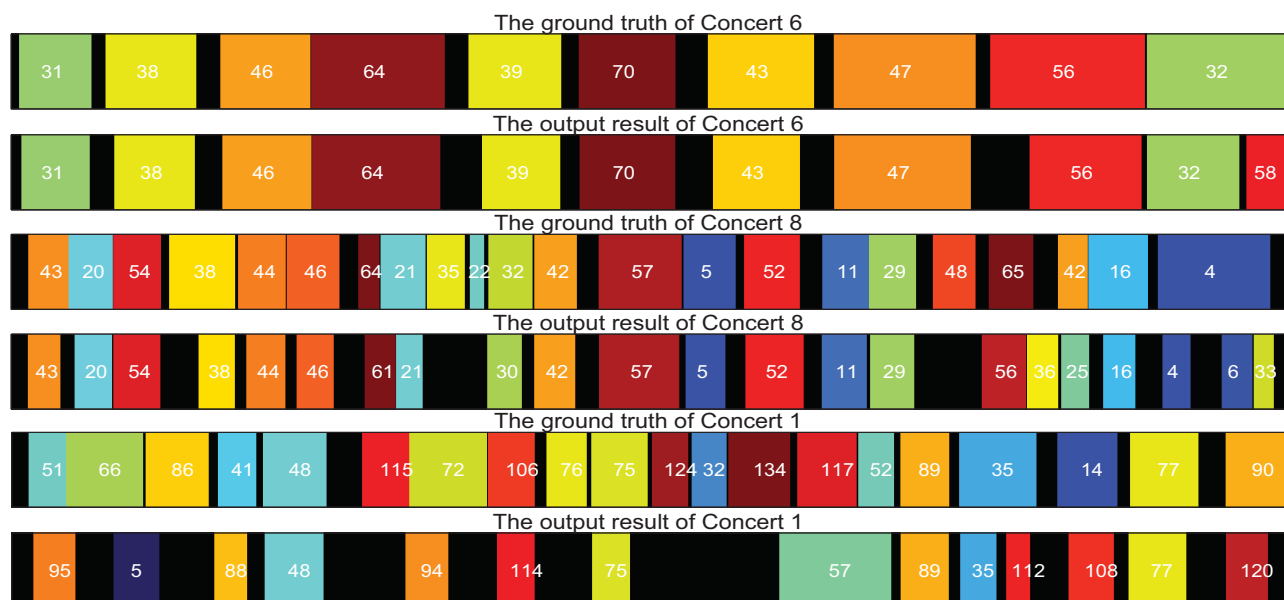
In this paper, we have proposed a novel MIR research problem with a new dataset and a new greedy approach to address the problem. We have also validated the effectiveness of the proposed approach via both quantitative and qualitative results. We are currently expanding the size of the dataset and conducting more in-depth signal-level analysis of the dataset. Due to the copyright issue on the studio track collection, however, it is not likely to distribute the dataset. We will propose this task to MIREX to call for more advanced approaches to tackle this problem.

## 7. ACKNOWLEDGEMENT

This work was supported by Academia Sinica–UCSD Postdoctoral Fellowship to Ju-Chiang Wang, and the Ministry of Science and Technology of Taiwan under Grants NSC 101-2221-E-001-019-MY3 and 102-2221-E-001-004-MY3.

## 8. REFERENCES

- [1] A. Bagri, F. Thudor, A. Ozerov, and P. Hellier. A scalable framework for joint clustering and synchronizing multi-camera videos. In *Proc. EUSIPCO*, 2013.
- [2] C. Cannam et al. MIREX 2013 entry: Vamp plugins from the centre for digital music. In *MIREX*, 2013.
- [3] P. Cano et al. A review of audio fingerprinting. *J. Sign. Process. Syst.*, 41(3):271–284, 2005.
- [4] M. A. Casey et al. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.
- [5] D. PW Ellis and G. E Poliner. Identifying ‘cover songs’ with chroma features and dynamic programming beat tracking. In *Proc. ICASSP*, pages IV–1429, 2007.



**Figure 1.** Qualitative result of three concerts, which represent the best (Concert 6, ‘Guns N’ Roses’), medium (Concert 8, ‘Linkin’ Park’), and worst (Concert 1, ‘Metallica’) output cases in the dataset. Black blocks correspond to no song. Different songs are marked by different colors. The number in a song block stands for the song index in the studio database. Note that Song 42 (‘Numb’) was sung twice in Concert 8, firstly by ‘Linkin’ Park’ and then by ‘featuring Jay-Z.’

- [6] E. Gómez et al. PHENICX: Performances as highly enriched and interactive concert experiences. In *Proc. SMC*, 2013.
- [7] P. Grosche, M. Müller, and J. Serrà. Audio content-based music retrieval. *Multimodal Music Processing*, 3:157–174, 2012.
- [8] P. Grosche, M. Müller, and J. Serrà. Towards cover group thumbnailing. In *Proc. ACM MM*, pages 613–616, 2013.
- [9] M. Guggenberger, M. Lux, and L. Boszormenyi. AudioAlign - Synchronization of A/V-streams based on audio data. In *Proc. IEEE ISM*, pages 382–383, 2012.
- [10] L. Guimarães et al. Creating personalized memories from social events: community-based support for multi-camera recordings of school concerts. In *Proc. ACM MM*, 2011.
- [11] L. Kennedy and M. Naaman. Less talk, more rock: Automated organization of community-contributed collections of concert videos. In *Proc. WWW*, pages 311–320, 2009.
- [12] J. H. Lee and N. M. Waterman. Understanding user requirements for music information services. In *Proc. ISMIR*, 2012.
- [13] Kjell Lemström. *String matching techniques for music retrieval*. Ph.D. Thesis, University of Helsinki, 2000.
- [14] Z. Li, J.-C. Wang, J. Cai, Z. Duan, H.-M. Wang, and Y. Wang. Non-reference audio quality assessment for online live music recordings. In *Proc. ACM MM*, pages 63–72, 2013.
- [15] B. Martin, P. Hanna, M. Robine, and P. Ferraro. Indexing musical pieces using their major repetition. In *Proc. JCDL*, pages 153–156, 2011.
- [16] M. Mauch, K. Noland, and S. Dixon. Using musical structure to enhance automatic chord transcription. In *Proc. ISMIR*, pages 231–236, 2009.
- [17] B. McFee and G. Lanckriet. The natural language of playlists. In *Proc. ISMIR*, pages 537–542, 2011.
- [18] M. Müller. *Information retrieval for music and motion*. 2007.
- [19] M. Müller and S. Ewert. Towards timbre-invariant audio features for harmony-based music. *IEEE Trans. Audio, Speech, and Lang. Process.*, 18(3):649–662, 2010.
- [20] M. Müller and S. Ewert. Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In *Proc. ISMIR*, pages 215–220, 2011.
- [21] M. Müller, F. Kurth, and M. Clausen. Audio matching via chroma-based statistical features. In *Proc. ISMIR*, 2005.
- [22] S. U. Naci and A. Hanjalic. Intelligent browsing of concert videos. In *Proc. ACM MM*, pages 150–151, 2007.
- [23] Z. Rafii, B. Coover, and J. Han. An audio fingerprinting system for live version identification using image processing techniques. In *Proc. ICASSP*, pages 644–648, 2014.
- [24] J. Serra, E. Gómez, and P. Herrera. Audio cover song identification and similarity: background, approaches, evaluation, and beyond. In *Advances in Music Information Retrieval*, pages 307–332. 2010.
- [25] J. Serra, X. Serra, and R. G Andrzejak. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11(9):093017, 2009.
- [26] J. Serrà, M. Zanin, and R. G Andrzejak. Cover song retrieval by cross recurrence quantification and unsupervised set detection. In *MIREX*, 2009.
- [27] P. Shrestha et al. Automatic mashup generation from multiple-camera concert recordings. In *Proc. ACM MM*, pages 541–550, 2010.
- [28] C. GM Snoek et al. The role of visual content and style for concert video indexing. In *Proc. ICME*, 2007.
- [29] K. Su, M. Naaman, A. Gurjar, M. Patel, and D. PW Ellis. Making a scene: alignment of complete sets of clips based on pairwise audio match. In *Proc. ICMR*, page 26, 2012.
- [30] A. Wang. An industrial strength audio search algorithm. In *Proc. ISMIR*, pages 7–13, 2003.
- [31] C.-C. Wang, J.-S. R. Jang, and W. Li. Speeding up audio fingerprinting over GPUs. In *Proc. ICALIP*, 2014.
- [32] J.-C. Wang, H.-M. Wang, and S.-K. Jeng. Playing with tagging: A real-time tagging music player. In *ICASSP*, 2012.
- [33] S.-Y. Wang, J.-C. Wang, Y.-H. Yang, and H.-M. Wang. Towards time-varying music auto-tagging based on CAL500 expansion. In *Proc. ICME*, 2014.

# ON INTER-RATER AGREEMENT IN AUDIO MUSIC SIMILARITY

Arthur Flexer

Austrian Research Institute for Artificial Intelligence (OFAI)

Freyung 6/6, Vienna, Austria

arthur.flexer@ofai.at

## ABSTRACT

One of the central tasks in the annual MIREX evaluation campaign is the "Audio Music Similarity and Retrieval (AMS)" task. Songs which are ranked as being highly similar by algorithms are evaluated by human graders as to how similar they are according to their subjective judgment. By analyzing results from the AMS tasks of the years 2006 to 2013 we demonstrate that: (i) due to low inter-rater agreement there exists an upper bound of performance in terms of subjective gradings; (ii) this upper bound has already been achieved by participating algorithms in 2009 and not been surpassed since then. Based on this sobering result we discuss ways to improve future evaluations of audio music similarity.

## 1. INTRODUCTION

Probably the most important concept in Music Information Retrieval (MIR) is that of *music similarity*. Proper modeling of music similarity is at the heart of every application allowing automatic organization and processing of music databases. Consequently, the "Audio Music Similarity and Retrieval (AMS)" task has been part of the annual "Music Information Retrieval Evaluation eXchange" (MIREX<sup>1</sup>) [2] since 2006. MIREX is an annual evaluation campaign for MIR algorithms allowing for a fair comparison in standardized settings in a range of different tasks. As such it has been of great value for the MIR community and an important driving force of research and progress within the community. The essence of the AMS task is to have human graders evaluate pairs of query/candidate songs. The query songs are randomly chosen from a test database and the candidate songs are recommendations automatically computed by participating algorithms. The human graders rate whether these query/candidate pairs "sound similar" using both a BROAD ("not similar", "somewhat similar", "very similar") and a FINE score (from 0 to 10 or from 0 to 100, depending on the year the AMS task was held, indicating degrees of similarity ranging from failure to perfection).

<sup>1</sup><http://www.music-ir.org/mirex>



© Arthur Flexer.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Arthur Flexer. "On inter-rater agreement in audio music similarity", 15th International Society for Music Information Retrieval Conference, 2014.

It is precisely this general notion of "sounding similar" which is the central point of criticism in this paper. A recent survey article on the "neglected user in music information retrieval research" [13] has made the important argument that users apply very different, individual notions of similarity when assessing the output of music retrieval systems. It seems evident that music similarity is a multi-dimensional notion including timbre, melody, harmony, tempo, rhythm, lyrics, mood, etc. Nevertheless most studies exploring music similarity within the field of MIR, which are actually using human listening tests, are restricted to overall similarity judgments (see e.g. [10] or [11, p. 82]) thereby potentially blurring the many important dimensions of musical expression. There is very little work on what actually are important dimensions for humans when judging music similarity (see e.g. [19]).

This paper therefore presents a meta analysis of all MIREX AMS tasks from 2006 to 2013 thereby demonstrating that: (i) there is a low inter-rater agreement due to the coarse concept of music similarity; (ii) as a consequence there exists an upper bound of performance that can be achieved by algorithmic approaches to music similarity; (iii) this upper bound has already been achieved years ago and not surpassed since then. Our analysis is concluded by making recommendations on how to improve future work on evaluating audio music similarity.

## 2. RELATED WORK

In our review on related work we focus on papers directly discussing results of the AMS task thereby addressing the problem of evaluation of audio music similarity.

After the first implementation of the AMS task in 2006, a meta evaluation of what has been achieved has been published [8]. Contrary to all subsequent editions of the AMS task, in 2006 each query/candidate pair was evaluated by three different human graders. Most of the study is concerned with the inter-rater agreement of the BROAD scores of the AMS task as well as the "Symbolic Melodic Similarity (SMS)" task, which followed the same evaluation protocol. To access the amount of agreement, the authors use Fleiss's Kappa [4] which ranges between 0 (no agreement) and 1 (perfect agreement). Raters in the AMS task achieved a Kappa of 0.21 for the BROAD task, which can be seen as a "fair" level of agreement. Such a "fair" level of agreement [9] is given if the Kappa result is between 0.21 and 0.40, therefore positioning the

BROAD result at the very low end of the range. Agreement in SMS is higher (Kappa of 0.37), which is attributed to the fact that the AMS task is "less well-defined" since graders are only informed that "works should sound similar" [8]. The authors also note that the FINE scores for query/candidate pairs, which have been judged as "somewhat similar", show more variance than the one judged as "very" or "not" similar. One of the recommendations of the authors is that "evaluating more queries and more candidates per query would more greatly benefit algorithm developers" [8], but also that a similar analysis of the FINE scores is also necessary.

For the AMS task 2006, the distribution of differences between FINE scores of raters judging the same query/candidate pair has already been analysed [13]. For over 50%, the difference between rater FINE scores is larger than 20. The authors also note that this is very problematic since the difference between the best and worst AMS 2012 systems was just 17.

In yet another analysis of the AMS task 2006, it has been reported [20] that a range of so-called "objective" measures of audio similarity are highly correlated with subjective ratings by human graders. These objective measures are based on genre information, which can be used to automatically rank different algorithms producing lists of supposedly similar songs. If the genre information of the query and candidate songs are the same, a high degree of audio similarity is achieved since songs within a genre are supposed to be more similar than songs from different genres. It has therefore been argued that, at least for large-scale evaluations, these objective measures can replace human evaluation [20]. However, this is still a matter of controversy within the music information retrieval community, see e.g. [16] for a recent and very outspoken criticism of this position.

A meta study of the 2011 AMS task explored the connection between statistical significance of reported results and how this relates to actual user satisfaction in a more realistic music recommendation setting [17]. The authors made the fundamental clarification that the fact of observing statistically significant differences is not sufficient. More important is whether this difference is noticeable and important to actual users interacting with the systems. Whereas a statistically significant difference can always be achieved by enlarging the sample size (i.e. the number of query/candidate pairs), the observed difference can nevertheless be so small that it is of no importance to users. Through a crowd-sourced user evaluation, the authors are able to show that there exists an upper bound of user satisfaction with music recommendation systems of about 80%. More concretely, in their user evaluation the highest percentage of users agreeing that two systems "are equally good" never exceeded 80%. This upper bound cannot be surpassed since there will always be users that disagree concerning the quality of music recommendations. In addition the authors are able to demonstrate that differences in FINE scores, which are statistically significant, are so small that they make no practical difference for users.

### 3. DATA

For our meta analysis of audio music similarity (AMS) we use the data from the "Audio Music Similarity and Retrieval" tasks from 2006 to 2013<sup>2</sup> within the annual MIREX [2] evaluation campaign for MIR algorithms.

For the AMS 2006 task, 5000 songs were chosen from the so-called "uspop", "uscrap" and "cover song" collections. Each of the participating 6 system then returned a 5000x5000 AMS distance matrix. From the complete set of 5000 songs, 60 songs were randomly selected as queries and the first 5 most highly ranked songs out of the 5000 were extracted for each query and each of the 6 systems (according to the respective distance matrices). These 5 most highly ranked songs were always obtained after filtering out the query itself, results from the same artist (i.e. a so-called artist filter was employed [5]) and members of the cover song collection (since this was essentially a separate task run together with the AMS task). The distribution for the 60 chosen random songs is highly skewed towards rock music: 22 ROCK songs, 6 JAZZ, 6 RAP&HIPHOP, 5 ELECTRONICA&DANCE, 5 R&B, 4 REGGAE, 4 COUNTRY, 4 LATIN, 4 NEWAGE. Unfortunately the distribution of genres across the 5000 songs is not available, but there is some information concerning the "excessively skewed distribution of examples in the database (roughly 50% of examples are labeled as Rock/Pop, while a further 25% are Rap & Hip-Hop)"<sup>3</sup>. For each query song, the returned results (candidates) from all participating systems were evaluated by human graders. For each individual query/candidate pair, three different human graders provided both a FINE score (from 0 (failure) to 10 (perfection)) and a BROAD score (not similar, somewhat similar, very similar) indicating how similar the songs are in their opinion. This altogether gives  $6 \times 60 \times 5 \times 3 = 5400$  human FINE and BROAD gradings. Please note that since some of the query/candidate pairs are identical for some algorithms (i.e. different algorithms returned identical candidates) and since such identical pairs were not graded repeatedly, the actual number of different FINE and BROAD gradings is somewhat smaller.

Starting with the AMS task 2007, a number of small changes to the overall procedure was introduced. Each participating algorithm was given 7000 songs chosen from the "uspop", "uscrap" and "american" "classical" and "sundry" collections. Therefore there is only a partial overlap in music collections ("uspop" and "uscrap") compared to AMS 2006. From now on 30 second clips instead of the full songs were being used both as input to the algorithms and as listening material for the human graders. For the subjective evaluation of music similarity, from now on 100 query songs were randomly chosen representing the 10 genres found in the database (i.e., 10 queries per genre). The whole database consists of songs from equally sized genre groups: BAROQUE, COUNTRY, EDANCE,

<sup>2</sup>The results and details can be found at: [http://www.music-ir.org/mirex/wiki/MIREX\\_HOME](http://www.music-ir.org/mirex/wiki/MIREX_HOME)

<sup>3</sup>This is stated in the 2006 MIREX AMS results: [http://www.music-ir.org/mirex/wiki/2006:Audio\\_Music\\_Similarity\\_and\\_Retrieval\\_Results](http://www.music-ir.org/mirex/wiki/2006:Audio_Music_Similarity_and_Retrieval_Results)

JAZZ, METAL, RAPHIPHOP, ROCKROLL, ROMANTIC, BLUES, CLASSICAL. Therefore there is only a partial overlap of genres compared to AMS 2006 (COUNTRY, EDANCE, JAZZ, RAPHIPHOP, ROCKROLL). As with AMS 2006, the 5 most highly ranked songs were then returned per query as candidates (after filtering for the query song and songs from the same artist). For AMS tasks 2012 and 2013, 50 instead of 100 query songs were chosen and 10 instead of 5 most highly ranked songs returned as candidates.

Probably the one most important change to the AMS 2006 task is the fact that from now on every query/candidate pair was only being evaluated by a single user. Therefore the degree of inter-rater agreement cannot be analysed anymore. For every AMS task, the subjective evaluation therefore results in  $a \times 100 \times 5$  human FINE and BROAD gradings, with  $a$  being the number of participating algorithms, 100 the number of query songs and 5 the number of candidate songs. For AMS 2012 and 2013 this changed to  $a \times 50 \times 10$ , which yields the same overall number. These changes are documented on the respective MIREX websites, but also in a MIREX review article covering all tasks of the campaign [3]. For AMS 2007 and 2009, the FINE scores range from 0 to 10, from AMS 2010 onwards from 0 to 100. There was no AMS task in MIREX 2008.

#### 4. RESULTS

In our meta analysis of the AMS tasks from years 2006 to 2013, we will focus on the FINE scores of the subjective evaluation conducted by the human graders. The reason is that the FINE scores provide more information than the BROAD scores which only allow for three categorical values. It has also been customary for the presentation of AMS results to mainly compare average FINE scores for the participating algorithms.

##### 4.1 Analysis of inter-rater agreement

Our first analysis is concerned with the degree of inter-rater agreement achieved in the AMS task 2006, which is the only year every query/candidate pair has been evaluated by three different human graders. Previous analysis of AMS results has concentrated on BROAD scores and used Fleiss's Kappa as a measure of agreement (see Section 2). Since the Kappa measure is only defined for the categorical scale, we use the Pearson correlation  $\rho$  between FINE scores of pairs of graders. As can be seen in Table 1, the average correlations range from 0.37 to 0.43. Taking the square of the observed values of  $\rho$ , we can see that only about 14 to 18 percent of the variance of FINE scores observed in one grader can be explained by the values observed for the respective other grader (see e.g. [1] on  $\rho^2$  measures). Therefore, this is the first indication that agreement between raters in the AMS task is rather low.

Next we plotted the average FINE score of a rater  $i$  for all query/candidate pairs, which he or she rated within a certain interval of FINE scores  $v$ , versus the average

	grader1	grader2	grader3
grader1	1.00	0.43	0.37
grader2		1.00	0.40
grader3			1.00

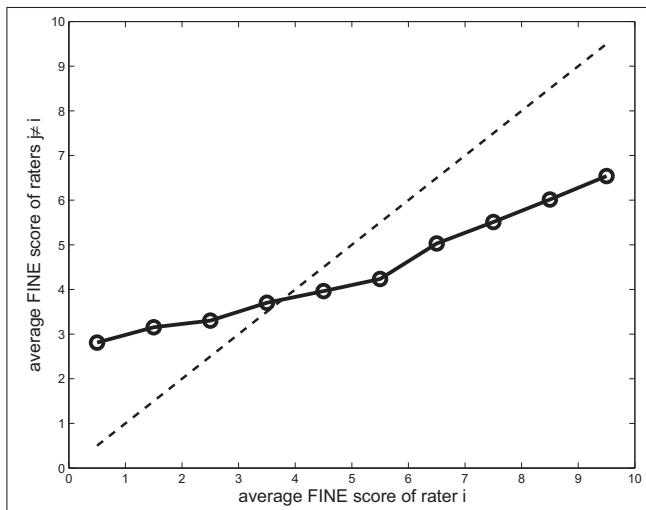
**Table 1.** Correlation of FINE scores between pairs of human graders.

	grader1	grader2	grader3
grader1	9.57	6.66	5.99
grader2	6.60	9.55	6.67
grader3	6.62	6.87	9.69

**Table 2.** Pairwise inter-rater agreement for FINE scores from interval  $v = [9, 10]$ .

FINE scores achieved by the other two raters  $j \neq i$  for the same query/candidate pairs. We therefore explore how human graders rate pairs of songs which another human grader rated at a specific level of similarity. The average results across all raters and for intervals  $v$  ranging from  $[0, 1)$ ,  $[1, 2)$ ... to  $[9, 10]$  are plotted in Figure 1. It is evident that there is a considerable deviation from the theoretical perfect agreement which is indicated as a dashed line. Pairs of query/candidate songs which are rated as being very similar (FINE score between 9 and 10) by one grader are on average only rated at around 6.5 by the two other raters. On the other end of the spectrum, query/candidate pairs rated as being not similar at all (FINE score between 0 and 1) receive average FINE scores of almost 3 by the respective other raters. The degree of inter-rater agreement for pairs of raters at the interval  $v = [9, 10]$  is given in Table 2. There are 333 pairs of songs which have been rated within this interval. The main diagonal gives the average rating one grader gave to pairs of songs in the interval  $v = [9, 10]$ . The off-diagonal entries show the level of agreement between different raters. As an example, query/candidate pairs that have been rated between 9 and 10 by *grader1* have received an average rating of 6.66 by *grader2*. The average of these pairwise inter-rater agreements given in Table 2 is 6.54 and is an upper bound for the average FINE scores of the AMS task 2006. This upper bound is the maximum of average FINE scores that can be achieved within such an evaluation setting. This upper bound is due to the fact that there is a considerable lack of agreement between human graders. What sounds very similar to one of the graders will on average not receive equally high scores by other graders.

The average FINE score achieved by the best participating system in AMS 2006 (algorithm EP) is  $4.30 \pm 8.8$  (mean  $\pm$  variance). The average upper bound inter-rater grading is  $6.54 \pm 6.96$ . The difference between the best FINE scores achieved by the system EP and the upper bound is significant according to a  $t$ -test:  $|t| = | -12.0612 | > t_{95, df=1231} = 1.96$  (confidence level of 95%, degrees of freedom = 1231). We can therefore conclude that for the AMS 2006 task, the upper bound on the av-



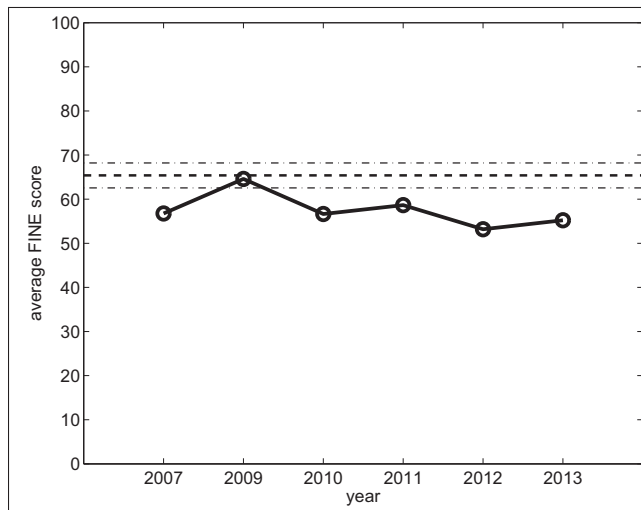
**Figure 1.** Average FINE score inter-rater agreement for different intervals of FINE scores (solid line). Dashed line indicates theoretical perfect agreement.

average FINE score had not yet been reached and that there still was room for improvement for future editions of the AMS task.

#### 4.2 Comparison to the upper bound

We will now compare the performance of the respective best participating systems in AMS 2007, 2009 to 2013 to the upper bound of average FINE scores we have retrieved in Section 4.1. This upper bound that can possibly be achieved due to the low inter-rater agreement results from the analysis of the AMS 2006 task. Although the whole evaluation protocol in all AMS tasks over the years is almost identical, AMS 2006 did use a song database that is only overlapping with that of subsequent years. It is therefore of course debatable how strictly the upper bound from AMS 2006 applies to the AMS results of later years. As outlined in Section 3, AMS 2006 has a genre distribution that is skewed to about 50% of rock music whereas all other AMS databases consist of equal amounts of songs from 10 genres. One could make the argument that in general songs from the same genre are being rated as being more similar than songs from different genres. As a consequence, agreement of raters for query/candidate pairs from identical genres might also be higher. Therefore inter-rater agreement within such a more homogeneous database should be higher than in a more diverse database and it can be expected that an upper bound of inter-rater agreement for AMS 2007 to 2013 is even lower than the one we obtained in Section 4.1. Of course this line of argument is somewhat speculative and needs to be further investigated.

In Figure 2 we have plotted the average FINE score of the highest performing participants of AMS tasks 2007, 2009 to 2013. These highest performing participants are the ones that achieved the highest average FINE scores in the respective years. In terms of statistical significance, the performance of these top algorithms is often at the same level as a number of other systems. We have also plotted



**Figure 2.** Average FINE score of best performing system (y-axis) vs. year (x-axis) plotted as solid line. Upper bound plus confidence interval plotted as dashed line.

year	system	mean	var	t
2007	PS	56.75	848.09	-4.3475
2009	PS2	64.58	633.76	-0.4415
2010	SSPK2	56.64	726.78	-4.6230
2011	SSPK2	58.64	687.91	-3.6248
2012	SSKS2	53.19	783.44	-6.3018
2013	SS2	55.21	692.23	-5.4604

**Table 3.** Comparison of best system vs. upper bound due to lack of inter-rater agreement.

the upper bound (dashed line) and a 95% confidence interval (dot-dashed lines). As can be seen the performance peaked in the year 2009 where the average FINE score reached the confidence interval. Average FINE scores in all other years are always a little lower. In Table 3 we show the results of a number of t-tests always comparing the performance to the upper bound. Table 3 gives the AMS year, the abbreviated name of the winning entry, the mean performance, its variance and the resulting t-value (with 831 degrees of freedom and 95% confidence). Only the best entry from year 2009 (PS2) reaches the performance of the upper bound, the best entries from all other years are statistically significant below the upper bound (critical value for all t-tests is again 1.96).

Interestingly, this system PS2 which gave the peak performance of all AMS years has also participated in 2010 to 2013. In terms of statistical significance (as measured via Friedman tests as part of the MIREX evaluation), PS2 has performed on the same level with the top systems of all following years. The systems PS2 has been submitted by Tim Pohle and Dominik Schnitzer and essentially consists of a timbre and a rhythm component [12]. Its main ingredients are MFCCs modeled via single Gaussians and Fluctuation patterns. It also uses the so-called P-norm normalization of distance spaces for combination of timbre and rhythm and to reduce the effect of hubness (anormal behavior of



distance spaces due to high dimensionality, see [6] for a discussion related to the AMS task and [14] on re-scaling of distance spaces to avoid these effects).

As outlined in Section 3, from 2007 on the same database of songs was used for the AMS tasks. However, each year a different set of 100 or 50 songs was chosen for the human listening tests. This fact can explain that the one algorithm participating from 2009 to 2013 did not always perform at the exact same level. After all, not only the choice of different human graders is a source of variance in the obtained FINE scores, but also the choice of different song material. However, the fact that the one algorithm that reached the upper bound has so far not been outperformed adds additional evidence that the upper bound that we obtained indeed is valid.

## 5. DISCUSSION

Our meta analysis of all editions of the MIREX "Audio Music Similarity and Retrieval" tasks conducted so far has produced somewhat sobering results. Due to the lack of inter-rater agreement there exists an upper bound of performance in subjective evaluation of music similarity. Such an upper bound will always exist when a number of different people have to agree on a concept as complex as that of music similarity. The fact that in the MIREX AMS task the notion of similarity is not defined very clearly adds to this general problem. After all, to "sound similar" does mean something quite different to different people listening to diverse music. As a consequence, an algorithm that has reached this upper bound of performance already in 2009 has not been outperformed ever since. Following our argumentation, this algorithm cannot be outperformed since any additional performance will be lost in the variance of the different human graders.

We now like to discuss a number of recommendations for future editions of the AMS task. One possibility is to go back to the procedure of AMS 2006 and again have more than one grader rate the same query/candidate pairs. This would allow to always also quantify the degree of inter-rater agreement and obtain upper bounds specific to the respective test songs. As we have argued above, we believe that the upper bound we obtained for AMS 2006 is valid for all AMS tasks. Therefore obtaining specific upper bounds would make much more sense if future AMS tasks would use an entirely different database of music. Such a change of song material would be a healthy choice in any case. Re-introducing multiple ratings per query/candidate pair would of course multiply the work load and effort if the number of song pairs to be evaluated should stay the same. However, using so-called "minimal test collections"-algorithms allows to obtain accurate estimates on much reduced numbers of query/candidate pairs as has already been demonstrated for the AMS task [18]. In addition rater-specific normalization should be explored. While some human graders use the full range of available FINE scores when grading similarity of song pairs, others might e.g. never rate song pairs as being very similar or not similar at all, thereby staying away from the extremes

of the scale. Such differences in rating style could add even more variance to the overall task and should therefore be taken care of via normalization.

However, all this would still not change the fundamental problem that the concept of music similarity is formulated in such a diffuse way that high inter-rater agreement cannot be expected. Therefore, it is probably necessary to research what the concept of music similarity actually means to human listeners. Such an exploration of what perceptual qualities are relevant to human listeners has already been conducted in the MIR community for the specific case of textural sounds [7]. Textural sounds are sounds that appear stationary as opposed to evolving over time and are therefore much simpler and constrained than real songs. By conducting mixed qualitative-quantitative interviews the authors were able to show that qualities like "high-low", "smooth-coarse" or "tonal-noisy" are important to humans discerning textural sounds. A similar approach could be explored for real song material, probably starting with a limited subset of genres. After such perceptual qualities have then been identified, future AMS tasks could ask human graders how similar pairs of songs are according to a specific quality of the music. Such qualities might not necessarily be straight forward musical concepts like melody, rhythm, or tempo, but rather more abstract notions like instrumentation, genre or specific recording effects signifying a certain style. Such a more fine-grained approach to music similarity would hopefully raise inter-rater agreement and make more room for improvements in modeling music similarity.

Last but not least it has been noted repeatedly that evaluation of abstract music similarity detached from a specific user scenario and corresponding user needs might not be meaningful at all [13]. Instead the MIR community might have to change to evaluation of complete music retrieval systems, thereby opening a whole new chapter for MIR research. Such an evaluation of a complete real life MIR system could center around a specific task for the users (e.g. building a playlist or finding specific music) thereby making the goal of the evaluation much clearer. Incidentally, this has already been named as one of the grand challenges for future MIR research [15]. And even more importantly, exactly such a user centered evaluation will happen at this year's tenth MIREX anniversary: the "MIREX Grand Challenge 2014: User Experience (GC14UX)"<sup>4</sup>. The task for participating teams is to create a web-based interface that supports users looking for background music for a short video. Systems will be rated by human evaluators on a number of important criteria with respect to user experience.

## 6. CONCLUSION

In our paper we have raised the important issue of the lack of inter-rater agreement in human evaluation of music information retrieval systems. Since human appraisal of phenomena as complex and multi-dimensional as music sim-

<sup>4</sup><http://www.music-ir.org/mirex/wiki/2014:GC14UX>

ilarity is highly subjective and depends on many factors such as personal preferences and past experiences, evaluation based on human judgments naturally shows high variance across subjects. This lack of inter-rater agreement presents a natural upper bound for performance of automatic analysis systems. We have demonstrated and analysed this problem in the context of the MIREX "Audio Music Similarity and Retrieval" task, but any evaluation of MIR systems that is based on ground truth annotated by humans has the same fundamental problem. Other examples from the MIREX campaign include such diverse tasks as "Structural Segmentation", "Symbolic Melodic Similarity" or "Audio Classification", which are all based on human annotations of varying degrees of ambiguity. Future research should explore upper bounds of performance for these many other MIR tasks based on human annotated data.

## 7. ACKNOWLEDGEMENTS

We would like to thank all the spiffy people who have made the MIREX evaluation campaign possible over the last ten years, including of course J. Stephen Downie and his people at IMIRSEL. This work was supported by the Austrian Science Fund (FWF, grants P27082 and Z159).

## 8. REFERENCES

- [1] Cohen J.: Statistical power analysis for the behavioral sciences, L. Erlbaum Associates, Second Edition, 1988.
- [2] Downie J.S.: The Music Information Retrieval Evaluation eXchange (MIREX), D-Lib Magazine, Volume 12, Number 12, 2006.
- [3] Downie J.S., Ehmann A.F., Bay M., Jones M.C.: The music information retrieval evaluation exchange: Some observations and insights, in *Advances in music information retrieval*, pp. 93-115, Springer Berlin Heidelberg, 2010.
- [4] Fleiss J.L.: Measuring nominal scale agreement among many raters, *Psychological Bulletin*, Vol. 76(5), pp. 378-382, 1971.
- [5] Flexer A., Schnitzer D.: Effects of Album and Artist Filters in Audio Similarity Computed for Very Large Music Databases, *Computer Music Journal*, Volume 34, Number 3, pp. 20-28, 2010.
- [6] Flexer A., Schnitzer D., Schlüter J.: A MIREX meta-analysis of hubness in audio music similarity, *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR'12)*, 2012.
- [7] Grill T., Flexer A., Cunningham S.: Identification of perceptual qualities in textural sounds using the repertory grid method, in *Proceedings of the 6th Audio Mostly Conference*, Coimbra, Portugal, 2011.
- [8] Jones M.C., Downie J.S., Ehmann A.F.: Human Similarity Judgments: Implications for the Design of Formal Evaluations, in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR'07)*, pp. 539-542, 2007.
- [9] Landis J.R., Koch G.G.: The measurement of observer agreement for categorical data, *Biometrics*, Vol. 33, pp. 159-174, 1977.
- [10] Novello A., McKinney M.F., Kohlrausch A.: Perceptual Evaluation of Music Similarity, *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, Victoria, Canada, 2006.
- [11] Pampalk E.: *Computational Models of Music Similarity and their Application to Music Information Retrieval*, Vienna University of Technology, Austria, Doctoral Thesis, 2006.
- [12] Pohle T., Schnitzer D., Schedl M., Knees P., Widmer G.: On Rhythm and General Music Similarity, *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR09)*, 2009.
- [13] Schedl M., Flexer A., Urbano J.: The neglected user in music information retrieval research, *Journal of Intelligent Information Systems*, 41(3), pp. 523-539, 2013.
- [14] Schnitzer D., Flexer A., Schedl M., Widmer G.: Local and Global Scaling Reduce Hubs in Space, *Journal of Machine Learning Research*, 13(Oct):2871-2902, 2012.
- [15] Serra X., Magas M., Benetos E., Chudy M., Dixon S., Flexer A., Gomez E., Gouyon F., Herrera P., Jorda S., Paytavi O., Peeters G., Schlüter J., Vinet H., Widmer G., *Roadmap for Music Information Research*, Peeters G. (editor), 2013.
- [16] Sturm B.L.: Classification accuracy is not enough, *Journal of Intelligent Information Systems*, 41(3), pp. 371-406, 2013.
- [17] Urbano J., Downie J.S., McFee B., Schedl M.: How Significant is Statistically Significant? The case of Audio Music Similarity and Retrieval, in *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR'12)*, pp. 181-186, 2012.
- [18] Urbano J., Schedl M.: Minimal test collections for low-cost evaluation of audio music similarity and retrieval systems, *International Journal of Multimedia Information Retrieval*, 2(1), pp. 59-70, 2013.
- [19] Vignoli F.: *Digital Music Interaction Concepts: A User Study*, *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, Barcelona, Spain, 2004.
- [20] West K.: *Novel techniques for audio music classification and search*, PhD thesis, University of East Anglia, 2008.



## Poster Session 2

This Page Intentionally Left Blank

# EMOTIONAL PREDISPOSITION OF MUSICAL INSTRUMENT TIMBRES WITH STATIC SPECTRA

**Bin Wu**

Department of Computer  
Science and Engineering  
Hong Kong University  
of Science and Technology  
Hong Kong  
bwuaa@cse.ust.hk

**Andrew Horner**

Department of Computer  
Science and Engineering  
Hong Kong University  
of Science and Technology  
Hong Kong  
horner@cse.ust.hk

**Chung Lee**

The Information Systems  
Technology and Design Pillar  
Singapore University  
of Technology and Design  
20 Dover Drive  
Singapore 138682  
im.lee.chung@gmail.com

## ABSTRACT

Music is one of the strongest triggers of emotions. Recent studies have shown strong emotional predispositions for musical instrument timbres. They have also shown significant correlations between spectral centroid and many emotions. Our recent study on spectral centroid-equalized tones further suggested that the even/odd harmonic ratio is a salient timbral feature after attack time and brightness. The emergence of the even/odd harmonic ratio motivated us to go a step further: to see whether the spectral shape of musical instruments alone can have a strong emotional predisposition. To address this issue, we conducted follow-up listening tests of static tones. The results showed that the even/odd harmonic ratio again significantly correlated with most emotions, consistent with the theory that static spectral shapes have a strong emotional predisposition.

## 1. INTRODUCTION

Music is one of the most effective media for conveying emotion. A lot of work has been done on emotion recognition in music, especially addressing melody [4], harmony [18], rhythm [23, 25], lyrics [15], and localization cues [11].

Some recent studies have shown that emotion is also closely related to timbre. Scherer and Oshinsky found that timbre is a salient factor in the rating of synthetic tones [24]. Peretz *et al.* showed that timbre speeds up discrimination of emotion categories [22]. Bigand *et al.* reported similar results in their study of emotion similarities between one-second musical excerpts [7]. It was also found that timbre is essential to musical genre recognition and discrimination [3, 5, 27].

Even more relevant to the current study, Eerola carried

out listening tests to investigate the correlation of emotion with temporal and spectral sound features [10]. The study confirmed strong correlations between features such as attack time and brightness and the emotion dimensions valence and arousal for one-second isolated instrument tones. Valence and arousal are measures of how pleasant and energetic the music sounds [31]. Asutay *et al.* also studied valence and arousal responses to 18 environmental sounds [2]. Despite the widespread use of valence and arousal in music research, composers may find them rather vague and difficult to interpret for composition and arrangement, and limited in emotional nuance. Using a different approach than Eerola, Ellermeier *et al.* investigated the unpleasantness of environmental sounds using paired comparisons [12].

Recently, we investigated the correlations between emotion and timbral features [30]. In our previous study, listening test subjects compared tones in terms of emotion categories such as Happy and Sad. We equalized the stimuli attacks and decays so that temporal features would not be factors. This modification isolated the effects of spectral features such as spectral centroid. Average spectral centroid significantly correlated for all emotions, and spectral centroid deviation significantly correlated for all emotions. This correlation was even stronger than average spectral centroid for most emotions. The only other correlation was spectral incoherence for a few emotions.

However, since average spectral centroid and spectral centroid deviation were so strong, listeners did not notice other spectral features much. This raised the question: if average spectral centroid was equalized in the tones, would spectral incoherence be more significant? Would other spectral characteristics emerge as significant? We tested this idea on spectral centroid normalized tones, and found that even/odd harmonic ratio was significant. This made us even more curious: if musical instruments tones only differed from one another in their spectral shapes, would they still have strong emotional predispositions? To answer this question, we conducted the follow-up experiment described in this paper using emotion responses for static spectra tones.



© Bin Wu, Andrew Horner, Chung Lee.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Bin Wu, Andrew Horner, Chung Lee. "Emotional Predisposition of Musical Instrument Timbres with Static Spectra", 15th International Society for Music Information Retrieval Conference, 2014.

## 2. LISTENING TEST

In our listening test, listeners compared pairs of eight instruments for eight emotions, using tones that were equalized for attack, decay, and spectral centroid.

### 2.1 Stimuli

#### 2.1.1 Prototype instrument sounds

The stimuli consisted of eight sustained wind and bowed string instrument tones: bassoon (Bs), clarinet (Cl), flute (Fl), horn (Hn), oboe (Ob), saxophone (Sx), trumpet (Tp), and violin (Vn). They were obtained from the McGill and Prosonus sample libraries, except for the trumpet, which had been recorded at the University of Illinois at Urbana-Champaign School of Music. The original of all these tones were used in a discrimination test carried out by Horner *et al.* [14], six of them were also used by McAdams *et al.* [20], and all of them used in our emotion-timbre test [30].

The tones were presented in their entirety. The tones were nearly harmonic and had fundamental frequencies close to 311.1 Hz (Eb4). The original fundamental frequencies deviated by up to 1 Hz (6 cents), and were synthesized by additive synthesis at 311.1 Hz.

Since loudness is potential factor in emotion, amplitude multipliers were determined by the Moore-Glasberg loudness program [21] to equalize loudness. Starting from a value of 1.0, an iterative procedure adjusted an amplitude multiplier until a standard loudness of  $87.3 \pm 0.1$  phons was achieved.

### 2.2 Stimuli Analysis and Synthesis

#### 2.2.1 Spectral Analysis Method

Instrument tones were analyzed using a phase-vocoder algorithm, which is different from most in that bin frequencies are aligned with the signal's harmonics (to obtain accurate harmonic amplitudes and optimize time resolution) [6]. The analysis method yields frequency deviations between harmonics of the analysis frequency and the corresponding frequencies of the input signal. The deviations are approximately harmonic relative to the fundamental and within  $\pm 2\%$  of the corresponding harmonics of the analysis frequency. More details on the analysis process are given by Beauchamp [6].

#### 2.2.2 Spectral Centroid Equalization

Different from our previous study [30], the average spectral centroid of the stimuli was equalized for all eight instruments. The spectra of each instrument was modified to an average spectral centroid of 3.7, which was the mean average spectral centroid of the eight tones. This modification was accomplished by scaling each harmonic amplitude by its harmonic number raised to a to-be-determined power:

$$A_k(t) \leftarrow k^p A_k(t) \quad (1)$$

For each tone, starting with  $p = 0$ ,  $p$  was iterated using Newton's method until an average spectral centroid was obtained within  $\pm 0.1$  of the 3.7 target value.

#### 2.2.3 Static Tone Preparation

The static tones were 0.5s in duration and were generated using the average steady-state spectrum of each spectral centroid equalized tone with linear 0.05s attacks and decays, and 0.4 sustains.

#### 2.2.4 Resynthesis Method

Stimuli were resynthesized from the time-varying harmonic data using the well-known method of time-varying additive sinewave synthesis (oscillator method) [6] with frequency deviations set to zero.

### 2.3 Subjects

32 subjects without hearing problems were hired to take the listening test. They were undergraduate students and ranged in age from 19 to 24. Half of them had music training (that is, at least five years of practice on an instrument).

### 2.4 Emotion Categories

As in our previous study [30], the subjects compared the stimuli in terms of eight emotion categories: Happy, Sad, Heroic, Scary, Comic, Shy, Joyful, and Depressed.

### 2.5 Listening Test Design

Every subject made pairwise comparisons of all eight instruments. During each trial, subjects heard a pair of tones from different instruments and were prompted to choose which tone more strongly aroused a given emotion. Each combination of two different instruments was presented in four trials for each emotion, and the listening test totaled  $C_2^8 \times 4 \times 8 = 896$  trials. For each emotion, the overall trial presentation order was randomized (i.e., all the Happy comparisons were first in a random order, then all the Sad comparisons were second, ...).

Before the first trial, the subjects read online definitions of the emotion categories from the Cambridge Academic Content Dictionary [1]. The listening test took about 1.5 hours, with breaks every 30 minutes.

The subjects were seated in a "quiet room" with less than 40 dB SPL background noise level. Residual noise was mostly due to computers and air conditioning. The noise level was further reduced with headphones. Sound signals were converted to analog by a Sound Blaster X-Fi Xtreme Audio sound card, and then presented through Sony MDR-7506 headphones at a level of approximately 78 dB SPL, as measured with a sound-level meter. The Sound Blaster DAC utilized 24 bits with a maximum sampling rate of 96 kHz and a 108 dB S/N ratio.

### 3. RESULTS

#### 3.1 Quality of Responses

The subjects' responses were first screened for inconsistencies, and two outliers were filtered out. Consistency was defined based on the four comparisons of a pair of instruments A and B for a particular emotion the same with our previous work [30]:

$$\text{consistency}_{A,B} = \frac{\max(v_A, v_B)}{4} \quad (2)$$

where  $v_A$  and  $v_B$  are the number of votes a subject gave to each of the two instruments. A consistency of 1 represents perfect consistency, whereas 0.5 represents approximately random guessing. The mean average consistency of all subjects was 0.74. Also, as in our previous work [30], we found that the two least consistent subjects had the highest outlier coefficients using White *et al.*'s method [28]. Therefore, they were excluded from the results.

We measured the level of agreement among the remaining 30 subjects with an overall Fleiss' Kappa statistic [16]. Fleiss' Kappa was 0.026, indicating a slight but statistically significant agreement among subjects. From this, we observed that subjects were self-consistent but less agreed in their responses than our previous study [30], since spectral shape was the only factor that could possibly affect emotion.

We also performed a  $\chi^2$  test [29] to evaluate whether the number of circular triads significantly deviated from the number to be expected by chance alone. This turned out to be insignificant for all subjects. The approximate likelihood ratio test [29] for significance of weak stochastic transitivity violations [26] was tested and showed no significance for all emotions.

##### 3.1.1 Emotion Results

Same with our previous work, we ranked the spectral centroid equalized instrument tones by the number of positive votes they received for each emotion, and derived scale values using the Bradley-Terry-Luce (BTL) model [8, 29] as shown in Figure 1. The likelihood-ratio test showed that the BTL model describes the paired-comparisons well for all emotions. We observe that: 1) The distribution of emotion ratings were much narrower than the original tones in our previous study [30]. The reason is that spectral shape was the only factor that could possibly affect emotion, which made it more difficult for subjects to distinguish. 2) Opposite of our previous study [30], the horn evoked positive emotions. It was ranked as the least Shy and Depressed, and among the most Heroic and Comic. 3) The clarinet and the saxophone were contrasting outliers for all emotions (except Scary).

Figure 2 shows BTL scale values and the corresponding 95% confidence intervals of the instruments for each emotion. The confidence intervals cluster near the line of indifference since it was difficult for listeners to make emotional distinctions. Table 1 shows the spectral characteristics of the static tones (time-domain spectral char-

acteristics are omitted since the tones are static). With all time-domain spectral characteristics removed, spectral shape features such as even/odd harmonic ratio became more salient. Specifically, even/odd ratio was calculated according to Caclin *et al.*'s method [9]. Pearson correlation between emotion and spectral characteristics are shown in Table 2. Both spectral irregularity and even/odd harmonic ratio are measures of spectral jaggedness, where even/odd harmonic ratio measures a particular, extreme type of spectral irregularity that is typical of the clarinet. In Table 2, even/odd harmonic ratio significantly correlated with nearly all emotions. The correlations were much stronger than in the original tones [30], and indicate that spectral shape by itself can arouse strong emotional responses.

### 4. DISCUSSION

These results are consistent with our previous results [30] and Eerola's Valence-Arousal results [10]. All these studies indicate that musical instrument timbres carry cues about emotional expression that are easily and consistently recognized by listeners. They show that spectral centroid/brightness is a significant component in music emotion. Beyond Eerola's and our previous findings, we have found that spectral shape by itself can have strong emotional predispositions, and even/odd harmonic ratio is the most salient timbral feature after attack time and brightness in static tones.

In hindsight, perhaps it is not so surprising that static spectra tones have emotional predispositions just as dynamic musical instrument tones do. It is somewhat analogous to viewers' emotional dispositions to primary colors [13, 17, 19].

Of course, just because static tones have emotional predispositions, it does not mean they are interesting to listen to. The dynamic spectra of real acoustic instruments are much more natural and life-like than any static tones, regardless of emotional predisposition. This is reflected in the wider range of emotion rankings of the original dynamic tones compared to the static tones.

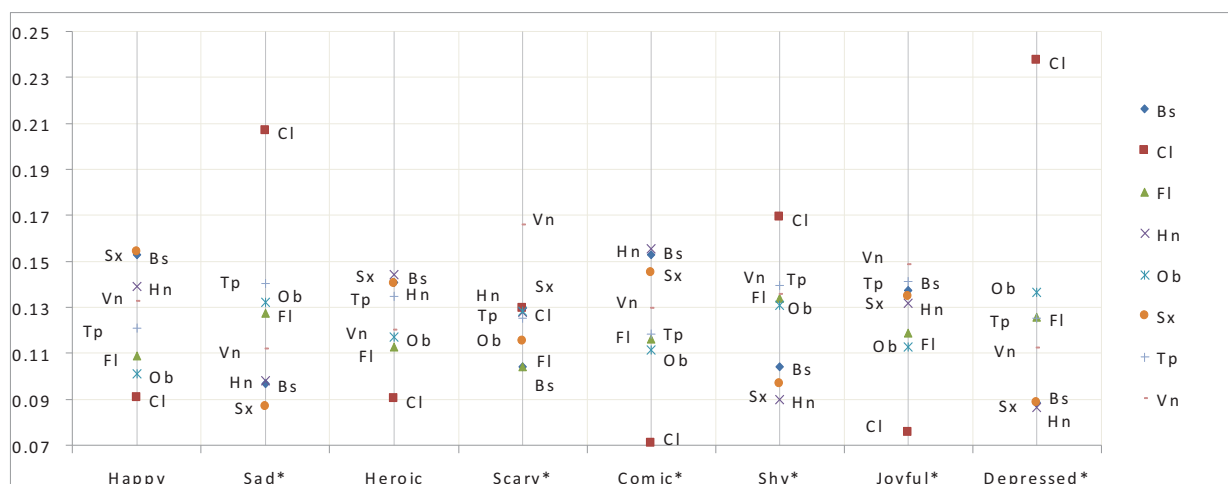
For future work, it will be fascinating to see how emotion varies with pitch, dynamic level, brightness, articulation, and cultural backgrounds.

### 5. ACKNOWLEDGMENT

This work has been supported by Hong Kong Research Grants Council grants HKUST613112.

### 6. REFERENCES

- [1] happy, sad, heroic, scary, comic, shy, joyful and depressed. *Cambridge Academic Content Dictionary*, 2013. Online: <http://goo.gl/v5xJZ> (17 Feb 2013).
- [2] Erkin Asutay, Daniel Västfjäll, Ana Tajadura-Jiménez, Anders Genell, Penny Bergman, and Mendel Kleiner. Emoacoustics: A Study of the Psychoacoustical and Psychological Dimensions of Emotional Sound

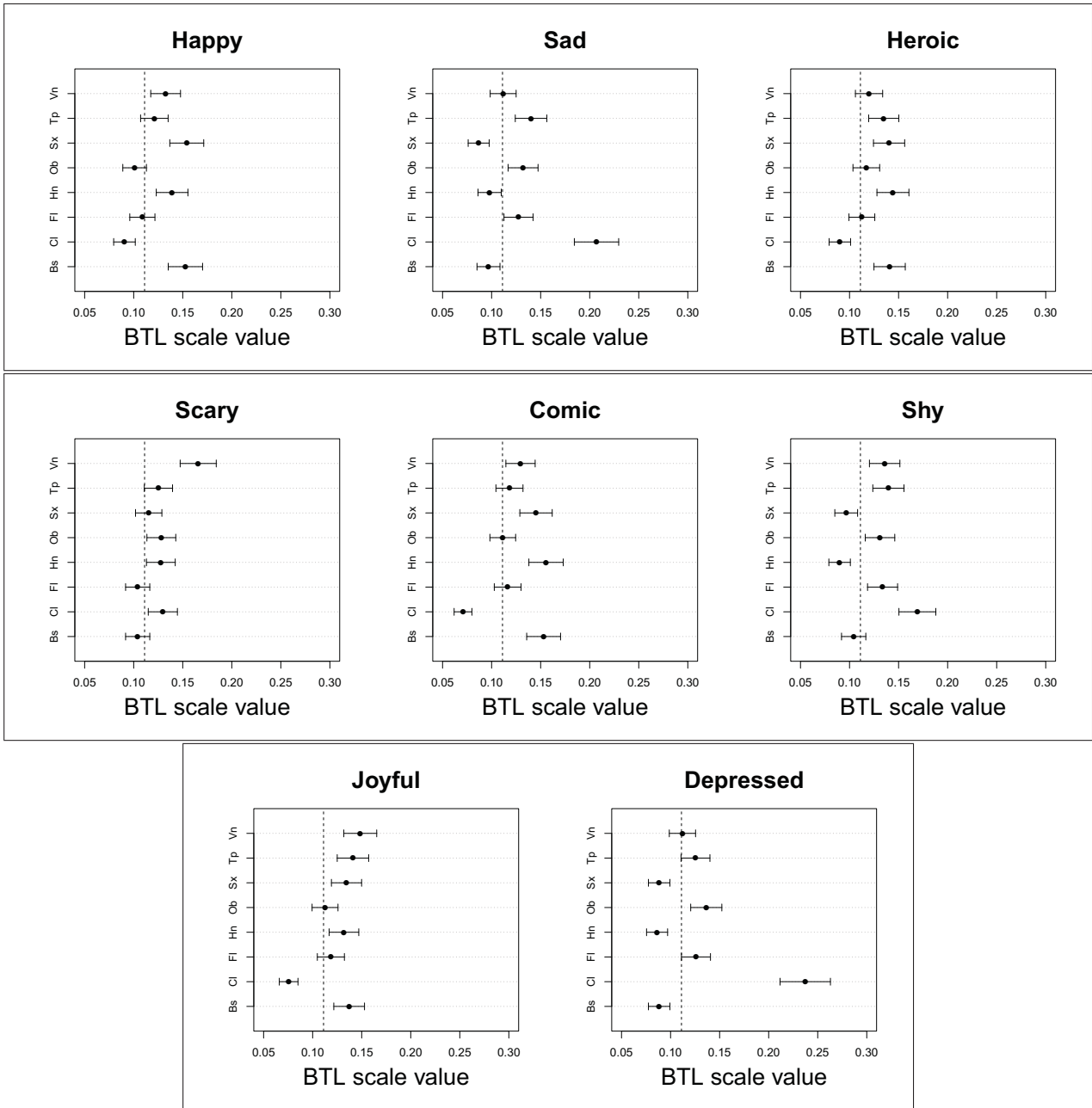


**Figure 1.** Bradley-Terry-Luce scale values of the static tones for each emotion.

- Design. *Journal of the Audio Engineering Society*, 60(1/2):21–28, 2012.
- [3] Jean-Julien Aucouturier, François Pachet, and Mark Sandler. The Way it Sounds: Timbre Models for Analysis and Retrieval of Music Signals. *IEEE Transactions on Multimedia*, 7(6):1028–1035, 2005.
- [4] Laura-Lee Balkwill and William Forde Thompson. A Cross-Cultural Investigation of the Perception of Emotion in Music: Psychophysical and Cultural Cues. *Music Perception*, 17(1):43–64, 1999.
- [5] Chris Baume. Evaluation of Acoustic Features for Music Emotion Recognition. In *Audio Engineering Society Convention 134*. Audio Engineering Society, 2013.
- [6] James W Beauchamp. Analysis and Synthesis of Musical Instrument Sounds. In *Analysis, Synthesis, and Perception of Musical Sounds*, pages 1–89. Springer, 2007.
- [7] E Bigand, S Vieillard, F Madurell, J Marozeau, and A Dacquet. Multidimensional Scaling of Emotional Responses to Music: The Effect of Musical Expertise and of the Duration of the Excerpts. *Cognition and Emotion*, 19(8):1113–1139, 2005.
- [8] Ralph A Bradley. Paired Comparisons: Some Basic Procedures and Examples. *Nonparametric Methods*, 4:299–326, 1984.
- [9] Anne Caclin, Stephen McAdams, Bennett K Smith, and Suzanne Winsberg. Acoustic Correlates of Timbre Space Dimensions: A Confirmatory Study Using Synthetic Tones. *Journal of the Acoustical Society of America*, 118:471, 2005.
- [10] Tuomas Eerola, Rafael Ferrer, and Vinoo Alluri. Timbre and Affect Dimensions: Evidence from Affect and Similarity Ratings and Acoustic Correlates of Isolated Instrument Sounds. *Music Perception*, 30(1):49–70, 2012.
- [11] Inger Ekman and Raine Kajastila. Localization Cues Affect Emotional Judgments—Results from a User Study on Scary Sound. In *Audio Engineering Society Conference: 35th International Conference: Audio for Games*. Audio Engineering Society, 2009.
- [12] Wolfgang Ellermeier, Markus Mader, and Peter Daniel. Scaling the Unpleasantness of Sounds According to the BTL Model: Ratio-scale Representation and Psychoacoustical Analysis. *Acta Acustica United with Acustica*, 90(1):101–107, 2004.
- [13] Michael Hemphill. A note on adults’ color–emotion associations. *The Journal of genetic psychology*, 157(3):275–280, 1996.
- [14] Andrew Horner, James Beauchamp, and Richard So. Detection of Random Alterations to Time-varying Musical Instrument Spectra. *Journal of the Acoustical Society of America*, 116:1800–1810, 2004.
- [15] Yajie Hu, Xiaou Chen, and Deshun Yang. Lyric-Based Song Emotion Detection with Affective Lexicon and Fuzzy Clustering Method. *Proceedings of ISMIR*, 2009.
- [16] Fleiss L Joseph. Measuring Nominal Scale Agreement among Many Raters. *Psychological Bulletin*, 76(5):378–382, 1971.
- [17] Naz Kaya and Helen H Epps. Relationship between color and emotion: A study of college students. *College student journal*, 38(3), 2004.
- [18] Judith Liebetrau, Sebastian Schneider, and Roman Jezierski. Application of Free Choice Profiling for the Evaluation of Emotions Elicited by Music. In *Proceedings of the 9th International Symposium on Computer Music Modeling and Retrieval (CMMR 2012): Music and Emotions*, pages 78–93, 2012.
- [19] Banu Manav. Color-emotion associations and color preferences: A case study for residences. *Color Research & Application*, 32(2):144–150, 2007.



- [20] Stephen McAdams, James W Beauchamp, and Suzanna Meneguzzi. Discrimination of Musical Instrument Sounds Resynthesized with Simplified Spectrotemporal Parameters. *Journal of the Acoustical Society of America*, 105:882, 1999.
- [21] Brian CJ Moore, Brian R Glasberg, and Thomas Baer. A Model for the Prediction of Thresholds, Loudness, and Partial Loudness. *Journal of the Audio Engineering Society*, 45(4):224–240, 1997.
- [22] Isabelle Peretz, Lise Gagnon, and Bernard Bouchard. Music and Emotion: Perceptual Determinants, Immediacy, and Isolation after Brain Damage. *Cognition*, 68(2):111–141, 1998.
- [23] Magdalena Plewa and Bozena Kostek. A Study on Correlation between Tempo and Mood of Music. In *Audio Engineering Society Convention 133*, Oct 2012.
- [24] Klaus R Scherer and James S Oshinsky. Cue Utilization in Emotion Attribution from Auditory Stimuli. *Motivation and Emotion*, 1(4):331–346, 1977.
- [25] Janto Skowronek, Martin McKinney, and Steven Van De Par. A Demonstrator for Automatic Music Mood Estimation. *Proceedings of the International Conference on Music Information Retrieval*, 2007.
- [26] Amos Tversky. Intransitivity of Preferences. *Psychological Review*, 76(1):31, 1969.
- [27] George Tzanetakis and Perry Cook. Musical Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.
- [28] Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier Movellan. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. *Advances in Neural Information Processing Systems*, 22(2035-2043):7–13, 2009.
- [29] Florian Wickelmaier and Christian Schmid. A Matlab Function to Estimate Choice Model Parameters from Paired-comparison Data. *Behavior Research Methods, Instruments, and Computers*, 36(1):29–40, 2004.
- [30] Bin Wu, Simon Wun, Chung Lee, and Andrew Horner. Spectral Correlates in Emotion Labeling of Sustained Musical Instrument Tones. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, November 4-8 2013.
- [31] Yi-Hsuan Yang, Yu-Ching Lin, Ya-Fan Su, and Homer H. Chen. A Regression Approach to Music Emotion Recognition. *IEEE TASLP*, 16(2):448–457, 2008.



**Figure 2.** BTL scale values and the corresponding 95% confidence intervals of the static tones for each emotion. The dotted line represents no preference.

Features \ Instruments	Instruments							
	Bs	Cl	Fl	Hn	Ob	Sx	Tp	Vn
Spectral Irregularity	0.0971	0.1818	0.143	0.0645	0.119	0.1959	0.0188	0.1176
Even/odd Ratio	1.2565	0.1775	0.9493	0.9694	0.4308	1.7719	0.7496	0.8771

**Table 1.** Spectral characteristics of the static instrument tones.

Features \ Emotion	Emotion							
	Happy	Sad	Heroic	Scary	Comic	Shy	Joyful	Depressed
Spectral Irregularity	-0.1467	0.1827	-0.4859	-0.0897	-0.3216	0.1565	-0.509	0.3536
Even/odd Ratio	<b>0.8901**</b>	<b>-0.8441**</b>	<b>0.7468**</b>	-0.3398	<b>0.8017**</b>	<b>-0.7942**</b>	<b>0.6524*</b>	<b>-0.7948**</b>

**Table 2.** Pearson correlation between emotion and spectral characteristics for static tones. \*\*:  $p < 0.05$ ; \*:  $0.05 < p < 0.1$ .

# PANAKO - A SCALABLE ACOUSTIC FINGERPRINTING SYSTEM HANDLING TIME-SCALE AND PITCH MODIFICATION

Six Joren, Marc Leman

Institute for Psychoacoustics and Electronic Music (IPEM),  
Department of Musicology, Ghent University

Ghent, Belgium

joren.six@ugent.be

## ABSTRACT

This paper presents a scalable granular acoustic fingerprinting system. An acoustic fingerprinting system uses condensed representation of audio signals, acoustic fingerprints, to identify short audio fragments in large audio databases. A robust fingerprinting system generates similar fingerprints for perceptually similar audio signals. The system presented here is designed to handle time-scale and pitch modifications. The open source implementation of the system is called Panako and is evaluated on commodity hardware using a freely available reference database with fingerprints of over 30,000 songs. The results show that the system responds quickly and reliably on queries, while handling time-scale and pitch modifications of up to ten percent.

The system is also shown to handle GSM-compression, several audio effects and band-pass filtering. After a query, the system returns the start time in the reference audio and how much the query has been pitch-shifted or time-stretched with respect to the reference audio. The design of the system that offers this combination of features is the main contribution of this paper.

## 1. INTRODUCTION

The ability to identify a small piece of audio by comparing it with a large reference audio database has many practical use cases. This is generally known as *audio fingerprinting* or *acoustic fingerprinting*. An acoustic fingerprint is a condensed representation of an audio signal that can be used to reliably identify identical, or recognize similar, audio signals in a large set of reference audio. The general process of an acoustic fingerprinting system is depicted in Figure 1. Ideally, a fingerprinting system only needs a short audio fragment to find a match in large set of reference audio. One of the challenges is to design a system in a way that the reference database can grow to contain millions of entries. Another challenge is that a robust finger-

printing should handle noise and other modifications well, while limiting the amount of false positives and processing time [5]. These modifications typically include dynamic range compression, equalization, added background noise and artifacts introduced by audio coders or A/D-D/A conversions.

Over the years several efficient acoustic fingerprinting methods have been introduced [1, 6, 8, 13]. These methods perform well, even with degraded audio quality and with industrial sized reference databases. However, these systems are not designed to handle queries with modified time-scale or pitch although these distortions can be present in replayed material. Changes in replay speed can occur either by accident during an analog to digital conversion or they are introduced deliberately.

Accidental replay speed changes can occur when working with physical, analogue media. Large music archive often consist of wax cylinders, magnetic tapes and gramophone records. These media are sometimes digitized using an incorrect or varying playback speed. Even when calibrated mechanical devices are used in a digitization process, the media could already have been recorded at an undesirable or undocumented speed. A fingerprinting system should therefore allow changes in replay speed to correctly detect duplicates in such music archives.

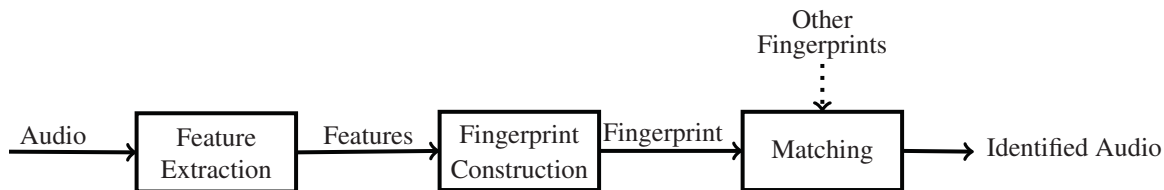
Deliberate time-scale manipulations are sometimes introduced as well. During radio broadcasts, for example, songs are occasionally played a bit faster to make them fit into a time slot. During a DJ-set pitch-shifting and time-stretching are present almost continuously. To correctly identify audio in these cases as well, a fingerprinting system robust against pitch-shifting and time-stretching is desired.

Some fingerprinting systems have been developed that take pitch-shifts into account [3, 7, 11] without allowing time-scale modification. Others are designed to handle both pitch and time-scale modification [10, 14]. The system by Zhu et al [14] employs an image processing algorithm on an auditory image to counter time-scale modification and pitch-shifts. Unfortunately, the system is computationally expensive, it iterates the whole database to find a match. The system by Malekesmaeili et al [10] allows extreme pitch-shifting and time-stretching, but has the same problem. To the best of our knowledge, a description of a practical acoustic fingerprinting system that allows sub-



© Six Joren, Marc Leman.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Six Joren, Marc Leman. "Panako - A Scalable Acoustic Fingerprinting System Handling Time-Scale and Pitch Modification", 15th International Society for Music Information Retrieval Conference, 2014.



**Figure 1: A generalized audio fingerprinter scheme.** Audio is fed into the system, features are extracted and fingerprints constructed. The fingerprints are consecutively compared with a database containing the fingerprints of the reference audio. The original audio is either identified or, if no match is found, labeled as unknown.

stantial pitch-shift and time-scale modification is nowhere to be found in the literature. This description is the main contribution of this paper.

## 2. METHOD

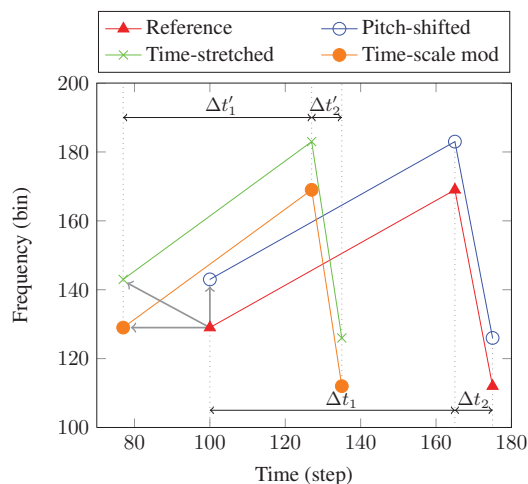
The proposed method is inspired by three works. Combining key components of those works results in a design of a granular acoustic fingerprinter that is robust to noise and substantial compression, has a scalable method for fingerprint storage and matching, and allows time-scale modification and pitch-shifting.

Firstly, the method used by Wang [13] establishes that local maxima in a time-frequency representation can be used to construct fingerprints that are *robust to quantization effects, filtering, noise and substantial compression*. The described exact-hashing method for *storing and matching fingerprints has proven to be very scalable*. Secondly, Artz et al. [2] describe a method to align performances and scores. Especially interesting is the way how triplets of events are used to search for performances with different timings. Thirdly, The method by Fenet et al. [7] introduces the idea to extract fingerprints from a Constant-Q [4] transform, a time-frequency representation that has a constant amount of bins for every octave. In their system a *fingerprint remains constant when a pitch-shift occurs*. However, since time is encoded directly within the fingerprint, the method does not allow time-scale modification.

Considering previous works, the method presented here uses local maxima in a spectral representation. It combines three event points, and takes time ratios to form time-scale invariant fingerprints. It leverages the Constant-Q transform, and only stores frequency differences for pitch-shift invariance. The fingerprints are designed with an exact hashing matching algorithm in mind. Below each aspect is detailed.

### 2.1 Finding Local Maxima

Suppose a time-frequency representation of a signal is provided. To locate the points where energy reaches a local maximum, a tiled two-dimensional peak picking algorithm is applied. First the local maxima for each spectral analysis frame are identified. Next each of the local maxima are iterated and put in the center of a tile with  $\Delta T \times \Delta F$  as dimensions. If the local maximum is also the maximum within the tile it is kept, otherwise it is discarded. Thus,



**Figure 2:** The effect of time-scale and pitch modifications on a fingerprint. It shows a single fingerprint extracted from reference audio (—▲—) and the same fingerprint extracted from audio after pitch-shifting (—○—), time-stretching (—●—) and time-scale modification (—×—).

making sure only one point is identified for every tile of  $\Delta T \times \Delta F$ . This approach is similar to [7, 13]. This results in a list of event points each with a frequency component  $f$ , expressed in bins, and a time component  $t$ , expressed in time steps.  $\Delta T$  and  $\Delta F$  are chosen so that there are between 24 and 60 event points every second.

A spectral representation of an audio signal has a certain granularity; it is essentially a grid with bins both in time as in frequency. When an audio signal is modified, the energy that was originally located in one single bin can be smeared over two or more bins. This poses a problem, since the goal is to be able to locate event points with maximum energy reliably. To improve reliability, a post processing step is done to refine the location of each event point by taking its energy and mixing it with the energy of the surrounding bins. The same thing is done for the surrounding bins. If a new maximum is found in the surroundings of the initial event point, the event point is relocated accordingly. Effectively, a rectangular blur with a  $3 \times 3$  kernel is applied at each event point and its surrounding bins.

Once the event points with local maximum energy are identified, the next step is to combine them to form a fingerprint. A fingerprint consists of three event points, as seen in Figure 2. To construct a fingerprint, each event point is combined with two nearby event points. Each event point can be part of multiple fingerprints. Only be-

tween 8 and 20 fingerprints are kept every second. Fingerprints with event points with the least cumulative energy are discarded. Now that a list of fingerprints has been created a method to encode time information in a fingerprint hash is needed.

## 2.2 Handling Time Stretching: Event Triplets

Figure 2 shows the effect of time stretching on points in the time-frequency domain. There, a fingerprint extracted from reference audio (Fig.2,  $\blacktriangle$ ) is compared with a fingerprint from time stretched audio (Fig.2,  $\bullet$ ). Both fingerprints are constructed using three local maxima  $e_1, e_2, e_3$  and  $e'_1, e'_2, e'_3$ . While the frequency components stay the same, the time components do change. However, the ratios between the time differences are constant as well. The following equation holds<sup>1</sup>:

$$\frac{t_2 - t_1}{t_3 - t_1} = \frac{t'_2 - t'_1}{t'_3 - t'_1} \quad (1)$$

With event point  $e_n$  having a time and frequency component  $(t_n, f_n)$  and the corresponding event points  $e'_n$  having the components  $(t'_n, f'_n)$ . Since  $t_3 - t_1 \geq t_2 - t_1$ , the ratio always resolves to a number in the range  $]0, 1]$ . This number, scaled and rounded, is a component of the eventual fingerprint hash (an approach similar to [2]).

Now that a way to encode time information, indifferent of time-stretching, has been found, a method to encode frequency, indifferent to pitch-shifting is desired.

## 2.3 Handling Pitch-Shifts: Constant-Q Transform

Figure 2 shows a comparison between a fingerprint from pitch shifted audio ( $\ominus$ ) with a fingerprint from reference audio ( $\blacktriangle$ ). In the time-frequency domain pitch shifting is a vertical translation and time information is preserved. Since every octave has the same number of bins [4] a pitch shift on event  $e_1$  will have the following effect on it's frequency component  $f_1$ , with  $K$  being a constant,  $f'_1 = f_1 + K$ . It is clear that the difference between the frequency components remains the same, before and after pitch shifting:  $f_1 - f_2 = (f'_1 + K) - (f'_2 + K)$  [7]. In the proposed system three event points are available, the following information is stored in the fingerprint hash:  $f_1 - f_2; f_2 - f_3; \tilde{f}_1; \tilde{f}_3$

The last two elements,  $\tilde{f}_1$  and  $\tilde{f}_3$  are sufficiently coarse locations of the first and third frequency component. They are determined by the index of the frequency band they fall into after dividing the spectrum into eight bands. They provide the hash with more discriminative power but also limit how much the audio can be pitch-shifted, while maintaining the same fingerprint hash.

## 2.4 Handling Time-Scale Modification

Figure 2 compares a fingerprint of reference audio (Fig.2,  $\blacktriangle$ ) with a fingerprint from the same audio that has been sped up (Fig.2,  $\times$ ). The figure makes clear that speed

<sup>1</sup> It is assumed that the time stretch factor is constant in the time interval  $t'_3 - t'_1$ . A reasonable assumption since  $t'_3 - t'_1$  is small.

change is a combination of both time-stretching and pitch-shifting. Since both are handled in with the previous measures, no extra precautions need to be taken. The next step is to combine the properties into a fingerprint that is efficient to store and match.

## 2.5 Fingerprint Hash

A fingerprint with a corresponding hash needs to be constructed carefully to maintain aforementioned properties. The result of a query should report the amount of pitch-shift and time-stretching that occurred. To that end, the absolute value of  $f_1$  and  $t_3 - t_1$  is stored, they can be used to compare with  $f'_1$  and  $t'_3 - t'_1$  from the query. The time offset at which a match was found should be returned as well, so  $t_1$  needs to be stored. The complete information to store for each fingerprint is:

$$\left( f_1 - f_2; f_2 - f_3; \tilde{f}_1; \tilde{f}_3; \frac{t_2 - t_1}{t_3 - t_1} \right); t_1; f_1; t_3 - t_1; id \quad (2)$$

The hash, the first element between brackets, can be packed into a 32bit integer. To save space,  $f_1$  and  $t_3 - t_1$  can be combined in one 32bit integer. An integer of 32bit is also used to store  $t_1$ . The reference audio identifier is also a 32bit identifier. A complete fingerprint consists of  $4 \times 32bit = 128bit$ . At eight fingerprints per second a song of four minutes is reduced to  $128bit \times 8 \times 60 \times 4 = 30kB$ . An industrial size data set of one million songs translates to a manageable  $28GB^2$ .

## 2.6 Matching Algorithm

The matching algorithm is inspired by [13], but is heavily modified to allow time stretched and pitch-shifted matches. It follows the scheme in Figure 1 and has seven steps.

1. Local maxima are extracted from a constant-Q spectrogram from the query. The local maxima are combined by three to form fingerprints, as explained in Sections 2.1, 2.3 and 2.4.
2. For each fingerprint a corresponding hash value is calculated, as explained in Section 2.5.
3. The set of hashes is matched with the hashes stored in the reference database, and each exact match is returned.
4. The matches are iterated while counting how many times each individual audio identifier occurs in the result set.
5. Matches with an audio identifier count lower than a certain threshold are removed, effectively dismissing random chance hits. In practice there is almost always only one item with a lot of matches, the rest being random chance hits. A threshold of three or four suffices.

<sup>2</sup> Depending on the storage engine used, storage of fingerprints together with an index of sorts introduces a storage overhead. Since the data to store is small, the index can be relatively large.

6. The residual matches are checked for alignment, both in frequency and time, with the reference fingerprints using the information that is stored along with the hash.
7. A list of audio identifiers is returned ordered by the amount of fingerprints that align both in pitch and frequency.

In step six, frequency alignment is checked by comparing the  $f_1$  component of the stored reference with  $f'_1$ , the frequency component of the query. If, for each match, the difference between  $f_1$  and  $f'_1$  is constant, the matches align.

Alignment in time is checked using the reference time information  $t_1$  and  $t_3 - t_1$ , and the time information of the corresponding fingerprint extracted from the query fragment  $t'_1$ ,  $t'_3 - t'_1$ . For each matching fingerprint the time offset  $t_o$  is calculated. The time offset  $t_o$  resolves to the amount of time steps between the beginning of the query and the beginning of the reference audio, even if a time modification took place. It stands to reason that  $t_o$  is constant for matching audio.

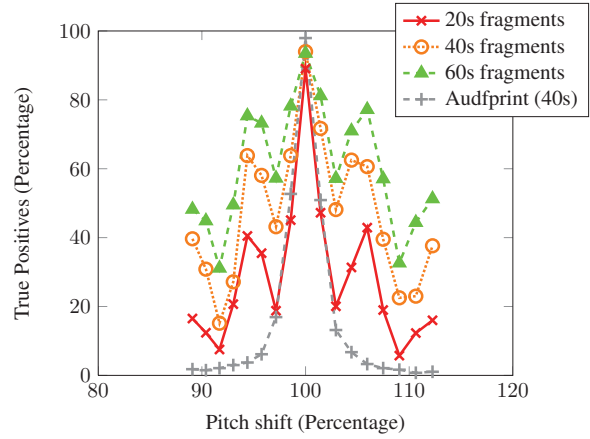
$$t_o = t_1 - t'_1 \times \frac{(t_3 - t_1)}{(t'_3 - t'_1)} \quad (3)$$

The matching algorithm also provides information about the query. The time offset tells at which point in time the query starts in the reference audio. The time difference ratio  $(t_3 - t_1)/(t'_3 - t'_1)$  represents how much time is modified, in percentages. How much the query is pitch-shifted with respect to the reference audio can be deduced from  $f'_1 - f_1$ , in frequency bins. To convert a difference in frequency bins to a percentage the following equation is used, with  $n$  the number of cents per bin,  $e$  Eulers number, and  $\ln$  the natural logarithm:  $e^{((f'_1 - f_1) \times n \times \ln(2)) / 1200}$

The matching algorithm ensures that random chance hits are very uncommon, the number of false positives can be effectively reduced to zero by setting a threshold on the number of aligned matches. The matching algorithm also provides the query time offset and the percentage of pitch-shift and time-scale modification of the query with respect to the reference audio.

### 3. RESULTS

To test the system, it was implemented in the Java programming language. The implementation is called Panako and is available under the *GNU Affero General Public License* on <http://panako.be>. The DSP is also done in Java using a DSP library [12]. To store and retrieve hashes, Panako uses a key-value store. Kyoto Cabinet, BerkeleyDB, Redis, LevelDB, RocksDB, Voldemort, and MapDB were considered. MapDB is an implementation of a storage backed B-Tree with efficient concurrent operations [9] and was chosen for its simplicity, performance and good Java integration. Also, the storage overhead introduced when storing fingerprints on disk is minimal. Panako is compared with Audfprint by Dan Ellis, an implementation of a fingerprinter system based on [13].



**Figure 3:** True positive rate after pitch-shifting. Note the fluctuating effect caused by the Constant-Q frequency bins.

The test data set consists of freely available music downloaded from Jamendo<sup>3</sup>. A reference database of about 30,000 songs, about  $10^6$  seconds of audio, was created. From this data set random fragments were selected, with a length of 20, 40 and 60 seconds. Each fragments was modified 54 times. The modifications included: pitch-shifting ( $-200$  tot  $200$  cents in steps of 25 cents), time-stretching ( $-16\%$  to  $+16\%$ , in steps of 2%), time-scale modification ( $-16\%$  to  $+16\%$ , in steps of 2%), echo, flanger, chorus and a band-pass filter<sup>4</sup>. Another set of fragments were created from audio not present in the reference database, in order to measure the number of correctly unidentified fragments. In total  $3$  (*durations*)  $\times$   $600$  (*excerpts*)  $\times$   $54$  (*modifications*) = 97,200 fragments were created.

Each fragment is presented to both Panako and Audfprint and the detection results are recorded. The systems are regarded as binary classifiers of which the amount of true positives ( $TP$ ), false positives ( $FP$ ), true negatives ( $TN$ ) and false negatives ( $FN$ ) are counted. During the experiment with Panako no false positives ( $FP$ ) were detected. Also, all fragments that are not present in the reference database were rejected correctly ( $TN$ ). So Panako's specificity is  $TN/(TN + FP) = 100\%$ . This can be explained by the design of the matching algorithm. A match is identified as such if a number of hashes, each consisting of three points in a spectrogram, align in time. A random match between hashes is rare, the chances of a random match between consecutively aligned hashes is almost non-existent, resulting in 100% specificity.

The sensitivity  $FP/(TP + FN)$  of the system, however, depends on the type of modification on the fragment. Figure 3 shows the results after pitch-shifting. It is clear that the amount of pitch-shift affects the performance, but

<sup>3</sup> <http://jamendo.com> is a website where artists share their work freely, under various creative commons licenses. To download the data set used in this paper, and repeat the experiment, please use the scripts provided at <http://panako.be>.

<sup>4</sup> The effects were applied using SoX, a command line audio editor. The scripts used to generate the queries can be found at the website <http://panako.be>

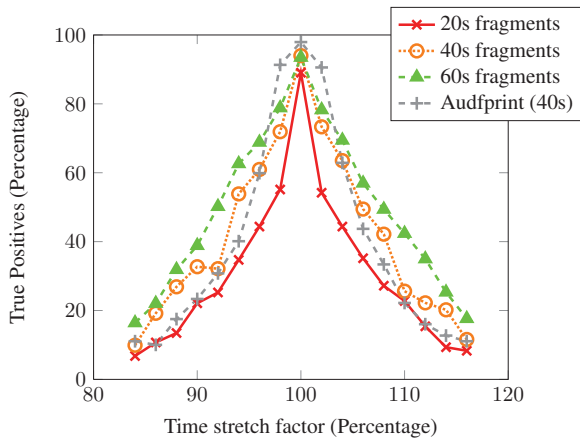


Figure 4: The true positive rate after time-stretching.

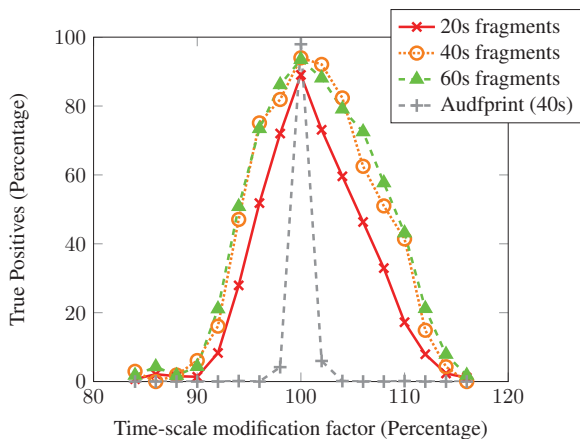


Figure 5: True positive rate after time-scale modification

in a fluctuating pattern. The effect can be explained by taking into account the Constant-Q bins. Here, a bin spans 33 cents, a shift of  $n \times 33/2$  cents spreads spectral information over two bins, if  $n$  is an odd number. So performance is expected to degrade severely at  $\pm 49.5$  cents (3%) and  $\pm 148.5$  cents (9%) an effect clearly visible in figure 3. The figure also shows that performance is better if longer fragments are presented to the system. The performance of Audfprint, however, does not recover after pitch-shifts of more than three percent.

Figure 4 shows the results after time stretching. Due to the granularity of the time bins, and considering that the step size stays the same for each query type, time modifications have a negative effect on the performance. Still, a more than a third of the queries is resolved correctly after a time stretching modification of 8%. Performance improves with the length of a fragment. Surprisingly, Audfprint is rather robust against time-stretching, thanks to the way time is encoded into a fingerprint.

Figure 5 shows the results after time-scale modification. The performance decreases severely above eight percent. The figure shows that there is some improvement when comparing the results of 20s fragments to 40s fragments, but going from 40s to 60s does not change much. Audiofprint is unable to cope with time-scale modification due to the changes in both frequency and time.

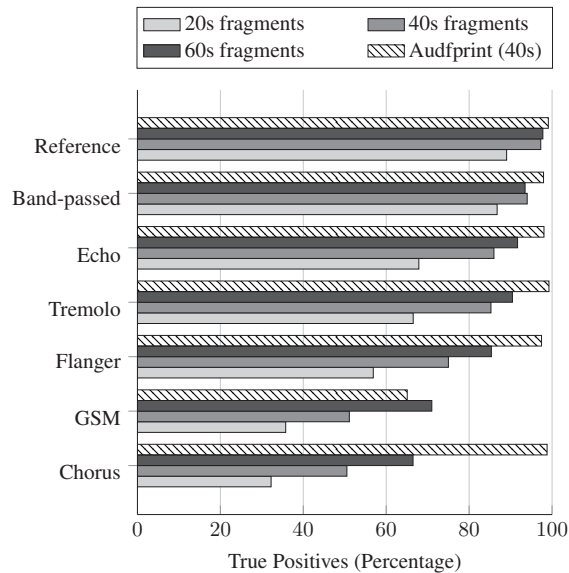


Figure 6: Effect of several attacks on true positive rate.

In Figure 6, the results for other modifications like echo, chorus, flanger, tremolo, and a band pass filter can be seen. The parameters of each effect are chosen to represent typical use, but on the heavy side. For example the echo effect applied has a delay line of 0.5 seconds and a decay of 30%. The system has the most problems with the chorus effect. Chorus has a blurring effect on a spectrogram, which makes it hard for the system to find matches. Still it can be said that the algorithm is rather robust against very present, clearly audible, commonly used audio effects. The result of the band pass filter with a center of 2000Hz is especially good. To test the systems robustness to severe audio compression a test was executed with GSM-compressed queries. The performance on 20s fragments is about 30% but improves a lot with query length, the 60s fragment yields 65%. The results for Audfprint show that there is room for improvement for the performance of Panako.

A practical fingerprinting system performs well, in terms of speed, on commodity hardware. With Panako extracting and storing fingerprints for 25s of audio is done in one second using a single core of a dated processor<sup>5</sup> The test data set was constructed in  $30,000 \times 4 \times 60s / 25 = 80$  processor hours. Since four cores were used, it took less than a full day. After the feature extraction, matching a 40s query with the test database with 30,000 songs is done within 75ms. The complete matching process for a 40s fragment takes about one second. Monitoring multiple streams in real-time poses no problem for the system. Building a fingerprint dataset with Audfprint is faster since fingerprints are extracted from an FFT which is less demanding than a Constant-Q transform. The matching step performance, however, is comparable.

<sup>5</sup> The testing machine has an Intel Core2 Quad CPU Q9650 @ 3.00GHz introduced in 2009. The processor has four cores.

Failure analysis shows that the system does not perform well on music with spectrograms either with very little energy or energy evenly spread across the range. Also extremely repetitive music, with a spectrogram similar to a series of dirac impulses, is problematic. Also, performance drops when time modifications of more than 8% are present. This could be partially alleviated by redesigning the time parameters used in the fingerprint hash, but this would reduce the discriminative power of the hash.

#### 4. CONCLUSION

In this paper a practical acoustic fingerprinting system was presented. The system allows fast and reliable identification of small audio fragments in a large set of audio, even when the fragment has been pitch-shifted and time-stretched with respect to the reference audio. If a match is found the system reports where in the reference audio a query matches, and how much time/frequency has been modified. To achieve this, the system uses local maxima in a Constant-Q spectrogram. It combines event points into groups of three, and uses time ratios to form a time-scale invariant fingerprint component. To form pitch-shift invariant fingerprint components only frequency differences are stored. For retrieval an exact hashing matching algorithm is used.

The system has been evaluated using a freely available data set of 30,000 songs and compared with a baseline system. The results can be reproduced entirely using this data set and the open source implementation of Panako. The scripts to run the experiment are available as well. The results show that the system's performance decreases with time-scale modification of more than eight percent. The system is shown to cope with pitch-shifting, time-stretching, severe compression, and other modifications as echo, flanger and band pass.

To improve the system further the constant-Q transform could be replaced by an efficient implementation of the non stationary Gabor transform. This is expected to improve the extraction of event points and fingerprints without effecting performance. Panako could also benefit from a more extensive evaluation and detailed comparison with other techniques. An analysis of the minimum, most discriminative, information needed for retrieval purposes could be especially interesting.

#### 5. REFERENCES

- [1] Eric Allamanche. Content-based identification of audio material using mpeg-7 low level description. In *Proceedings of the 2nd International Symposium on Music Information Retrieval (ISMIR 2001)*, 2001.
- [2] Andreas Arzt, Sebastian Böck, and Gerhard Widmer. Fast identification of piece and score position via symbolic fingerprinting. In Fabien Gouyon, Perfecto Herrera, Luis Gustavo Martins, and Meinard Mller, editors, *Proceedings of the 13th International Symposium on Music Information Retrieval (ISMIR 2012)*, pages 433–438, 2012.
- [3] Carlo Bellettini and Gianluca Mazzini. Reliable automatic recognition for pitch-shifted audio. In *Proceedings of 17th International Conference on Computer Communications and Networks (ICCCN 2008)*, pages 838–843. IEEE, 2008.
- [4] Judith Brown and Miller S. Puckette. An Efficient Algorithm for the Calculation of a Constant Q Transform. *Journal of the Acoustical Society of America*, 92(5):2698–2701, November 1992.
- [5] Pedro Cano, Eloi Batlle, Ton Kalker, and Jaap Haitsma. A review of audio fingerprinting. *The Journal of VLSI Signal Processing*, 41:271–284, 2005.
- [6] Dan Ellis, Brian Whitman, and Alastair Porter. Echoprint - an open music identification service. In *Proceedings of the 12th International Symposium on Music Information Retrieval (ISMIR 2011)*, 2011.
- [7] Sébastien Fenet, Gaël Richard, and Yves Grenier. A Scalable Audio Fingerprint Method with Robustness to Pitch-Shifting. In *Proceedings of the 12th International Symposium on Music Information Retrieval (ISMIR 2011)*, pages 121–126, 2011.
- [8] Jaap Haitsma and Ton Kalker. A highly robust audio fingerprinting system. In *Proceedings of the 3th International Symposium on Music Information Retrieval (ISMIR 2002)*, 2002.
- [9] Philip L. Lehman and s. Bing Yao. Efficient locking for concurrent operations on b-trees. *ACM Transactions Database Systems*, 6(4):650–670, 1981.
- [10] Mani Malekesmaeili and Rabab K. Ward. A local fingerprinting approach for audio copy detection. *Computing Research Repository (CoRR)*, abs/1304.0793, 2013.
- [11] M. Ramona and G. Peeters. AudioPrint: An efficient audio fingerprint system based on a novel cost-less synchronization scheme. In *Proceedings of the 2013 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP 2013)*, pages 818–822, 2013.
- [12] Joren Six, Olmo Cornelis, and Marc Leman. Tarsos-DSP, a Real-Time Audio Processing Framework in Java. In *Proceedings of the 53rd AES Conference (AES 53rd)*. The Audio Engineering Society, 2014.
- [13] Avery L. Wang. An Industrial-Strength Audio Search Algorithm. In *Proceedings of the 4th International Symposium on Music Information Retrieval (ISMIR 2003)*, pages 7–13, 2003.
- [14] Bilei Zhu, Wei Li, Zhurong Wang, and Xiangyang Xue. A novel audio fingerprinting method robust to time scale modification and pitch shifting. In *Proceedings of the international conference on Multimedia (MM 2010)*, pages 987–990. ACM, 2010.



# PERCEPTUAL ANALYSIS OF THE F-MEASURE FOR EVALUATING SECTION BOUNDARIES IN MUSIC

Oriol Nieto<sup>1</sup>, Morwaread M. Farbood<sup>1</sup>, Tristan Jehan<sup>2</sup>, and Juan Pablo Bello<sup>1</sup>

<sup>1</sup>Music and Audio Research Lab, New York University, {oriol, mfarbood, jpbello}@nyu.edu

<sup>2</sup>The Echo Nest, tristan@echonest.com

## ABSTRACT

In this paper, we aim to raise awareness of the limitations of the F-measure when evaluating the quality of the boundaries found in the automatic segmentation of music. We present and discuss the results of various experiments where subjects listened to different musical excerpts containing boundary indications and had to rate the quality of the boundaries. These boundaries were carefully generated from state-of-the-art segmentation algorithms as well as human-annotated data. The results show that humans tend to give more relevance to the *precision* component of the F-measure rather than the *recall* component, therefore making the classical F-measure not as perceptually informative as currently assumed. Based on the results of the experiments, we discuss the potential of an alternative evaluation based on the F-measure that emphasizes precision over recall, making the section boundary evaluation more expressive and reliable.

## 1. INTRODUCTION

Over the past decade, significant effort has been made toward developing methods that automatically extract large-scale structures in music. In this paper, we use the term *musical structure analysis* to refer to the task that identifies the different sections (or segments) of a piece. In Western popular music, these sections are commonly labeled as *verse*, *chorus*, *bridge*, etc. Given that we now have access to vast music collections, this type of automated analysis can be highly beneficial for organizing and exploring these collections.

Musical structure analysis is usually divided into two subtasks: the identification of section boundaries and the labeling of these sections based on their similarity. Here, we will only focus on the former. Section boundaries usually occur when salient changes in various musical qualities (such as harmony, timbre, rhythm, or tempo) take place. See [9] for a review of some of the state of the art in musical structure analysis.

Typically, researchers make use of various human-annotated datasets to measure the accuracy of their analysis algorithms. The standard methodology for evaluating the accuracy of estimated section boundaries is to compare those estimations with ground truth data by means of the F-measure (also referred to as the hit rate), which gives equal weight to the values of precision (proportion of the boundaries found that are correct) and recall (proportion of correct boundaries that are located). However, it is not entirely clear that humans perceive the type of errors those two metrics favor or the penalties they impose as equally important, calling into question the perceptual relevance of the F-measure for evaluating long-term segmentation. To the best of our knowledge, no empirical evidence or formal study exists that can address such a question in the context of section boundary identification. This work is an effort to redress that.

Our work is motivated by a preliminary study we ran on two subjects showing a preference for high precision results, thus making us reconsider the relevance of precision and recall for the evaluation of section boundary estimations. As a result, in this paper we present two additional experiments aimed at validating and expanding those preliminary findings including a larger subject population and more controlled conditions. In our experiments, we focus on the analysis of Western popular songs since this is the type of data most segmentation algorithms in the MIR literature operate on, and since previous studies have shown that most listeners can confidently identify structure in this type of music [1].

The rest of this paper is organized as follows. We present a review of the F-measure and a discussion of the preliminary study in section 2. We describe the design of two experiments along with discussions of their results in sections 3 and 4. We explore an alternative F-measure based on our experimental findings that could yield more expressive and perceptually relevant outcomes in section 5. Finally, we draw conclusions and discuss future work in section 6.

## 2. THE F-MEASURE FOR MUSIC BOUNDARIES

### 2.1 Review of the F-measure

In order to evaluate automatically computed music boundaries, we have to define how we accept or reject an estimated boundary given a set of annotated ones (i.e., find the intersection between these two sets). Traditionally, re-



searchers consider an estimated boundary *correct* as long as its maximum deviation to its closest annotated boundary is  $\pm 3$  seconds [8] (in MIREX, <sup>1</sup> inspired by [16], an evaluation that uses a shorter window of  $\pm 0.5$  seconds is also performed). Following this convention, we use a  $\pm 3$ -second window in our evaluation.

Let us assume that we have a set of correctly estimated boundaries given the annotated ones (hits), a set of annotated boundaries that are not estimated (false negatives), and a set of estimated boundaries that are not in the annotated dataset (false positives). Precision is the ratio between hits and the total number of estimated elements (e.g., we could have 100% precision with an algorithm that only returns exactly one boundary and this boundary is correct). Recall is the ratio between hits and the total number of annotated elements (e.g. we could have a 100% recall with an algorithm that returns one boundary every 3 seconds, since all the annotated boundaries will be sufficiently close to an estimated one). Precision and recall are defined formally as

$$P = \frac{|\text{hits}|}{|\text{bounds}_e|}; \quad R = \frac{|\text{hits}|}{|\text{bounds}_a|} \quad (1)$$

where  $|\cdot|$  represents the cardinality of the set  $\cdot$ ,  $\text{bounds}_e$  is the set of estimated boundaries and  $\text{bounds}_a$  is the set of annotated ones. Finally, the F-measure is the harmonic mean between  $P$  and  $R$ , which weights these two values equally, penalizes small outliers, and mitigates the impact of large ones:

$$F = 2 \frac{P \cdot R}{P + R} \quad (2)$$

When listening to the output of music segmentation algorithms, it is immediately apparent that false negatives and false positives are perceptually very different (an initial discussion about assessing a *synthetic* precision of 100% when evaluating boundaries can be found in [14]). Thus, in the process of developing novel methods for structure segmentation, we decided to informally assess the relative effect that different types of errors had on human evaluations of the accuracy of the algorithms' outputs. The following section describes the resulting preliminary study.

## 2.2 Preliminary Study

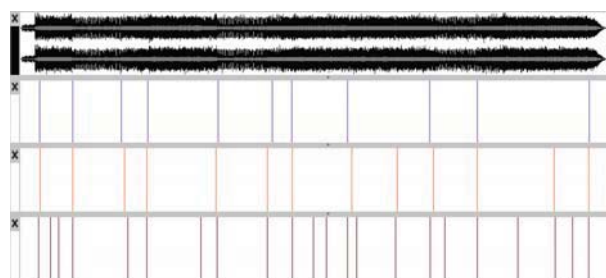
For this study we compared three algorithms, which we will term  $\mathcal{A}$ ,  $\mathcal{B}$  and  $\mathcal{C}$ .  $\mathcal{A}$  is an unpublished algorithm currently in development that relies on homogeneous repeated section blocks;  $\mathcal{B}$  is an existing algorithm that uses novelty in audio features to identify boundaries; and  $\mathcal{C}$  combines the previous two methods. All three methods were optimized to maximize their F-measure performance on the structure-annotated Levy dataset [5]. Table 1 shows each method's average F-measure, precision, and recall values across the entire set. Note how  $\mathcal{C}$  maximizes the F-measure, mostly by increasing recall, while  $\mathcal{A}$  shows maximum precision.

We asked two college music majors to rank the three algorithms for every track in the Levy set. The goal was

Preliminary Study			
Algorithm	F	P	R
$\mathcal{A}$	49%	57%	47%
$\mathcal{B}$	44%	46%	46%
$\mathcal{C}$	51%	47%	64%

**Table 1.** Algorithms and their ratings used to generate the input for the preliminary study. These ratings are averaged across the 60 songs of the Levy dataset.

not to compare the results of the algorithms to the annotated ground truth, but to compare the algorithms with each other and determine the best one from a perceptual point of view. The participants were asked to listen to each of the algorithm outputs for all the songs and rank the algorithms by the quality of their estimated section boundaries; no particular constraints were given on what to look for. We used Sonic Visualiser [3] to display the waveform and three section panels for each of the algorithms in parallel (see Figure 1). While playing the audio, listeners could both see the sections and hear the boundaries indicated by a distinctive percussive sound. The section panels were organized at random for each song so listeners could not easily tell which algorithm they were choosing.



**Figure 1.** Screenshot of Sonic Visualiser used in the preliminary experiment. The song is “Smells Like Teen Spirit” by Nirvana. In this case, algorithms are ordered as  $\mathcal{A}$ ,  $\mathcal{B}$ , and  $\mathcal{C}$  from top to bottom.

Analysis of the results showed that 68.3% of the time, the two participants chose the same best algorithm. In 23.3% of the cases, they disagreed on the best, and in just 8.3% of the cases, they chose opposite rankings. When they actually agreed on the best algorithm, they chose  $\mathcal{A}$  58.5% of the time.  $\mathcal{A}$  did not have the highest F-measure but the highest precision. Perhaps more surprising, they chose  $\mathcal{C}$  only 14.6% of the time even though that algorithm had the highest F-measure.

These results raised the following questions: Is the F-measure informative enough to evaluate the accuracy of automatically estimated boundaries in a perceptually-meaningful way? Is precision more important than recall when assessing music boundaries? Would the observed trends remain when tested on a larger population of subjects? Can these results inform more meaningful evaluation measures? We decided to address these questions by running two more formal experiments in order to better understand this apparent problem and identify a feasible solution.

<sup>1</sup> [http://www.music-ir.org/mirex/wiki/MIREX\\_HOME](http://www.music-ir.org/mirex/wiki/MIREX_HOME)

### 3. EXPERIMENT 1: RATING BOUNDARIES

#### 3.1 Motivation

The results of the preliminary study suggested that precision is more relevant than recall when perceiving boundaries. However, to fully explore this hypothesis, these two values had to be carefully manipulated. For this experiment, a set of boundaries was synthesized by setting specific values for precision and recall while maintaining a near-constant F-measure. Moreover, we wanted to ensure that the findings were robust across a larger pool of subjects. With these considerations in mind, the experiment was designed to be both shorter in time and available on line.

#### 3.2 Methodology

We selected five track excerpts from the Levy catalog by finding the one-minute segments containing the highest number of boundaries across the 60 songs of the dataset. By having short excerpts instead of full songs, we could reduce the duration of the entire experiment with negligible effect on the results—past studies have shown that boundaries are usually perceived locally instead of globally [15]. We decided to use only five excerpts with the highest number of boundaries in order to maintain participants’ attention as much as possible. For each track excerpt, we synthesized three different segmentations: ground truth boundaries (GT) with an F-measure of 100%; high precision (HP) boundaries with a precision of 100% and recall of around 65%; and high recall (HR) boundaries with a recall of 100% and precision of around 65%. The extra boundaries for the HR version were randomly distributed (using a normal distribution) across a 3 sec window between the largest regions between boundaries. For the HP version, the boundaries that were most closely spaced were removed. Table 2 presents F-measure, precision, and recall values for the five tracks along with the average values across excerpts. Note the closeness between F-measure values for HP and HR.

Experiment 1 Excerpt List						
Song Name (Artist)	HP			HR		
	F	P	R	F	P	R
Black & White (Michael Jackson)	.809	1	.68	.794	.658	1
Drive (R.E.M.)	.785	1	.647	.791	.654	1
Intergalactic (Beastie Boys)	.764	1	.619	.792	.656	1
Suds And Soda (Deus)	.782	1	.653	.8	.666	1
Tubthumping (Chumbawamba)	.744	1	.593	.794	.659	1
Average	.777	1	.636	.794	.659	1

**Table 2.** Excerpt list with their evaluations for experiment 1. The F-measure of GT is 100% (not shown in the table).

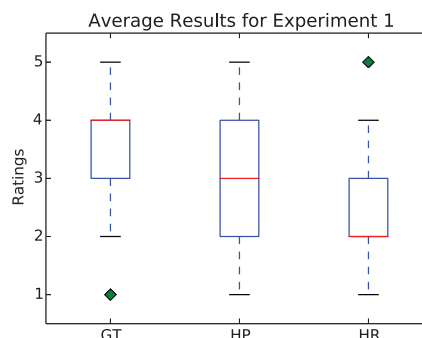
Subjects had to rate the “quality” of the boundaries for each version of the five tracks by choosing a discrete value between 1 and 5 (lowest and highest ratings respectively). Although this might arguably bias the subjects towards the

existing boundaries only (reducing the influence of the missing ones), it is unclear how to design a similar experiment that would avoid this. Excerpts were presented in random order. Participants were asked to listen to all of the excerpts before submitting the results. As in the preliminary experiment, auditory cues for the section boundaries were added to the original audio signal in the form of a salient sharp sound. For this experiment, no visual feedback was provided because the excerpts were short enough for listeners to retain a general perception of the accuracy of the boundaries. The entire experiment lasted around 15 minutes (5 excerpts  $\times$  3 versions  $\times$  one minute per excerpt) and was available on line<sup>2</sup> as a web survey in order to facilitate participation.

An announcement to various specialized mailing lists was sent in order to recruit participants. As such, most subjects had a professional interest in music, and some were even familiar with the topic of musical structure analysis. A total number of 48 participants took part in the experiment; subjects had an average of  $3.1 \pm 1.6$  years of musical training and  $3.7 \pm 3.3$  years of experience playing an instrument.

#### 3.3 Results and Discussion

Box plots of accuracy ratings across versions can be seen in Figure 2. These experimental results show that higher accuracy ratings were assigned to GT followed by HP, and then HR.



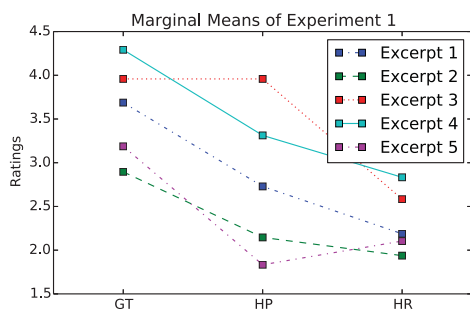
**Figure 2.** Average ratings across excerpts for Experiment 1; GT = ground truth; HP = high precision; HR = high recall.

A two-way, repeated-measures ANOVA was performed on the accuracy ratings with type (ground truth, high precision, high recall) and excerpt (the five songs) as factors. There were 48 data points in each Type  $\times$  Excerpt category. The main effects of type,  $F(2, 94) = 90.74$ ,  $MSE = 1.10$ ,  $p < .001$ , and excerpt,  $F(4, 188) = 59.84$ ,  $MSE = 0.88$ ,  $p < .001$ , were significant. There was also an interaction effect,  $F(6.17, 290.01) = 9.42$ ,  $MSE = 0.74$ ,  $p < .001$  (Greenhouse-Geisser corrected), indicating that rating profiles differed based on excerpt. Mean ratings by type and excerpt are shown in Figure 3.

Looking at the data for each excerpt, there was a clear pattern showing that subjects preferred segmentations with

<sup>2</sup> <http://urinieto.com/NYU/BoundaryExperiment/>

high precision over high recall (Figure 3). Post-hoc multiple comparisons indicated that differences between means of all three types were significant. The only excerpt where precision was not rated more highly than recall was in Excerpt 5 (Tubthumping), a difference that contributed primarily to the interaction. In this case, the excerpt contains a distinctive chorus where the lyrics “I get knocked down” keep repeating. This feature is likely the reason some subjects were led to interpret every instance of this refrain as a possible section beginning even though the harmony underneath follows a longer sectional pattern that is annotated in the ground truth. On the other hand, Excerpt 3 (Intergalactic) obtained similar ratings for ground truth and high precision, likely due to the high number of different sections and silences it contains. This can become problematic when extra boundaries are added (therefore obtaining poor ratings for the high-recall version). Nevertheless, given the subjectivity of this task [2] and the multi-layer organization of boundaries [10], it is not surprising that this type of variability appears in the results.



**Figure 3.** Means for excerpt and version of the results of Experiment 1.

The results of this experiment show that precision is more perceptually relevant than recall for the evaluation of boundaries, validating the preliminary findings (Section 2.2) in a controlled scenario and with a much larger population of subjects. Nevertheless, the number of tracks employed in this experiment was limited. As a follow-up, we explored these findings using a larger dataset in Experiment 2.

## 4. EXPERIMENT 2: CHOOSING BOUNDARIES

### 4.1 Motivation

The results of Experiment 1 show the relative importance of precision over recall for a reduced dataset of five tracks. However, it remains to be seen whether the F-measure, precision, and recall can predict a listener’s preference when faced with a real-world evaluation scenario (i.e., boundaries not synthesized but estimated from algorithms). How this information can be used to redesign the metric to be more perceptually relevant is another question. In Experiment 2, we used excerpts sampled from a larger set of music, boundaries computed with state-of-the-art algorithms (thus recreating a real-world evaluation *à la* MIREX), and limited the evaluation to pairwise preferences.

### 4.2 Methodology

The analysis methods used to compute the boundaries included structural features (SF, [12]), convex non-negative matrix factorization (C-NMF, [7]), and shift-invariant probabilistic latent component analysis (SI-PLCA, [17]). These three algorithms yield ideal results for our experimental design since SF provides one of the best results reported so far on boundaries recognition (high precision and high recall) footnote Recently challenged by Ordinal Linear Discriminant Analysis [6]. C-NMF tends to over segment (higher recall than precision), and SI-PLCA, depending on parameter choices, tends to under segment (higher precision than recall).

We ran these three algorithms on a database of 463 songs composed of the conjunction of the TUT Beatles dataset,<sup>3</sup> the Levy catalogue [5], and the freely available songs of the SALAMI dataset [13]. Once computed, we filtered the results based on the following criteria for each song: (1) at least two algorithm outputs have a similar F-measure (within a 5% threshold); (2) the F-measure of both algorithms must be at least 45%; (3) at least a 10% difference between the precision and recall values of the two selected algorithm outputs exists.

We found 41 out of 463 tracks that met the above criteria. We made a qualitative selection of these filtered tracks (there are many free tracks in the SALAMI dataset that are live recordings with poor audio quality or simply speech), resulting in a final set of 20 songs. The number of these carefully selected tracks is relatively low, but we expect it to be representative enough to address our research questions. Given the two algorithmic outputs maximizing the difference between precision and recall, two differently segmented versions were created for each track: high precision (HP) and high recall (HR). Moreover, similar to Experiment 1, only one minute of audio from each track was utilized, starting 15 seconds into the song.

Table 3 shows average metrics across the 20 selected tracks. The F-measures are the same, while precision and recall vary.

Boundaries Version	F	P	R
HP	.65	.82	.56
HR	.65	.54	.83

**Table 3.** Average F-measure, precision, and recall values for the two versions of excerpts used in Experiment 2.

As in Experiment 1, the interface for Experiment 2 was on line<sup>4</sup> to facilitate participation. Each participant was presented with five random excerpts selected from the set of 20. Instead of assessing the accuracy on a scale, listeners had to choose the version they found more accurate. In order uniformly distribute excerpts across total trials, selection of excerpts was constrained by giving more priority to those excerpts with fewer collected responses. We obtained an average of 5.75 results per excerpt. The two versions were presented in random order, and subjects had

<sup>3</sup> [http://www.cs.tut.fi/sgn/arg/paulus/beatles\\_sections.TUT.zip](http://www.cs.tut.fi/sgn/arg/paulus/beatles_sections.TUT.zip)

<sup>4</sup> <http://cognition.smusic.nyu.edu/boundaryExperiment2/>

to listen to the audio at least once before submitting the results. Boundaries were marked with a salient sound like in the prior experiments.

A total 23 subjects, recruited from professional mailing lists, participated in the experiment. Participants had an average of  $2.8 \pm 1.4$  years of musical training and  $3.2 \pm 2.9$  years of experience playing an instrument.

### 4.3 Results and Discussion

We performed binary logistic regression analysis [11] on the results with the goal of understanding what specific values of the F-measure were actually useful in predicting subject preference (the binary values representing the versions picked by the listeners). Logistic regression enables us to compute the following probability:

$$P(Y|X_1, \dots, X_n) = \frac{e^{k+\beta_1 X_1 + \dots + \beta_n X_n}}{1 + e^{k+\beta_1 X_1 + \dots + \beta_n X_n}} \quad (3)$$

where  $Y$  is the dependent, binary variable,  $X_i$  are the predictors,  $\beta_i$  are the weights for these predictors, and  $k$  is a constant value. Parameters  $\beta_i$  and  $k$  are learned through the process of training the regressor. In our case,  $Y$  tells us whether a certain excerpt was chosen or not according to the following predictors: the F-measure ( $X_1$ ), the signed difference between precision and recall ( $X_2$ ), and the absolute difference between precision and recall ( $X_3$ ).

Since 23 subjects took part in the experiment and there were five different tracks with two versions per excerpt, we had a total of  $23 \times 5 \times 2 = 230$  observations as input to the regression with the parameters defined above. We ran the Hosmer & Lemeshow test [4] in order to understand the predictive ability of our input data. If this test is not statistically significant ( $p > 0.05$ ), we know that logistic regression can indeed help us predict  $Y$ . In our case, we obtain a value of  $p = .763$  ( $\chi^2 = 4.946$ , with 8 degrees of freedom) which tells us that the data for this type of analysis fits well, and that the regressor has predictive power.

The analysis of the results of the learned model is shown in Table 4. As expected, the F-measure is not able to predict the selected version ( $p = .992$ ), providing clear evidence that the metric is inexpressive and perceptually irrelevant for the evaluation of segmentation algorithms. Furthermore, we can see that  $P - R$  can predict the results in a statistically significant manner ( $p = .000$ ), while the absolute difference  $|P - R|$ , though better than the F-measure, has low predictive power ( $p = .482$ ). This clearly illustrates the asymmetrical relationship between P and R: it is not sufficient that P and R are different, but the sign matters: P has to be higher than R.

Based on this experiment we can claim that, for these set of tracks, (1) the F-measure does not sufficiently characterize the perception of boundaries, (2) precision is clearly more important than recall, and (3) there might be a better parameterization of the F-measure that encodes relative importance. We attempt to address this last point in the next section.

Logistic Regression Analysis of Experiment 2						
Predictor	$\beta$	S.E. $\beta$	Wald's $\chi^2$	df	$p$	$e^\beta$
F-measure	-.012	1.155	.000	1	.992	.988
$P - R$	2.268	.471	23.226	1	.000	1.023
$ P - R $	-.669	.951	.495	1	.482	.512
$k$	.190	.838	.051	1	.821	1.209

**Table 4.** Analysis of Experiment 2 data using logistic regression. According to these results,  $P - R$  can predict the version of the excerpt that subjects will choose.

## 5. ENHANCING THE F-MEASURE

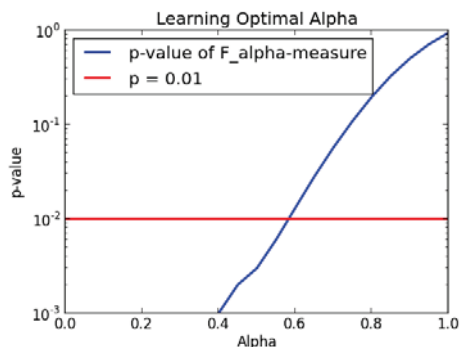
Based on our experiments, we have empirical evidence that high precision is perceptually more relevant than high recall for the evaluation of segmentation algorithms. We can then leverage these findings to obtain a more expressive and perceptually informative version of the F-measure for benchmarking estimated boundaries.

The F-measure is, in fact, a special case of the  $F_\alpha$ -measure:

$$F_\alpha = (1 + \alpha^2) \frac{P \cdot R}{\alpha^2 P + R} \quad (4)$$

where  $\alpha = 1$ , resulting in P and R having the same weight. However, it is clear from the equation that we should impose  $\alpha < 1$  in order to give more importance to  $P$  and make the F-measure more perceptually relevant. Note that an algorithm that outputs fewer boundaries does not necessarily increase its  $F_\alpha$ -measure, since the fewer predicted boundaries could still be incorrect. Regardless, the question remains: how is the value of  $\alpha$  determined?

A possible method to answer this question is to sweep  $\alpha$  from 0 to 1 using a step size of 0.05 and perform logistic regression analysis at each step using the  $F_\alpha$ -measure as the only predictor ( $X_1 = F_\alpha$ ,  $n = 1$ ). The  $p$ -value of the  $F_\alpha$ -measure predicting subject preference in Experiment 2 across all  $\alpha$  is shown in Figure 4.



**Figure 4.** Statistical significance of the  $F_\alpha$ -measure predicting the perceptual preference of a given evaluation for  $\alpha \in [0, 1]$

It is important to note that the data from Experiment 2 is limited as it does not include information at the limits of the difference between precision and recall. As a result, our model predicts that decreases of  $\alpha$  always lead to highest predictive power. Naturally, this is undesirable since we will eventually remove all influence from recall in the measure and favor the trivial solutions discussed at

the beginning of this paper. At some point, as  $P - R$  increases, we expect subject preference to decrease, as preserving a minimum amount of recall becomes more important. Therefore, we could choose the first value of  $\alpha$  (0.58) for which  $F_\alpha$ -based predictions of subject preference become accurate at the statistically significant level of 0.01.

We can re-run the evaluation of Experiments 1 and 2 using the  $F_{0.58}$ -measure (i.e.  $\alpha = 0.58$ ) to illustrate that it behaves as expected. For Experiment 1, we obtain 83.3% for HP and 72.1% for HR (instead of 77.7% and 79.4% respectively). For Experiment 2, the values of HP and HR become 71.8% and 58.9% respectively, whereas they were both 65.0% originally. This shows how the new approximated measure is well coordinated with the preferences of the subjects from Experiments 1 and 2, therefore making this evaluation of section boundaries more expressive and perceptually relevant.

This specific  $\alpha$  value is highly dependent on the empirical data, and we are aware of the limitations of using reduced data sets as compared to the real world—in other words, we are likely overfitting to our data. Nonetheless, based on our findings, there must be a value of  $\alpha < 1$  that better represents the relative importance of precision and recall. Future work, utilizing larger datasets and a greater number of participants, should focus on understanding the upper limit of the difference between precision and recall in order to find the specific inflection point at which higher precision is not perceptually relevant anymore.

## 6. CONCLUSIONS

We presented a series of experiments concluding that precision is perceived as more relevant than recall when evaluating boundaries in music. The results of the two main experiments discussed here are available on line.<sup>5</sup> Moreover, we have noted the shortcomings of the current F-measure when evaluating results in a perceptually meaningful way. By using the general form of the F-measure, we can obtain more relevant results when precision is emphasized over recall ( $\alpha < 1$ ). Further steps should be taken in order to determine a more specific and generalizable value of  $\alpha$ .

## 7. ACKNOWLEDGMENTS

This work was partially funded by Fundación Caja Madrid and by the National Science Foundation, under grant IIS-0844654.

## 8. REFERENCES

- [1] G. Boutard, S. Goldszmidt, and G. Peeters. Browsing Inside a Music Track, The Experimentation Case Study. In *Proc. of the Workshop of Learning the Semantics of Audio Signals*, pages 87–94, 2006.
- [2] M. J. Bruderer, M. F. McKinney, and A. Kohlrausch. The Perception of Structural Boundaries in Melody Lines of Western Popular Music. *MusicaScientia*, 13(2):273–313, 2009.
- [3] C. Cannam, C. Landone, M. Sandler, and J. P. Bello. The Sonic Visualiser: A Visualisation Platform for Semantic Descriptors from Musical Signals. In *Proc. of the 7th International Conference on Music Information Retrieval*, pages 324–327, Victoria, BC, Canada, 2006.
- [4] D. W. Hosmer and S. Lemeshow. *Applied Logistic Regression*. John Wiley & Sons, 2004.
- [5] M. Levy and M. Sandler. Structural Segmentation of Musical Audio by Constrained Clustering. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):318–326, Feb. 2008.
- [6] B. McFee and D. P. W. Ellis. Learning to Segment Songs with Ordinal Linear Discriminant Analysis. In *Proc. of the 39th IEEE International Conference on Acoustics Speech and Signal Processing*, Florence, Italy, 2014.
- [7] O. Nieto and T. Jehan. Convex Non-Negative Matrix Factorization For Automatic Music Structure Identification. In *Proc. of the 38th IEEE International Conference on Acoustics Speech and Signal Processing*, pages 236–240, Vancouver, Canada, 2013.
- [8] B. S. Ong and P. Herrera. Semantic Segmentation of Music Audio Contents. In *Proc. 32nd of the International Computer Music Conference*, Barcelona, Spain, 2005.
- [9] J. Paulus, M. Müller, and A. Klapuri. Audio-Based Music Structure Analysis. In *Proc of the 11th International Society of Music Information Retrieval*, pages 625–636, Utrecht, Netherlands, 2010.
- [10] G. Peeters and E. Deruty. Is Music Structure Annotation Multi-Dimensional? A Proposal for Robust Local Music Annotation. In *Proc. of the 3rd International Workshop on Learning Semantics of Audio Signals*, pages 75–90, Graz, Austria, 2009.
- [11] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll. An Introduction to Logistic Regression Analysis and Reporting. *The Journal of Educational Research*, 96(1):3–14, Sept. 2002.
- [12] J. Serrà, M. Müller, P. Grosche, and J. L. Arcos. Unsupervised Music Structure Annotation by Time Series Structure Features and Segment Similarity. *IEEE Transactions on Multimedia, Special Issue on Music Data Mining*, 2014.
- [13] J. B. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure, and J. S. Downie. Design and Creation of a Large-Scale Database of Structural Annotations. In *Proc. of the 12th International Society of Music Information Retrieval*, pages 555–560, Miami, FL, USA, 2011.
- [14] J. B. L. Smith. *A Comparison And Evaluation Of Approaches To The Automatic Formal Analysis Of Musical Audio*. Master’s thesis, McGill University, 2010.
- [15] B. Tillmann and E. Bigand. Global Context Effect in Normal and Scrambled Musical Sequences. *Journal of Experimental Psychology: Human Perception and Performance*, 27(5):1185–1196, 2001.
- [16] D. Turnbull, G. Lanckriet, E. Pampalk, and M. Goto. A Supervised Approach for Detecting Boundaries in Music Using Difference Features and Boosting. In *Proc. of the 5th International Society of Music Information Retrieval*, pages 42–49, Vienna, Austria, 2007.
- [17] R. Weiss and J. P. Bello. Unsupervised Discovery of Temporal Structure in Music. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1240–1251, 2011.

<sup>5</sup> <http://www.urinieto.com/NYU/ISMIR14-BoundariesExperiment.zip>

# KEYWORD SPOTTING IN A-CAPELLA SINGING

Anna M. Kruspe

Fraunhofer IDMT, Ilmenau, Germany

Johns Hopkins University, Baltimore, MD, USA

kpe@idmt.fhg.de

## ABSTRACT

Keyword spotting (or spoken term detection) is an interesting task in Music Information Retrieval that can be applied to a number of problems. Its purposes include topical search and improvements for genre classification. Keyword spotting is a well-researched task on pure speech, but state-of-the-art approaches cannot be easily transferred to singing because phoneme durations have much higher variations in singing. To our knowledge, no keyword spotting system for singing has been presented yet.

We present a keyword spotting approach based on keyword-filler Hidden Markov Models (HMMs) and test it on a-capella singing and spoken lyrics. We test Mel-Frequency Cepstral Coefficients (MFCCs), Perceptual Linear Predictive Features (PLPs), and Temporal Patterns (TRAPs) as front ends. These features are then used to generate phoneme posteriors using Multilayer Perceptrons (MLPs) trained on speech data. The phoneme posteriors are then used as the system input. Our approach produces useful results on a-capella singing, but depend heavily on the chosen keyword. We show that results can be further improved by training the MLP on a-capella data.

We also test two post-processing methods on our phoneme posteriors before the keyword spotting step. First, we average the posteriors of all three feature sets. Second, we run the three concatenated posteriors through a fusion classifier.

## 1. INTRODUCTION

Keyword spotting is the task of searching for certain words or phrases (spoken term detection) in acoustic data. In contrast to text data, we cannot directly search for these words, but have to rely on the output of speech recognition systems in some way.

In speech, this problem has been a topic of research since the 1970's [1] and has since seen a lot of development and improvement [11]. For singing, however, we are not aware of any fully functional keyword spotting systems.

Music collections of both professional distributors and private users have grown exponentially since the switch to a digital format. For these large collections, efficient search

methods are necessary. Keyword spotting in music collections has beneficial applications for both user groups. Using keyword spotting, users are able to search their collections for songs with lyrics about certain topics. As an example, professional users might use this in the context of synch licensing [4] (e.g., "I need a song containing the word 'freedom' for a car commercial".) Private users could, for example, use keyword spotting for playlist generation ("Generate a playlist with songs that contain the word 'party'.")

In this paper, we present our approach to a keyword spotting system for a-capella singing. We will first look at the current state of the art in section 2. We then present our data set in section 3. In section 4, we describe our own keyword spotting system. A number of experiments on this system and their results are presented in section 5. Finally, we draw conclusions in section 6 and give an outlook on future work in section 7.

## 2. STATE OF THE ART

### 2.1 Keyword spotting principles

As described in [13], there are three basic principles that have been developed over the years for keyword spotting in speech:

**LVCSR-based keyword spotting** For this approach, full Large Vocabulary Continuous Speech Recognition (LVCSR) is performed on the utterances. This results in a complete text transcription, which can then be searched for the required keywords. LVCSR-based systems lack tolerance for description errors - i.e., if a keyword is not correctly transcribed from the start, it cannot be found later. Additionally, LVCSR systems are complex and expensive to implement.

**Acoustic keyword spotting** As in LVCSR-based keyword spotting, acoustic keyword spotting employs Viterbi search to find the requested keyword in a given utterance. In this approach, however, the system does not attempt to transcribe each word, but only searches for the specific keyword. Everything else is treated as "filler". This search can be performed directly on the audio features using an acoustic example, or on phoneme posteriorgrams generated by an acoustic model. In the second case, the algorithm searches for the word's phonemes.

This approach is easy to implement and provides some pronunciation tolerance. Its disadvantage is



© Anna M. Kruspe.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Anna M. Kruspe. "Keyword spotting in a-capella singing", 15th International Society for Music Information Retrieval Conference, 2014.

the lack of integration of a-priori language knowledge (i.e. knowledge about plausible phoneme and word sequences) that could improve performance.

**Phonetic search keyword spotting** Phonetic search keyword spotting starts out just like LVCSR-based keyword spotting, but does not generate a word transcription of the utterance. Instead, phoneme lattices are saved. Phonetic search for the keyword is then performed on these lattices. This approach combines the advantages of LVCSR-based keyword spotting (a-priori knowledge in the shape of language models) and acoustic keyword spotting (flexibility and robustness).

## 2.2 Keyword spotting in singing

The described keyword spotting principles cannot easily be transferred to music. Singing, in contrast to speech, presents a number of additional challenges, such as larger pitch fluctuation, more pronunciation variation, and different vocabulary (which means existing models cannot easily be transferred).

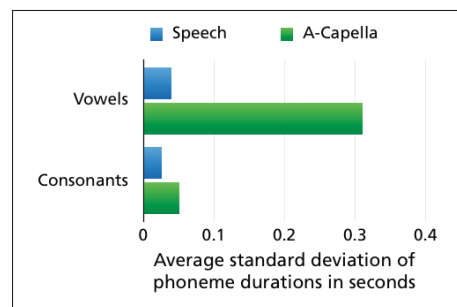
Another big difference is the higher variation of phoneme durations in singing. Both LVCSR-based keyword spotting and Phonetic search keyword spotting depend heavily on predictable phoneme durations (within certain limits). When a certain word is pronounced, its phonemes will usually have approximately the same duration across speakers. The language model employed in both approaches will take this information into account.

We compared phoneme durations in the TIMIT speech database [7] and our own a-capella singing database (see section 3). The average standard deviations for vowels and consonants are shown in figure 1. It becomes clear that the phoneme durations taken from TIMIT do not vary a lot, whereas some the a-capella phonemes show huge variations. It becomes clear that this especially concerns vowels (AA, AW, EH, IY, AE, AH, AO, EY, AY, ER, UW, OW, UH, IH, OY). This observation has a foundation in music theory: Drawn-out notes are usually sung on vowels.

For this reason, acoustic keyword spotting appears to be the most feasible approach to keyword spotting in singing. To our knowledge, no full keyword spotting system for singing has been presented yet. In [2], an approach based on sub-sequence Dynamic Time Warping (DTW) is suggested. This is similar to the acoustic approach, but does not involve a full acoustic model. Instead, example utterances of the keyword are used to find similar sequences in the tested utterance.

In [5], a phoneme recognition system for singing is presented. It extracts Mel-Frequency Cepstral Coefficients (MFCCs) and Temporal Patterns (TRAPs) which are then used as inputs to a Multilayer Perceptron (MLP). The phonetic output of such a system could serve as an input to a keyword spotting system.

There are also some publications where similar principles are applied to lyrics alignment and Query by Humming [12] [3].



**Figure 1:** Average standard deviations for vowels and consonants in the TIMIT speech databases (blue) and our a-capella singing data set (green).

## 3. DATA SET

Our data set is the one presented in [5]. It consists of the vocal tracks of 19 commercial pop songs. They are studio quality with some post-processing applied (EQ, compression, reverb). Some of them contain choir singing. These 19 songs are split up into clips that roughly represent lines in the song lyrics.

Twelve of the songs were annotated with time-aligned phonemes. The phoneme set is the one used in CMU Sphinx<sup>1</sup> and TIMIT [7] and contains 39 phonemes. All of the songs were annotated with word transcriptions. For comparison, recordings of spoken recitations of all song lyrics were also made. These were all performed by the same speaker.

We selected 51 keywords for testing our system. Most of them were among the most frequent words in the provided lyrics. A few were selected because they had a comparatively large number of phonemes. An overview is given in table 1.

## 4. PROPOSED SYSTEM

Figure 2 presents an overview of our system.

**1. Feature extraction** We extract Mel-Frequency Cepstral Coefficients (MFCCs), Perceptual Linear Predictive features (PLPs), and Temporal Patterns (TRAPs) [6]. We keep 20 MFCC coefficients and 39 PLP coefficients (13 direct coefficients plus deltas and double-deltas). For the TRAPs, we use 8 linearly spaced spectral bands and a temporal context of 20 frames and keep 8 DCT coefficients.

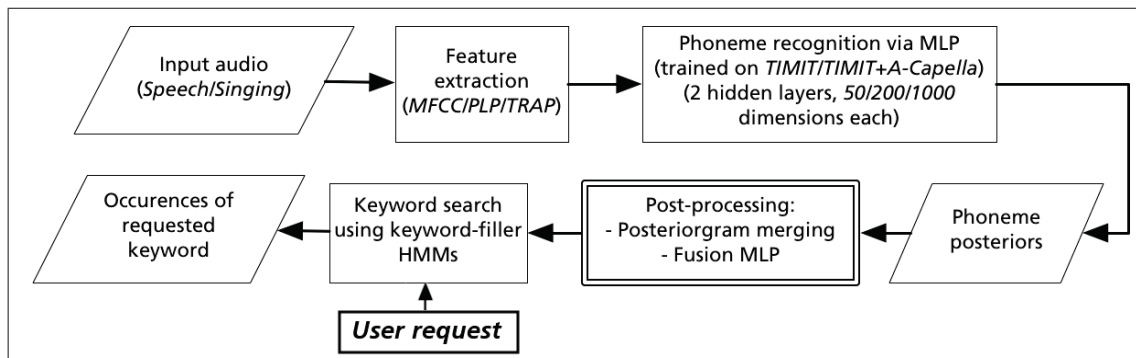
**2. MLP training and phoneme recognition** Using each feature data set, we train Multi-Layer Perceptrons (MLPs). MLPs are commonly used to train acoustic models for the purpose of phoneme recognition. We chose a structure with two hidden layers and tested three different dimension settings: 50, 200, and 1000 dimensions per layer. MLPs were trained solely on TIMIT data first, then on a mix of TIMIT and a-capella in a second experiment. The resulting MLPs are then used to recognize phonemes in our a-capella dataset, thus generating phoneme posteriorgrams.

<sup>1</sup> <http://cmusphinx.sourceforge.net/>



Number of Phonemes	Keywords
2	way, eyes
3	love, girl, away, time, over, home, sing, kiss, play, other
4	hello, trick, never, hand, baby, times, under, things, world, think, heart, tears, lights
5	always, inside, drink, nothing, rehab, forever, rolling, feeling, waiting, alright, tonight
6	something, denial, together, morning, friends, leaving, sunrise
7	umbrella, afternoon, stranger, somebody, entertain, everyone
8	beautiful, suicidal

**Table 1:** All 51 tested keywords, ordered by number of phonemes.



**Figure 2:** Overview of our keyword spotting system. Variable parameters are shown in italics.

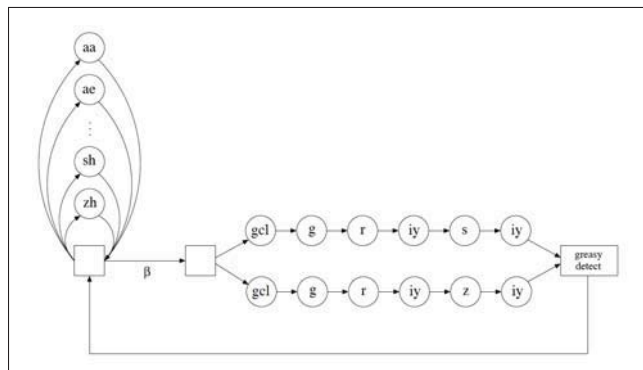
The following two points described optional post-processing steps on the phoneme posteriorgrams.

**3a. Posteriorgram merging** For this post-processing step, we take the phoneme posterior results that were obtained using different feature sets and average them. We tested both the combinations of PLP+MFCC, PLP+TRAP, and PLP+MFCC+TRAP.

**3b. Fusion MLP classifier** As a second post-processing option, we concatenate phoneme posteriors obtained by using different feature sets and run them through a fusion MLP classifier to create better posteriors. We again tested the combinations PLP+MFCC, PLP+TRAP, and PLP+MFCC+TRAP.

**4. Keyword spotting** The resulting phoneme posteriorgrams are then used to perform the actual keyword spotting. As mentioned above, we employ an acoustic approach. It is based on keyword-filler Hidden Markov Models (HMMs) and has been described in [14] and [8].

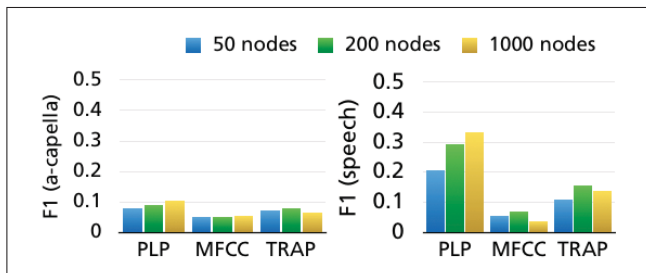
In general, two separate HMMs are created: One for the requested keyword, and one for all non-keyword regions (=filler). The keyword HMM is generated using a simple left-to-right topology with one state per keyword phoneme, while the filler HMM is a fully connected loop of states for all phonemes. These two HMMs are then joined. Using this composite HMM, a Viterbi decode is performed on the phoneme posteriorgrams. Whenever the Viterbi path passes through the keyword HMM, the keyword is detected. The likelihood of this path can then be compared to an alternative path through the filler HMM, resulting in a detection score. A threshold



**Figure 3:** Keyword-filler HMM for the keyword “greasy” with filler path on the left hand side and two possible keyword pronunciation paths on the right hand side. The parameter  $\beta$  determines the transition probability between the filler HMM and the keyword HMM. [8]

can be employed to only return highly scored occurrences. Additionally, the parameter  $\beta$  can be tuned to adjust the model. It determines the likelihood of transitioning from the filler HMM to the keyword HMM. The whole process is illustrated in figure 3.

We use the  $F_1$  measure for evaluation. Results are considered to be true positives when a keyword is spotted somewhere in an expected utterance. Since most utterances contain one to ten words, we consider this to be sufficiently exact. Additionally, we evaluate the precision of the results. For the use cases described in section 1, users will usually only require a number of correct results, but not necessarily all the occurrences of the keyword in the whole database. We consider a result to be correct when the keyword is found as part of another word with the same pronunciation. The reasoning behind this is that a user who searched



**Figure 4:**  $F_1$  measures for a-capella data (left) and speech (right) when using PLP, MFCC, or TRAP features. The MLPs for phoneme recognition had two hidden layers with 50, 200, or 1000 nodes each.

for the keyword “time” might also accept occurrences of the word “times” as correct.

## 5. EXPERIMENTS

### 5.1 Experiment 1: Oracle search

As a precursor to the following experiments, we first tested our keyword spotting approach on oracle posteriorgrams for the a-capella data. This was done to test the general feasibility of the algorithm for keyword spotting on singing data with its highly variable phoneme durations.

The oracle posteriorgrams were generated by converting the phoneme annotations to posteriorgram format by setting the likelihoods of the annotated phonemes to 1 during the corresponding time segment and everything else to 0. A keyword search on these posteriorgrams resulted in  $F_1$  measures of 1 for almost all keywords. In cases where the result was not 1, we narrowed the reasons down to annotation errors and pronunciation variants that we did not account for. We conclude that our keyword-filler approach is generally useful for keyword spotting on a-capella data, and our focus in the following experiments is on obtaining good posteriorgrams from the audio data.

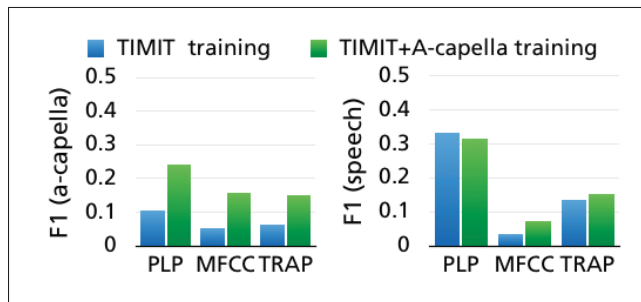
### 5.2 Experiment 2: A-Capella vs. Speech

For our first experiment, we run our keyword spotting system on the a-capella singing data, and on the same utterances spoken by a single speaker. We evaluate all three feature datasets (MFCC, PLP, TRAP) separately. The recognition MLP is trained on TIMIT speech data only. We also test three different sizes for the two hidden MLP layers: 50 nodes, 200 nodes, and 1000 nodes in each layer. The results are shown in figure 4.

As described in section 2.2, we expected keyword spotting on singing to be more difficult than on pure speech because of a larger pitch range, more pronunciation variations, etc. Our results support this assumption: In speech, keywords are recognized with an average  $F_1$  measure of 33% using only PLP features, while the same system results in an average  $F_1$  of only 10% on a-capella singing.

For both data sets, an MLP with 200 nodes in the hidden layers shows a notable improvement over one with just 50. When using 1000 nodes, the result still improves by a few percent in most cases.

When looking at the features, PLP features seem to work



**Figure 5:**  $F_1$  measures for a-capella data (left) and speech (right) when the recognition is trained only on TIMIT speech data (blue) or on a mix of TIMIT and a-capella data (green).

best by a large margin, with TRAPs coming in second. It is notable, however, that some keywords can be detected much better when using MFCCs or TRAPs than PLPs (e.g. “sing”, “other”, “hand”, “world”, “tears”, “alright”). As described in [5] and [10], certain feature sets represent some phonemes better than others and can therefore balance each other out. A combination of the features might therefore improve the whole system.

Evaluation of the average precision (instead of  $F_1$  measure) shows the same general trend. The best results are again obtained when using PLP features and the largest MLP. The average precision in this configuration is 16% for a-capella singing and 37% for speech. (While the difference is obvious, the result is still far from perfect for speech. This demonstrates the difficulty of the recognition process without a-priori knowledge.)

### 5.3 Experiment 3: Training including a-capella data

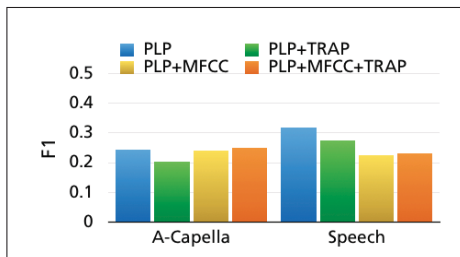
As a measure to improve the phoneme posteriorgrams for a-capella singing, we next train our recognition MLP with both TIMIT and a part of the a-capella data. We mix in about 50% of the a-capella clips with the TIMIT data. They make up about 10% of the TIMIT speech data. The results are shown in figure 5 (only the results for the largest MLP are shown).

This step improves the keyword recognition on a-capella data massively in all feature and MLP configurations. The best result still comes from the biggest MLP when using PLP features and is now an average  $F_1$  of 24%. This step makes the recognition MLP less specific to the properties of pure speech and therefore does not improve the results for the speech data very much. It actually degrades the best result somewhat.

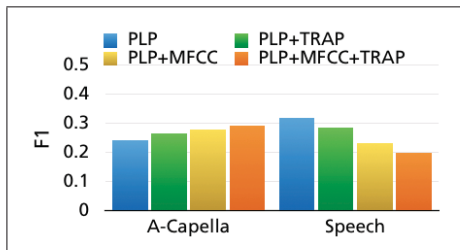
The effect on the average precision is even greater. The a-capella results are improved by 10 to 15 percentage points for each feature set. On speech data, the PLP precision decreases by 7 percentage points.

### 5.4 Experiment 4: Posterior merging

As mentioned in experiment 2, certain feature sets seem to represent some keywords better than others. We therefore concluded that combining the results for all features could improve the recognition result.



**Figure 6:**  $F_1$  measures for a-capella data (left) and speech (right) when posteriorgrams for two or three features are merged. The configurations PLP+MFCC, PLP+TRAP, and PLP+MFCC+TRAP are shown and compared to the PLP only result.



**Figure 7:**  $F_1$  measures for a-capella data (left) and speech (right) when posteriorgrams for two or three features are fed into a fusion classifier. The configurations PLP+MFCC, PLP+TRAP, and PLP+MFCC+TRAP are shown and compared to the PLP only result.

To this end, we tested merging the phoneme posteriorgrams between the MLP phoneme recognition step and the HMM keyword spotting step. In order to do this, we simply calculated the average values across the posteriors obtained using the three different feature data sets. This was done for all phonemes and time frames. Keyword spotting was then performed on the merged posteriorgrams. We tested the configurations PLP+MFCC, PLP+TRAP, and PLP+MFCC+TRAP. The results are shown in figure 6.

Posterior merging seems to improve the results for a-capella singing somewhat and works best when all three feature sets are used. The  $F_1$  measure on a-capella singing improves from 24% (PLP) to 27%. It does not improve the speech result, where PLP remains the best feature set.

### 5.5 Experiment 5: Fusion classifier

After the posterior merging, we tested a second method of combining the feature-wise posteriorgrams. In this second method, we concatenated the posteriorgrams obtained from two or all three of the feature-wise MLP recognizers and ran them through a second MLP classifier. This fusion MLP was trained on a subset of the a-capella data. This fusion classifier generates new, hopefully improved phoneme posteriorgrams. HMM keyword spotting is then performed on these new posteriorgrams. We again tested the configurations PLP+MFCC, PLP+TRAP, and PLP+MFCC+TRAP. The results are shown in figure 7. The fusion classifier improves the  $F_1$  measure for a-capella singing by 5 percentage points. The best result of 29% is obtained when all three feature sets are used. Precision

improves from 24% to 31%. However, the fusion classifier makes the system less specific towards speech and therefore decreases the performance on speech data.

### 5.6 Variation across keywords

The various results we presented in the previous experiments varies widely across the 51 keywords. This is a common phenomenon in keyword spotting. In many approaches, longer keywords are recognized better than shorter ones because the Viterbi path becomes more reliable with each additional phoneme. This general trend can also be seen in our results, but even keywords with the same number of phonemes vary a lot. The precisions vary similarly, ranging between 2% and 100%.

When taking just the 50% of the keywords that can be recognized best, the average  $F_1$  measure for the best approach (fusion MLP) jumps from 29% to 44%. Its precision increases from 31% to 46%. We believe the extremely bad performance of some keywords is in part due to the small size of our data set. Some keywords occurred in just one of the 19 songs and were, for example, not recognized because the singer used an unusual pronunciation in each occurrence or had an accent that the phoneme recognition MLP was not trained with. We therefore believe these results could improve massively when more training data is used.

## 6. CONCLUSION

In this paper, we demonstrated a first keyword spotting approach for a-capella singing. We ran experiments for 51 keywords on a database of 19 a-capella pop songs and recordings of the spoken lyrics. As our approach, we selected acoustic keyword spotting using keyword-filler HMMs. Other keyword spotting approaches depend on learning average phoneme durations, which vary a lot more in a-capella singing than in speech. These approaches therefore cannot directly be transferred.

As a first experiment, we tested our approach on oracle phoneme posteriorgrams and obtained almost perfect results. We then produced “real world” posteriorgrams using MLPs with two hidden layers which had been trained on TIMIT speech data. We tested PLP, MFCC, and TRAP features. The training yielded MLPs with 50, 200, and 1000 nodes per hidden layer. We observed that the 200 node MLP produced significantly better results than the 50 node MLPs in all cases ( $p < 0.0027$ ), while the 1000 node MLPs only improved upon this result somewhat. PLP features performed significantly better than the two other feature sets. Finally, keywords were detected much better in speech than in a-capella singing. We expected this result due to the specific characteristics of singing data (higher variance of frequencies, more pronunciation variants).

We then tried training the MLPs with a mixture of TIMIT speech data and a portion of our a-capella data. This improved the results for a-capella singing greatly.

We noticed that some keywords were recognized better when MFCCs or TRAPs were used instead of PLPs. We therefore tried two approaches to combine the results for

all three features: Posterior merging and fusion classifiers. Both approaches improved the results on the a-capella data. The best overall result for a-capella data was produced by a fusion classifier that combined all three features (29%).

As expected, keyword spotting on a-capella singing proved to be a harder task than on speech. The results varied widely between keywords. Some of the very low results arise because the keyword in question only occurred in one song where the singer used an unusual pronunciation or had an accent. The small size of our data set also poses a problem when considering the limited number of singers. The acoustic model trained on speech data and a part of the a-capella data might be subject to overfitting to the singers' vocal characteristics.

In contrast, the recognition worked almost perfectly for keywords with more training data. Keyword length also played a role. When using only the 50% best keywords, the average  $F_1$  measure increased by 15 percentage points. Finally, there are many applications where precision plays a greater role than recall, as described in section 4. Our system can be tuned to achieve higher precisions than  $F_1$  measures and is therefore also useful for these applications. We believe that the key to better keyword spotting results lies in better phoneme posteriorgrams. A larger a-capella data set would therefore be very useful for further tests and would provide more consistent results.

## 7. FUTURE WORK

As mentioned in section 2, more sophisticated keyword spotting systems for speech incorporate knowledge about plausible phoneme durations (e.g. [9]). In section 2.2, we showed why this approach is not directly transferable to singing: The vowel durations vary too much. However, consonants are not affected. We would therefore like to start integrating knowledge about average consonant durations in order to improve our keyword spotting system. In this way, we hope to improve the results for the keywords that were not recognized well by our system.

Following this line of thought, we could include even more language-specific knowledge in the shape of a language model that also contains phonotactic information, word frequencies, and phrase frequencies. We could thus move from a purely acoustic approach to a phonetic (lattice-based) approach.

We will also start applying our approaches to polyphonic music instead of a-capella singing. To achieve good results on polyphonic data, pre-processing will be necessary (e.g. vocal activity detection and source separation).

## 8. REFERENCES

- [1] J. S. Bridle. An efficient elastic-template method for detecting given words in running speech. In *Brit. Acoust. Soc. Meeting*, pages 1 – 4, 1973.
- [2] C. Dittmar, P. Mercado, H. Grossmann, and E. Cano. Towards lyrics spotting in the SyncGlobal project. In *3rd International Workshop on Cognitive Information Processing (CIP)*, 2012.
- [3] H. Fujihara and M. Goto. Three techniques for improving automatic synchronization between music and lyrics: Fricative detection, filler model, and novel feature vectors for vocal activity detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 69–72, Las Vegas, NV, USA, 2008.
- [4] H. Grossmann, A. Kruspe, J. Abesser, and H. Lukashovich. Towards cross-modal search and synchronization of music and video. In *International Congress on Computer Science Information Systems and Technologies (CSIST)*, Minsk, Belarus, 2011.
- [5] J. K. Hansen. Recognition of phonemes in a-cappella recordings using temporal patterns and mel frequency cepstral coefficients. In *9th Sound and Music Computing Conference (SMC)*, pages 494–499, Copenhagen, Denmark, 2012.
- [6] H. Hermansky and S. Sharma. Traps – classifiers of temporal patterns. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*, pages 1003–1006, Sydney, Australia, 1998.
- [7] J. S. Garofolo et al. TIMIT Acoustic-Phonetic Continuous Speech Corpus. Technical report, Linguistic Data Consortium, Philadelphia, 1993.
- [8] A. Jansen and P. Niyogi. An experimental evaluation of keyword-filler hidden markov models. Technical report, Department of Computer Science, University of Chicago, 2009.
- [9] K. Kintzley, A. Jansen, K. Church, and H. Hermansky. Inverting the point process model for fast phonetic keyword search. In *INTERSPEECH*. ISCA, 2012.
- [10] A. M. Kruspe, J. Abesser, and C. Dittmar. A GMM approach to singing language identification. In *53rd AES Conference on Semantic Audio*, London, UK, 2014.
- [11] A. Mandal, K. R. P. Kumar, and P. Mitra. Recent developments in spoken term detection: a survey. *International Journal of Speech Technology*, 17(2):183–198, June 2014.
- [12] A. Mesaros and T. Virtanen. Automatic recognition of lyrics in singing. *EURASIP Journal on Audio, Speech, and Music Processing*, 2010(4), January 2010.
- [13] A. Moyal, V. Aharonson, E. Tetariy, and M. Gishri. *Phonetic Search Methods for Large Speech Databases*, chapter 2: Keyword spotting methods. Springer, 2013.
- [14] I. Szoeké, P. Schwarz, P. Matejka, L. Burget, M. Karafiat, and J. Cernocký. Phoneme based acoustics keyword spotting in informal continuous speech. In V. Matousek, P. Mautner, and T. Pavelka, editors, *TSD*, volume 3658 of *Lecture Notes in Computer Science*, pages 302–309. Springer, 2005.

# THE IMPORTANCE OF F0 TRACKING IN QUERY-BY-SINGING-HUMMING

Emilio Molina, Lorenzo J. Tardón, Isabel Barbancho, Ana M. Barbancho

Universidad de Málaga, ATIC Research Group, Andalucía Tech,

ETSI Telecomunicación, Campus de Teatinos s/n, 29071 Málaga, SPAIN

emm@ic.uma.es, lorenzo@ic.uma.es, ibp@ic.uma.es, abp@ic.uma.es

## ABSTRACT

In this paper, we present a comparative study of several state-of-the-art F0 trackers applied to the context of query-by-singing-humming (QBSH). This study has been carried out using the well known, freely available, MIR-QBSH dataset in different conditions of added pub-style noise and smartphone-style distortion. For audio-to-MIDI melodic matching, we have used two state-of-the-art systems and a simple, easily reproducible baseline method. For the evaluation, we measured the QBSH performance for 189 different combinations of F0 tracker, noise/distortion conditions and matcher. Additionally, the overall accuracy of the F0 transcriptions (as defined in MIREX) was also measured. In the results, we found that F0 tracking overall accuracy correlates with QBSH performance, but it does not totally measure the suitability of a pitch vector for QBSH. In addition, we also found clear differences in robustness to F0 transcription errors between different matchers.

## 1. INTRODUCTION

Query-by-singing-humming (QBSH) is a music information retrieval task where short hummed or sung audio clips act as queries. Nowadays, several successful commercial applications for QBSH have been released, such as MusicRadar<sup>1</sup> or SoundHound<sup>2</sup>, and it is an active field of research. Indeed, there is a task for QBSH in MIREX since 2006, and every year novel and relevant approaches can be found.

Typically, QBSH approaches firstly extract the F0 contour and/or a note-level transcription for a given vocal query, and then a set of candidate melodies are retrieved from a large database using a melodic matcher module. In the literature, many different approaches for matching in QBSH can be found: statistical, note vs. note, frame vs. note, frame vs. frame. Generally, state-of-the-art systems for QBSH typically combines different approaches in order to achieve more reliable results [3, 12].

<sup>1</sup> www.doreso.com

<sup>2</sup> www.soundhound.com



© Emilio Molina, Lorenzo J. Tardón, Isabel Barbancho, Ana M. Barbancho.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Emilio Molina, Lorenzo J. Tardón, Isabel Barbancho, Ana M. Barbancho. "The importance of F0 tracking in query-by-singing-humming", 15th International Society for Music Information Retrieval Conference, 2014.

However, even state-of-the-art systems for QBSH have not a totally satisfactory performance in many real-world cases [1], so there is still room for improvement. Nowadays, some challenges related to QBSH are [2]: reliable pitch tracking in noisy environments, automatic song database preparation (predominant melody extraction and transcription), efficient search in very large music collections, dealing with errors of intonation and rhythm in amateur singers, etc.

In this paper, we analyse the performance of various state-of-the-art F0 trackers for QBSH in different conditions of background noise and smartphone-style distortion. For this study, we have considered three different melodic matchers: two state-of-the-art systems (one of which obtained the best results in MIREX 2013), and a simple, easily reproducible baseline method based on frame-to-frame matching using dynamic time warping (DTW). In Figure 1, we show a scheme of our study.

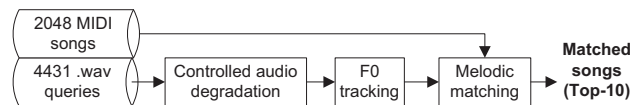


Figure 1. Overall scheme of our study

This paper is organized as follows: Section 2 and Section 3 present the studied algorithms for F0 tracking and melodic matching, respectively. The evaluation strategy is presented in Section 4. Section 5 presents the obtained results and Section 6 draws some conclusions about the present study.

## 2. F0 TRACKERS

In this section, we describe the F0 trackers considered in our study, together with their specific set of parameters. The literature reports a wide set of algorithms oriented to either monophonic or polyphonic audio, so we have focused on well-known, commonly used algorithms (e.g. Yin [4] or Praat-AC [8]), and some recently published algorithms for F0 estimation (e.g. pYin [6] or MELODIA [15]). Most of the algorithms analysed address F0 estimation in monophonic audio, but we have also studied the performance of MELODIA, which is a method for predominant melody extraction in polyphonic audio, using monophonic audio in noisy conditions. Regarding the used set of parameters, when possible, they have been adjusted by trial and error using ten audio queries. The considered methods for F0 tracking are the following ones:

## 2.1 YIN

The Yin algorithm was developed by de Cheveigné and Kawahara in 2002 [4]. It resembles the idea of the autocorrelation method [5] but it uses the cumulative mean normalized difference function, which peaks at the local period with lower error rates than the traditional autocorrelation function. In our study, we have used Matthias Mauch's VAMP plugin<sup>3</sup> in Sonic Annotator tool<sup>4</sup>.

*Parameters used in YIN:* step size = 80 samples (0.01 seconds), Block size = 512 samples, Yin threshold = 0.15.

## 2.2 pYIN

The pYin method has been published by Mauch in 2014 [6], and it basically adds a HMM-based F0 tracking stage in order to find a "smooth" path through the fundamental frequency candidates obtained by Yin. Again, we have used the original Matthias Mauch's VAMP plugin<sup>3</sup> in Sonic Annotator tool<sup>4</sup>.

*Parameters used in PYIN:* step size = 80 samples (0.01 seconds), Block size = 512 samples, Yin threshold distribution = Beta (mean 0.15).

## 2.3 AC-DEFAULT and AC-ADJUSTED (Praat)

Praat is a well-known tool for speech analysis [7], which includes several methods for F0 estimation. In our case, we have chosen the algorithm created by P. Boersma in 1993 [8]. It is based on the autocorrelation method, but it improves it by considering the effects of the window during the analysis and by including a F0 tracking stage based on dynamic programming. This method has 9 parameters that can be adjusted to achieve a better performance for a specific application. According to [9], this method significantly improves its performance when its parameters are adapted to the input signal. Therefore, we have experimented not only with the default set of parameters (AC-DEFAULT), but also with an adjusted set of parameters in order to limit octave jumps and false positives during the voicing process (AC-ADJUSTED). In our case, we have used the implementation included in the console Praat tool.

*Parameters used in AC-DEFAULT:* Time step = 0.01 seconds, Pitch floor = 75Hz, Max. number of candidates = 15, Very accurate = off, Silence threshold = 0.03, Voicing threshold = 0.45, Octave cost = 0.01, Octave-jump cost = 0.35, Voiced / unvoiced cost = 0.15, Pitch ceiling = 600 Hz.

*Parameters used in AC-ADJUSTED:* Time step = 0.01 seconds, Pitch floor = 50Hz, Max. number of candidates = 15, Very accurate = off, Silence threshold = 0.03, Voicing threshold = 0.45, Octave cost = 0.1, Octave-jump cost = 0.5, Voiced / unvoiced cost = 0.5, Pitch ceiling = 700 Hz.

## 2.4 AC-LEIWANG

In our study we have also included the exact F0 tracker used in Lei Wang's approach for QBSH [3], which obtained the best results for most of the datasets in MIREX 2013. It is based on P. Boersma's autocorrelation method

[8], but it uses a finely tuned set of parameters and a post-processing stage in order to mitigate spurious and octave errors. This F0 tracker is used in the latest evolution of a set of older methods [11, 12] also developed by Lei Wang (an open source C++ implementation is available<sup>5</sup>).

## 2.5 SWIPE'

The Swipe' algorithm was published by A. Camacho in 2007 [10]. This algorithm estimates the pitch as the fundamental frequency of the sawtooth waveform whose spectrum best matches the spectrum of the input signal. The algorithm proved to outperform other well-known F0 estimation algorithms, and it is used in the F0 estimation stage of some state-of-the-art query-by-humming systems [13]. In our study, we have used the original author's Matlab implementation<sup>6</sup>. The Matlab code does not provide a voiced / unvoiced classification of frames, but it outputs a strength vector  $S$  which has been used for it. Specifically, a frame is considered voiced if its strength is above a threshold  $S_{th}$ , otherwise they are considered unvoiced.

*Parameters used in SWIPE':* DT (hop-size) = 0.01 seconds, pmin = 50 Hz, pmax = 700Hz, dlog2p = 1/48 (default), dERBs = 0.1 (default), woverlap = 0.5 (default), voicing threshold  $S_{th}$  = 0.3.

## 2.6 MELODIA-MONO and MELODIA-POLY

MELODIA is a system for automatic melody extraction in polyphonic music signals developed by Salamon in 2012 [15]. This system is based on the creation and characterisation of pitch contours, which are time continuous sequences of pitch candidates grouped using auditory streaming cues. Melodic and non-melodic contours are distinguished depending on the distributions of its characteristics. The used implementation is MELODIA VAMP plugin<sup>7</sup> in Sonic Annotator tool<sup>4</sup>. This plugin has two default sets of parameters, adapted to deal with monophonic or polyphonic audio. We have experimented with both of them, and therefore we have defined two methods: MELODIA-MONO and MELODIA-POLY.

*Parameters used in MELODIA-MONO:* Program = Monophonic, Min Frequency = 55Hz, Max Frequency = 700Hz, Voicing Tolerance = 3,00, Monophonic Noise Filter = 0,00, Audio block size = 372 (not configurable), Window increment = 23 (not configurable).

*Parameters used in MELODIA-POLY:* Program = Polyphonic, Min Frequency = 55Hz, Max Frequency = 700Hz, Voicing Tolerance = 0,20, Monophonic Noise Filter = 0,00, Audio block size = 372 (not configurable), Window increment = 23 (not configurable).

Note that the time-step in this case can not be directly set to 0.01 seconds. Therefore, we have linearly interpolated the pitch vector in order to scale it to a time-step of 0.01 seconds.

<sup>5</sup> <http://www.atc.uma.es/ismir2014qbs/>

<sup>6</sup> <http://www.cise.ufl.edu/acamacho/publications/swipep.m>

<sup>7</sup> <http://mtg.upf.edu/technologies/melodia>

<sup>3</sup> <http://code.soundsoftware.ac.uk/projects/pyin>

<sup>4</sup> <http://www.vamp-plugins.org/sonic-annotator/>

### 3. AUDIO-TO-MIDI MELODIC MATCHERS

In this section, we describe the three considered methods for audio-to-MIDI melodic matching: a simple baseline (Section 3.1) and two state-of-the-art matchers (Sections 3.2 and 3.3).

#### 3.1 Baseline approach

We have implemented a simple, freely available<sup>5</sup> baseline approach based on dynamic time warping (DTW) for melodic matching. Our method consists of four steps (a scheme is shown in Figure 2):

(1) *Model building*: We extract one pitch vector  $\mathbf{P}^k$  (in MIDI number) for every target MIDI song  $k \in 1 \dots N_{\text{songs}}$  using a hop-size of 0.01 seconds. Then we replace unvoiced frames (rests) in  $\mathbf{P}^k$  by the pitch value of the previous note, except for the case of initial unvoiced frames, which are directly removed (these processed pitch vectors are labelled as  $\mathbf{P}^{*k}$ ). Then, each pitch vector  $\mathbf{P}^{*k} \forall k \in 1 \dots N_{\text{songs}}$  is truncated to generate 7 pitch vectors with lengths [500, 600, 700, 800, 900, 1000, 1100] frames (corresponding to the first 5, 6, 7, 8, 9, 10 and 11 seconds of the target MIDI song, which are reasonable durations for an user query). We label these pitch vectors as  $\mathbf{P}_{5s}^{*k}$ ,  $\mathbf{P}_{6s}^{*k}$ , ...,  $\mathbf{P}_{11s}^{*k}$ . Finally, all these pitch vectors are resampled (through linear interpolation) to a length of 50 points, and then zero-mean normalized (for a common key transposition), leading to  $\mathbf{P}_{Duration}^{50*k} \forall Duration \in 5s \dots 11s$  and  $\forall k \in 1 \dots N_{\text{songs}}$ . These vectors are then stored for later usage. Note that this process must be done only once.

(2) *Query pre-processing*: The pitch vector  $\mathbf{P}^Q$  of a given .wav query is loaded (note that all pitch vectors are computed with a hopsize equal to 0.01 seconds). Then, as in step (1), unvoiced frames are replaced by the pitch value of the previous note, except for the case of initial unvoiced frames, which are directly removed. This processed vector is then converted to MIDI numbers with 1 cent resolution, and labelled as  $\mathbf{P}^{*Q}$ . Finally,  $\mathbf{P}^{*Q}$  is resampled (using linear interpolation) to a length  $L = 50$  and zero-mean normalized (for a common key transposition), leading to  $\mathbf{P}^{50*Q}$ .

(3) *DTW-based alignment*: Now we find the optimal alignment between  $\mathbf{P}^{50*Q}$  and all pitch vectors  $\mathbf{P}_{Duration}^{50*k} \forall Duration \in 5s \dots 11s$  and  $\forall k \in 1 \dots N_{\text{songs}}$  using dynamic time warping (DTW). In our case, each cost matrix  $\mathcal{C}^{Duration,k}$  is built using the squared difference:

$$\mathcal{C}^{Duration,k}(i,j) = (P^{50*Q}(i) - P_{Duration}^{50*k}(j))^2 \quad (1)$$

Where  $k$  is the target song index,  $Duration$  represents the truncation level (from 5s to 11s), and  $i, j$  are the time indices of the query pitch vector  $\mathbf{P}^{50*Q}$  and the target pitch vector  $\mathbf{P}_{Duration}^{50*k}$ , respectively. The optimal path is now found using Dan Ellis' Matlab implementation for DTW [16] (`dpfast.m` function), with the following allowed steps and associated cost weights  $[\Delta i, \Delta j, W]$ :  $[1, 1, 1]$ ,  $[1, 0, 30]$ ,  $[0, 1, 30]$ ,  $[1, 2, 5]$ ,  $[2, 1, 5]$ . The allowed steps and weights have been selected in order to penalize 0 or 90 angles in the optimal path (associated to unnatural alignments), and although they lead to acceptable results, they may not be optimal.

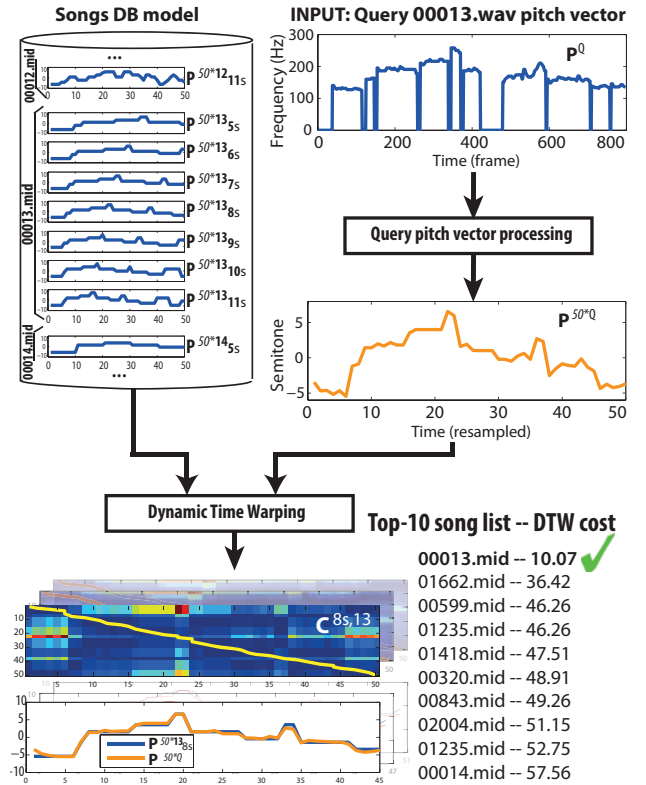


Figure 2. Scheme of the proposed baseline method for audio-to-MIDI melody matching.

(4) *Top-10 report*: Once the  $\mathbf{P}^{50*Q}$  has been aligned with all target pitch vectors (a total of  $7 \times N_{\text{songs}}$  vectors, since we use 7 different durations), the matched pitch vectors are sorted according to their alignment total cost (this value consists of the matrix  $\mathcal{D}$  produced by `dpfast.m` evaluated in the last position of the optimal path,  $T_{\text{cost}} = \mathcal{D}(p(\text{end}), q(\text{end}))$ ). Finally, the 10 songs with minimum cost are reported.

#### 3.2 Music Radar's approach

MusicRadar [3] is a state-of-the-art algorithm for melodic matching, which participated in MIREX 2013 and obtained the best accuracy in all datasets, except for the case of IOA-CAS<sup>8</sup>. It is the latest evolution of a set of systems developed by Lei Wang since 2007 [11, 12]. The system takes advantage of several matching methods to improve its accuracy. First, Earth Mover's Distance (EMD), which is note-based and fast, is adopted to eliminate most unlikely candidates. Then, Dynamic Time Warping (DTW), which is frame-based and more accurate, is executed on these surviving candidates. Finally, a weighted voting fusion strategy is employed to find the optimal match. In our study, we have used the exact melody matcher tested in MIREX 2013, provided by its original author.

#### 3.3 NetEase's approach

NetEase's approach [13] is a state-of-the-art algorithm for melodic matching, which participated in MIREX 2013 and

<sup>8</sup> [http://www.music-ir.org/mirex/wiki/2013:Query\\_by\\_-Singing/Humming](http://www.music-ir.org/mirex/wiki/2013:Query_by_-Singing/Humming)

obtained the first position for IOACAS dataset<sup>8</sup>, as well as relevant results in the rest of datasets. This algorithm adopts a two-stage cascaded solution based on Locality Sensitive Hashing (LSH) and accurate matching of frame-level pitch sequence. Firstly, LSH is employed to quickly filter out songs with low matching possibilities. In the second stage, Dynamic Time Warping is applied to find the  $N$  (set to 10) most matching songs from the candidate list. Again, the original authors of NetEase’s approach (who also authored some older works on query-by-humming [14]) collaborated in this study, so we have used the exact melody matcher tested in MIREX 2013.

#### 4. EVALUATION STRATEGY

In this section, we present the datasets used in our study (Section 4.1), the way in which we have combined F0 trackers and melody matchers (Section 4.2) and the chosen evaluation measures (Section 4.3).

##### 4.1 Datasets

We have used the public corpus MIR-QBSH<sup>8</sup> (used in MIREX since 2005), which includes 4431 .wav queries corresponding to 48 different MIDI songs. The audio queries are 8 seconds length, and they are recorded in mono 8 bits, with a sample rate of 8kHz. In general, the audio queries are monophonic with no background noise, although some of them are slightly noisy and/or distorted. This dataset also includes a manually corrected pitch vector for each .wav query. Although these annotations are fairly reliable, they may not be totally correct, as stated in MIR-QBSH documentation.

In addition, we have used the Audio Degradation Toolbox [17] in order to recreate common environments where a QBSH system could work. Specifically, we have combined three levels of pub-style added background noise (PubEnvironment1 sound) and smartphone-style distortion (smartPhoneRecording degradation), leading to a total of seven evaluation datasets: (1) Original MIR-QBSH corpus (2) 25 dB SNR (3) 25 dB SNR + smartphone distortion (4) 15 dB SNR (5) 15 dB SNR + smartphone distortion (6) 5 dB SNR (7) 5 dB SNR + smartphone distortion. Note that all these degradations have been checked in order to ensure perceptually realistic environments.

Finally, in order to replicate MIREX conditions, we have included 2000 extra MIDI songs (randomly taken from ESSEN collection<sup>9</sup>) to the original collection of 48 MIDI songs, leading to a songs collection of 2048 MIDI songs. Note that, although these 2000 extra songs fit the style of the original 48 songs, they do not correspond to any .wav query of Jang’s dataset.

##### 4.2 Combinations of F0 trackers and melody matchers

For each of the 7 datasets, the 4431 .wav queries have been transcribed using the 8 different F0 trackers mentioned in Section 2. Additionally, each dataset also includes the 4431 manually corrected pitch vectors of MIR-QBSH as a reference, leading to a total of 7 datasets  $\times$  (8

F0 trackers + 1 manual annotation)  $\times$  4431 queries =  $63 \times 4431$  queries = 279153 pitch vectors. Then, all these pitch vectors have been used as input to the 3 different melody matchers mentioned in Section 3, leading to 930510 lists of top-10 matched songs. Finally, these results have been used to compute a set of meaningful evaluation measures.

##### 4.3 Evaluation measures

In this section, we present the evaluation measures used in this study:

**(1) Mean overall accuracy of F0 tracking ( $\overline{\text{Acc}_{\text{ov}}}$ ):** For each pitch vector we have computed an evaluation measure defined in MIREX Audio Melody Extraction task: *overall accuracy* ( $\text{Acc}_{\text{ov}}$ ) (a definition can be found in [15]). The *mean overall accuracy* is then defined as  $\overline{\text{Acc}_{\text{ov}}} = (1/N) \sum_{i=1}^N \text{Acc}_{\text{ov}_i}$ , where  $N$  is the total number of queries considered and  $\text{Acc}_{\text{ov}_i}$  is the overall accuracy of the pitch vector of the  $i$ :th query. We have selected this measure because it considers both voicing and pitch, which are important aspects in QBSH. For this measure, our ground truth consists of the manually corrected pitch vectors of the .wav queries, which are included in the original MIR-QBSH corpus.

**(2) Mean Reciprocal Rank (MRR):** This measure is commonly used in MIREX Query By Singing Humming task<sup>8</sup>, and it is defined as:  $\text{MRR} = (1/N) \sum_{i=1}^N r_i^{-1}$ , where  $N$  is the total number of queries considered and  $r_i$  is the rank of the correct answer in the retrieved melodies for  $i$ :th query.

## 5. RESULTS & DISCUSSION

In this section, we present the obtained results and some relevant considerations about them.

##### 5.1 $\overline{\text{Acc}_{\text{ov}}}$ and MRR for each F0 tracker - Dataset - Matcher

In Table 1, we show the  $\overline{\text{Acc}_{\text{ov}}}$  and the MRR obtained for the whole dataset of 4431 .wav queries in each combination of F0 tracker-dataset-matcher (189 combinations in total). Note that these results are directly comparable to MIREX Query by Singing/Humming task<sup>8</sup> (Jang Dataset). As expected, the manually corrected pitch vectors produce the best MRR in most cases (the overall accuracy is 100% because it has been taken as the ground truth for such measure). Note that, despite manual annotations are the same in all datasets, NetEase and MusicRadar matchers do not produce the exact same results in all cases. It is due to the generation of the indexing model (used to reduced the time search), which is not a totally deterministic process.

Regarding the relationship between  $\overline{\text{Acc}_{\text{ov}}}$  and MRR in the rest of F0 trackers, we find a somehow contradictory result: the best  $\overline{\text{Acc}_{\text{ov}}}$  does not always correspond with the best MRR. This fact may be due to two different reasons. On the one hand, the meaning of  $\overline{\text{Acc}_{\text{ov}}}$  may be distorted due to annotation errors in the ground truth (as mentioned in Section 4.1), or to eventual intonation errors in the dataset. However, the manual annotations produce the best MRR, what suggests that the amount of these types

<sup>9</sup> www.esac-data.org/



F0 tracker	Clean dataset	25dB SNR	25 dB SNR + distortion	15dB SNR	15 dB SNR + distortion	5dB SNR	5 dB SNR + distortion
(A)	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.95	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.89 / 0.96	100 / 0.82 / 0.88 / 0.95
(B)	89 / <b>0.80</b> / <b>0.89</b> / <b>0.96</b>	89 / <b>0.80</b> / <b>0.89</b> / <b>0.96</b>	<b>88</b> / <b>0.80</b> / <b>0.88</b> / <b>0.95</b>	88 / <b>0.79</b> / <b>0.88</b> / <b>0.94</b>	84 / 0.71 / 0.86 / 0.94	78 / 0.50 / 0.73 / 0.85	67 / 0.33 / 0.57 / 0.73
(C)	<b>90</b> / 0.74 / 0.85 / 0.94	90 / 0.71 / 0.85 / 0.92	86 / 0.72 / 0.84 / 0.92	89 / 0.71 / 0.84 / 0.92	85 / 0.66 / 0.81 / 0.89	72 / 0.49 / 0.58 / 0.70	64 / 0.26 / 0.39 / 0.51
(D)	90 / 0.71 / 0.83 / 0.92	<b>90</b> / 0.74 / 0.85 / 0.93	85 / 0.74 / 0.85 / 0.94	<b>90</b> / 0.78 / 0.87 / 0.94	<b>85</b> / <b>0.77</b> / <b>0.87</b> / <b>0.94</b>	79 / <b>0.69</b> / <b>0.79</b> / <b>0.87</b>	72 / <b>0.58</b> / <b>0.69</b> / <b>0.81</b>
(E)	89 / 0.71 / 0.83 / 0.92	89 / 0.71 / 0.84 / 0.92	84 / 0.66 / 0.80 / 0.91	88 / 0.72 / 0.84 / 0.93	83 / 0.65 / 0.80 / 0.91	75 / 0.67 / 0.67 / 0.82	66 / 0.48 / 0.53 / 0.73
(F)	86 / 0.62 / 0.81 / 0.89	86 / 0.70 / 0.83 / 0.92	81 / 0.64 / 0.78 / 0.89	82 / 0.60 / 0.77 / 0.88	75 / 0.50 / 0.67 / 0.82	48 / 0.03 / 0.08 / 0.04	44 / 0.04 / 0.04 / 0.03
(G)	88 / 0.56 / 0.81 / 0.88	87 / 0.47 / 0.79 / 0.86	83 / 0.47 / 0.76 / 0.85	86 / 0.39 / 0.78 / 0.87	81 / 0.35 / 0.73 / 0.82	70 / 0.11 / 0.32 / 0.52	63 / 0.04 / 0.20 / 0.38
(H)	87 / 0.66 / 0.83 / 0.87	87 / 0.67 / 0.82 / 0.87	83 / 0.64 / 0.78 / 0.84	86 / 0.66 / 0.81 / 0.84	82 / 0.58 / 0.74 / 0.80	83 / 0.51 / 0.73 / 0.75	73 / 0.32 / 0.55 / 0.62
(I)	84 / 0.62 / 0.76 / 0.86	84 / 0.62 / 0.76 / 0.86	79 / 0.50 / 0.64 / 0.74	84 / 0.63 / 0.76 / 0.86	79 / 0.50 / 0.65 / 0.75	<b>83</b> / 0.60 / 0.73 / 0.83	<b>75</b> / 0.39 / 0.55 / 0.65

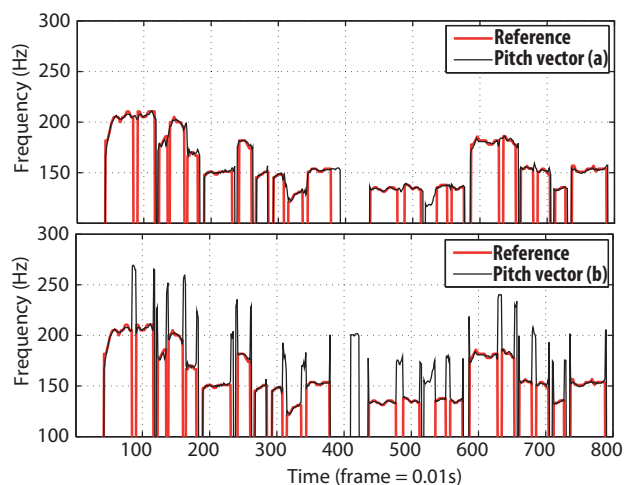
**Table 1:** F0 overall accuracy and MRR obtained for each case. F0 trackers: (A) *MANUALLY CORRECTED* (B) *AC-LEIWANG* (C) *AC-ADJUSTED* (D) *PYIN* (E) *SWIPE'* (F) *YIN* (G) *AC-DEFAULT* (H) *MELODIA-MONO* (I) *MELODIA-POLY*. The format of each cell is:  $\overline{\text{Acc}}_{\text{ov}}(\%) / \text{MRR-baseline} / \text{MRR-NetEase} / \text{MRR-MusicRadar}$ .

of errors are low. On the other hand, the measure  $\overline{\text{Acc}}_{\text{ov}}$  itself may not be totally representative of the suitability of a pitch vector for QBSH. Indeed, after analysing specific cases, we observed that two pitch vectors with same F0 tracking accuracy (according to MIREX measures) may not be equally suitable for query-by-humming. For instance, we analysed the results produced by the baseline matcher using two different pitch vectors (Figure 3) with exactly the same evaluation measures in MIREX Audio Melody Extraction task: *voicing recall* = 99.63%, *voicing false-alarm* = 48.40%, *raw pitch accuracy* = 97.41%, *raw-chroma accuracy* = 97.41% and *overall accuracy* = 82.91%. However, we found that pitch vector (a) matches the right song with rank  $r_i = 1$  whereas pitch vector (b) does not matches the right song at all ( $r_i \geq 11$ ). The reason is that MIREX evaluation measures do not take into account the pitch values of false positives, but in fact they are important for QBSH. Therefore, we conclude that the high MRR achieved by some F0 trackers (AC-LEIWANG when background noise is low, and PYIN for highly degraded signals), is not only due to the amount of errors made by them, but also to the type of such errors.

Additionally, we observed that, in most cases, the queries are matched either with rank  $r_i = 1$  or  $r_i \geq 11$  (intermediate cases such as rank  $r_i = 2$  or  $r_i = 3$  are less frequent). Therefore, the variance of ranks is generally high, their distribution is not Gaussian.

## 5.2 MRR vs. $\overline{\text{Acc}}_{\text{ov}}$ for each matcher

In order to study the robustness of each melodic matcher to F0 tracking errors, we have represented the MRR obtained by each one for different ranges of  $\overline{\text{Acc}}_{\text{ov}}$  (Figure 4). For this experiment, we have selected only the .wav queries which produce the right answer in first rank for the three matchers considered (baseline, Music Radar and NetEase) when manually corrected pitch vectors are used (around a 70% of the dataset matches this condition). In this way, we ensure that bad singing or a wrong manual annotation is not affecting the variations of MRR in the plots. Note that, in this case, the results are not directly comparable to the ones computed in MIREX (in contrast to the results shown in Section 5.1).

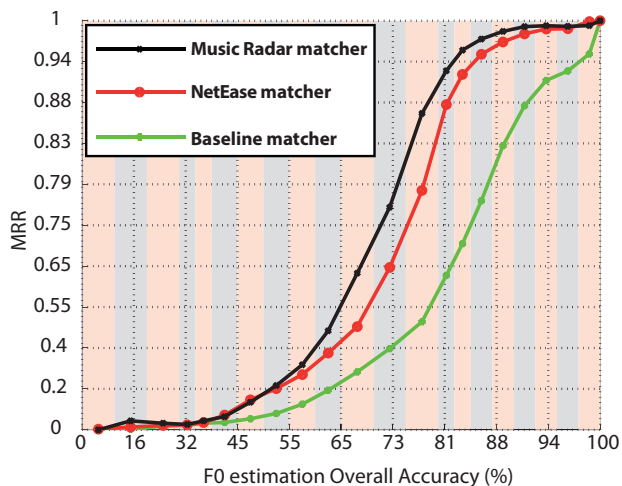


**Figure 3.** According to MIREX measures, these two pitch vectors (manually manipulated) are equally accurate; however, they are not equally suitable for QBSH.

Regarding the obtained results (shown in Figure 4), we observe clear differences in the robustness to F0 estimation errors between matchers, which is coherent with the results presented in Table 1. The main difference is found in the baseline matcher with respect to both NetEase and Music Radar. Given that the baseline matcher only uses DTW, whereas the other two matchers use a combination of various searching methods (see Sections 3.2 and 3.3), we hypothesise that such combination may improve their robustness to F0 tracking errors. However, further research is needed to really test this hypothesis.

## 6. CONCLUSIONS

In this paper, eight different state-of-the-art F0 trackers were evaluated for the specific application of query-by-humming-singing in different conditions of pub-style added noise and smartphone-style distortion. This study was carried out using three different matching methods: a simple, freely available baseline (a detailed description has been provided in Section 3.1) and two state-of-the-art matchers. In our results, we found that Boersma's AC method [8], with an appropriate adjustment and a smoothing stage



**Figure 4.** MRR obtained for each range of Overall Accuracy (each range is marked with coloured background rectangles). We have considered only the .wav queries which, using manually corrected F0 vectors, produce MRR = 1 in all matchers.

achieves the best results when the audio is not very degraded. In contrast, when the audio is highly degraded, the best results are obtained with pYIN [6], even without further smoothing. Considering that pYIN is a very recent, open source approach, this result is promising in order to improve the noise robustness of future QBSH systems. Additionally, we found that F0 trackers perform differently on QBSH depending on the type of F0 tracking errors made. Due to this, MIREX measures do not fully represent the suitability of a pitch vector for QBSH purposes, so the development of novel evaluation measures in MIREX is encouraged to really measure the suitability of MIR systems for specific applications. Finally, we observed clear differences between matchers regarding their robustness to F0 estimation errors. However, further research is needed for a deeper insight into these differences.

## 7. ACKNOWLEDGEMENTS

Special thanks to Doreso<sup>1</sup> team (especially to Lei Wang and Yuhang Cao) and to Peng Li for their active collaboration in this study. This work has been funded by the Ministerio de Economía y Competitividad of the Spanish Government under Project No. TIN2013-47276-C6-2-R and by the Junta de Andalucía under Project No. P11-TIC-7154. The work has been done at Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

## 8. REFERENCES

- [1] A. D. Brown and Brighthand staff: “SoundHound for Android OS Review: ‘Name That Tune,’ But At What Price?”, *Brighthand Smartphone News & Review*, 2012. Online: [www.brighthand.com](http://www.brighthand.com) [Last Access: 28/04/2014]
- [2] J. -S. Roger Jang: “QBSH and AFP as Two Successful Paradigms of Music Information Retrieval” Course in *RuSSIR*, 2013. Available at: <http://mirilab.org/jang/> [Last Access: 28/04/2014]
- [3] Doreso Team ([www.doreso.com](http://www.doreso.com)): “MIREX 2013 QBSH Task: Music Radar’s Solution” *Extended abstract for MIREX*, 2013.
- [4] A. De Cheveigné and H. Kawahara: “YIN, a fundamental frequency estimator for speech and music,” *Journal of the Acoustic Society of America*, Vol. 111, No. 4, pp. 1917-1930, 2002.
- [5] L. Rabiner: “On the use of autocorrelation analysis for pitch detection,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, Vol. 25, No.1, pp. 24-33. 1977.
- [6] M. Mauch, and S. Dixon, “pYIN: A fundamental frequency estimator using probabilistic threshold distributions,” *Proceedings of ICASSP*, 2014.
- [7] P. Boersma and D. Weenink: “Praat: a system for doing phonetics by computer,” *Glott international*, Vol. 5, No. 9/10, pp. 341-345, 2002. Software available at: [www.praat.org](http://www.praat.org) [Last access: 28/04/2014]
- [8] P. Boersma: “Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound,” *Proceedings of the Institute of Phonetic Sciences*, Vol. 17, No. 1193, pp 97-110, 1993.
- [9] E. Keelan, C. Lai, K. Zechner: “The importance of optimal parameter setting for pitch extraction,” *Journal of Acoustical Society of America*, Vol. 128, No. 4, pp. 2291–2291, 2010.
- [10] A. Camacho: “SWIPE: A sawtooth waveform inspired pitch estimator for speech and music,” PhD dissertation, University of Florida, 2007.
- [11] L. Wang, S. Huang, S. Hu, J. Liang and B. Xu: “An effective and efficient method for query-by-humming system based on multi-similarity measurement fusion,” *Proceedings of ICALIP*, 2008.
- [12] L. Wang, S. Huang, S. Hu, J. Liang and B. Xu: “Improving searching speed and accuracy of query by humming system based on three methods: feature fusion, set reduction and multiple similarity measurement rescoring,” *Proceedings of INTERSPEECH*, 2008.
- [13] P. Li, Y. Nie and X. Li: “MIREX 2013 QBSH Task: Netease’s Solution” *Extended abstract for MIREX*, 2013.
- [14] P. Li, M. Zhou, X. Wang and N. Li: “A novel MIR system based on improved melody contour definition,” *Proceedings of the International Conference on Multi-Media and Information Technology (MMIT)*, 2008.
- [15] J. Salamon and E. Gómez: “Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics,” *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 20, No. 6, pp. 1759–1770, 2012.
- [16] D. Ellis: “Dynamic Time Warp (DTW) in Matlab”, 2003. Web resource, available: [www.ee.columbia.edu/~dpwe/resources/matlab/dtw/](http://www.ee.columbia.edu/~dpwe/resources/matlab/dtw/) [Last Access: 28/04/2014]
- [17] M. Mauch and S. Ewert: “The Audio Degradation Toolbox and its Application to Robustness Evaluation,” *Proceedings of ISMIR*, 2013.

# VOCAL SEPARATION USING SINGER-VOWEL PRIORS OBTAINED FROM POLYPHONIC AUDIO

Shrikant Venkataramani<sup>1</sup>, Nagesh Nayak<sup>2</sup>, Preeti Rao<sup>1</sup>, and Rajbabu Velmurugan<sup>1</sup>

<sup>1</sup>Department of Electrical Engineering, IIT Bombay, Mumbai 400076

<sup>2</sup>Sensibol Audio Technologies Pvt. Ltd.

<sup>1</sup>{vshrikant, prao, rajbabu}@ee.iitb.ac.in

<sup>2</sup>nageshsnayak@sensibol.com

## ABSTRACT

Single-channel methods for the separation of the lead vocal from mixed audio have traditionally included harmonic-sinusoidal modeling and matrix decomposition methods, each with its own strengths and shortcomings. In this work we use a hybrid framework to incorporate prior knowledge about singer and phone identity to achieve the superior separation of the lead vocal from the instrumental background. Singer specific dictionaries learned from available polyphonic recordings provide the soft mask that effectively attenuates the bleeding-through of accompanying melodic instruments typical of purely harmonic-sinusoidal model based separation. The dictionary learning uses NMF optimization across a training set of mixed signal utterances while keeping the vocal signal bases constant across the utterances. A soft mask is determined for each test mixed utterance frame by imposing sparseness constraints in the NMF partial co-factorization. We demonstrate significant improvements in reconstructed signal quality arising from the more accurate estimation of singer-vowel spectral envelope.

## 1. INTRODUCTION

Source separation techniques have been widely applied in the suppression of the lead vocal in original songs to obtain the orchestral background for use in karaoke and remix creation. In stereo and multichannel recordings, spatial cues can contribute significantly to vocal separation from the original mixtures. However this separation is not complete, depending on the manner in which the multiple instruments are panned in the mix. Further, an important category of popular music recordings, dating until the 1950s in the West and even later in the rest of the world, are purely monophonic. Single-channel methods for the separation of the lead vocal from the instrumental background

include harmonic sinusoidal modeling and matrix decomposition methods. Of these, harmonic sinusoidal modeling has found success in situations where no clean data is available for supervised learning [6], [10]. Based on the assumption that the vocal is dominant in the mixture, predominant pitch detection methods are applied to obtain the vocal pitch and hence the predicted vocal harmonic locations at each instant in time. Harmonic sinusoidal modeling is then applied to reconstruct the vocal component based on assigning a magnitude and phase to each reconstructed harmonic from a detected sinusoidal peak in the corresponding spectral neighborhood of the mixed signal short-time Fourier transform (STFT). The vocal signal is reconstructed by the amplitude and phase interpolation of the harmonic component tracks. The instrumental background is obtained by the subtraction of the reconstructed vocal from the original mixture. A high degree of vocal separation is obtained when the assumption of vocal dominance holds for the mixture. However some well-known artifacts remain viz. (i) “bleeding through” of some of the melodic instrumentation due to the blind assignment of the total energy in the mixed signal in the vocal harmonic location to the corresponding reconstructed harmonic; this artifact is particularly perceptible in the sustained vowel regions of singing, (ii) improper cancellation of the unvoiced consonants and breathy voice components due to the limitations of sinusoidal modeling of noise and (iii) residual of vocal reverb if present in the original [14]. To address the first shortcoming, recent methods rely on the availability of non-overlapping harmonics of the same source anywhere in the entire audio [3]. We propose to replace the binary mask (implicit in the harmonic-sinusoidal modeling) applied to the vocal harmonics before reconstruction by a soft-mask (a form of Wiener filtering). An effective soft mask would be based on an accurate estimate of the vocal signal spectrum at any time-instant [2], [14]. This would improve the reconstructed vocal signal and lead to more complete suppression in the estimated background.

The vocal signal spectrum depends on several factors such as the singer’s voice, the phone being uttered, the pitch and the vocal effort. We cannot assume the availability of clean data for supervised training (i.e., unaccompanied voice of the particular singer). However popular singers typically have a large number of songs to their



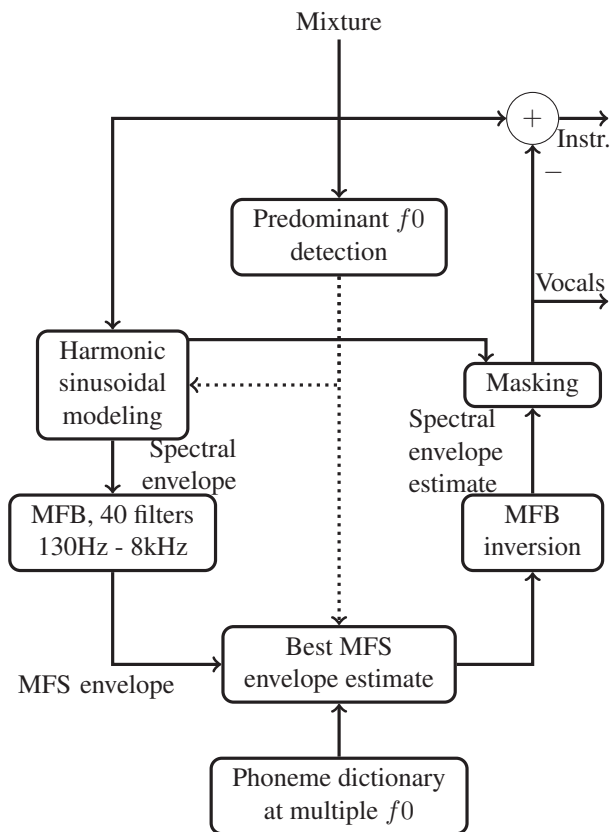
© Shrikant Venkataramani, Nagesh Nayak, Preeti Rao, Rajbabu Velmurugan.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Shrikant Venkataramani, Nagesh Nayak, Preeti Rao, Rajbabu Velmurugan. “VOCAL SEPARATION USING SINGER-VOWEL PRIORS OBTAINED FROM POLYPHONIC AUDIO”, 15th International Society for Music Information Retrieval Conference, 2014.

credit, and therefore a method for learning a dictionary of soft masks for the singer from such a training data set could be useful. The training set thus has original single-channel polyphonic songs where the vocal characteristics correspond to the singer but the background orchestration is diverse. We apply non-negative matrix factorization (NMF) methods to estimate the invariant set of basis vectors across multiple instances of the singer’s phones in different songs. In the recent past, several systems have been proposed that qualify as modifications of NMF for improved performance in various scenarios where specific prior knowledge about the data are available [5] (and references therein). In the present work, we attempt to formulate a NMF approach to obtain basis elements corresponding to the singer’s utterances by providing audio corresponding to a particular singer. Given the very diverse spectra of the different phones in a language, the quality of the decomposition can be improved by restricting the optimization to within a phone class [11]. We exploit the availability of song-synchronized lyrics data available in karaoke applications to achieve this. Our main contribution is to combine the advantages of harmonic-sinusoidal modeling in localizing the vocal components in time-frequency with that of soft-masking based on spectral envelope estimates from a NMF decomposition on polyphonic audio training data. Prior knowledge about singer identity and underlying phone transcription of the training and test audio are incorporated in the proposed framework. We develop and evaluate the constrained NMF optimization required for the training across instances where a common basis function set corresponds to the singer-vowel. On the test data, partial co-factorization with a sparseness constraint helps obtain the correct basis decomposition for the mixed signal at any time instant, and thus a reliable spectral envelope estimate of the vowel for use in the soft mask. Finally, the overall system is evaluated based on the achieved vocal and orchestral background separation using objective measures and informal listening. In the next sections, we present the overall system for vocal separation, followed by the proposed NMF-based singer-vowel dictionary learning, estimation of the soft mask for test mixed polyphonic utterances and experimental evaluation of system performance.

## 2. PROPOSED HYBRID SYSTEM

A block diagram of the proposed hybrid system for vocal separation is shown in Figure 1. The single-channel audio mixture considered for vocal separation is assumed to have the singing voice, when present, as the dominant source in the mix. We assume that the sung regions are annotated at the syllable level, as expected from music audio prepared for karaoke use. A predominant pitch tracker [9] is applied to the sung regions to detect vocal pitch at 10 ms intervals throughout the sung regions of the audio. Sinusoidal components are tracked in the computed short-time magnitude spectrum after biasing trajectory information towards the harmonic locations based on the detected pitch [8]. The pitch salience and total harmonic energy are used to locate the vowel region within the syllable. The vocal signal can



**Figure 1.** Block diagram of the proposed vocal separation system.

be reconstructed from the harmonic sinusoidal component trajectories obtained by amplitude and phase interpolation of the frame-level estimates from the STFT. An estimate of the instantaneous spectral envelope of the singer’s voice provides a soft mask to re-shape the harmonic amplitudes before vocal reconstruction. The mel-filtered spectral envelope (MFS) is computed by applying a 40-band mel-filter bank to the log-linearly interpolated envelope of the mixture harmonic amplitudes. By using the spectral envelope, we eliminate pitch dependence in the soft mask to a large extent. The phoneme dictionary consists of a set of basis vectors for each vowel, at various pitches. A linear combination of these basis vectors may be used to estimate the MFS envelope of the vocal component of the mixture, from the MFS envelope of the mixture. These spectral envelope vectors are learnt from multiple polyphonic mixtures of the phoneme as explained in Section 3. The MFS is used as a low-dimensional perceptually motivated representation. The reconstructed vocal signal is subtracted in the time-domain from the polyphonic mixture to obtain the vocal-suppressed music background.

## 3. SPECTRAL ENVELOPE DICTIONARY LEARNING USING NMF

To obtain the singer specific soft mask mentioned in the previous section, we create a dictionary of basis vectors corresponding to each of the vowels of the language. This

dictionary is created from polyphonic song segments, containing the vowel, of the singer under consideration. While spectral envelope of a vowel depends on the vowel identity, there are prominent dependencies on (i) the singer, whose physiological characteristics and singing style affect the precise formant locations and bandwidths for a given vowel. This is especially true of the higher formants (4th and 5th), which depend primarily on the singer rather than on the vowel; (ii) pitch, specifically in singing where the articulation can vary with large changes in pitch due to the ‘‘formant tuning’’ phenomenon [12]; (iii) loudness or vocal effort. Raising the vocal effort reduces spectral tilt, increasing the relative amplitudes of the higher harmonics and consequently the brightness of the voice.

In the proposed dictionary learning, pitch dependence is accounted for by separate dictionary entries corresponding to 2 or 3 selected pitch ranges across the 2 octaves span of a singer. Since the pitch and vowel identity are known for the test song segment, the correct dictionary can be selected at any time. The basis vectors for any pitch range of the singer-vowel capture the variety of spectral envelopes that arise from varying vocal effort and vowel context. We have training data of several instances of a particular singer uttering a common vowel. These utterances have been obtained from different songs and hence, we may assume that the accompaniments in the mixtures are different. The MFS envelopes, reviewed in the previous section, are extracted for each training vowel instance in the polyphonic audio. Based on the assumption of additivity of the spectral envelopes of vocal and instrumental background, there is a common partial factor corresponding to the singer-vowel across the mixtures with changing basis vectors for the accompaniment.

We use NMF to extract common features (singer-vowel spectra) across multiple song segments. The conventional use of NMF is similar to the phoneme-dependent NMF used for speech separation in [7] where the bases are estimated from clean speech. We extend the scope of NMF further, using non-negative matrix partial co-factorization (NMPCF) [4] equivalent to NMF for multiblock data [15] techniques. NMPCF and its variants have been used in drum source separation [4], where one of the training signals is the solo drums audio. Here, we use NMPCF for multiple MFS matrices of mixed signals across segments of the polyphonic audio of the singer, without the use of clean vocal signal. This will yield a common set of bases representing the singer-vowel and other varying bases representative of the accompaniments.

We now describe the NMPCF algorithm for learning the singer-vowel basis. The MFS representation for one specific annotated segment of a polyphonic music is represented as  $\mathbf{V}_i$ . This section has the vowel of interest and instrumental accompaniments. We have MFS of  $M$  such mixtures for  $i = 1, \dots, M$  represented as [15],

$$\mathbf{V}_i = \mathbf{V}_{c,i} + \mathbf{V}_{a,i}, \quad i = 1, \dots, M. \quad (1)$$

where  $\mathbf{V}_{c,i}$  and  $\mathbf{V}_{a,i}$  denote the MFS of the common singer-vowel and accompaniment, respectively. Using NMF de-

composition for the MFS spectra we have,

$$\mathbf{V}_i = \mathbf{W}_c \mathbf{H}_{c,i} + \mathbf{W}_{a,i} \mathbf{H}_{a,i}, \quad i = 1, \dots, M. \quad (2)$$

where  $\mathbf{W}_c \in \mathbb{R}_+^{F \times N_c}$  denotes the basis vectors corresponding to the common vowel shared by the  $M$  mixtures and  $\mathbf{W}_{a,i} \in \mathbb{R}_+^{F \times N_a}$  are the basis vectors corresponding to the accompaniments. Here  $F$  is the number of mel-filters (40) used,  $N_c$  and  $N_a$  are the number of basis vectors for the vowel and accompaniments, respectively. The matrices  $\mathbf{H}_{c,i}$  and  $\mathbf{H}_{a,i}$  are the activation matrices for the vowel and accompaniment basis vectors, respectively. Our objective is to obtain the basis vectors  $\mathbf{W}_c$  corresponding to the common vowel across these  $M$  mixtures. We achieve this by minimizing the Frobenius norm  $\| \cdot \|_F^2$  of the discrepancy between the given mixtures and their factorizations, simultaneously. Accordingly, the cost function,

$$D = \sum_{i=1}^M \frac{1}{2} \| \mathbf{V}_i - \mathbf{W}_c \mathbf{H}_{c,i} - \mathbf{W}_{a,i} \mathbf{H}_{a,i} \|_F^2 + \frac{\lambda_1}{2} \| \mathbf{W}_{a,i} \|_F^2, \quad (3)$$

is to be minimized with respect to  $\mathbf{W}_c$ ,  $\mathbf{W}_{a,i}$ ,  $\mathbf{H}_{c,i}$ , and  $\mathbf{H}_{a,i}$ . The regularizer  $\| \mathbf{W}_{a,i} \|_F^2$  and  $\lambda_1$  the Lagrange multiplier lead to dense  $\mathbf{W}_{a,i}$  and  $\mathbf{W}_c$  matrices [15]. The basis vectors thus obtained are a good representation of both the common vowel and the accompaniments, across the mixture. In this work, we choose  $\lambda_1 = 10$  for our experimentation as it was found to result in the sparsest  $\mathbf{H}_{c,i}$  matrix for varying values of  $\lambda_1$ . We solve (3) using the multiplicative update algorithm. The multiplicative update for a parameter  $\mathbf{P}$  in solving the NMF problem takes the general form,

$$\mathbf{P} = \mathbf{P} \odot \frac{\nabla_{\mathbf{P}}^-(D)}{\nabla_{\mathbf{P}}^+(D)}, \quad (4)$$

where  $\nabla_{\mathbf{X}}^-(D)$  and  $\nabla_{\mathbf{X}}^+(D)$  represent the negative and positive parts of the derivative of the cost  $D$  w.r.t. the parameter  $\mathbf{X}$ , respectively,  $\odot$  represents the Hadamard (element-wise) product and the division is also element-wise. Correspondingly, the multiplicative update for the parameter  $\mathbf{W}_c$  in (3) is,

$$\mathbf{W}_c = \mathbf{W}_c \odot \frac{\nabla_{\mathbf{W}_c}^-(D)}{\nabla_{\mathbf{W}_c}^+(D)}, \quad (5)$$

where,

$$\nabla_{\mathbf{W}_c}^-(D) = \sum_{i=1}^M (\mathbf{W}_c \mathbf{H}_{c,i} + \mathbf{W}_{a,i} \mathbf{H}_{a,i} - \mathbf{V}_i) \mathbf{H}_{c,i}^T. \quad (6)$$

Similarly, the update equation for other terms in (3) are,

$$\mathbf{H}_{c,i} = \mathbf{H}_{c,i} \odot \frac{\mathbf{W}_c^T \mathbf{V}_i}{\mathbf{W}_c^T \mathbf{W}_c \mathbf{H}_{c,i} + \mathbf{W}_c^T \mathbf{W}_{a,i} \mathbf{H}_{a,i}}, \quad (7)$$

$$\mathbf{H}_{a,i} = \mathbf{H}_{a,i} \odot \frac{\mathbf{W}_{a,i}^T \mathbf{V}_i}{\mathbf{W}_{a,i}^T \mathbf{W}_c \mathbf{H}_{c,i} + \mathbf{W}_{a,i}^T \mathbf{W}_{a,i} \mathbf{H}_{a,i}}, \quad (8)$$

$$\mathbf{W}_{a,i} = \frac{\mathbf{V}_i \mathbf{H}_{a,i}^T}{\mathbf{W}_c \mathbf{H}_{c,i} \mathbf{H}_{a,i}^T + \mathbf{W}_{a,i} \mathbf{H}_{a,i} \mathbf{H}_{a,i}^T + \lambda_1 \times \mathbf{W}_{a,i}} \odot \mathbf{W}_{a,i} \quad (9)$$

for  $i = 1, \dots, M$ . The basis vectors  $\mathbf{W}_c$  for the various phonemes form the dictionary and act as a prior in the spectral envelope estimation. Each dictionary entry is associated with a vowel and pitch range. We denote each entry in the dictionary as  $\mathbf{W}_c(/p/, f_0)$  for each vowel  $/p/$  at pitch  $f_0$ .

#### 4. SOFT MASK ESTIMATION USING SINGER-VOWEL DICTIONARY

In this section, we describe the approach to estimate the frame-wise soft mask for a test polyphonic vowel mixture segment. We first obtain the MFS envelope for the mixture as mentioned in Section 2. With the vowel label and pitch range known, we obtain the corresponding set of basis vectors  $\mathbf{W}_c(/p/, f_0)$  from the dictionary. Given this MFS representation of the mixture and the basis vectors, our objective is to separate the vocal component from the mixture. We do this by minimizing the cost function

$$D_T = \frac{1}{2} \|\mathbf{V}_T - \mathbf{W}_c \mathbf{H}_{c,T} - \mathbf{W}_{a,T} \mathbf{H}_{a,T}\|_F^2 + \frac{\lambda_2}{2} \|\mathbf{H}_{c,T}\|_F^2, \quad (10)$$

where the subscript  $T$  refers to the test case. The minimization is done with the dictionary bases  $\mathbf{W}_c$  kept fixed and using multiplicative updates for  $\mathbf{H}_{c,T}$ ,  $\mathbf{W}_{a,T}$  and  $\mathbf{H}_{a,T}$ . The sparsity constraint on  $\mathbf{H}_{c,T}$  in (10) accounts for the fact that the best set of bases representing the vowel would result in the sparsest temporal matrix  $\mathbf{H}_{c,T}$ . Under this formulation,  $\mathbf{W}_c \mathbf{H}_{c,T}$  will give an estimate of the vowel's MFS envelope  $\mathbf{V}_c$  (as in (1)) for the mixture. An alternate way is to use Wiener filtering to estimate  $\mathbf{V}_c$  as,

$$\hat{\mathbf{V}}_c = \frac{\mathbf{W}_c \mathbf{H}_{c,T}}{\mathbf{W}_c \mathbf{H}_{c,T} + \mathbf{W}_{a,T} \mathbf{H}_{a,T}} \odot \mathbf{V}_T. \quad (11)$$

This estimated vowel MFS can be used to reconstruct the spectral envelope of the vowel  $\hat{\mathbf{C}}$ . This is done by multiplying  $\hat{\mathbf{V}}_c$  with the pseudoinverse of the DFT matrix  $\mathbf{M}$  of the mel filter bank [1] as  $\hat{\mathbf{C}} = \mathbf{M}^\dagger \hat{\mathbf{V}}_c$ . A soft mask corresponding to this spectral envelope can be obtained using the Gaussian radial basis function [2],

$$\mathbf{G}_b(f, t) = \exp \left( - \frac{(\log \mathbf{X}(f, t) - \log \hat{\mathbf{C}}(f, t))^2}{2\sigma^2} \right) \quad (12)$$

where,  $\sigma$  is the Gaussian spread,  $\mathbf{X}$  is the magnitude spectrum of the mixed signal. The soft mask (12) is evaluated with  $\sigma = 1$ , in a 50 Hz band around the pitch ( $f_0$ ) and its harmonics [14].

Having obtained the soft mask, the vocals track is reconstructed by multiplying the soft mask with the harmonic

amplitudes of the sinusoidally modeled signal. The resynthesized signal then corresponds to the reconstructed vocals. The accompaniment can be obtained by performing a time-domain subtraction of the reconstructed vocals from the original mixture.

## 5. EXPERIMENTS AND PARAMETER CHOICES

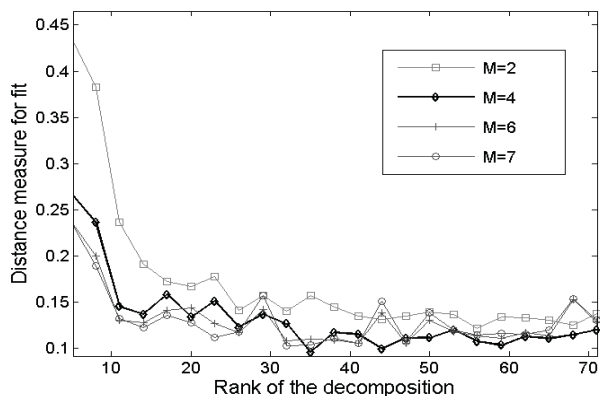
Given a polyphonic vowel segment, the vocal is separated by applying the generated soft mask corresponding to the given mixture. We compare the separated vocal with the ground truth to evaluate the performance. The performance evaluation of the proposed system is carried out in two steps. The first step is to choose the parameters of the system using the distance in the MFS space between the estimated and ground-truth MFS vectors obtained from the clean utterance. The second step is the computation of signal-to-distortion (SDR) measure (in dB) on the separated vocal and instrumental time-domain signals which will be given in Section 6. We present the training and test data used in the experiments next.

### 5.1 Description of the Dataset

The training dataset comprised of nine instances of three vowels viz., /a/, /i/, /o/ at two average pitches of 200 Hz and 300 Hz and sung by a male singer over three different songs with their accompaniments, annotated at the phoneme level. The training data was chosen so as to have different accompaniments across all the instances of a vowel. The training audios thus contained the vowel utterances throughout in the presence of background accompaniments. The training mixtures were pre-emphasised using a filter with a zero located at 0.7 to better represent the higher formants. A dictionary of bases was created for all the vowels for the two pitch ranges using the NMCPF optimization procedure discussed in Section 3. The performance was evaluated over a testing dataset of 45 test mixtures with 15 mixtures for each vowel over the two pitch ranges. The mixtures used for testing were distinct from the training mixtures. Since the audios were obtained directly from full songs, there was a significant variation in terms of the pitch of the vowel utterances around the average pitch ranges and in terms of coarticulation. The training and testing mixtures were created in a karaoke singing context and hence, we had available, the separate vocal and accompaniment tracks to be used as ground truth in the performance evaluation. All the mixtures had durations in the range of 400 ms - 2.2 s and were sampled at a frequency of 16 kHz. The window size and hop size used for the 1024 point STFT were 40 ms and 10 ms, respectively.

### 5.2 Choice of Parameters

There are several training parameters likely to influence the performance of the system. These include the ranks of the matrices in the decomposition ( $\mathbf{W}_c$ ,  $\mathbf{W}_a$ ) and the number of mixtures  $M$ . We obtain these parameters experimentally using a goodness-of-fit measure. The goodness-of-fit is taken to be the normalised Frobenius norm of the



**Figure 2.** Goodness-of-fit, averaged over all phoneme bases, with increasing number of mixtures ( $M$ ) used in the parallel optimization procedure for different decomposition ranks. The distance measure decreases indicating an improving fit as the number of mixtures and rank increase.

difference between the ideal envelope of a vowel in the MFS domain  $\mathbf{V}_c$  and its best estimate  $\hat{\mathbf{V}}_c$  obtained as a linear combination of the bases, for a mixture containing the vowel and accompaniment. This estimate can be calculated as explained in Section 4. Lower the value of this distance measure, closer is the envelope estimate to the ideal estimate. The various bases may be compared by calculating  $D_i$  for different bases  $\mathbf{W}_{c_i}$  using

$$D_i = \frac{\|\mathbf{V}_c - \hat{\mathbf{V}}_{c_i}\|_F^2}{\|\mathbf{V}_c\|_F^2} = \frac{\|\mathbf{V}_c - \frac{\mathbf{W}_{c_i}\mathbf{H}_{c_i}}{\mathbf{W}_{c_i}\mathbf{H}_{c_i} + \mathbf{W}_{a_i}\mathbf{H}_{a_i}} \odot \mathbf{V}_T\|_F^2}{\|\mathbf{V}_c\|_F^2}, \quad (13)$$

and comparing the same. To account for variabilities in the evaluation mixtures, the distance measure is evaluated and averaged over a number of mixtures and combinations, for each set of bases  $\mathbf{W}_{c_i}$ . The goodness-of-fit is used only to choose the appropriate parameter values for the system. The performance of the overall system, however, is evaluated in terms of SDR.

As shown in Figure 2, the goodness-of-fit measure decreases with increasing rank of the decomposition (number of vocal basis vectors) for a given  $M$ . The decreasing trend flattens out and then shows a slight increase beyond rank 35. For a fixed rank, the goodness-of-fit improves with increasing number of mixtures. Of the configurations tried, the distance measure is minimum when four mixtures ( $M = 4$ ) are used in the NMPCF optimization to obtain the dictionary. Thus, a rank 35 decomposition with  $M = 4$  is chosen for each singer-vowel dictionary for system performance evaluation.

As for the rank of the accompaniment basis, it is observed that the regularization term in the joint optimization (3) seems to make the algorithm robust to choice of number of basis vectors for the accompaniment. Eight basis vectors were chosen for each mixture term in the joint optimization. Although the number of accompani-

Separated track	Binary mask	Soft mask	
		Original singer	Alternate singer
Vocal	8.43	9.06	8.66
Instrumental	13.63	14.16	14.10

**Table 1.** SDR values (in dB) for separated vocals and instruments obtained using a binary mask, soft mask from the original singer’s training mixtures and soft mask from an alternate singer’s vocals.

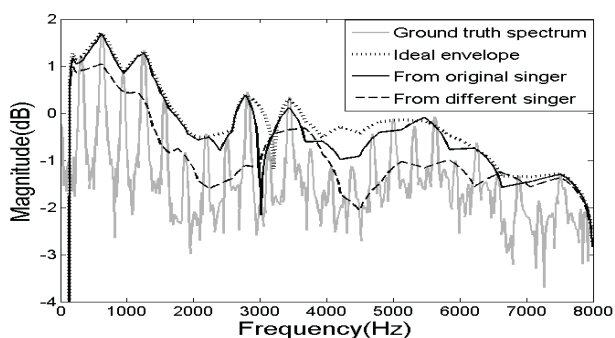
ment bases seems to be comparatively low, eight bases were sufficient to reconstruct the accompaniment signals from the mixtures. A high value for  $\lambda_2$  in the test optimization problem of (10) results in a decomposition involving a linear combination of the least number of bases per time frame. This sparse decomposition may not necessarily lead to the best reconstruction in more challenging scenarios involving articulation variations. Thus a small value of  $\lambda_2 = 0.1$  was chosen.

## 6. RESULTS AND DISCUSSION

We evaluate the performance of the system using the SDR. The SDR is evaluated using the BSS\_eval toolbox [13]. The SDR values averaged across 45 vowel test mixtures, separately for the reconstructed vocals and instrumental background are given in Table 1. To appreciate the improvement, if any, the SDR is also computed for the harmonic sinusoidal model without soft masking (i.e., binary masking only). While the proposed soft masking shows an increase in SDR, closer examination revealed that the improvements were particularly marked for those mixtures with overlapping vocal and instrumental harmonics (accompaniments) in some spectral regions. This was also borne out by informal listening. When we isolated these samples, we observed SDR improvements of up to 4 dB in several instances. This is where the selective attenuation of the harmonic amplitudes in accordance with the estimated vowel spectral envelope is expected to help most. The harmonics in the non-formant regions are retained in the instrumental background rather than being canceled out as in binary masking, contributing to higher SDR<sup>1</sup>.

To understand the singer dependence of the dictionary, we carried out soft mask estimation from the polyphonic test mixtures using the basis vectors of an alternate singer. This basis was a set of clean vowel spectral envelopes obtained from another male singer’s audio with the same vowels and pitches corresponding to our training dataset. We observe from Table 1 that the alternate singer soft mask does better than the binary mask, since it brings in the vowel dependence of the soft mask. However, it does not perform as well as the original singer’s soft mask even though the latter is obtained from clean vowel utterances. As depicted in Figure 3 (for a sample case), the envelope obtained using the original singer’s data closely follows the

<sup>1</sup> Audio examples are available at [http://www.ee.iitb.ac.in/student/~daplab/ISMIR\\_webpage/webpageISMIR.html](http://www.ee.iitb.ac.in/student/~daplab/ISMIR_webpage/webpageISMIR.html).



**Figure 3.** Comparison of the reconstructed phoneme envelope of the phoneme /a/ obtained from training mixtures of the same singer and with that obtained from pure vocals of a different singer.

ideal envelope of the phoneme.

Although the NMF optimization converges slowly, the number of iterations to be carried out to obtain the envelope is low, for both training and testing procedures. It is observed that the bases and envelopes attain their final structure after 4000 and 1000 iterations, respectively.

## 7. CONCLUSION

Soft masks derived from a dictionary of singer-vowel spectra are used to improve upon the vocal-instrumental music separation achieved by harmonic sinusoidal modeling for polyphonic music of the particular singer. The main contribution of this work is an NMF based framework that exploits the amply available original polyphonic audios of the singer as training data for learning the dictionary of singer spectral envelopes. Appropriate constraints are introduced in the NMF optimization for training and test contexts. The availability of lyric-aligned audio (and therefore phone labels) helps to improve the homogeneity of the training data and have a better model with fewer basis vectors. Significant improvements in reconstructed signal quality are obtained over binary masking. Further it is demonstrated that a vowel-dependent soft mask obtained from clean data of a different available singer is not as good as the singer-vowel dependent soft mask even if the latter is extracted from polyphonic audio.

## 8. ACKNOWLEDGEMENT

Part of the work was supported by Bharti Centre for Communication in IIT Bombay.

## 9. REFERENCES

- [1] L. Boucheron and P. De Leon. On the inversion of mel-frequency cepstral coefficients for speech enhancement applications. In *Int. Conf. Signals Electronic Systems, 2008.*, pages 485–488, 2008.
- [2] D. Fitzgerald. Vocal separation using nearest neighbours and median filtering. In *Irish Signals Systems Conf.*, 2012.
- [3] J. Han and B. Pardo. Reconstructing completely overlapped notes from musical mixtures. In *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process., 2011 (ICASSP '11.)*, pages 249 – 252, 2011.
- [4] M. Kim, J. Yoo, K. Kang, and S. Choi. Nonnegative matrix partial co-factorization for spectral and temporal drum source separation. *IEEE Journal Selected Topics Signal Process.*, 5(6):1192 – 1204, 2011.
- [5] A. Lefevre, F. Bach, and C. Fevotte. Semi-supervised NMF with time-frequency annotations for single-channel source separation. In *Proc. Int. Soc. Music Information Retrieval (ISMIR 2012)*, pages 115–120, 2012.
- [6] Y. Li and D. Wang. Separation of singing voice from music accompaniment for monaural recordings. *IEEE Trans. Audio, Speech, Lang. Process.*, 15(4):1475–1487, 2007.
- [7] B. Raj, R. Singh, and T. Virtanen. Phoneme-dependent NMF for speech enhancement in monaural mixtures. In *Proc. Interspeech*, pages 1217–1220, 2011.
- [8] V. Rao, C. Gupta, and P. Rao. Context-aware features for singing voice detection in polyphonic music. In *Proc. Adaptive Multimedia Retrieval*, Barcelona, Spain, 2011.
- [9] V. Rao and P. Rao. Vocal melody extraction in the presence of pitched accompaniment in polyphonic music. *IEEE Trans. Audio, Speech, Lang. Process.*, 18(8):2145–2154, 2010.
- [10] M. Ryynanen, T. Virtanen, J. Paulus, and A. Klauri. Accompaniment separation and karaoke application based on automatic melody transcription. In *IEEE Int. Conf. Multimedia Expo*, pages 1417–1420, 2008.
- [11] M. Schmidt and R. Olsson. Single-channel speech separation using sparse non-negative matrix factorization. In *Proc. of Interspeech*, pages 2617–2614, 2006.
- [12] J. Sundberg. The science of the singing voice. *Music Perception: An Interdisciplinary Journal*, 7(2):187 – 195, 1989.
- [13] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *IEEE Trans. Audio, Speech, Lang. Process.*, 14(4):1462–1469, 2006.
- [14] T. Virtanen, A. Mesaros, and M. Ryyänen. Combining pitch-based inference and non-negative spectrogram factorization in separating vocals from polyphonic music. In *ICSA Tutorial and Research Workshop on Statistical and Perceptual Audition*, Brisbane, Australia, 2008.
- [15] G. Zhou, A. Cichocki, Q. Zhao, and S. Xie. Nonnegative matrix and tensor factorizations: An algorithmic perspective. *IEEE Signal Process. Magazine*, pages 54–65, 2014.



# IMPROVED QUERY-BY-TAPPING VIA TEMPO ALIGNMENT

**Chun-Ta Chen**

Department of Computer Science  
National Tsing Hua University  
Hsinchu, Taiwan  
chun-ta.chen@mirlab.org

**Jyh-Shing Roger Jang**

Department of Computer Science  
National Taiwan University  
Taipei, Taiwan  
jang@mirlab.org

**Chun-Hung Lu**

Innovative Digitech-Enabled Applications & Services Institute (IDEAS), Institute for Information Industry, Taiwan  
enricoghlu@iii.org.tw

## ABSTRACT

Query by tapping (QBT) is a content-based music retrieval method that can retrieve a song by taking the user's tapping or clapping at the note onsets of the intended song in the database for comparison. This paper proposes a new query-by-tapping algorithm that aligns the IOI (inter-onset interval) vector of the query sequence with songs in the dataset by building an IOI ratio matrix, and then applies a dynamic programming (DP) method to compute the optimum path with minimum cost. Experiments on different datasets indicate that our algorithm outperforms other previous approaches in accuracy (top-10 and MRR), with a speedup factor of 3 in computation. With the advent of personal handheld devices, QBT provides an interesting and innovative way for music retrieval by shaking or tapping the devices, which is also discussed in the paper.

## 1. INTRODUCTION

QBT is a mechanism for content-based music retrieval which extracts the note onset time from recordings of users' input tapping or symbolic signals, which it then compares against a song database to retrieve the correct song. Unlike query-by-singing/humming (QBSH) [1, 2], which takes the user's melody pitch for comparison, QBT only uses the note duration for comparison, with no pitch information. This makes QBT more difficult to implement than QBSH, because the note onset in QBT contains less information than the musical pitch in QBSH, raising the likelihood of collision. For example, musical pieces with different melodies but similar rhythmic patterns may be characterized by the same onset sequence.

One may argue that QBT is not a popular way of music retrieval. Some people may even think it is not useful. However, with the advent of personal handheld devices, we can think QBT as a novel way of human-computer interface. For instance, with the use of QBT, one may shake or click his/her mobile phones in order to retrieve a song. Moreover, one can even use a personal style of shaking or clicking as the password to unlock a phone. These innovative ways of human-machine interface indicate that QBT, though not the most popular way of music

retrieval, is itself interesting and could pave the way for other innovative applications [10].

QBT system algorithms are based on the estimation of the similarity between two onset sequences. For example, G. Eisenberg proposed a simple algorithm called "Direct Measure" to accomplish such comparisons [3, 4]. R. Typke presented a variant of the earth mover's distance appropriate for searching rhythmic patterns [5]. Among these algorithms, the techniques of dynamic programming (DP) have been widely used, such as R. Jang's Dynamic Time Warping (DTW) [6], G. Peters's edit distance algorithm [7, 8], and P. Hanna's adaptation of local alignment algorithm [9].

In this paper, we propose and test a new QBT algorithm. In Section 2, we discuss the general frameworks of QBT and existing QBT methods. Section 3 describes the proposed method. Experiments with different QBT techniques are described in Section 4. Finally, Section 5 concludes this paper.

## 2. THE QBT SYSTEM

Fig. 1 illustrates the flowchart of our query-by-tapping system. In general, there are 2 kinds of inputs to a QBT system:

- Symbolic input: The onset time of the tapping event is provided symbolically with little or no ambiguity. For instance, the user may tap on a PC's keyboard or an iPad's touch panel to give the onset time.
- Acoustic input: The onset time is extracted from acoustic input of the user's tapping on a microphone. This input method requires additional onset detection to extract the onset time of the acoustic input. For example, we can estimate the onset time by local-maximum-picking of the input audio's intensity as in [5], or by detecting the transients of kurtosis variation as in [7].

The input onset sequence can be obtained as the inter onset interval (IOI) vector whose elements are the difference between two successive onset times. The note onset sequences extracted from the monophonic MIDIs (or the melody track of polyphonic MIDIs) in the song database are also converted into IOIs in advance. We can then apply a QBT algorithm to compare the query IOI vector to those in the database in order to retrieve the most similar song from the database. A QBT algorithm usually needs to perform IOI vector normalization before similarity comparison. Normalization can take care of tempo devia-



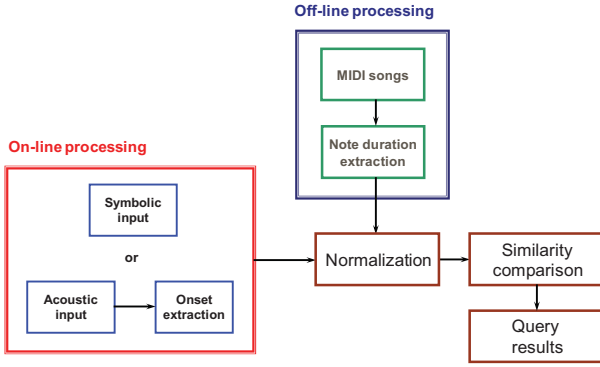


Fig. 1. QBT system flowchart

tion, which similarity comparison can handle possible insertion/deletion errors. Once normalization is performed, we can apply similarity comparison to find the similarity between the IOI query vector and that of each database song. The system can then return a ranked list of all database songs according to their similarity to the query input.

Normalization and the similarity comparison are detailed in the following sections.

### 2.1 Normalization of IOI Vectors

In most cases, the tempo of the user's query input is different from those of the candidate songs in the database. To deal with this problem, we need to normalize the IOI vectors of the input query and the candidate songs. There are 2 common methods for normalization. The first one is to convert the summation of all IOI to a constant value [5].

$$\begin{aligned}\tilde{q}_i &= q_i / \sum_{k=1}^m q_k \\ \tilde{r}_j &= r_j / \sum_{k=1}^{\tilde{n}} r_k\end{aligned}\quad (1)$$

where  $\{q_i, i=1\sim m\}$  is the input query IOI vector, and  $\{r_j, j=1\sim \tilde{n}\}$  is a reference IOI vector from the song database. Note that the reference IOI vector of a song is truncated to a variety of lengths in order to match the query IOI. For instance,  $\tilde{n}$  may be set to a value from  $m-2$  to  $m+2$  in order to deal with possible insertions and deletions in the query input. Thus all these variant normalized versions of the IOI vectors for a song must be compared for similarity with the query IOI vector. The second method is to represent the normalized IOI vector as the ratio of the current IOI element to its preceding element [7]. That is:

$$\begin{cases} \tilde{s}_1 = 1 \\ \tilde{s}_i = s_i / s_{i-1}, \text{ if } i \geq 2 \end{cases}\quad (2)$$

where  $\{s_i\}$  is the original input query or reference IOI vector, and  $\{\tilde{s}_i\}$  is its normalized version. The advantage of this method is that computation-wise it is much simpler than the first one. However, this method is susceptible to the problem of magnified insertion and deletion

errors of the original IOI vectors, if any. For example, an IOI vector is  $[1, 2, 1]$ , then its normalized vector is  $[1, 2, 0.5]$ . If this IOI vector is wrongly tapped as  $[1, 1, 1, 1]$  (i.e., with one insertion in the second IOI), the normalized will become  $[1, 1, 1, 1]$ , which has a larger degree of difference from the groundtruth after normalization. This kind of amplified difference is harder to recover in the step of similarity comparison.

### 2.2 Similarity Comparison

A robust QBT system should be able to handle insertion and deletion errors since most of the common users are not likely to tap the correct note sequence of the intended song precisely. In particular, a common user is likely to lose one or several notes when the song has a fast tempo, which leads to deletion errors. On the other hand, though less likely, a user may have a wrong impression of the intended song and taps more notes instead, which lead to insertion errors. Several methods have been proposed to compare IOI vectors for QBT, including the earth mover's distance [4] and several DP-based methods [5], [6], [7] which can deal with two input vectors of different lengths. In general, the earth mover's distance is faster than DP-based methods, but its retrieval accuracy is not as good [11]. Our goal is to obtain a good accuracy with a reasonable amount of computing time. Therefore, the proposed method is based on a highly efficient DP-based method for better accuracy.

## 3. THE SHIFTED ALIGNMENT ALGORITHM

This section presents the proposed method to QBT. The method can also be divided into two stages of IOI normalization and similarity comparison. We shall describe these two steps and explain the advantages over the state-of-art QBT methods.

**Normalization:** In QBT, though the query IOI vector and its target song IOI vector are not necessarily of the same size, the ratio of their tempos should be close to a constant. In other words, the ratios of an IOI element of a query to the corresponding one of the target song should be close to a constant. To take advantage of this fact, we can shift the query IOI vector (relatively to the target song IOI vector) to construct an IOI ratio matrix in order to find the optimum mapping between IOI elements of these two sequences. An example is shown in Fig. 2(a), where the input query IOI vector is represented by  $\{q_i, i=1\sim m\}$ , and the reference IOI vector from the song database by  $\{r_j, j=1\sim n\}$ . As displayed in the figure, the reference IOI vector is shown at the top and the shifted query IOI vectors are shown below. Each element of a shifted query IOI vector is mapped to that of the reference IOI vector in the same column. Take the first shifted query IOI vector as an example, its second element  $q_2$  is mapped to  $r_1$  of the reference IOI vector,  $q_3$  is mapped to  $r_2$ , etc. For each matched element pair, we divide the

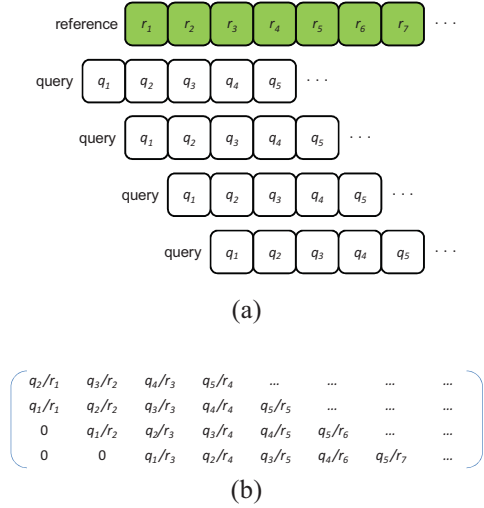


Fig. 2. Example of the normalization step of the shifted alignment algorithm: (a) Reference IOI vector and the shifted query IOI vectors. (b) IOI ratio matrix.

query IOI by its mapping reference IOI to construct an IOI ratio matrix  $M$  according to the following formula:

$$M_{i,j} = \begin{cases} q_{i-s-i+j+1} / r_j & , \text{ if } 1 \leq i-s-i+j+1 \leq \min(m,n) \\ 0 & , \text{ otherwise} \end{cases} \quad (3)$$

where the size of the matrix  $M$  is  $\min(m,n) * (i_s + i_e + 1)$ .  $i_s$  and  $i_e$  are the left- and right-shift amount of the query IOI vector, respectively. Fig. 2(b) is the IOI ratio matrix of fig. 2(a). In this example,  $i_s$  and  $i_e$  are 1 and 2, respectively. Since the length of the query is usually shorter,  $m$  is generally much less than  $n$ . Besides, in practice, if the anchor position is the beginning of a song, then we can improve the computation efficiency by truncating a reference IOI vector to a length slightly longer (e.g., 5-element longer) than the length of query IOI vector.

Unlike the equation (1) which requires many different versions of normalized reference IOI vectors for similarity comparison, the proposed approach requires only one-time normalization to generate a single IOI ratio table for computing the similarity. So the proposed approach is guaranteed to more efficient.

**Similarity comparison:** In order to handle insertions and deletions in a flexible yet robust manner, we propose a dynamic programming method to compute the similarity between the query and the reference IOI vectors. The basic principle is to identify a path over the IOI ratio matrix  $M$  where the elemental values along the path should be as close as possible to one another. In other words, the accumulated IOI ratio variation should be minimal along the optimal path. Fig. 3 illustrates two typical numeric examples that involve insertion and deletion in the optimal path. In fig. 3(a), query IOI vector and reference IOI vector have the same tempo, so their elements are pretty much the same except that there is an insertion in the query. That is, the fourth element of the reference IOI

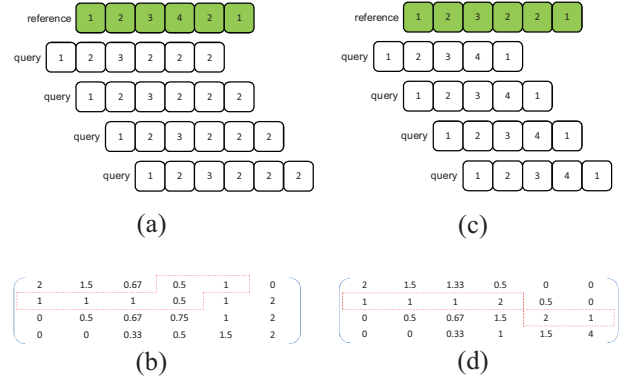


Fig. 3. Typical examples of the shifted alignment algorithm: (a) is an example where the query IOI vector has an insertion; (b) is the corresponding IOI ratio matrix; (c) is another example where the query IOI vector has a deletion; and (d) is the corresponding IOI ratio matrix. The path enclosed by dashed line in (b) and (d) represents the optimal DP path.

vector is equally split into 2 elements in the query. Fig. 3(b) is the IOI ratio matrix derived from the fig. 3(a), with the optimal path surrounded by dashed lines. The horizontal direction within the optimal path represent one-to-one sequential mapping between the two vectors without insertion or deletion. The vertical direction within the path indicates an insertion, where the 4th and 5th query IOI elements should be mapped to the 4th reference IOI element. On the other hand, Fig. 3(c) demonstrates an example of deletion where the query misses the 4th onset of the reference vector. Fig. 3(d) shows the corresponding IOI ratio matrix with the optimal path surrounded by dashed lines. The vertical shift of the path indicates a deletion where the 4th query IOI element should be mapped to the 4th and 5th reference IOI elements.

If there is no insertion or deletion in the query, each element along the optimal path should have a value close to its preceding element. With insertion or deletion, then the optimal path exhibits some specific behavior. Therefore our goal is to find the optimal path with minimal variations between neighboring elements in the path, with special consideration for specific path behavior to accommodate insertion and deletion. The variation between neighboring IOI ratio elements can be represented as the deviation between 1 and the ratio of one IOI ratio element to the preceding modified IOI ratio element, which takes into consideration the specific path behavior for accommodating insertion and deletion. The resulting recurrent equation for the optimum-value function  $D_{i,j}$  for DP is shown next:

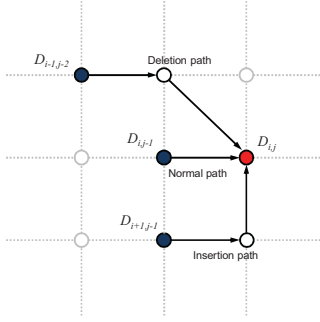


Fig. 4. Constrained DP paths

$$D_{i,j} = \min \begin{cases} D_{i,j-1} + \left| \frac{M_{i,j}}{H_{i,j-1}} - 1 \right| \\ D_{i+1,j-1} + \left| \frac{M_{i+1,j} + M_{i,j}}{H_{i+1,j-1}} - 1 \right| + \eta_1 \\ D_{i-1,j-2} + \left| \frac{M_{i-1,j-1} \cdot M_{i,j}}{M_{i-1,j-1} + M_{i,j}} \cdot \frac{1}{H_{i-1,j-2}} - 1 \right| + \eta_2 \end{cases} \quad (4)$$

where  $H_{i,j}$  is the modified IOI ratio element with values in the set  $\{M_{i,j}, M_{i+1,j}+M_{i,j}, M_{i-1,j-1}M_{i,j}/(M_{i-1,j-1}+M_{i,j})\}$ . The actual value of  $H_{i,j}$  depends on the backtrack index in the above formula. Specifically,  $H_{i,j}$  will respectively be set to the first, second or third element in the set if  $D_{i,j}$  takes its value from the first, second or third row of the above recurrent formula. The first row in the formula indicates one-by-one sequential mapping of the query and the reference IOI. The second row considers the case when the user commits an insertion error by taking one note as two, with the addition of a constant  $\eta_1$  as its penalty. The third row considers the case when the user commits a deletion error by taking two notes as one, with the addition of a constant  $\eta_2$  as its penalty. Fig. 4 illustrates these three conditions with three allowable local paths in the DP matrix  $D$ . Note that equation (4) does not consider some special cases of n-to-1 insertion or 1-to-n deletion when n is greater than 2. We can easily modify the equation in order to take such considerations, but we choose not to do so since these special cases rarely occur. Moreover, we want to keep the formula simple for straightforward implementation and better efficiency.

The degree of similarity between two IOI vectors can thus be determined from the matrix  $D$ . The strategy compares the elements in the corresponding positions of the last non-zeros element in each row of the matrix  $M$ . For example, if the DP matrix  $D$  is derived from the IOI ratio matrix  $M$  in Fig. 2(b), we need to compare the next-to-last element of the first row with the last element of the other rows in  $D$ . The optimal cost is the minimal value of these elements. The size of the DP matrix is  $\min(m,n) \cdot (i_s + i_e + 1)$ , which is less than the size  $(m \cdot n)$  of the DP algorithms in [6], [7], [9]. In addition, our algo-

rithm can be easily extended to the QBT system with “anywhere” anchor positions by setting the  $i_e$  to the length of the reference IOI vector.

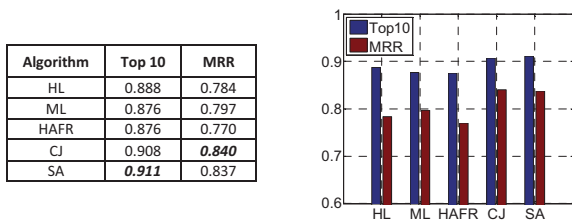
#### 4. PERFORMANCE EVALUATION

To evaluate the proposed method, we design 3 experiments and compare the performance with that of the state-of-art algorithm. The first experiment compares the recognition rate with algorithms in MIREX QBT task. The second experiment compares their computing speeds. The third experiment demonstrates the robustness of the proposed method using a larger dataset. These experiments are described in the following sub-sections.

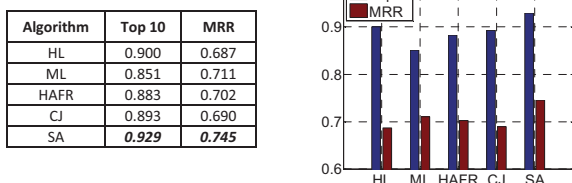
##### 4.1 MIREX QBT Evaluation Task

We have submitted our algorithm to the 2012 MIREX QBT task [12], which involves two subtasks for symbolic and acoustic inputs, respectively. Because the onset detection of acoustic input is not the focus of this paper, the following experiments only consider the case of queries with symbolic input. There are 2 datasets of symbolic input, including Jang's dataset of 890 queries (with groundtruth onsets to be used as the symbolic input) and 136 monophonic MIDIs, and Hsiao's dataset of 410 symbolic queries and 143 monophonic MIDIs. The queries of both datasets are all tapped from the beginning of the target song. These datasets are published in 2009 and can be downloaded from the MIREX QBT webpage. The top-10 hit rate and the mean reciprocal rank (MRR) are used as the performance indices of a submitted QBT method. Fig. 5 shows the performance of 5 submitted algorithms, with (a) and (b) are respectively the results of Jang's and Hsiao's datasets. Out of these five submissions, “HL” and “ML” do not have clear descriptions about their algorithms in the MIREX abstracts. Therefore, these 2 algorithms are not included in the experiments in section 4.2 and 4.3. “HAFR” is the implementation of [9], which claimed that its results outperformed other submissions, including the methods of [5] and [6], in MIREX 2008. The algorithm “CJ” is an improved version of [6]. The submission of “SA” is the proposed algorithm in this paper.

As shown in fig. 5(a), our algorithm outperforms almost all the other submissions except for the MRR in Jang's dataset where our submission is ranked the second. In fact, the MRR of our algorithm is only 0.3% lower than that of “CJ”. On the other hand, the top-10 hit rate of our submission is 0.3% higher than that of “CJ”. So the performances of “CJ” and “SA” are very close in this dataset. From fig. 5(b), it is obvious that our algorithm simply outperforms all the other submission in both MRR and top-10 hit rate. As a whole, the proposed method obtains good results in MIREX QBT contest.



(a) Result 1: Jang's dataset



(b) Result 2: Hsiao's dataset

Fig. 5. Results of MIREX QBT evaluation task

## 4.2 Evaluation of Computation Efficiency

In this experiment, we want to compare the efficiency of several QBT algorithms. We implemented three submissions (including ours) to 2012 MIREX QBT tasks in C language. The “ML” and “HL” algorithms were not included in this experiment due to the lack of clear descriptions about their algorithms in the MIREX abstracts. The experiment was conducted on a PC with an AMD Athlon 2.4GHz CPU and 1G RAM. Each algorithm was repeated 10 times over Jang's dataset to obtain the average computing time of a single query. The results are shown in Table 1 which indicates that our algorithm is at least 3 times faster than the other two algorithms. This is due to the fact that our algorithm has an efficient way of normalization for IOI vectors (as described in section 3), leading to a smaller table for DP optimal path finding.

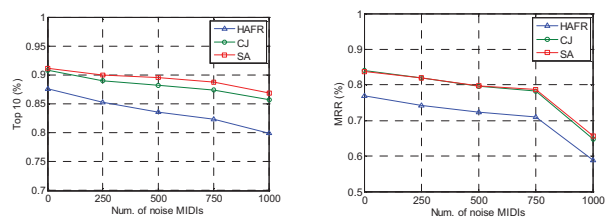
Algorithm	Avg. time (ms)
HAFR	421
CJ	213
SA	65

Table 1. Speed comparison of QBT algorithms

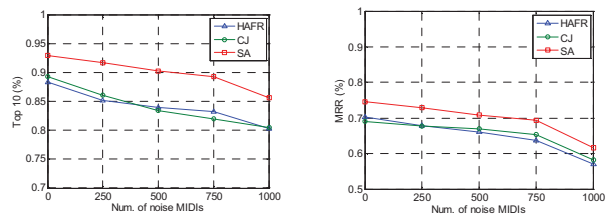
From these two experiments, we can claim that our algorithm strike a good balance between the recognition rate and computation efficiency.

## 4.3 Experiment with Larger Databases

The MIREX QBT datasets are well organized for QBT research. However, both datasets contain song databases of slightly more than 100 songs. These small database sizes lead to high accuracy for all submissions in MIREX QBT task. Therefore, we designed another experiment to demonstrate how the performance varies with the dataset



(a) Result 1: Jang's dataset



(b) Result 2: Hsiao's dataset

Fig. 6. Results of the performance versus database sizes. (a) is the performance of top-10 hit rate (left subplot) and MRR (right subplot) using Jang's dataset. (b) is the performance of top-10 hit rate (left subplot) and MRR (right subplot) using Hsiao's dataset.

sizes. We collected 1000 MIDIs which are different from the MIDIs in the MIREX QBT datasets. And we enlarge the original databases by adding 250 noise MIDIs each time, and evaluate the performance in both MRR and top-10 hit rate.

Fig. 6 shows the experimental results. As the number of noise MIDIs increases, the recognition rate of each algorithm gradually decreases. In Jang's dataset of the fig. 6(a), the top-10 hit rate of “SA” is the best among all algorithms (left subplot). However, the MRR of “SA” and “CJ” are very close and the value of one is slightly higher than the other in different number of noise MIDIs (right subplot). In fig. 6(b), our algorithm notably outperforms the others in both top-10 hit rate (left subplot) and MRR (right subplot). It is interesting to note that the decay of the top-10 hit rate of “SA” is slower than the others in both datasets, especially in Jang's dataset. This indicates that our algorithm has better resistance to these noise MIDIs in top-10 hit rate. In both datasets, “SA” still had >85% top-10 rate and >60% MRR. Therefore we can conclude that the proposed method is more robust in dealing with a large song database.

## 5. CONCLUSION

In this paper, we have proposed a shifted-alignment algorithm for QBT by constructing an IOI ratio matrix, in which each element is the ratio of relative IOI elements of the query and a reference song. The similarity comparison is based on DP to deal with possible insertions and deletions of query IOI vectors. We evaluated the performance of the proposed method with two datasets. The experimental results showed that our algorithm exhibited

an overall better accuracy than other submissions to 2012 MIREX query-by-taping task. Moreover, the computation time is at least 3 times faster than others. We also conducted an experiment to demonstrate that our algorithm performs better and more robustly than other existing QBT algorithms in the case of large databases. In particular, our algorithm has a top-10 hit rate larger than 85% and MRR larger than 60% in both databases when the number of noise MIDIs is as high as 1000.

Although the proposed method performs well in the experiments, the recognition rate still has room for further improvement, especially in the case of “anywhere” anchor position, that is, the user is allowed to start tapping from anywhere in the middle of a song. From the experimental results, we can observe that each algorithm has its strength and weakness in dealing with different queries and database songs. Therefore, one direction of our immediate future work is to find an optimal way to combine these methods for better accuracy.

## 6. ACKNOWLEDGEMENT

This study is conducted under the "NSC 102-3114-Y-307-026 A Research on Social Influence and Decision Support Analytics" of the Institute for Information Industry which is subsidized by the National Science Council.

## 7. REFERENCES

- [1] R. B. Dannenberg, W. P. Birmingham, B. Pardo, N. Hu, C. Meek, and G. Tzanetakis: “A Comparative Evaluation of Search Techniques for Query-by-Humming Using the MUSART Testbed,” *Journal of the American Society for Information Science and Technology* (JASIST), vol. 58, no. 5, pp. 687–701, 2007.
- [2] J.-S. Roger Jang, H. R. Lee, and M. Y. Kao: “Content-based Music Retrieval Using Linear Scaling and Branch-and-bound Tree Search,” *IEEE International Conference on Multimedia and Expo*, pp. 289-292, 2001.
- [3] G. Eisenberg, J. Batke, and T. Sikora: “Beatbank - an MPEG-7 Compliant Query by Tapping System,” *116th Convention of the Audio Engineering Society*, Berlin, Germany, pp.189-192, May 2004.
- [4] G. Eisenberg, J. M. Batke, and T. Sikora: “Efficiently Computable Similarity Measures for Query by Tapping System,” *Proc. of the 7th Int. Conference on Digital Audio Effects* (DAFx'04), October, 2004.
- [5] R. Typke, and A. W. Typke: “A Tunneling-Vantage Indexing Method for Non-Metrics,” *9th International Conference on Music Information Retrieval*, Philadelphia, USA, pp683-688, September 14-18, 2008
- [6] J.-S. Roger Jang, H. R. Lee, C. H. Yeh: “Query by Tapping A New Paradigm for Content-Based Music Retrieval from Acoustic input,” *Second IEEE Pacific-Rim Conference on Multimedia*, pp590-597, October, 2001
- [7] G. Peters, C. Anthony, and M. Schwartz: “Song Search And Retrieval by Tapping,” *Proceedings of AAAI'05 Proceedings of the 20th national conference on Artificial intelligence*, pp. 1696-1697, 2005
- [8] G. Peters, D. Cukierman, C. Anthony, and M. Schwartz: “Online Music Search by Tapping,” *Ambient Intelligence in Everyday Life*, pages 178–197. Springer, 2006.
- [9] P. Hanna and M. Robine: “Query By Tapping System Based On Alignment Algorithm,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), pp. 1881-1884, 2009.
- [10] B. Kaneshiro, H. S. Kim, J. Herrera, J. Oh, J. Berger and M. Slaney. “QBT-Extended: An Annotated Dataset of Melodically Contoured Tapped Queries,” *Proceedings of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, November, 2013.
- [11] L. Wang: “MIREX 2012 QBSH Task: YINLONG’s Solution,” *Music Information Retrieval Evaluation eXchange 2012*.
- [12] The Music Information Retrieval Evaluation eXchange evaluation task of query by tapping: [http://www.music-ir.org/mirex/wiki/2012:Query\\_by\\_Tapping](http://www.music-ir.org/mirex/wiki/2012:Query_by_Tapping)

# AUTOMATIC TIMBRE CLASSIFICATION OF ETHNOMUSICOLOGICAL AUDIO RECORDINGS

Dominique Fourer, Jean-Luc Rouas, Pierre Hanna, Matthias Robine

LaBRI - CNRS UMR 5800 - University of Bordeaux

{fourer, rouas, hanna, robine}@labri.fr

## ABSTRACT

Automatic timbre characterization of audio signals can help to measure similarities between sounds and is of interest for automatic or semi-automatic databases indexing. The most effective methods use machine learning approaches which require qualitative and diversified training databases to obtain accurate results. In this paper, we introduce a diversified database composed of worldwide non-western instruments audio recordings on which is evaluated an effective timbre classification method. A comparative evaluation based on the well studied Iowa musical instruments database shows results comparable with those of state-of-the-art methods. Thus, the proposed method offers a practical solution for automatic ethnomusicological indexing of a database composed of diversified sounds with various quality. The relevance of audio features for the timbre characterization is also discussed in the context of non-western instruments analysis.

## 1. INTRODUCTION

Characterizing musical timbre perception remains a challenging task related to the human auditory mechanism and to the physics of musical instruments [4]. This task is full of interest for many applications like automatic database indexing, measuring similarities between sounds or for automatic sound recognition. Existing psychoacoustical studies model the timbre as a multidimensional phenomenon independent from musical parameters (e.g. pitch, duration or loudness) [7, 8]. A quantitative interpretation of instrument's timbre based on acoustic features computed from audio signals was first proposed in [9] and pursued in more recent studies [12] which aim at organizing audio timbre descriptors efficiently. Nowadays, effective automatic timbre classification methods [13] use supervised statistical learning approaches based on audio signals features computed from analyzed data. Thus, the performance obtained with such systems depends on the taxonomy, the size and the diversity of training databases. However, most

of existing research databases (e.g. RWC [6], Iowa [5]) are only composed of common western instruments annotated with specific taxonomies. In this work, we revisit the automatic instrument classification problem from an ethnomusicological point of view by introducing a diversified and manually annotated research database provided by the *Centre de Recherche en Ethno-Musicologie* (CREM). This database is daily supplied by researchers and has the particularity of being composed of uncommon non-western musical instrument recordings from around the world. This work is motivated by practical applications to automatic indexing of online audio recordings database which have to be computationally efficient while providing accurate results. Thus, we aim at validating the efficiency and the robustness of the statistical learning approach using a constrained standard taxonomy, applied to recordings of various quality. In this study, we expect to show the database influence, the relevance of timbre audio features and the choice of taxonomy for the automatic instrument classification process. A result comparison and a cross-database evaluation is performed using the well-studied university of Iowa musical instrument database. This paper is organized as follows. The CREM database is introduced in Section 2. The timbre quantization principle based on mathematical functions describing audio features is presented in Section 3. An efficient timbre classification method is described in Section 4. Experiments and results based on the proposed method are detailed in Section 5. Conclusion and future works are finally discussed in Section 6.

## 2. THE CREM ETHNOMUSICOLOGICAL DATABASE

The CREM research database<sup>1</sup> is composed of diversified sound samples directly recorded by ethnomusicologists in various conditions (i.e. no recording studio) and from diversified places all around the world. It contains more than 7000 hours of audio data recorded since 1932 to nowadays using different supports like magnetic tapes or vinyl discs. The vintage audio recordings of the database were carefully digitized to preserve the authenticity of the originals and contain various environment noise. The more recent audio recordings can be directly digital recorded with a high-quality. Most of the musical instruments which com-



© Dominique Fourer, Jean-Luc Rouas, Pierre Hanna, Matthias Robine.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Dominique Fourer, Jean-Luc Rouas, Pierre Hanna, Matthias Robine. "Automatic timbre classification of ethnomusicological audio recordings", 15th International Society for Music Information Retrieval Conference, 2014.

<sup>1</sup> CREM audio archives freely available online at: <http://archives.crem-cnrs.fr/>

pose this database are non-western and can be uncommon while covering a large range of musical instrument families (see Figure 1(a)). Among uncommon instruments, one can find the lute or the Ngbaka harp as cordophones. More uncommon instruments like Oscillating bamboo, struck machete and struck girder were classified by ethnomusicologists as idiophones. In this paper, we restricted our study to the solo excerpts (where only one monophonic or polyphonic instrument is active) to reduce the interference problems which may occur during audio analysis. A description of the selected CREM sub-database is presented in Table 1. According to this table, one can observe that this database is actually inhomogeneous. The aerophones are overrepresented while membranophones are underrepresented. Due to its diversity and the various quality of the composing sounds, the automatic ethnomusicological classification of this database may appear as challenging.

Class name	Duration (s)		#	
aerophones-blown	1,383		146	
cordophones-struck	357	1,229	37	128
cordophones-plucked	715		75	
cordophones-bowed	157		16	
idiophones-struck	522	753	58	82
idiophones-plucked	137		14	
idiophones-clinked	94		10	
membranophones-struck	170		19	
Total	3,535		375	

**Table 1.** Content of the CREM sub-database with duration and number of 10-seconds segmented excerpts.

### 3. TIMBRE QUANTIZATION AND CLASSIFICATION

#### 3.1 Timbre quantization

Since preliminaries works on the timbre description of perceived sounds, Peeters *et al.* proposed in [12] a large set of audio features descriptors which can be computed from audio signals. The audio descriptors define numerical functions which aim at providing cues about specific acoustic features (e.g. brightness is often associated with the spectral centroid according to [14]). Thus, the audio descriptors can be organized as follows:

- Temporal descriptors convey information about the time evolution of a signal (e.g. log attack time, temporal increase, zero-crossing rate, etc.).
- Harmonic descriptors are computed from the detected pitch events associated with a fundamental frequency ( $F_0$ ). Thus, one can use a prior waveform model of quasi-harmonic sounds which have an equally spaced Dirac comb shape in the magnitude spectrum. The tonal part of sounds can be isolated from signal mixture and be described (e.g. noisiness, inharmonicity, etc.).
- Spectral descriptors are computed from signal time-frequency representation (e.g. Short-Term Fourier

Transform) without prior waveform model (e.g. spectral centroid, spectral decrease, etc.)

- Perceptual descriptors are computed from auditory-filtered bandwidth versions of signals which aim at approximating the human perception of sounds. This can be efficiently computed using Equivalent Rectangular Bandwidth (ERB) scale [10] which can be combined with gammatone filter-bank [3] (e.g. loudness, ERB spectral centroid, etc.)

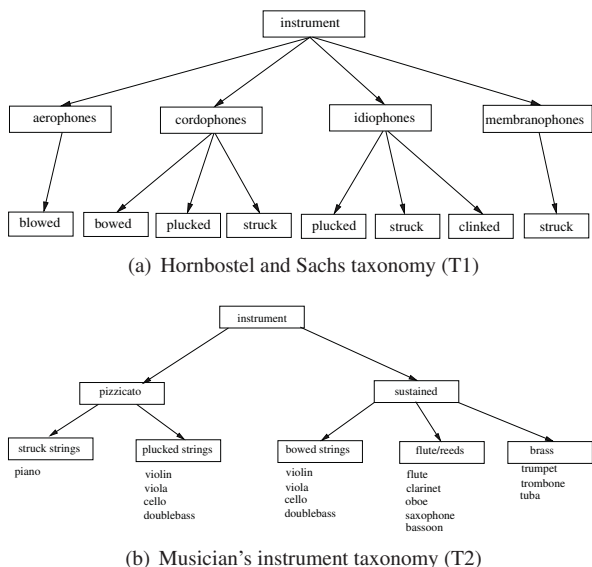
In this study, we focus on the sound descriptors listed in table 2 which can be estimated using the timbre toolbox<sup>2</sup> and detailed in [12]. All descriptors are computed for each analyzed sound excerpt and may return null values. The harmonic descriptors of polyphonic sounds are computed using the prominent detected  $F_0$  candidate (single  $F_0$  estimation). To normalize the duration of analyzed sound, we separated each excerpt in 10-seconds length segments without distinction of silence or pitch events. Thus, each segment is represented by a real vector where the corresponding time series of each descriptor is summarized by a statistic. The median and the Inter Quartile Range (IQR) statistics were chosen for their robustness to outliers.

Acronym	Descriptor name	#
Att	Attack duration (see ADSR model [15])	1
AttSlp	Attack slope (ADSR)	1
Dec	Decay duration (ADSR)	1
DecSlp	Decay slope (ADSR)	1
Rel	Release duration (ADSR)	1
LAT	Log Attack Time	1
Tcent	Temporal centroid	1
Edur	Effective duration	1
FreqMod, AmpMod	Total energy modulation (frequency,amplitude)	2
RMSenv	RMS envelope	2
ACor	Signal Auto-Correlation function (12 first coef.)	24
ZCR	Zero-Crossing Rate	2
HCent	Harmonic spectral centroid	2
HSprd	Harmonic spectral spread	2
Hskew	Harmonic skewness	2
HKurt	Harmonic kurtosis	2
HSlp	Harmonic slope	2
HDec	Harmonic decrease	2
HRoff	Harmonic rolloff	2
HVar	Harmonic variation	2
HErg, HNErg, HFErg,	Harmonic energy, noise energy and frame energy	6
HNois	Noisiness	2
HF0	Fundamental frequency $F_0$	2
HinH	Inharmonicity	2
HTris	Harmonic trispectrum	6
HodevR	Harmonic odd to even partials ratio	2
Hdev	Harmonic deviation	2
SCent, ECent	Spectral centroid of the magnitude and energy spectrum	4
SSprd, ESprd	Spectral spread of the magnitude and energy spectrum	4
SSkew, ESkew	Spectral skewness of the magnitude and energy spectrum	4
SKurt, EKurt	Spectral kurtosis of the magnitude and energy spectrum	4
SSlp, ESLp	Spectral slope of the magnitude and energy spectrum	4
SDec, EDec	Spectral decrease of the magnitude and energy spectrum	4
SRoff, ERoff	Spectral rolloff of the magnitude and energy spectrum	4
SVar, EVar	Spectral variation of the magnitude and energy spectrum	4
SFErg, EFErg	Spectral frame energy of the magnitude and energy spectrum	4
Sflat, ESflat	Spectral flatness of the magnitude and energy spectrum	4
Scre, EScre	Spectral crest of the magnitude and energy spectrum	4
ErbCent, ErbGCent	ERB scale magnitude spectrogram / gammatone centroid	4
ErbSprd, ErbGSprd	ERB scale magnitude spectrogram / gammatone spread	4
ErbSkew, ErbGskew	ERB scale magnitude spectrogram / gammatone skewness	4
ErbKurt, ErbGKurt	ERB scale magnitude spectrogram / gammatone kurtosis	4
ErbSlp, ErbGSlp	ERB scale magnitude spectrogram / gammatone slope	4
ErbDec, ErbGDec	ERB scale magnitude spectrogram / gammatone decrease	4
ErbRoff, ErbGRoff	ERB scale magnitude spectrogram / gammatone rolloff	4
ErbVar, ErbGVar	ERB scale magnitude spectrogram / gammatone variation	4
ErbFErg, ErbGFErg	ERB scale magnitude spectrogram / gammatone frame energy	4
ErbSflat, ErbGSflat	ERB scale magnitude spectrogram / gammatone flatness	4
ErbScre, ErbGScre	ERB scale magnitude spectrogram / gammatone crest	4
Total		164

**Table 2.** Acronym, name and number of the used timbre descriptors.

<sup>2</sup> MATLAB code available at <http://www.cirmmt.org/research/tools>





**Figure 1.** Taxonomies used for the automatic classification of musical instruments as proposed by Hornbostel and Sachs taxonomy in [16] (a) and Peeters in [13] (b).

### 3.2 Classification taxonomy

In this study, we use two databases which can be annotated using different taxonomies. Due to its diversity, the CREM database was only annotated using the Hornbostel and Sachs taxonomy [16] (T1) illustrated in Figure 1(a) which is widely used in ethnomusicology. This hierarchical taxonomy is general enough to classify uncommon instruments (e.g. struck bamboo) and conveys information about sound production materials and playing styles. From an another hand, the Iowa musical instruments database [5] used in our experiments was initially annotated using a musician's instrument taxonomy (T2) as proposed in [13] and illustrated in Figure 1(b). This database is composed of common western pitched instruments which can easily be annotated using T1 as described in Table 3. One can notice that the Iowa database is only composed of aerophones and cordophones instruments. If we consider the playing style, only 4 classes are represented if we apply T1 taxonomy to the Iowa database.

T1 class name	T2 equivalence	Duration (s)	#
aero-blown	reed/flute and brass	5,951	668
cordo-struck	struck strings	5,564	646
cordo-plucked	plucked strings	5,229	583
cordo-bowed	bowed strings	7,853	838
Total		24,597	2,735

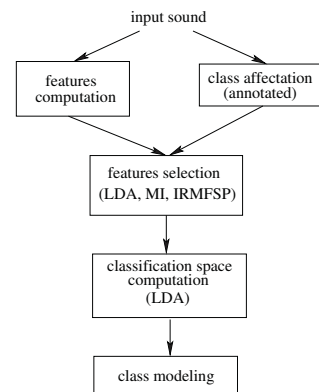
**Table 3.** Content of the Iowa database using musician's instrument taxonomy (T2) and equivalence with the Hornbostel and Sachs taxonomy (T1).

## 4. AUTOMATIC INSTRUMENT TIMBRE CLASSIFICATION METHOD

The described method aims at estimating the corresponding taxonomy class name of a given input sound.

### 4.1 Method overview

Here, each sound segment (cf. Section 3.1) is represented by vector of length  $p = 164$  where each value corresponds to a descriptor (see Table 2). The training step of this method (illustrated in Figure 2) aims at modeling each timbre class using the best projection space for classification. A features selection algorithm is first applied to efficiently reduce the number of descriptors to avoid statistical over-learning. The classification space is computed using discriminant analysis which consists in estimating optimal weights over the descriptors allowing the best discrimination between timbre classes. Thus, the classification task consists in projecting an input sound into the best classification space and to select the most probable timbre class using the learned model.



**Figure 2.** Training step of the proposed method.

### 4.2 Linear discriminant analysis

The goal of Linear Discriminant Analysis (LDA) [1] is to find the best projection or linear combination of all descriptors which maximizes the average distance between classes (inter-class distance) while minimizing distance between individuals from the same class (intra-class distance). This method assumes that the class affectation of each individual is a priori known. Its principle can be described as follows. First consider the  $n \times p$  real matrix  $M$  where each row is a vector of descriptors associated to a sound (individual). We assume that each individual is a member of a unique class  $k \in [1, K]$ . Now we define  $W$  as the intra-class variance-covariance matrix which can be estimated by:

$$W = \frac{1}{n} \sum_{k=1}^K n_k W_k, \quad (1)$$

where  $W_k$  is the variance-covariance matrix computed from the  $n_k \times p$  sub-matrix of  $M$  composed of the  $n_k$  individuals included into the class  $k$ .

We also define  $B$  the inter-class variance-covariance matrix expressed as follows:

$$B = \frac{1}{n} \sum_{k=1}^K n_k (\mu_k - \mu)(\mu_k - \mu)^T, \quad (2)$$

where  $\mu_k$  corresponds to the mean vector of class  $k$  and  $\mu$  is the mean vector of the entire dataset. According to [1], it can be shown that the eigenvectors of matrix  $D = (B + W)^{-1}B$  solve this optimization problem. When the matrix  $A = (B + W)$  is not invertible, a computational solution consists in using pseudoinverse of matrix  $A$  which can be calculated using  $A^T(AA^T)^{-1}$ .

### 4.3 Features selection algorithms

Features selection aims at computing the optimal relevance of each descriptor which can be measured with a weight or a rank. The resulting descriptors subset has to be the most discriminant as possible with the minimal redundancy. In this study, we investigate the three approaches described below.

#### 4.3.1 LDA features selection

The LDA method detailed in Section 4.2 can also be used for selecting the most relevant features. In fact, the computed eigenvectors which correspond to linear combination of descriptors convey a relative weight applied to each descriptor. Thus, the significance (or weight)  $S_d$  of a descriptor  $d$  can be computed using a summation over a defined range  $[1, R]$  of the eigenvectors of matrix  $D$  as follows:

$$S_d = \sum_{r=1}^R |v_{r,d}|, \quad (3)$$

where  $v_{r,d}$  is the  $d$ -th coefficient of the  $r$ -th eigenvector associated to the eigenvalues sorted by descending order (i.e.  $r = 1$  corresponds to the maximal eigenvalue of matrix  $D$ ). In our implementation, we fixed  $R = 8$ .

#### 4.3.2 Mutual information

Features selection algorithms aim at computing a subset of descriptors that conveys the maximal amount of information to model classes. From a statistical point of view, if we consider classes and feature descriptors as realizations of random variables  $C$  and  $F$ . The relevance can be measured with the mutual information defined by:

$$I(C, F) = \sum_c \sum_f P(c, f) \frac{P(c, f)}{P(c)P(f)}, \quad (4)$$

where  $P(c)$  denotes the probability of  $C = c$  which can be estimated from the approximated probability density functions (pdf) using a computed histogram. According to Bayes theorem one can compute  $P(c, f) = P(f|c)P(c)$  where  $P(f|c)$  is the pdf of the feature descriptor value  $f$  into class  $c$ . This method can be improved using [2] by reducing simultaneously the redundancy by considering the mutual information between previously selected descriptors.

#### 4.3.3 Inertia Ratio Maximisation using features space projection (IRMFSP)

This algorithm was first proposed in [11] to reduce the number of descriptors used by timbre classification methods. It consists in maximizing the relevance of the de-

scriptors subset for the classification task while minimizing the redundancy between the selected ones. This iterative method ( $\iota \leq p$ ) is composed of two steps. The first one selects at iteration  $\iota$  the non-previously selected descriptor which maximizes the ratio between inter-class inertia and the total inertia expressed as follow:

$$\hat{d}^{(\iota)} = \arg \max_d \frac{\sum_{k=1}^K n_k (\mu_{d,k} - \mu_d)(\mu_{d,k} - \mu_d)^T}{\sum_{i=1}^n (f_{d,i}^{(\iota)} - \mu_d)(f_{d,i}^{(\iota)} - \mu_d)^T}, \quad (5)$$

where  $f_{d,i}^{(\iota)}$  denotes the value of descriptor  $d \in [1, p]$  affected to the individual  $i$ .  $\mu_{d,k}$  and  $\mu_d$  respectively denote the average value of descriptor  $d$  into the class  $k$  and for the total dataset. The second step of this algorithm aims at orthogonalizing the remaining data for the next iteration as follows:

$$f_d^{(\iota+1)} = f_d^{(\iota)} - \left( f_d^{(\iota)} \cdot g_{\hat{d}} \right) g_{\hat{d}} \quad \forall d \neq \hat{d}^{(\iota)}, \quad (6)$$

where  $f_{\hat{d}}^{(\iota)}$  is the vector of the previously selected descriptor  $\hat{d}^{(\iota)}$  for all the individuals of the entire dataset and  $g_{\hat{d}} = f_{\hat{d}}^{(\iota)} / \|f_{\hat{d}}^{(\iota)}\|$  is its normalized form.

### 4.4 Class modeling and automatic classification

Each instrument class is modeled into the projected classification space resulting from the application of LDA. Thus, each class can be represented by its gravity center  $\hat{\mu}_k$  which corresponds to the vector of the averaged values of the projected individuals which compose the class  $k$ . The classification decision which affect a class  $\hat{k}$  to an input sound represented by a projected vector  $\hat{x}$  is simply performed by minimizing the Euclidean distance with the gravity center of each class as follows:

$$\hat{k} = \arg \min_k \|\hat{\mu}_k - \hat{x}\|_2 \quad \forall k \in [1, K], \quad (7)$$

where  $\|v\|_2$  denotes the  $l_2$  norm of vector  $v$ . Despite its simplicity, this method seems to obtain good results comparable with those of the literature [12].

## 5. EXPERIMENTS AND RESULTS

In this section we present the classification results obtained using the proposed method described in Section 4.

### 5.1 Method evaluation based on self database classification

In this experiment, we evaluate the classification of each distinct database using different taxonomies. We applied the 3-fold cross validation methodology which consists in partitioning the database in 3 distinct random subsets composed with 33% of each class (no collision between sets). Thus, the automatic classification applied on each subset is based on training applied on the remaining 66% of the

database. Figure 5.1 compares the classification accuracy obtained as a function of the number of used descriptors. The resulting confusion matrix of the CREM database using 20 audio descriptors is presented in Table 4 and shows an average classification accuracy of 80% where each instrument is well classified with a minimal accuracy of 70% for the aerophones. These results are good and seems comparable with those described in the literature [11] using the same number of descriptor. The most relevant feature descriptors (selected among the top ten) estimated by the IRMSFP and used for the classification task are detailed in Table 7. This result reveals significant differences between the two databases. As an example, harmonic descriptors are only discriminative for the CREM database but not for the Iowa database. This may be explained by the presence of membranophone in the CREM database which are not present in the Iowa database. Contrarily, spectral and perceptual descriptors seems more relevant for the Iowa database than for the CREM database. Some descriptors appear to be relevant for both database like the Spectral flatness (Sflat) and the ERB scale frame energy (ErbFErg) which describe the spectral envelope of signal.

	aero	c-struc	c-pluc	c-bowed	i-pluc	i-struc	i-clink	membr
aero	70	3	9	5		7		5
c-struc	6	92		3				
c-pluc	5	8	73	4		8		1
c-bowed			13	80	7			
i-pluc					79	14		7
i-struc	9	2	5		2	79		4
i-clink							100	
membr			11			17		72

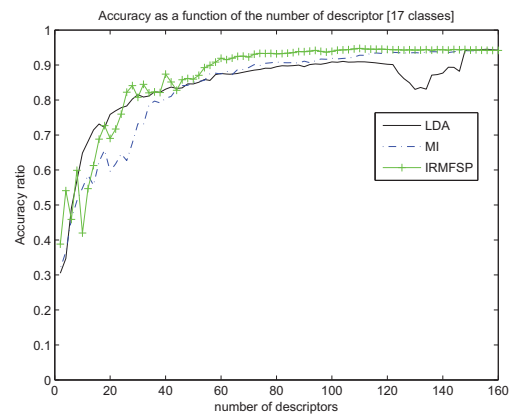
**Table 4.** Confusion matrix (expressed in percent of the sounds of the original class listed on the left) of the CREM database using the 20 most relevant descriptors selected by IRMSFP.

## 5.2 Cross-database evaluation

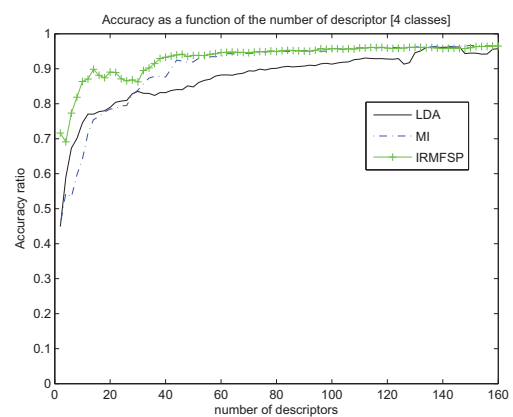
In this experiments (see Table 5), we merged the two databases and we applied the 3-fold cross validation method based on the T1 taxonomy to evaluate the classification accuracy on both database. The resulting average accuracy is about 68% which is lower than the accuracy obtained on the distinct classification of each database. The results of cross-database evaluation applied between databases using the T1 taxonomy are presented in Table 6 and obtain a poor average accuracy of 30%. This seems to confirm our intuition that the Iowa database conveys insufficient information to distinguish the different playing styles between the non-western cordophones instruments of the CREM database.

## 6. CONCLUSION AND FUTURE WORKS

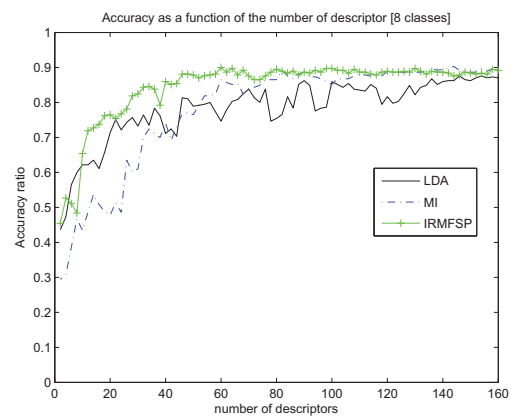
We applied a computationally efficient automatic timbre classification method which was successfully evaluated on an introduced diversified database using an ethnomusicological taxonomy. This method obtains good classification results (> 80% of accuracy) for both evaluated databases which are comparable to those of the literature. However,



(a) Iowa database using T2



(b) Iowa database using T1



(c) CREM database using T1

**Figure 3.** Comparison of the 3-fold cross validation classification accuracy as a function of the number of optimally selected descriptors.

the cross-database evaluation shows that each database cannot be used to infer a classification to the other. This can be explained by significant differences between these databases. Interestingly, results on the merged database obtain an acceptable accuracy of about 70%. As shown in previous work [11], our experiments confirm the efficiency of IRMFSP algorithm for automatic features selection applied to timbre classification. The interpretation of the

	aero	c-struct	c-pluc	c-bowed	i-pluc	i-struct	i-clink	membr
aero	<b>74</b>	14	5	3	2	1		
c-struct	12	<b>69</b>	10	5	1			2
c-pluc	1	7	<b>58</b>	29	1	2		2
c-bowed	3	6	33	<b>52</b>	1	3		
i-pluc		7		14	<b>79</b>			
i-struct	2	2	4	11	2	<b>51</b>		30
i-clink	11						<b>89</b>	
membr				6		17		<b>78</b>

**Table 5.** Confusion matrix (expressed in percent of the sounds of the original class listed on the left) of the evaluated fusion between the CREM and the Iowa database using the 20 most relevant descriptors selected by IRMSFP.

	aero	c-struct	c-pluc	c-bowed
aero	<b>72</b>	9	10	9
c-struct	12	<b>12</b>	34	42
c-pluc	23	47	<b>28</b>	3
c-bowed	28	34	24	<b>14</b>

**Table 6.** Confusion matrix (expressed in percent of the sounds of the original class listed on the left) of the CREM database classification based on Iowa database training.

CREM T1	Iowa T1	Iowa T2	CREM-Iowa T1
Edur Acor	AttSlp Dec	AttSlp Acor ZCR	AmpMod Acor RMSenv
Hdev Hnois HTris3			
Sflat	SFErg ERoff	Sflat SRoff SSkew	Sflat SVar SKurt Scre
ErbGKurt	ErbKurt ErbFErg ErbRoff ErbSlp ErbGCent	ErbSprd ErbFErg ErbGSprd	ErbFErg ErbRoff

**Table 7.** Comparison of the most relevant descriptors estimated by IRMFSP.

most relevant selected features shows a significant effect of the content of database rather than on the taxonomy. However the timbre modeling interpretation applied to timbre classification remains difficult. Future works will consist in further investigating the role of descriptors by manually constraining selection before the classification process.

## 7. ACKNOWLEDGMENTS

This research was partly supported by the French ANR (*Agence Nationale de la Recherche*) DIADEMS (*Description, Indexation, Acces aux Documents Ethnomusicologiques et Sonores*) project (ANR-12-CORD-0022).

## 8. REFERENCES

- [1] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley-Blackwell, New York, USA, 1958.
- [2] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. on Neural Networks*, 5(4):537–550, Jul. 1994.
- [3] E. Ambikairajah, J. Epps, and L. Lin. Wideband speech and audio coding using gammatone filter banks. In *Proc. IEEE ICASSP'01*, volume 2, pages 773–776, 2001.
- [4] N. F. Fletcher and T. D. Rossing. *The Physics of Musical Instruments*. Springer-Verlag, 1998.
- [5] L. Fritts. Musical instrument samples. Univ. Iowa Electronic Music Studios, 1997. [Online]. Available: <http://theremin.music.uiowa.edu/MIS.html>.
- [6] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. Rwc music database: Music genre database and musical instrument sound database. In *Proc. ISMIR*, pages 229–230, Oct. 2003.
- [7] J. M. Grey and J. W. Gordon. Perceptual effects of spectral modifications on musical timbre. *Journal of Acoustic Society of America (JASA)*, 5(63):1493–1500, 1978.
- [8] S. McAdams, S. Winsberg, S. Donnadieu, G. Soete, and J. Krimphoff. Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological Research*, 58(3):177–192, 1995.
- [9] N. Misdariis, K. Bennett, D. Pressnitzer, P. Susini, and S. McAdams. Validation of a multidimensional distance model for perceptual dissimilarities among musical timbres. In *Proc. ICA & ASA*, volume 103, Seattle, USA, Jun. 1998.
- [10] B.C.J. Moore and B.R. Glasberg. Suggested formulae for calculating auditory-filter bandwidths and excitation patterns. *Journal of the Acoustical Society of America*, 74:750–753, 1983.
- [11] G. Peeters. Automatic classification of large musical instrument databases using hierarchical classifiers with inertia ratio maximization. In *115th convention of AES*, New York, USA, Oct. 2003.
- [12] G. Peeters, B. Giordano, P. Susini, N. Misdariis, and S. McAdams. The timbre toolbox: Audio descriptors of musical signals. *Journal of Acoustic Society of America (JASA)*, 5(130):2902–2916, Nov. 2011.
- [13] G. Peeters and X. Rodet. Automatically selecting signal descriptors for sound classification. In *Proc. ICMC*, Göteborg, Sweden, 2002.
- [14] E. Schubert, J. Wolfe, and A. Tarnopolsky. Spectral centroid and timbre in complex, multiple instrumental textures. In *Proc. 8th Int. Conf. on Music Perception & Cognition (ICMPC)*, Evanston, Aug. 2004.
- [15] G. Torelli and G. Caironi. New polyphonic sound generator chip with integrated microprocessor-programmable adsr envelope shaper. *IEEE Trans. on Consumer Electronics*, CE-29(3):203–212, 1983.
- [16] E. v. Hornbostel and C. Sachs. The classification of musical instruments. *Galpin Society Journal*, 3(25):3–29, 1961.

# MUSIC ANALYSIS AS A SMALLEST GRAMMAR PROBLEM

Kirill Sidorov

Andrew Jones

David Marshall

Cardiff University, UK

{K.Sidorov, Andrew.C.Jones, Dave.Marshall}@cs.cardiff.ac.uk

## ABSTRACT

In this paper we present a novel approach to music analysis, in which a grammar is automatically generated explaining a musical work's structure. The proposed method is predicated on the hypothesis that the shortest possible grammar provides a model of the musical structure which is a good representation of the composer's intent. The effectiveness of our approach is demonstrated by comparison of the results with previously-published expert analysis; our automated approach produces results comparable to human annotation. We also illustrate the power of our approach by showing that it is able to locate errors in scores, such as introduced by OMR or human transcription. Further, our approach provides a novel mechanism for intuitive high-level editing and creative transformation of music. A wide range of other possible applications exists, including automatic summarization and simplification; estimation of musical complexity and similarity, and plagiarism detection.

## 1. INTRODUCTION

In his Norton Lectures [1], Bernstein argues that music can be analysed in linguistic terms, and even that there might be "a worldwide, inborn musical grammar". Less specifically, the prevalence of musical form analyses, both large-scale (*e.g.* sonata form) and at the level of individual phrases, demonstrates that patterns, motifs, *etc.*, are an important facet of a musical composition, and a grammar is certainly one way of capturing these artefacts.

In this paper we present a method for automatically deriving a compact grammar from a musical work and demonstrate its effectiveness as a tool for analysing musical structure. A key novelty of this method is that it operates *automatically*, yet generates insightful results. We concentrate in this paper on substantiating our claim that generating parsimonious grammars is a useful analysis tool, but also suggest a wide range of scenarios to which this approach could be applied.

Previous research into grammar-based approaches to modelling music has led to promising results. Treating harmonic phenomena as being induced by a generative grammar has been proposed in [9, 24, 27], and the explanatory

power of such grammars has been demonstrated. The use of musical grammar based of the Generative Theory of Tonal Music [19] has been proposed in [11–13], for the analysis of music. Similarly, there is a number of grammar-based approaches to automatic composition, including some which automatically learn stochastic grammars or derive grammars in an evolutionary manner, although some researchers continue to craft grammars for this purpose by hand [7].

However, in these works the derivation of grammar rules themselves is performed manually [27] or semi-automatically [11–13] from heuristic musicological considerations. In some cases generative grammars (including stochastic ones) are derived or learned automatically, but they describe general patterns in a corpus of music, *e.g.* for synthesis [16, 22], rather than being precise analyses of individual works. In a paper describing research carried out with a different, more precise aim of visualising semantic structure of an individual work, the authors remark that they resorted to manual retrieval of musical structure data from descriptive essays "since presently there is no existing algorithm to parse the high-level structural information automatically from MIDI files or raw sound data" [5].

In this paper we present a method which addresses the above concern expressed by Chan *et al.* in [5], but which at the same time takes a principled, information-theoretical approach. We argue that the best model explaining a given piece of music is the most compact one. This is known as Minimum Description Length principle [23] which, in turn, is a formal manifestation of the Occam's razor principle: the best explanation for data is the most compressive one. Hence, given a piece of music, we seek to find the shortest possible context free grammar that generates this piece (and only this piece). The validity of our compressive modelling approach in this particular domain is corroborated by evidence from earlier research in predictive modelling of music [6] and from perception psychology [14, 25]: humans appear to find strongly compressible music (which therefore has a compact grammar) appealing.

## 2. COMPUTING THE SMALLEST GRAMMAR

Given a piece of music, we treat it as a sequence(s) of symbols (see Section 2.1) and we seek to find the *shortest possible* context-free grammar that generates this (and only this) piece. Following [21], we define the size of a grammar  $G$  to be the total length of the right hand sides of all the production rules  $R_i$  plus one for each rule (length of a separator or cost of introducing a new rule):



© Kirill Sidorov, Andrew Jones, David Marshall.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Kirill Sidorov, Andrew Jones, David Marshall. "Music Analysis as a Smallest Grammar Problem", 15th International Society for Music Information Retrieval Conference, 2014.

$$|G| = \sum_i (|R_i| + 1). \quad (1)$$

Searching for such a grammar is known as the *smallest grammar problem*. It has recently received much attention due its importance in compression and analysis of DNA sequences, see *e.g.* [4]. For an overview of the smallest grammar problem the reader is referred to [3].

Computing the smallest grammar is provably NP-hard (see [18] Theorem 3.1), therefore in practice we are seeking an approximation to the smallest grammar.

Various heuristics have been proposed in order to tackle the smallest grammar problem in tractable time by greedy algorithms [4, 21]. A fast on-line (linear time) algorithm called SEQUITUR has been proposed in [20]. While the focus of [20] was fast grammar inference for large sequences, rather than strong compression, [20] contains an early mention that such techniques may be applied to parsing of music. (In Section 3 we compare grammars produced by SEQUITUR with our approach.)

In [4, 21], a class of algorithms involving iterative replacement of a repeated substring is considered (termed there iterative repeat replacement (IRR)). We employ a similar procedure here, summarised in Alg. 1. First, the grammar  $G$  is initialised with top level rule(s), whose right-hand sides initially are simply the input string(s). Then, in the original IRR scheme, a candidate substring  $c$  is selected according to some scoring function  $F$ . All non-overlapping occurrences of this substring in the grammar are replaced with a new symbol  $R_{n+1}$ , and a new rule is added to the grammar:  $R_{n+1} \rightarrow c$ . Replacement of a substring of length  $m$  in a string of length  $n$  can be done using the Knuth-Morris-Pratt algorithm [17] in  $O(m+n)$  time. The replacement procedure repeats until no further improvement is possible.

In [4, 21] the various heuristics according to which such candidate substitutions can be selected are examined; the conclusion is that the “locally most compressive” heuristic results in the shortest final grammars. Suppose a substring of length  $L$  occurring  $N$  times is considered for replacement with a new rule. The resulting saving is, therefore [21]:

$$F = \Delta|G| = (LN) - (L + 1 + N). \quad (2)$$

Hence, we use Eq. (2) (the locally most compressive heuristic) as our scoring function when selecting candidate substrings (in line 2 of Alg. 1). We found that the greedy iterative replacement scheme of [21] does not always produce optimal grammars. We note that a small decrease in grammar size may amount to a substantial change in the grammar’s *structure*, therefore we seek to improve the compression performance.

To do so, instead of greedily making a choice at each iteration, we recursively evaluate (line 9) multiple ( $w$ ) candidate substitutions (line 2) with backtracking, up to a certain depth  $d_{\max}$  (lines 9–15). Once the budgeted search depth has been exhausted, the remaining substitutions are done greedily (lines 4–7) as in [21]. This allows us to control the greediness of the algorithm from completely greedy ( $d_{\max} = 0$ ) to exhaustive search ( $d_{\max} = \infty$ ). We observed that using more than 2–3 levels of backtracking usually does not yield any further reduction in the size of the grammar.

---

**Algorithm 1** CompGram (Compress grammar)
 

---

**Require:** Grammar  $G$ ; search depth  $d$ .

(Tuning constants: max depth  $d_{\max}$ ; width  $w$ )

```

1: loop
2:   Find  $w$  best candidate substitutions  $C = \{c_i\}$  in  $G$ .
3:   if  $C = \emptyset$  then return  $G$ ; end if
4:   if recursion depth  $d > d_{\max}$  then
5:     Greedily choose best  $c_{\text{best}}$ 
6:      $G' := \text{replace}(G, c_{\text{best}}, \text{new symbol})$ 
7:     return CompGram( $G'$ ,  $d + 1$ )
8:   else
9:     Evaluate candidates:
10:    for  $c_i \in C$  do
11:       $G' := \text{replace}(G, c_i, \text{new symbol})$ 
12:       $G''_i := \text{CompGram}(G', d + 1)$ 
13:    end for
14:     $b := \arg \min_i |G''_i|$ 
15:    return  $G''_b$ 
16:   end if
17: end loop

```

---

Selecting a candidate according to Eq. (2) in line 2 involves maximising the number of non-overlapping occurrences of a substring, which is known as the *string statistics problem*, the solutions to which are not cheap [2]. Therefore, as in [4] we approximate the maximal number of non-overlapping occurrences with the number of *maximal repeats* [10]. All  $z$  maximal repeats in a string (or a set of strings) of total length  $n$  can be found very fast (in  $O(n + z)$  time) using suffix arrays [10]. In principle, it is possible to construct an example in which this number will be drastically different from the true number of non-overlapping occurrences (*e.g.* a long string consisting of a repeated symbol). However, this approximation was shown to work well in [4] and we have confirmed this in our experiments. Further, this concern is alleviated by the backtracking procedure we employ.

## 2.1 Representation of Music

In this paper, we focus on music that can be represented as several monophonic voices (such as voices in a fugue, or orchestral parts), that is, on the horizontal aspects of the music. We treat each voice, or orchestral part, as a string. We use (diatonic) intervals between adjacent notes, ignoring rests, as symbols in our strings. For ease of explanation of our algorithm we concentrate on the melodic information only, ignoring rhythm (note durations). Rhythmic invariance may be advantageous when melodic analysis is the prime concern. However, it is trivial to include note durations, and potentially even chord symbols and other musical elements, as symbols in additional (top level) strings.

Note that even though we take no special measures to model the relationship between the individual voices, this is happening *automatically*: indeed, all voices are encompassed in the same grammar and are considered for the iterative replacement procedure on equal rights as the grammar is updated.

Rule	$R_9$	$R_8$	$R_5$	$R_{18}$	$R_{10}$	$R_{11}$	$R_4$	$R_{19}$	$R_7$	$R_{20}$
Freq.	2	2	4	2	5	5	2	2	3	3
Len.	16	12	6	7	5	5	4	6	5	4
E. len.	99	94	24	47	11	11	37	34	17	16
<b>Comp.</b>	<b>96</b>	<b>91</b>	<b>67</b>	<b>44</b>	<b>38</b>	<b>38</b>	<b>34</b>	<b>31</b>	<b>30</b>	<b>28</b>

**Table 1.** Grammar statistics for Fugue №10. The ten *most compressing* rules are shown. For each rule  $R_i$ : *Freq.* is the number of times a rule occurs in the grammar, *Len.* is its right hand side length, *E. len.* is the length of the rule’s expansion, and *Comp.* is the total saving due to this rule.

### 3. RESULTS AND APPLICATIONS

#### 3.1 Automatic Structural Analysis

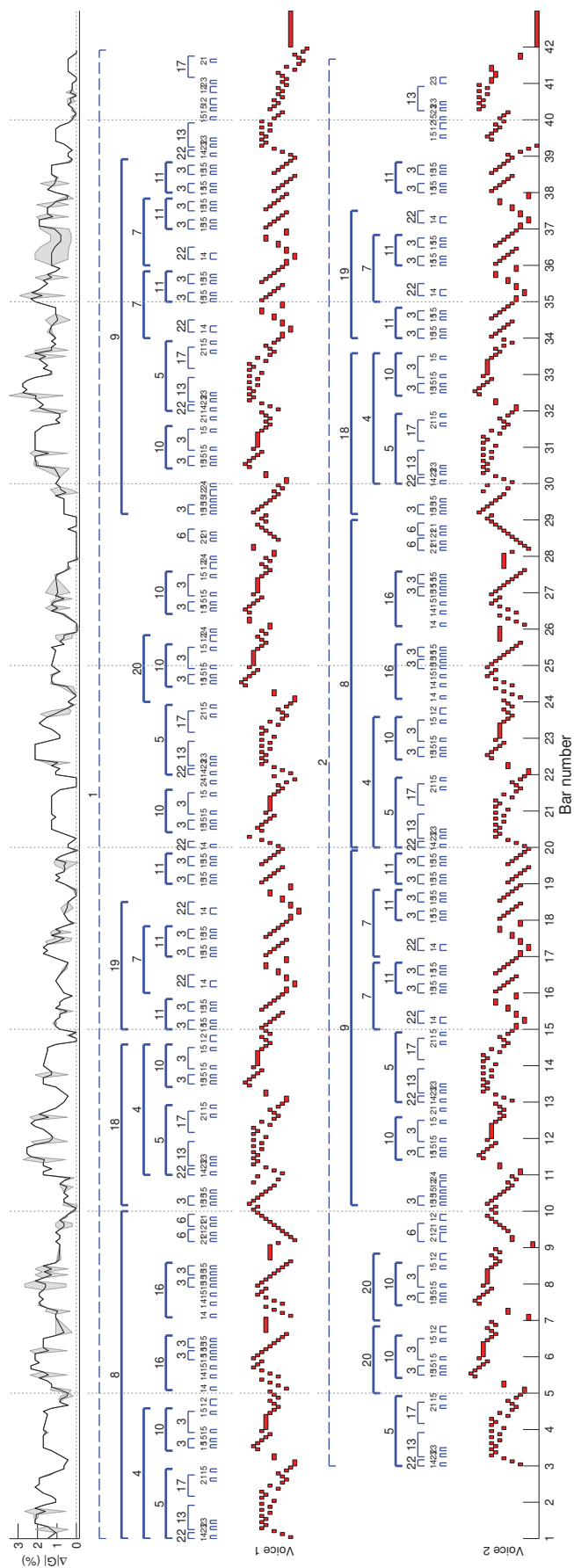
We have applied our method to automatically detect the structure of a selection of Bach’s fugues. (Eventually we intend to analyse all of them in this way.) Figure 1 shows one example of such analysis. We show the voices of the fugue in piano roll representation, with the hierarchy of the grammar on top: rules are represented by brackets labelled by rule number. For completeness, we give the entire grammar (of size  $|G| = 217$ ) obtained for Fugue №10 later, in Fig. 7. Figure 2 zooms in onto a fragment of the score with the rules overlaid. For comparison, we show manual analysis by a musicologist [26] in Fig. 3. Observe that all the main structural elements of the fugue have been correctly identified by our method (*e.g.* exp. and 1st dev. in rule  $R_8$ , re-exp. and 4th dev. in rule  $R_9$ , variant of re-exp. and 2nd dev in  $R_{18}$  and  $R_{19}$ ) and our automatic analysis is comparable to that by a human expert.

It is possible to use structures other than individual notes or intervals as symbols when constructing grammars. Figure 4 shows the simplified grammar for Fugue №10 generated using entire bars as symbols. In this experiment we first measured pairwise similarity between all bars (using Levenshtein distance [8]) and denoted each bar by a symbol, with identical or almost identical bars being denoted by the same symbol. The resulting grammar (Fig. 4) can be viewed as a coarse-grained analysis. Observe again that it closely matches human annotation (Fig. 3).

Our approach can also be used to detect prominent high-level features in music. We can compute the usage frequency for each rule and the corresponding savings in grammar size (as shown in Table 1 for Fugue №10). Most compressing rules, we argue, correspond to structurally important melodic elements. The present example illustrates our claim: rule  $R_8$  corresponds to the fugue’s exposition,  $R_9$  to re-exposition, and  $R_5$  to the characteristic chromatic figure in the opening (*cf.* Figs. 1 to 3 and the score).

In addition to high-level analysis, our approach can be used to detect the smallest constituent building blocks of a piece. For example, Fig. 5 shows the lowest level rules (that use only terminals) produced in analysis of Fugue №10, and the frequency of each rule. These are the elementary “bricks” from which Bach has constructed this fugue.

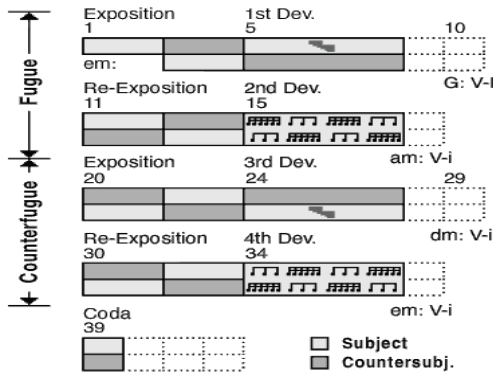
In [20], SEQUITUR is applied to two Bach chorales. In Fig. 6 we replicate the experiment from [20] and com-



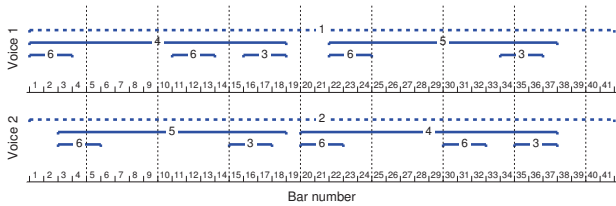
**Figure 1.** Automatic analysis of Bach’s Fugue №10 from WTK book I. *On top:* sensitivity to point errors as measured by the increase in grammar size  $\Delta|G|$ .



**Figure 2.** Close-up view of the first four bars: rules  $R_4$ ,  $R_5$ ,  $R_{10}$ , and  $R_{12}$  overlaid with the score (lower level rules are not shown).



**Figure 3.** Manual analysis of Fugue №10 by a musicologist (from [26] with permission).



**Figure 4.** Simplified automatic analysis of Fugue №10 using whole bars as symbols.

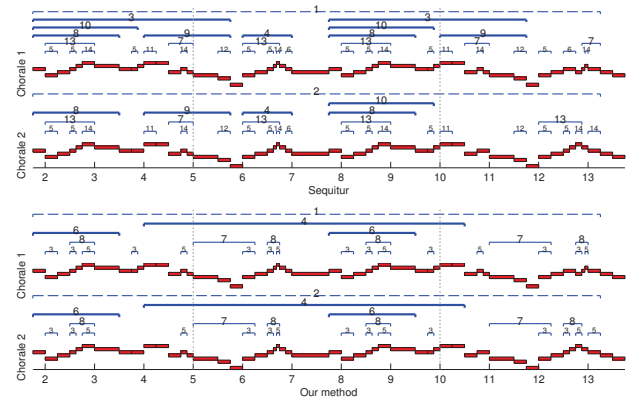
pare the grammars of these two chorales generated by SEQUITUR and by our approach. The chorales are very similar except for a few subtle differences. We note that our method was able to produce a shorter grammar ( $|G_{our}| = 50$  vs.  $|G_{Sequitur}| = 59$ ) and hence revealed more of the relevant structure, while the grammar of the more greedy (and hence less compressing) SEQUITUR was compromised by the small differences between the chorales.

### 3.2 Error Detection and Spell-checking

We investigated the sensitivity of the grammars generated by our method to alterations in the original music. In one experiment, we systematically altered each note in turn (introducing a point error) in the 1st voice of Fugue №10 and constructed a grammar for each altered score. The change in grammar size relative to that of the unaltered score is plotted in Fig. 1 (top) as a function of the alteration’s position. Observe that the grammar is more sensitive to alterations in structurally dense regions and less sensitive elsewhere, *e.g.* in between episodes. Remarkably, with



**Figure 5.** Atomic (lowest level) rules with their frequencies in brackets.



**Figure 6.** The grammars for the Bach chorales (from [20]) produced by SEQUITUR (above),  $|G_{Sequitur}| = 59$ , and by the proposed approach (below),  $|G_{our}| = 50$ .

very few exceptions (*e.g.* bars 10, 26) altering the piece consistently results in the grammar size increasing. We observed a similar effect in other Bach fugues and even in 19th century works (see below). We propose, only partially in jest, that this indicates that Bach’s fugues are close to structural perfection which is ruined by even the smallest alteration.

Having observed the sensitivity of the grammar size to point errors (at least in highly structured music), we propose that grammar-based modelling can be used for musical “spell-checking” to correct errors in typesetting (much like a word processor does for text), or in optical music recognition (OMR). This is analogous to compressive sensing which is often used in signal and image processing (see *e.g.* [15]) for denoising: noise compresses poorly. We can regard errors in music as noise and use the grammar-based model for locating such errors. We investigated this possibility with the following experiment.

As above, we introduce a point error (replacing one note) at a random location in the score to simulate a “typo” or an OMR error. We then systematically alter every note in the score and measure the resulting grammar size in each case. When the error is thus undone by one of the modifications, the corresponding grammar size should be noticeably smaller, and hence the location of the error may thus be revealed. We rank the candidate error positions by grammar size and consider suspected error locations with grammar size less than or equal to that of the ground truth error, *i.e.* the number of locations that would need to be manually examined to pin-point the error, as false positives. We then report the number of false positives as a fraction of the total number of notes in the piece. We repeat the experiment for multiple randomly chosen error



$S_1 \rightarrow R_8 \uparrow R_{18} R_{12} \uparrow R_{19} \downarrow R_{11} \uparrow R_{22} \downarrow R_{10} \downarrow R_{24} \downarrow R_5 R_{20}$   
 $R_{24} \downarrow R_{10} R_{12} R_{24} \downarrow R_6 \downarrow R_6 \downarrow \uparrow \downarrow \uparrow \downarrow \uparrow R_9 R_{22} R_{13} R_{15} \uparrow R_{15} R_{12} \downarrow$   
 $R_{12} R_{23} R_{17} \downarrow \uparrow \downarrow \uparrow$   
 $S_2 \rightarrow R_5 R_{20} \downarrow R_{20} \downarrow \uparrow R_6 R_{12} \downarrow \uparrow R_9 R_8 \uparrow R_{18} \downarrow \downarrow \uparrow \uparrow R_{19}$   
 $7 \downarrow \uparrow R_{11} \uparrow \downarrow \downarrow \downarrow \downarrow \uparrow \uparrow R_{15} \downarrow R_{12} R_{15} \uparrow R_{13} R_{23} \downarrow \downarrow \downarrow$   
 $R_3 \rightarrow R_{15} R_{15}$   $R_4 \rightarrow R_5 \uparrow \downarrow R_{10}$   
 $R_5 \rightarrow R_{22} R_{13} \uparrow \downarrow R_{17} R_{15}$   $R_6 \rightarrow R_{21} R_{21} \uparrow$   
 $R_7 \rightarrow \downarrow R_{22} \downarrow \uparrow R_{11}$   
 $R_8 \rightarrow R_4 R_{12} \uparrow \downarrow R_{16} \downarrow R_{16} \downarrow \downarrow R_6 R_6 \downarrow$   
 $R_9 \rightarrow R_3 R_{15} R_{12} R_{24} \uparrow R_{10} \downarrow R_{21} \downarrow R_5 R_7 \downarrow R_7 \uparrow R_{11} \uparrow$   
 $R_{10} \rightarrow \uparrow R_3 \uparrow R_3 \uparrow$   
 $R_{11} \rightarrow R_3 \downarrow \uparrow R_3 \downarrow$   $R_{12} \rightarrow \uparrow \downarrow \uparrow$   
 $R_{13} \rightarrow R_{23} R_{23} \downarrow \downarrow \downarrow \uparrow \downarrow \uparrow \downarrow$   $R_{14} \rightarrow \uparrow \downarrow \uparrow$   
 $R_{15} \rightarrow \downarrow \downarrow$   
 $R_{16} \rightarrow R_{14} \uparrow \downarrow R_{14} \uparrow R_{15} \uparrow R_3 R_3 \uparrow$   
 $R_{17} \rightarrow \uparrow \downarrow \downarrow \downarrow \uparrow \downarrow R_{21} \downarrow$   
 $R_{18} \rightarrow R_3 R_{15} \uparrow \downarrow \downarrow \uparrow \downarrow R_4$   
 $R_{19} \rightarrow R_{11} \downarrow R_7 \downarrow \downarrow R_{22}$   $R_{20} \rightarrow \uparrow \downarrow \uparrow R_{10} R_{12}$   
 $R_{21} \rightarrow \uparrow \uparrow$   $R_{22} \rightarrow R_{14} \uparrow$   
 $R_{23} \rightarrow \downarrow \uparrow$   $R_{24} \rightarrow \downarrow \downarrow \uparrow$

**Figure 7.** Automatically generated shortest grammar for Fugue №10. Here,  $R_i$  are production rules ( $S_i$  are the top level rules corresponding to entire voices), numbers with arrows are terminal symbols (diatonic intervals with the arrows indicating the direction).

Piece	F/P	Piece	F/P
Fugue №1 <sup>1</sup>	6.07%	Fugue №2	2.28%
Fugue №10	1.55%	Fugue №9	9.07%
Bvn. 5th str.	2.28%	Elgar Qrt.	14.22%
Blz. SF. str.	16.71%	Mndsn. Heb.	16.28%

<sup>1</sup>Fugues are from WTC book I.

**Table 2.** Spell-checking performance: fraction of false positives.



**Figure 8.** Selecting between two editions of Fugue №10 using grammar size.

locations and report median performance over 100 experiments in Table 2. We have performed this experiment on Bach fugues, romantic symphonic works (Beethoven’s 5th symphony 1st mvt., Berlioz’s “Symphonie Fantastique” 1st mvt., Mendelssohn’s “Hebrides”) and Elgar’s Quartet 3rd mvt. We observed impressive performance (Table 2) on Bach’s fugues (error location narrowed down to just a few percent of the score’s size), and even in supposedly less-structured symphonic works the algorithm was able to substantially narrow down the location of potential errors. This suggests that our approach can be used to effectively locate errors in music: for example a notation editor using our method may highlight potential error locations, thus warning the user, much like word processors do for text.

A variant of the above experiment is presented in Fig. 8. We want to select between two editions of Fugue №10 in which bar 33 differs. We measured the total grammar size for the two editions and concluded that the variant in

**Figure 9.** High-level editing. Above: automatic analysis of Fugue №16 (fragment); middle: original; below: rules  $R_6$  and  $R_8$  were edited with our method to obtain a new fugue.

Edition B is more logical as it results in smaller grammar size  $|G| = 208$  (vs.  $|G| = 217$  for Edition A).

### 3.3 High-level Editing

A grammar automatically constructed for a piece of music can be used as a means for high-level editing. For example, one may edit the right-hand sides of individual rules to produce a new similarly-structured piece, or, by operating on the grammar tree, alter the structure of a whole piece. We illustrate such editing in Fig. 9. We have automatically analysed Fugue №16 with our method and then edited two of the detected rules ( $R_6$  and  $R_8$ ) to obtain a new fugue (expanding the grammar back). This new fugue is partially based on new material, yet maintains the structural perfection of the original fugue. We believe this may be a useful and intuitive next-generation method for creatively transforming scores.

### 3.4 Further Applications

We speculate that in addition to the applications discussed above, the power of our model may be used in other ways: for estimation of complexity and information content in a musical piece; as means for automatic summarisation by analysing the most compressive rules; for improved detection of similarity and plagiarism (including structural similarity); for automatic simplification of music (by transforming a piece so as to decrease its grammar size); and for classification of music according to its structural properties.

Having observed that size of grammar is a good measure of the “amount of structure” in a piece, we suggest that our model can even be used to tell good music from bad music.

Hypothetically, a poor composition would remain poor even when (random) alterations are made to it and hence its grammar size would be insensitive to such alterations, while well-constructed works (like those of Bach in our examples) would suffer, in terms of grammar size, from perturbation.

#### 4. CONCLUSIONS AND FUTURE WORK

We have posed the analysis of music as a smallest grammar problem and have demonstrated that building parsimonious context-free grammars is an appealing tool for analysis of music, as grammars give insights into the underlying structure of a piece. We have discussed how such grammars may be efficiently constructed and have illustrated the power of our model with a number of applications: automatic structural analysis, error detection and spell-checking (without prior models), high-level editing.

Future work would include augmenting the presented automatic grammatical analysis to allow inexact repetitions (variations or transformations of material) to be recognised in the grammar, and, in general, increasing the modelling power by recognising more disguised similarities in music.

#### 5. REFERENCES

- [1] L. Bernstein. Norton lectures. <http://www.leonardbernstein.com/norton.htm>.
- [2] G. Brodal, R. B. Lyngsø, Anna Östlin, and Christian N. S. Pedersen. Solving the string statistics problem in time  $O(n \log n)$ . In *Proc. ICALP '02*, pages 728–739, 2002.
- [3] R. Carrascosa, F. Coste, M. Gallé, and G. G. I. López. The smallest grammar problem as constituents choice and minimal grammar parsing. *Algorithms*, 4(4):262–284, 2011.
- [4] R. Carrascosa, F. Coste, M. Gallé, and G. G. I. López. Searching for smallest grammars on large sequences and application to DNA. *J. Disc. Alg.*, 11:62–72, 2012.
- [5] W.-Y. Chan, H. Qu, and W.-H. Mak. Visualizing the semantic structure in classical music works. *IEEE Trans. Vis. and Comp. Graph.*, 16(1):161–173, 2010.
- [6] D. Conklin and I. H. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24:51–73, 1995.
- [7] J. D. Fernandez and F. J. Vico. AI methods in algorithmic composition: A comprehensive survey. *J. Artif. Intell. Res. (JAIR)*, 48:513–582, 2013.
- [8] M. Grachten, J. L. Arcos, and R. L. de Mántaras. Melodic similarity: Looking for a good abstraction level. In *Proc. ISMIR*, 2004.
- [9] M. Granroth-Wilding and M. Steedman. Statistical parsing for harmonic analysis of jazz chord sequences. In *Proc. ICMC*, pages 478–485, 2012.
- [10] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, 1997.
- [11] M. Hamanaka, K. Hirata, and S. Tojo. Implementing a generative theory of tonal music. *Journal of New Music Research*, 35(4):249–277, 2006.
- [12] M. Hamanaka, K. Hirata, and S. Tojo. FATTA: Full automatic time-span tree analyzer. In *Proc. ICMC*, pages 153–156, 2007.
- [13] M. Hamanaka, K. Hirata, and S. Tojo. Grouping structure generator based on music theory GTTM. *J. of Inf. Proc. Soc. of Japan*, 48(1):284–299, 2007.
- [14] N. Hudson. Musical beauty and information compression: Complex to the ear but simple to the mind? *BMC Research Notes*, 4(1):9+, 2011.
- [15] J. Jin, B. Yang, K. Liang, and X. Wang. General image denoising framework based on compressive sensing theory. *Computers & Graphics*, 38(0):382–391, 2014.
- [16] R. M. Keller and D. R. Morrison. A grammatical approach to automatic improvisation. In *Proc. SMC '07*, pages 330–337, 2007.
- [17] D. E. Knuth, J. H. Morris, and V. R. Pratt. Fast pattern matching in strings. *SIAM Journal of Computing*, 6(2):323–350, 1977.
- [18] E. Lehman and A. Shelat. Approximation algorithms for grammar-based compression. In *Proc. ACM-SIAM SODA '02*, pages 205–212, 2002.
- [19] F. Lerdahl and R. Jackendoff. *A generative theory of tonal music*. The MIT Press, 1983.
- [20] C. G. Nevill-Manning and I. H. Witten. Identifying hierarchical structure in sequences: A linear-time algorithm. *J. Artif. Intell. Res. (JAIR)*, 7:64–82, 1997.
- [21] C.G. Nevill-Manning and I.H. Witten. Online and offline heuristics for inferring hierarchies of repetitions in sequences. In *Proc. IEEE*, volume 88, pages 1745–1755, 2000.
- [22] D. Quick and P. Hudak. Grammar-based automated music composition in Haskell. In *Proc. ACM SIGPLAN FARM'13*, pages 59–70, 2013.
- [23] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.
- [24] M. Rohrmeier. Towards a generative syntax of tonal harmony. *J. of Math. and Music*, 5(1):35–53, 2011.
- [25] J. Schmidhuber. Low-complexity art. *Leonardo*, 30(2):97–103, 1997.
- [26] Tim Smith. Fugues of the Well-Tempered Clavier. <http://bach.nau.edu/clavier/nature/fugues/Fugue10.html>, 2013.
- [27] M. J. Steedman. A generative grammar for jazz chord sequences. *Music Perception: An Interdisciplinary Journal*, 2(1):52–77, 1984.

# FRAME-LEVEL AUDIO SEGMENTATION FOR ABRIDGED MUSICAL WORKS

Thomas Prätzlich, Meinard Müller  
International Audio Laboratories Erlangen

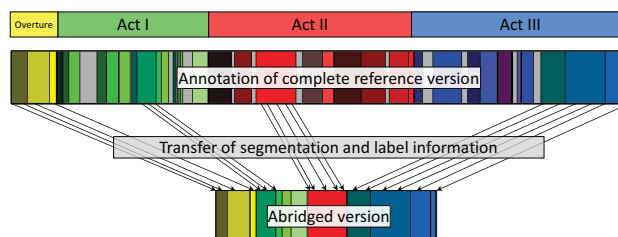
{thomas.praetzelich, meinard.mueller}@audiolabs-erlangen.de

## ABSTRACT

Large-scale musical works such as operas may last several hours and typically involve a huge number of musicians. For such compositions, one often finds different arrangements and abridged versions (often lasting less than an hour), which can also be performed by smaller ensembles. Abridged versions still convey the flavor of the musical work containing the most important excerpts and melodies. In this paper, we consider the task of automatically segmenting an audio recording of a given version into semantically meaningful parts. Following previous work, the general strategy is to transfer a reference segmentation of the original complete work to the given version. Our main contribution is to show how this can be accomplished when dealing with strongly abridged versions. To this end, opposed to previously suggested segment-level matching procedures, we adapt a frame-level matching approach for transferring the reference segment information to the unknown version. Considering the opera “Der Freischütz” as an example scenario, we discuss how to balance out flexibility and robustness properties of our proposed frame-level segmentation procedure.

## 1. INTRODUCTION

Over the years, many musical works have seen a great number of reproductions, ranging from reprints of the sheet music to various audio recordings of performances. For many works this has led to a wealth of co-existing versions including arrangements, adaptations, cover versions, and so on. Establishing semantic correspondences between different versions and representations is an important step for many applications in Music Information Retrieval. For example, when comparing a musical score with an audio version, the goal is to compute an alignment between measures or notes in the score and points in time in the audio version. This task is motivated by applications such as score following [1], where the score can be used to navigate through a corresponding audio version and vice versa. The aligned score information can also be used to parameterize an audio processing algorithm such as in score-



**Figure 1.** Illustration of the proposed method. Given the annotated segments on a complete reference version of a musical work, the task is to transfer the segment information to an abridged version.

informed source separation [4, 12]. When working with two audio versions, alignments are useful for comparing different performances of the same piece of music [2, 3]. In cover song identification, alignments can be used to compute the similarity between two recordings [11]. Alignment techniques can also help to transfer meta data and segmentation information between recordings. In [7], an unknown recording is queried against a database of music recordings to identify a corresponding version of the same musical work. After a successful identification, alignment techniques are used to transfer the segmentation given in the database to the unknown recording.

A similar problem was addressed in previous work, where the goal was to transfer a labeled segmentation of a reference version onto an unknown version of the same musical work [10]. The task was approached by a segment-level matching procedure, where one main assumption was that a given reference segment either appears more or less in the same form in the unknown version or is omitted completely.

In abridged versions of an opera, however, this assumption is often not valid. Such versions strongly deviate from the original by omitting a large portion of the musical material. For example, given a segment in a reference version, one may no longer find the start or ending sections of this segment in an unknown version, but only an intermediate section. Hence, alignment techniques that account for structural differences are needed. In [5], a music synchronization procedure accounting for structural differences in recordings of the same piece of music is realized with an adaptation of the Needleman-Wunsch algorithm. The algorithm penalizes the skipping of frames in the alignment by adding an additional cost value for each skipped frame.



© Thomas Prätzlich, Meinard Müller.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Thomas Prätzlich, Meinard Müller. “Frame-Level Audio Segmentation for Abridged Musical Works”, 15th International Society for Music Information Retrieval Conference, 2014.

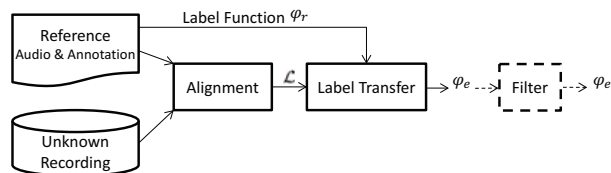
Thus, the cost for skipping a sequence of frames is dependent on the length of the sequence. In abridged versions, however, omission may occur on an arbitrary scale, ranging from several musical measures up to entire scenes of an opera. In such a scenario, a skipping of long sequences should not be more penalized as a skipping of short sequences. In this work, we will therefore use a different alignment strategy.

In this paper, we address the problem of transferring a labeled reference segmentation onto an unknown version in the case of abridged versions, see Figure 1. As our main contribution, we show how to approach this task with a frame-level matching procedure, where correspondences between frames of a reference version and frames of an unknown version are established. The labeled segment information of the reference version is then transferred to the unknown version only for frames for which a correspondence has been established. Such a frame-level procedure is more flexible than a segment-level procedure. However, on the downside, it is less robust. As a further contribution, we show how to stabilize the robustness of the frame-level matching approach while preserving most of its flexibility.

The remainder of this paper is structured as follows: In Section 2, we discuss the relevance of abridged music recordings and explain why they are problematic in a standard music alignment scenario. In Section 3, we review the segment-level matching approach from previous work (Section 3.2), and then introduce the proposed frame-level segmentation pipeline (Section 3.3). Subsequently, we present some results of a qualitative (Section 4.2) and a quantitative (Section 4.3) evaluation and conclude the paper with a short summary (Section 5).

## 2. MOTIVATION

For many musical works, there exists a large number of different versions such as cover songs or different performances in classical music. These versions can vary greatly in different aspects such as the instrumentation or the structure. Large-scale musical works such as operas usually need a huge number of musicians to be performed. For these works, one often finds arrangements for smaller ensembles or piano reductions. Furthermore, performances of these works are usually very long. Weber’s opera “Der Freischütz”, for example, has an average duration of about two hours. Taking it to an extreme, Wagner’s epos “Der Ring der Nibelungen”, consists of four operas having an overall duration of about 15 hours. For such large-scale musical works, one often finds abridged versions. These versions usually present the most important material of a musical work in a strongly shortened and structurally modified form. Typically, these structural modifications include omissions of repetitions and other “non-essential” musical passages. Abridged versions were very common in the early recording days due to space constraints of the sound carriers. The opera “Der Freischütz” would have filled 18 discs on a shellac record. More recently, abridged versions or excerpts of a musical work can often be found as bonus tracks on CD records. In a standard alignment



**Figure 2.** Illustration of the proposed frame-level segmentation pipeline. A reference recording with a reference label function  $\varphi_r$  is aligned with an unknown version. The alignment  $\mathcal{L}$  is used to transfer  $\varphi_r$  to the unknown version yielding  $\varphi_e$ .

scenario, abridged versions are particularly problematic as they omit material on different scales, ranging from the omission of several musical measures up to entire parts.

## 3. METHODS

In this section, we show how one can accomplish the task of transferring a given segmentation of a reference version, say  $X$ , onto an unknown version, say  $Y$ . The general idea is to use alignment techniques to find corresponding parts between  $X$  and  $Y$ , and then to transfer on those parts the given segmentation from  $X$  to  $Y$ .

After introducing some basic notations on alignments and segmentations (Section 3.1), we review the segment-level matching approach from our previous work (Section 3.2). Subsequently, we introduce our frame-level segmentation approach based on partial matching (Section 3.3).

### 3.1 Basic Notations

#### 3.1.1 Alignments, Paths, and Matches

Let  $[1 : N] := \{1, 2, \dots, N\}$  be an index set representing the *time line* of a discrete signal or feature sequence  $X = (x_1, x_2, \dots, x_N)$ . Similarly, let  $[1 : M]$  be the time line of a second sequence  $Y = (y_1, \dots, y_M)$ . An *alignment* between two time lines  $[1 : N]$  and  $[1 : M]$  is modeled as a set  $\mathcal{L} = (p_1, \dots, p_L) \subseteq [1 : N] \times [1 : M]$ . An element  $p_\ell = (n_\ell, m_\ell) \in \mathcal{L}$  is called a *cell* and encodes a correspondence between index  $n_\ell \in [1 : N]$  of the first time line and index  $m_\ell \in [1 : M]$  of the second one. In the following, we assume  $\mathcal{L}$  to be in lexicographic order.  $\mathcal{L}$  is called a *match* if  $(p_{\ell+1} - p_\ell) \in \mathbb{N} \times \mathbb{N}$  for  $\ell \in [1 : L - 1]$ . Note that this condition implies strict monotonicity and excludes the possibility to align an index of the first time line with many indices of the other and vice versa. An alignment can also be constrained by requiring  $(p_{\ell+1} - p_\ell) \in \Sigma$  for a given set  $\Sigma$  of admissible step sizes. A typical choice for this set is  $\Sigma = \{(1, 1), (1, 0), (0, 1)\}$ , which allows to align an index of one time line to many indices of another, and vice versa. Sometimes other sets such as  $\Sigma = \{(1, 1), (1, 2), (2, 1)\}$  are used to align sequences which are assumed to be structurally and temporally mostly consistent. If  $\mathcal{L}$  fulfills a given step size condition,  $\mathcal{P} = \mathcal{L}$  is called a *path*. Note that alignments that fulfill  $\Sigma_1$  and  $\Sigma_2$  are both paths, but only an alignment fulfilling  $\Sigma_2$  is also a match.

### 3.1.2 Segments and Segmentation

We formally define a *segment* to be a set  $\alpha = [s : t] \subseteq [1 : N]$  specified by its start index  $s$  and its end index  $t$ . Let  $|\alpha| := t - s + 1$  denote the length of  $\alpha$ . We define a (partial) *segmentation* of size  $K$  to be a set  $\mathcal{A} := \{\alpha_1, \alpha_2, \dots, \alpha_K\}$  of pairwise disjoint segments:  $\alpha_k \cap \alpha_j = \emptyset$  for  $k, j \in [1 : K], k \neq j$ .

### 3.1.3 Labeling

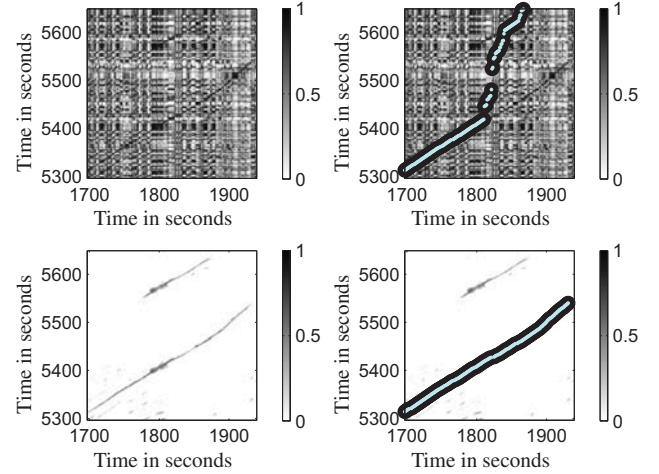
Let  $[0 : K]$  be a set of labels. The label 0 plays a special role and is used to label everything that has not been labeled otherwise. A *label function*  $\varphi$  maps each index  $n \in [1 : N]$  to a label  $k \in [0 : K]$ :

$$\varphi : [1 : N] \rightarrow [0 : K].$$

The pair  $([1 : N], \varphi)$  is called a *labeled time line*. Let  $n \in [1 : N]$  be an index,  $\alpha = [s : t]$  be a segment, and  $k \in [0 : K]$  be a label. Then the pair  $(n, k)$  is called a *labeled index* and the pair  $(\alpha, k)$  a *labeled segment*. A labeled segment  $(\alpha, k)$  induces a labeling of all indices  $n \in \alpha$ . Let  $\mathcal{A} := \{\alpha_1, \alpha_2, \dots, \alpha_K\}$  be a segmentation of  $[1 : N]$  and  $[0 : K]$  be the label set. Then the set  $\{(\alpha_k, k) \mid k \in [1 : K]\}$  is called a *labeled segmentation* of  $[1 : N]$ . From a labeled segmentation one obtains a label function on  $[1 : N]$  by setting  $\varphi(n) := k$  for  $n \in \alpha_k$  and  $\varphi(n) := 0$  for  $n \in [1 : N] \setminus \bigcup_{k \in [1 : K]} \alpha_k$ . Vice versa, given a label function  $\varphi$ , one obtains a labeled segmentation in the following way. We call consecutive indices with the same label a *run*. A segmentation of  $[1 : N]$  is then derived by considering runs of maximal length. We call this segmentation the *segmentation induced by  $\varphi$* .

### 3.2 Segment-Level Matching Approach

The general approach in [10] is to apply segment-level matching techniques based on dynamic time warping (DTW) to transfer a labeled reference segmentation to an unknown version. Given a labeled segmentation  $\mathcal{A}$  of  $X$ , each  $\alpha_k \in \mathcal{A}$  is used as query to compute a ranked list of matching candidates in  $Y$ . The matching candidates are derived by applying a subsequence variant of the DTW algorithm using the step size conditions  $\Sigma = \{(1, 1), (1, 2), (2, 1)\}$ , see [8, Chapter 5]. The result of the subsequence DTW procedure is a matching score and an alignment path  $\mathcal{P} = (p_1, \dots, p_L)$  with  $p_\ell = (n_\ell, m_\ell)$ .  $\mathcal{P}$  encodes an alignment of the segment  $\alpha_k := [n_1 : n_L] \subseteq [1 : N]$  and the corresponding segment  $[m_1 : m_L] \subseteq [1 : M]$  in  $Y$ . To derive a final segmentation, one segment from each matching candidate list is chosen such that the sum of the alignment scores of all chosen segments is maximized by simultaneously fulfilling the following constraints. First, the chosen segments have to respect the temporal order of the reference segmentation and second, no overlapping segments are allowed in the final segmentation. Furthermore, the procedure is adapted to be robust to tuning differences of individual segments, see [10] for further details.



**Figure 3.** Excerpt of similarity matrices of the reference Kl1e1973 and Kna1939 before (top) and after enhancement (bottom), shown without match (left) and with match (right).

### 3.3 Frame-Level Segmentation Approach

The basic procedure of our proposed frame-level segmentation is sketched in Figure 2. First, we use a *partial matching* algorithm (Section 3.3.1) to compute an alignment  $\mathcal{L}$ . Using  $\mathcal{L}$  and the reference label function  $\varphi_r$  obtained from the reference annotation  $\mathcal{A}$  of  $X$ , an *induced label function*  $\varphi_e$  to estimate the labels on  $Y$  is derived (Section 3.3.2). Finally, we apply a mode filter (Section 3.3.3) and a filling up strategy (Section 3.3.4) to derive the final segmentation result.

#### 3.3.1 Partial Matching

Now we describe a procedure for computing a partial matching between two sequences as introduced in [8]. To compare the two feature sequences  $X$  and  $Y$ , we compute a similarity matrix  $S(n, m) := s(x_n, y_m)$ , where  $s$  is a suitable similarity measure. The goal of the partial matching procedure is to find a score-maximizing match through the matrix  $S$ . To this end, we define the *accumulated score matrix*  $D$  by  $D(n, m) := \max\{D(n-1, m), D(n, m-1), D(n-1, m-1) + S(n, m)\}$  with  $D(0, 0) := D(n, 0) := D(0, m) := 0$  for  $1 \leq n \leq N$  and  $1 \leq m \leq M$ . The score maximizing match can then be derived by backtracking through  $D$ , see [8, Chapter 5]. Note that only diagonal steps contribute to the accumulated score in  $D$ . The partial matching algorithm is more flexible in aligning two sequences than the subsequence DTW approach, as it allows for skipping frames at any point in the alignment. However, this increased flexibility comes at the cost of losing robustness. To improve the robustness, we apply path-enhancement (smoothing) on  $S$ , and suppress other noise-like structures by thresholding techniques [9, 11]. In this way, the algorithm is less likely to align small scattered fragments. Figure 3 shows an excerpt of a similarity matrix before and after path-enhancement together with the computed matches.

### 3.3.2 Induced Label Function

Given a labeled time line ( $[1 : N], \varphi_r$ ) and an alignment  $\mathcal{L}$ , we derive a label function  $\varphi_e$  on  $[1 : M]$  by setting:

$$\varphi_e(m) := \begin{cases} \varphi_r(n) & \text{if } (n, m) \in \mathcal{L} \\ 0 & \text{else,} \end{cases}$$

for  $m \in [1 : M]$ . See Figure 4 for an illustration.

### 3.3.3 Local Mode Filtering

The framewise transfer of the labels may lead to very short and scattered runs. Therefore, to obtain longer runs and a more homogeneous labeling, especially at segment boundaries, we introduce a kind of smoothing step by applying a mode filter. The *mode* of a sequence  $\mathcal{S} = (s_1, s_2, \dots, s_N)$  is the most frequently appearing value and is formally defined by  $\text{mode}(\mathcal{S}) := \arg \max_{s \in \mathcal{S}} |\{n \in [1 : N] : s_n = s\}|$ . A *local mode filter* of length  $L = 2q + 1$  with  $q \in \mathbb{N}$  replaces each element  $s_n \in \mathcal{S}$ ,  $n \in [1 : N]$ , in a sequence by the mode of its neighborhood  $(s_{n-q}, \dots, s_{n+q})$ :

$$\text{modefilt}_q(\mathcal{S})(n) := \text{mode}(s_{n-q}, \dots, s_{n+q}).$$

Note that the mode may not be unique. In this case, we apply the following strategy in the mode filter. If the element  $s_n$  is one of the modes,  $s_n$  is left unmodified by the filter. Otherwise, one of the modes is chosen arbitrarily.

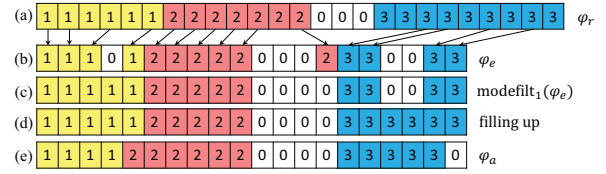
In our scenario, we apply the local mode filter on a labeled time line ( $[1 : N], \varphi_e$ ) by inputting the sequence  $\varphi_e([1 : N]) := (\varphi_e(1), \varphi_e(2), \dots, \varphi_e(N))$  into the filter, see Figure 4 for an illustration. The reason to use the mode opposed to the median to filter segment labels, is that labels are nominal data and therefore have no ordering (integer labels were only chosen for the sake of simplicity).

### 3.3.4 From Frames to Segments (Filling Up)

In the last step, we derive a segmentation from the label function  $\varphi_e$ . As indicated in Section 3.1.3, we could simply detect maximal runs and consider them as segments. However, even after applying the mode filter, there may still be runs sharing the same label that are interrupted by non-labeled parts (labeled zero). In our scenario, we assume that all segments have a distinct label and occur in the same succession as in the reference. Therefore, in the case of a sequence of equally labeled runs that are interrupted by non-labeled parts, we can assume that the runs belong to the same segment. Formally, we assign an index in between two indices with the same label (excluding the zero label) to belong to the same segment as these indices. To construct the final segments, we iterate over each  $k \in [1 : K]$  and construct the segments  $\alpha_k = [s_k : e_k]$ , such that  $s_k = \min\{m \in [1 : M] : \varphi_e(m) = k\}$ , and  $e_k = \max\{m \in [1 : M] : \varphi_e(m) = k\}$ , see Figure 4 for an example.

## 4. EVALUATION

In this section, we compare the previous segment-level matching procedure with our novel frame-level segmenta-



**Figure 4.** Example of frame-level segmentation. The arrows indicate the match between the reference version and the unknown version. **(a):** Reference label function. **(b):** Induced label function. **(c):** Mode filtered version of (b) with length  $L = 3$ . **(d):** Filling up on (c). **(e):** Ground truth label function.

tion approach based on experiments using abridged versions of the opera “Der Freischütz”. First we give an overview of our test set and the evaluation metric (Section 4.1). Subsequently, we discuss the results of the segment-level approach and the frame-level procedure on the abridged versions (Section 4.2). Finally, we present an experiment where we systematically derive synthetic abridged versions from a complete version of the opera (Section 4.3).

### 4.1 Tests Set and Evaluation Measure

In the following experiments, we use the recording of Carlos Kleiber performed in 1973 with a duration of 7763 seconds as reference version. The labeled reference segmentation consists of 38 musical segments, see Figure 5. Furthermore, we consider five abridged versions that were recorded between 1933 and 1994. The segments of the opera that are performed in these versions are indicated by Figure 5. Note that the gray parts in the figure correspond to dialogue sections in the opera. In the following experiments, the dialogue sections are considered in the same way as non-labeled (non-musical) parts such as applause, noise or silence. In the partial matching algorithm, they are excluded from the reference version (by setting the similarity score in these regions to minus infinity), and in the segment-level matching procedure, the dialogue parts are not used as queries.

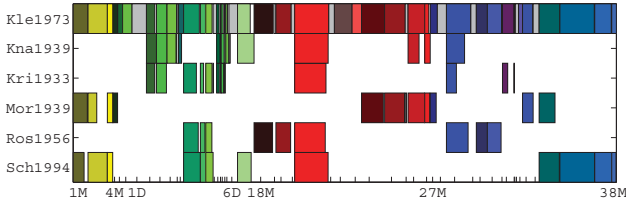
Throughout all experiments, we use CENS features which are a variant of chroma features. They are computed with a feature rate of 1 Hz (derived from 10 Hz pitch features with a smoothing length of 41 frames and a down-sampling factor of 10), see [8]. Each feature vector covers roughly 4.1 seconds of the original audio.

In our subsequent experiments, the following segment-level matching (M4) and frame-level segmentation (F1–F4) approaches are evaluated:

**(M4)** – Previously introduced segment-level matching, see Section 3.2 and [10] for details.

**(F1)** – Frame-level segmentation using a similarity matrix computed with the cosine similarity  $s$  defined by  $s(x, y) = \langle x, y \rangle$  for features  $x$  and  $y$ , see Section 3.3.

**(F2)** – Frame-level segmentation using a similarity matrix with enhanced path structures using the SM Toolbox [9]. For the computation of the similarity matrix, we used forward/backward smoothing with a smoothing length of 20



**Figure 5.** Visualization of relative lengths of the abridged versions compared to the reference version K1e1973. The gray segments indicate dialogues whereas the colored segments are musical parts.

frames (corresponding to 20 seconds) with relative tempi between  $0.5 - 2$ , sampled in 15 steps. Afterwards, a thresholding technique that retained only 5% of the highest values in the similarity matrix and a scaling of the remaining values to  $[0, 1]$  is applied. For details, we refer to [9] and Section 3.3.

**(F3)** – The same as in F2 with a subsequent mode filtering using a filter length  $L = 21$  frames, see Section 3.3.3 for details.

**(F4)** – The segmentation derived from F3 as described in Section 3.3.4.

#### 4.1.1 Frame Accuracy

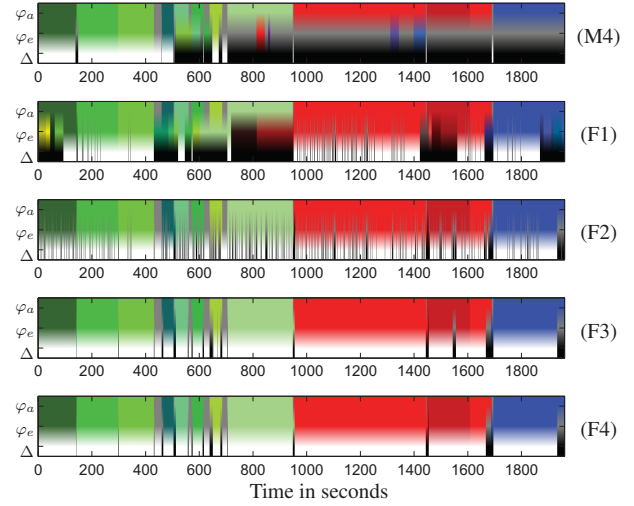
To evaluate the performance of the different segmentation approaches, we calculate the *frame accuracy*, which is defined as the ratio of correctly labeled frames and the total number of frames in a version. Given a ground truth label function  $\varphi_a$  and an induced label function  $\varphi_e$ , the frame accuracy  $A_f$  is computed as following:

$$A_f := \frac{\sum_{k \in [0:K]} |\varphi_a^{-1}(k) \cap \varphi_e^{-1}(k)|}{\sum_{k \in [0:K]} |\varphi_a^{-1}(k)|}$$

We visualize the accuracy by means of an *agreement sequence*  $\Delta(\varphi_a, \varphi_e)$  which we define as  $\Delta(\varphi_a, \varphi_e)(m) := 1$  (white) if  $\varphi_a(m) = \varphi_e(m)$  and  $\Delta(\varphi_a, \varphi_e)(m) := 0$  (black) otherwise. The sequences  $\Delta(\varphi_a, \varphi_e)$  visually correlates well with the values of the frame accuracy  $A_f$ , see Table 1 and the Figure 6. Note that in structural segmentation tasks, it is common to use different metrics such as the pairwise precision, recall, and f-measure [6]. These metrics disregard the absolute labeling of a frame sequence by relating equally labeled pairs of frames in an estimate to equally labeled frames in a ground truth sequence. However, in our scenario, we want to consider frames that are differently labeled in the ground truth and the induced label function as wrong. As the pairwise f-measure showed the same tendencies as the frame accuracy (which can be easily visualized), we decided to only present the frame accuracy values.

## 4.2 Qualitative Evaluation

In this section, we qualitatively discuss the results of our approach in more detail by considering the evaluation of the version Kna1939. For each of the five approaches, the results are visualized in a separate row of Figure 6,



**Figure 6.** Segmentation results on Kna1939 showing the ground truth label function  $\varphi_a$ , the induced label function  $\varphi_e$ , and the agreement sequence  $\Delta := \Delta(\varphi_a, \varphi_e)$ . White encodes an agreement and black a disagreement between  $\varphi_a$  and  $\varphi_e$ . **(M4),(F1),(F2),(F3),(F4)**: See Section 4.1.

showing the ground truth  $\varphi_a$ , the induced label function  $\varphi_e$  and the agreement sequence  $\Delta(\varphi_a, \varphi_e)$ .

For Kna1939, the segment-level matching approach M4 does not work well. Only 28% of the frames are labeled correctly. The red segment, for example, at around 1500 seconds is not matched despite the fact that it has roughly the same overall duration as the corresponding segment in the reference version, see Figure 5. Under closer inspection, it becomes clear that it is performed slower than the corresponding segment in the reference version, and that some material was omitted at the start, in the middle and the end of the segment. The frame-level matching approach F1 leads to an improvement, having a frame accuracy of  $A_f = 0.520$ . However, there are still many frames wrongly matched. For example, the overture of the opera is missing in Kna1939, but frames from the overture (yellow) of the reference are matched into a segment from the first act (green), see Figure 6. Considering that the opera consists of many scenes with harmonically related material and that the partial matching allows for skipping frames at any point in the alignment, it sometimes occurs that not the semantically corresponding frames are aligned, but harmonically similar ones. This problem is better addressed in approach F2, leading to an improved frame accuracy of 0.788. The enhancement of path structures in the similarity matrix in this approach leads to an increased robustness of the partial matching. Now, all high similarity values are better concentrated in path structures of the similarity matrix.

As a result, the algorithm is more likely to follow sequences of harmonically similar frames, see also Figure 3. However, to follow paths that are not perfectly diagonal, the partial matching algorithm needs to skip frames in the alignment, which leads to a more scattered label function. This is approached by F3 which applies a mode filter on the label function from F2, resulting in an improved frame

	<i>dur.</i> (s)	<b>M4</b>	<b>F1</b>	<b>F2</b>	<b>F3</b>	<b>F4</b>
Kna1939	1965	0.283	0.520	0.788	0.927	0.934
Kri1933	1417	0.390	0.753	0.777	0.846	0.870
Mor1939	1991	0.512	0.521	0.748	0.841	0.919
Ros1956	2012	0.887	0.749	0.817	0.850	0.908
Sch1994	2789	0.742	0.895	0.936	0.986	0.989
mean	2035	0.563	0.687	0.813	0.890	0.924

**Table 1.** Frame accuracy values on abridged versions. M4: Segment-level matching, F1: Frame-level segmentation, F2: Frame-level segmentation with path-enhanced similarity matrix, F3: Mode filtering with  $L = 21$  seconds on F2. F4: Derived Segmentation on F4.

accuracy of 0.927. In F4, the remaining gaps in the label function of F3 are filled up, which leads to a frame accuracy of 0.934.

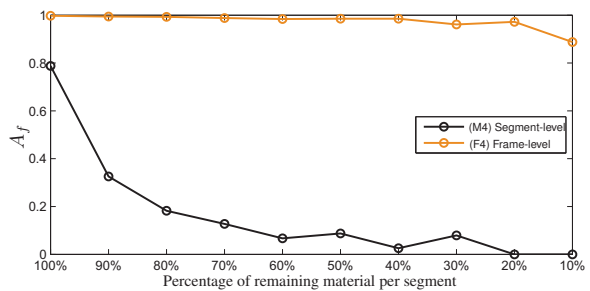
### 4.3 Quantitative Evaluation

In this section, we discuss the results of Table 1. Note that all abridged versions have less than 50% of the duration of the reference version (7763 seconds). From the mean frame accuracy values for all approaches, we can conclude that the segment-level matching (0.563) is not well suited for dealing with abridged versions, whereas the different strategies in the frame-level approaches F1 (0.687) – F4 (0.924) lead to a subsequent improvement of the frame accuracy. Using the segment-level approach, the frame accuracies for the versions *Ros1956* (0.887) and *Sch1994* (0.742) stand out compared to the other versions. The segments that are performed in these versions are not shortened and therefore largely coincide with the segments of the reference version. This explains why the segment-level matching still performs reasonably well on these versions.

In Figure 7, we show the frame accuracy results for the approaches M4 and F4 obtained from an experiment on a set of systematically constructed abridged versions. The frame accuracy values at 100% correspond to a subset of 10 segments (out of 38) that were taken from a complete recording of the opera “Der Freischütz” recorded by Keilberth in 1958. From this subset, we successively removed 10% of the frames from each segment by removing 5% of the frames at the start, and 5% of the frames at the end sections of the segments. In the last abridged version, only 10% of each segment remains. This experiment further supports the conclusions that the segment-level approach is not appropriate for dealing with abridged versions, whereas the frame-level segmentation approach stays robust and flexible even in the case of strong abridgments.

## 5. CONCLUSIONS

In this paper, we approached the problem of transferring the segmentation of a complete reference recording onto an abridged version of the same musical work. We compared the proposed frame-level segmentation approach based on partial matching with a segment-level matching strategy. In experiments with abridged recordings, we have shown that our frame-level approach is robust and flexible when



**Figure 7.** Performance of segment-level approach (M4) versus frame-level approach (F4) on constructed abridged versions. See Section 4.3

enhancing the path structure of the used similarity matrix and applying a mode filter on the labeled frame sequence before deriving the final segmentation.

**Acknowledgments:** This work has been supported by the BMBF project *Freischütz Digital* (Funding Code 01UG1239A to C). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer IIS.

## 6. REFERENCES

- [1] Roger B. Dannenberg and Ning Hu. Polyphonic audio matching for score following and intelligent audio editors. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 27–34, San Francisco, USA, 2003.
- [2] Simon Dixon and Gerhard Widmer. MATCH: A music alignment tool chest. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, London, GB, 2005.
- [3] Sebastian Ewert, Meinard Müller, Verena Konz, Daniel Müllensiefen, and Geraint Wiggins. Towards cross-version harmonic analysis of music. *IEEE Transactions on Multimedia*, 14(3):770–782, 2012.
- [4] Sebastian Ewert, Bryan Pardo, Meinard Müller, and Mark D. Plumbley. Score-informed source separation for musical audio recordings: An overview. *IEEE Signal Processing Magazine*, 31(3):116–124, May 2014.
- [5] Maarten Grachten, Martin Gasser, Andreas Arzt, and Gerhard Widmer. Automatic alignment of music performances with structural differences. In *ISMIR*, pages 607–612, 2013.
- [6] Hanna Lukashevich. Towards quantitative measures of evaluating song segmentation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 375–380, Philadelphia, USA, 2008.
- [7] Nicola Montecchio, Emanuele Di Buccio, and Nicola Orio. An efficient identification methodology for improved access to music heritage collections. *Journal of Multimedia*, 7(2):145–158, 2012.
- [8] Meinard Müller. *Information Retrieval for Music and Motion*. Springer Verlag, 2007.
- [9] Meinard Müller, Nanzhu Jiang, and Harald Grohgan. SM Toolbox: MATLAB implementations for computing and enhancing similarity matrices. In *Proceedings of the AES Conference on Semantic Audio*, London, GB, 2014.
- [10] Thomas Prätzlich and Meinard Müller. Freischütz digital: A case study for reference-based audio segmentation of operas, to appear. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 589–594, Curitiba, Brazil, 2013.
- [11] Joan Serra, Emilia Gómez, Perfecto Herrera, and Xavier Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech and Language Processing*, 16:1138–1151, 2008.
- [12] John Woodruff, Bryan Pardo, and Roger B. Dannenberg. Remixing stereo music with score-informed source separation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 314–319, Victoria, Canada, 2006.



# CREATING A CORPUS OF JINGJU (BEIJING OPERA) MUSIC AND POSSIBILITIES FOR MELODIC ANALYSIS

**Rafael Caro Repetto**

Music Technology Group,  
Universitat Pompeu Fabra, Barcelona  
rafael.caro@upf.edu

**Xavier Serra**

Music Technology Group,  
Universitat Pompeu Fabra, Barcelona  
xavier.sera@upf.edu

## ABSTRACT

Jingju (Beijing opera) is a Chinese traditional performing art form in which theatrical and musical elements are intimately combined. As an oral tradition, its musical dimension is the result of the application of a series of pre-defined conventions and it offers unique concepts for musicological research. Computational analyses of jingju music are still scarce, and only a few studies have dealt with it from an MIR perspective. In this paper we present the creation of a corpus of jingju music in the framework of the CompMusic project that is formed by audio, editorial metadata, lyrics and scores. We discuss the criteria followed for the acquisition of the data, describe the content of the corpus, and evaluate its suitability for computational and musicological research. We also identify several research problems that can take advantage of this corpus in the context of computational musicology, especially for melodic analysis, and suggest approaches for future work.

## 1. INTRODUCTION

Jingju (also known as Peking or Beijing opera) is one of the most representative genres of *xiqu*, the Chinese traditional form of performing arts. Just as its name suggests, it consists of a theatrical performance, *xi*, in which the main expressive element is the music, *qu*. Although it has commonalities with theatre and opera, it cannot be fully classified as any of those. In *xiqu* there are not equivalent figures to that of the theatre director or opera composer; instead, the actor is the main agent for creativity and performance. Each of the skills that the actor is expected to master, encompassing poetry, declamation, singing, mime, dance and martial arts, is learned and executed as pre-defined, well established conventions. It is precisely the centrality of the actor and the acting through conventions what make *xiqu* unique. Its musical content is also created by specific sets of such conventions.

*Xiqu* genres developed as adaptations of the general common principles of the art form to a specific region, especially in terms of dialect and music. The adoption of local dialects was a basic requirement for the intelligibil-

ity of the performance by local audiences. The phonetic features of these dialects, including intonation and especially linguistic tones, establish a melodic and rhythmic framework for the singing. The musical material itself derives from local tunes, which is precisely the literal meaning of *qu*. This implies that music in *xiqu* is not an original creation by the actors, but an adaptation of pre-existing material. Furthermore, each genre employs also the most representative instruments of the region for accompaniment, conveying the regional filiation also timbrally. These local features are what define each *xiqu* genre's individuality. Jingju is then the regional genre of *xiqu* that formed in Beijing during the 19<sup>th</sup> Century, achieving one of the highest levels of refinement and complexity.

Despite the uniqueness of this tradition, the interesting aspects for musicological analysis it offers, and its international recognition, jingju music has barely been approached computationally. Most of the few studies of jingju in MIR have focused on its acoustic and timbral characteristics. Zhang and Zhou have drawn on these features for classification of jingju in comparison with other music traditions [18, 19] and other *xiqu* genres [20]. Sundberg et al. have analyzed acoustically the singing of two role-types [14], whilst Tian et al. have extracted timbral features for onset detection of percussion instruments [15]. More recently, Zhang and Wang have integrated domain knowledge for musically meaningful segmentation of jingju arias [21]. Related to melodic analysis, Chen [3] has implemented a computational analysis of jingju music for the characterization of pitch intonation.

The main concepts that define jingju music are *shengqiang*, *banshi* and role-type. As stated previously, the melodic material used in *xiqu* genres is not original, but derived from local tunes. These tunes share common features that allow them to be recognized as pertaining to that specific region, such as usual scale, characteristic timbre, melodic structure, pitch range and tessitura, ornamentation, etc. This set of features is known as *shengqiang*, usually translated into English as 'mode' or 'modal system' [16]. Each *xiqu* genre can use one or more *shengqiang*, and one single *shengqiang* can be shared by different genres. There are two main *shengqiang* in jingju, namely *xipi* and *erhuang* (see Table 2). Their centrality in the genre is such that jingju music as a whole has been also named by the combination of these two terms, *pihuang*.



© Rafael Caro Repetto, Xavier Serra.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Rafael Caro Repetto, Xavier Serra. "Creating a corpus of jingju (Beijing opera) music and possibilities for melodic analysis", 15th International Society for Music Information Retrieval Conference, 2014.

The melodic features determined by the *shengqiang* are rhythmically rendered through a series of metrical patterns called *banshi*. These *banshi* are individually labelled and defined by a unit of metre, a tempo value and a degree of melodic density; they are associated as well to an expressive function. The system of *banshi* is conceived as derived from an original one, called *yuanban*, so that the rest of them are expansions, reductions or free realizations of the first one [8]. The *banshi* system in jingju consists of a core of eight patterns commonly used, plus some variants.

Each of the characters of a play is assigned to one specific, pre-defined acting class, according to their gender, age, social status, psychological profile and emotional behavior. These acting classes are known as *hangdang* or role-types, and each actor is specialized in the performance of one of them. Each role-type determines the specific set of conventions that must be used for the creation of the character, including those attaining to music. Consequently, *shengqiang* and *banshi* will be expressed differently by each role-type, so that these concepts cannot be studied without referencing each other. In jingju there are four general categories of role-types, with further subdivisions. We consider that the five main role-types regarding musical expression are *sheng* (male characters), *dan* (female characters), *jing* (painted-face), *xiaosheng* (young males), and *laodan* (old females). They are usually classified into two styles of singing, the male style, characterized for using chest voice, used by *sheng*, *jing* and *laodan*, and the female one, sung in falsetto and higher register, used by *dan* and *xiaosheng*.

The fullest expression of such melodic concepts occurs in the singing sections called *changduan*, which can be compared, but not identified, with the concept of aria in Western opera (to ease readability, we will use the term ‘aria’ throughout the paper). Consequently, we have determined the aria as our main research object, and it has been the main concern for the creation of our corpus and the analyses suggested.

In this paper we present a corpus of jingju music that we have gathered for its computational analysis. We explain the criteria followed for the collection of its different types of data, describe the main features of the corpus and discuss its suitability for research. Thereupon we explore the possibilities that the corpus offers to computational musicology, focusing in melodic analysis, specifically in the concepts of *shengqiang* and role-type.

## 2. JINGJU MUSIC RESEARCH CORPUS

In order to undertake a computational analysis of jingju music, and to exploit the unique musical concepts of this tradition from an MIR perspective, we have gathered in the CompMusic project a research corpus [12] that includes audio, editorial metadata, lyrics and scores. We introduce here the criteria for the selection of the data, describe its content and offer a general evaluation.

### 2.1 Criteria for data collection

For the collection of audio recordings, which is the core of the corpus, we have considered three main criteria: repertoire to be covered, sound quality and recording unit. In order to take maximum advantage of the unique features of jingju music, we have gathered recordings of mostly traditional repertoire, as well as some modern compositions based on the traditional methods. The so-called contemporary plays, since they have been created integrating compositional techniques from the Western tradition, have been disregarded for our corpus. Regarding the sound quality needed for a computational analysis, and considering the material to which we have had access, the recordings that best suited our requirements have been commercial CDs released in the last three decades in China. Finally, since our main research object is the aria, we have acquired CDs releases of single arias per track. This means that full play or full scene CDs, and video material in VCD and DVD have not been considered. These CDs have been the source from which we have extracted the editorial metadata contained in the corpus, which have been stored in MusicBrainz.<sup>1</sup> This platform assigns one unique ID to each entity in the corpus, so that they can be easily searchable and retrievable.

Our aim has been to include for each audio recording, whenever possible, its corresponding lyrics and music score. All releases gathered included the lyrics in their leaflets for the arias recorded. However, since they are not usable for computational purposes, we get them from specialized free repositories in the web.<sup>2</sup> As for the scores, an explanation of its function in the tradition is first needed. Since jingju music has been created traditionally by actors, no composers, drawing on pre-existing material, scores appeared only as an aide-mémoire and a tool for preserving the repertoire. Although barely used by professional actors, scores have been widely spread among amateur singers and aficionados, and are a basic resource for musicological research. In fact, in the last decades there has been a remarkable effort to publish thoroughly edited collections of scores. Although many scores are available in the web, they have not been systematically and coherently stored, what makes them not easily retrievable. Furthermore, they consist of image or pdf files, not usable computationally. Consequently, we have acquired printed publications that meet the academic standards of edition, but that will require to be converted into a machine readable format.

### 2.2 Description of the corpus

The audio data collection of the corpus is formed by 78 releases containing 113 CDs, consisting of collections of single arias per track. Besides these, due to the fact that

<sup>1</sup><http://musicbrainz.org/collection/40d0978b-0796-4734-9fd4-2b3ebe0f664c>

<sup>2</sup>Our main sources for lyrics are the websites <http://www.jingju.com>, <http://www.jingju.net> and <http://www.xikao.com>.

many of the consumers of these recordings are amateur singers, many releases contain extra CDs with just the instrumental accompaniment of the arias recorded in the main ones. Consequently, the corpus also contains 19 CDs with just instrumental accompaniments.

Although we do not have complete figures yet, we have computed statistics for different aspects of the corpus. The releases contain recordings by 74 singers, belonging to 7 different role-types, as indicated in Table 1.

<i>Laosheng</i>	20	<i>Xiaosheng</i>	9
<i>Jing</i>	7	<i>Wusheng</i>	3
<i>Laodan</i>	8	<i>Chou</i>	3
<i>Dan</i>	24		

**Table 1.** Number of singers per role-type in the corpus.

As for the works, the corpus covers 653 arias from 215 different plays. Table 2 shows the distribution of these arias according to role-type and *shengqiang*. Since the number of *banshi* is limited and all of them frequently used, an estimation of its appearance in the corpus is not meaningful. As shown in Table 2, the corpus contains highly representative samples for the research of the two main *shengqiang* and the five main role-types as described in section 1. Table 3 displays more detailed numbers concerning these specific entities. The editorial metadata stored in MusicBrainz include textual information as well as cover art. For the former the original language has been maintained, that is, Chinese in simplified characters, with romanizations in the Pinyin system, stored either as pseudo-releases or aliases.

Role-types		<i>Shengqiang</i>	
<i>Laosheng</i>	224	<i>Xipi</i>	324
<i>Jing</i>	55	<i>Erhuang</i>	200
<i>Laodan</i>	66	<i>Fan'erhuang</i>	31
<i>Dan</i>	257	<i>Nanbangzi</i>	25
<i>Xiaosheng</i>	43	<i>Sipingdiao</i>	23
<i>Wusheng</i>	3	<i>Others</i>	45
<i>Chou</i>	5	<i>Unknown</i>	5

**Table 2.** Distribution of the arias in the corpus according to role-type and *shengqiang*.

Regarding the scores, the corpus contains two collections of full play scores [5, 22] and an anthology of selected arias [6]. The two collections contain a total of 155 plays, 26 of which appear in both publications; the anthology contains 86 scores. This material offers scores for 317 arias of the corpus, that is 48.5% of the total.

Apart from the research corpus, but related to it, specific test corpora will be developed, consisting of collections of data used as ground truth for specific research tasks, as defined by Serra [12]. The test corpora created in the framework of the CompMusic project are accessible from the website <http://compmusic.upf.edu/datasets>. To date there are two test corpora related to the jingju music corpus, namely the *Beijing opera percussion in-*

*strument dataset*,<sup>3</sup> which contains 236 audio samples of jingju percussion instruments, used by Tian et al. [15] for onset detection of these instruments, and the *Beijing opera percussion pattern dataset*,<sup>4</sup> formed by 133 audio samples of five jingju percussion patterns, supported by transcriptions both in staff and syllable notations. Srinivasamurthy et al. [13] have used this dataset for the automatic recognition of such patterns in jingju recordings.

## 2.3 Evaluation of the corpus

For the evaluation of the corpus, we will draw on some of the criteria defined by Serra [12] for the creation of culture specific corpora, specifically coverage and completeness, and discuss as well the usability of the data for computational analyses.

### 2.3.1. Coverage

Assessing the coverage of the jingju music corpus is not an easy task, since, to the best of our knowledge, there is no reference source that estimates the number of plays in this tradition. However, compared with the number of total plays covered in our collections of full play scores, which are considered to be the most prominent publications in this matter, the number of plays represented in our corpus is considerable higher. Besides, these releases have been purchased in the specialized bookshop located in the National Academy of Chinese Theatre Arts, Beijing, the only institution of higher education in China dedicated exclusively to the training of *xiqu* actors, and one of the most acclaimed ones for jingju. Our corpus contains all the releases available in this bookshop at the time of writing this paper that met the criteria settled in section 2.1. Regarding the musical concepts represented in the corpus, Table 2 shows that both systems of role-type and *shengqiang* are equally fully covered, with unequal proportion according to their relevance in the tradition, as explained in the introduction. As for the *banshi*, as stated previously, they are fully covered due to their limited number and varied use. Consequently, we argue that the coverage of our corpus is highly satisfactory, in terms of variety of repertoire, availability in the market and representation of musical entities.

### 2.3.2. Completeness

Considering the musicological information needed for each recording according to our purposes, the editorial metadata contained in the releases are fully complete, with the exception of the 5 arias mentioned in Table 2 (0.8% of the total), which lack information about *shengqiang* and *banshi*. One important concept for aficionados of this tradition is the one of *liupai*, or performing schools. However, this concept is far from being well defined, and depends both on the play and on the per-

<sup>3</sup><http://compmusic.upf.edu/bo-perc-dataset>

<sup>4</sup><http://compmusic.upf.edu/bopp-dataset>

Role-type	Singers	<i>Xipi</i>		<i>Erhuang</i>		Total	
		Recordings	Duration	Recordings	Duration	Recordings	Duration
<b>Laosheng</b>	18	179	12h 09m 47s	147	13h 30m 51s	326	25h 40m 38s
<b>Jing</b>	6	41	3h 04m 30s	43	3h 21m 39s	84	6h 26m 09s
<b>Laodan</b>	8	30	2h 00m 54s	52	4h 37m 54s	82	6h 38m 48s
<b>Dan</b>	24	224	17h 26m 00s	101	11h 13m 02s	325	28h 39m 02s
<b>Xiaosheng</b>	9	40	3h 19m 00s	7	41m 14s	47	4h 00m 14s
<i>Total</i>	65	514	38h 00m 11s	350	33h 24m 40s	864	71h 24m 51s

**Table 3.** Data in our corpus for the analysis of the two main *shengqiang* and five main role-types.

former, and usually is not specifically stated in the releases. Finally, the information related to the publication of the recordings is not consistent. Usually, the dates of recording and releasing are not available from the CDs. However, the releasing period has been restricted to the last three decades by our criteria, as stated in section 2.1, although in some rare cases some of these releases may contain recordings from earlier periods.

### 2.3.3. Usability

The data contained in the corpus are fully suitable for analysis of jingju music according to the musical concepts explained in section 1. However, not all the data are equally usable. The main difficulty is presented by the scores, to date available only in printed edition. Consequently, for their computational exploitation they need to be converted into a machine readable format. In the CompMusic project we intend to use MusicXML, maintaining the so called *jianpu* notation used in the originals. As for the lyrics, although most of them are freely accessible on the web, due to the fact that singers may make some changes according to their needs, some problems for the recognition of related lyrics for a specific aria might rise.

To access the corpus for research purposes, we refer to the website <http://compmusic.upf.edu/corpora>. The corpus will eventually be also available through Dunya [9],<sup>5</sup> a web based browsing tool developed by the CompMusic project, which also displays content-based analyses carried out in its framework for each of the culturally specific corpora that it has gathered.

## 3. RESEARCH POSSIBILITIES FOR THE JINGJU MUSIC CORPUS

In this section we introduce research issues of relevance for each data type in our corpus with a special focus on the melodic analysis. We discuss the application of state of the art analytic approaches to our corpus, and propose specific future work.

### 3.1 Analyses of audio, lyrics and scores

According to the research objectives in the CompMusic project, in whose framework our corpus has been gath-

ered, audio data is the main research object, supported by information from metadata, lyrics and scores. For the analysis of the musical elements described in the first section, the vocal line of the arias is the most relevant element, since it renders the core melody of the piece. Consequently, segmentation of the vocal part and extraction of its pitch are needed steps. However, the timbral and textural characteristics of jingju music pose important challenges for these tasks. The timbre of the main accompanying instrument, the jinghu, a small, two-stringed spike fiddle, is very similar to that of the voice. Besides, the typical heterophonic texture results in the simultaneous realization of different versions of the same melody. These features make the extraction of the vocal line from the accompaniment difficult. Besides, octave errors are still frequent to state of the art algorithms for predominant melody extraction, especially for role-types of the male style of singing.

If audio is the main research object, the other types of data in our corpus offer equally interesting and challenging tasks for computational analysis. The delivery of the lyrics is the main goal of singing in jingju; therefore their analysis is essential for the understanding of the genre. Of special importance for its musical implications is the analysis of the poetic structure of the lyrics, since it determines the musical one, as well as their meaning, what would help to better define the expressive function of *shengqiang* and *banshi*. Methods from natural language processing can be applied for the identification of poetic formulae, commonly used by actors for the creation of new lyrics. As for the scores, their analysis will be beneficial for the computation of intervallic preferences, creation of cadential schemata and detection of stable pitches.

However, as stated previously, the main use of lyrics and scores according to our research purposes will be as supporting elements for audio analysis. To that aim, the main computational task is the alignment of both data types to audio. This is a challenging task, since the actors, in a tradition without the authority of a composer or playwright, have certain margins to modify lyrics and melody according to their own interpretation, as far as the main features of the aria, as sanctioned by the tradition, are maintained. In the case of lyrics, this task is even more complex due to the fact that jingju uses an art language of its own, that combines linguistic features from two dialects, the Beijing dialect and the Huguang dialect

<sup>5</sup><http://dunya.compmusic.upf.edu>

from the South [8, 16]. This combination is far from being systematic and consistent, what in many cases poses difficulties to the prediction of the phonetic representation required for lyrics to audio alignment. The music traditions researched in the CompMusic project present similar problems for these tasks, and specific approaches have been proposed by Şentürk et al. [11] and Dzhambazov et al. [4]. We intend to benefit from these works for the development of specific methods for jingju music.

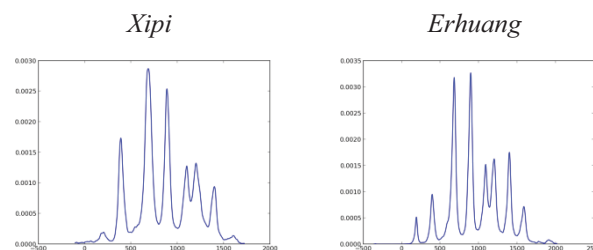
Alignment of lyrics and scores to audio will be an important step for several analytical tasks. The information from these data types combined with the audio will allow a musically informed segmentation of the recording, either in vocal or instrumental sections, or in different structural units, from the couplet, a poetic structure which is the basic unit also for the musical one, to the syllable level. The absolute pitch value of the first degree can be computed by the combined information from the score and the pitch track. Finally, an especially interesting topic is the study of how the tonal information of the syllable is expressed in the melody. Zhang et al. [17] have applied computational methods to this issue within the context of the CompMusic project.

### 3.2. Characterization of *shengqiang* and role-type

As stated in the introduction, the two more relevant concepts for the melodic aspect of jingju are *shengqiang* and role-type. Chen [3] has attempted a characterization of these entities by means of pitch histograms. For the classification of audio fragments as vocal and non-vocal Chen drew on machine learning, and extracted the pitch of the vocal line with the algorithm proposed by Salamon and Gómez [10]. In order to overcome some limitations of the results in this work, we have carried out an initial experiment in which we have extracted pitch tracks for a subset of 30 arias from our corpus that have been manually pre-processed. The sample contains three arias for each of the ten combinations of the two main *shengqiang* and the five main role-types. We use mp3 mono files with a sampling rate of 44,100 Hz, and have annotated them with Praat [1] to the syllable level for segmentation. Pitch tracks have been obtained with the aforementioned algorithm [10] implemented in Essentia [2], whose parameters have been manually set for each aria.

The obtained pitch tracks have been used for the computation of pitch histograms. Koduri et al. have successfully characterized Carnatic ragas by histogram peak parametrization [7]. Chen has applied this methodology for the characterization of male and female styles of singing in jingju. We have expanded the same approach to our subset of 30 arias, with the aim of characterizing the ten combinations of *shengqiang* and role-types. Our initial observations give some evidence that pitch histograms will help describe some aspects of *shengqiang* and role-types as stated in the musicological literature, such as modal center, register with respect to the first degree,

range and hierarchy of scale degrees, so that differences can be established between each category. Our results also show that the approach is efficient for the characterization of different role-types for the same *shengqiang*. However, differences are not meaningful when the two *shengqiang* are compared for one single role-type. Figure 1 shows how the modal center for both *xipi* and *erhuang* in a *dan* role-type is located around the fifth and sixth degrees, register with respect to the first degree and range are practically identical, and the differences in the hierarchy of scale degrees are not relevant enough.



**Figure 1.** Pitch histograms for the *dan* role-type in the two *shengqiang*.

In our future work we propose to expand this approach by integrating the information obtained from the lyrics and the scores. In the specific case of *shengqiang*, a work of melodic similarity between arias of the *shengqiang* according to their musical structure, specially determined by the *banshi*, will shed light on the melodic identity of these entities. As for the role-type, we argue that an analysis of timbre, dynamics and articulation for each category, especially at the syllable level, will offer characterizing features that complete the information obtained from the pitch histograms.

### 3.3. Other research tasks

Beyond the tasks described previously, jingju music offer a wide range of research possibilities. One important aspect is the rhythmic component of the arias, mainly determined by the concept of *banshi*. An automatic identification of *banshi* and segmentation of the aria in these sections is a musically meaningful, but computational challenging task, due to the different renditions of the same *banshi* by different role-types and even different actors, as well as to the rhythmic flexibility that characterizes jingju music. The instrumental sections of the jingju arias are also an interesting research topic, especially regarding their relationship with the melody of the vocal part and how *shengqiang* and *banshi* define their features. To this task, the CDs with only accompaniment tracks will be valuable. In the framework of the CompMusic project, Srinivasamurthy et al. [13] have presented a computational model for the automatic recognition of percussion patterns in jingju. Finally, due to the importance of the acting component of this genre, and its intimate relationship with the music, jingju is a perfect case for a com-

bined research of visual and musical features, integrating computational analysis of video and audio material. Should this task be undertaken, our corpus should be expanded to include also video material.

#### 4. SUMMARY

In this paper we have presented a corpus of jingju music, gathered with the purpose of researching its musical features from an MIR methodology. After discussing the criteria for its creation, describing its different data types and offering a general evaluation, we have suggested analytical tasks for its computational exploitation, especially focused on melodic analysis. Some state of the art approaches have been applied to a small sample of the corpus, in order to analyze their results and propose consequently further work and future tasks.

#### 5. ACKNOWLEDGEMENTS

This research was funded by the European Research Council under the European Union's Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583). We are thankful to G. K. Koduri for providing and helping with his code.

#### 6. REFERENCES

- [1] P. Boersma and D. Weenink: "Praat, a system for doing phonetics by computer," *Glott International*, Vol. 5, No. 9/10, pp. 341–345.
- [2] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra: "ESSENTIA: an Audio Analysis Library for Music Information Retrieval," *ISMIR 2013*, pp. 423–498, 2013.
- [3] K. Chen: *Characterization of Pitch Intonation of Beijing Opera*, Master thesis, Universitat Pompeu Fabra, Barcelona, 2013.
- [4] G. Dzhabazov, S. Şentürk, and X. Serra: "Automatic Lyrics-to-Audio Alignment in Classical Turkish Music," *The 4<sup>th</sup> International Workshop on Folk Music Analysis*, pp. 61–64, 2014.
- [5] *Jingju qupu jicheng* 京剧曲谱集成 (Collection of jingju scores), 10 vols., Shanghai wenyi chubanshe, Shanghai, 1998.
- [6] *Jingju qupu jingxuan* 京剧曲谱精选 (Selected scores of jingju), 2 vols., Shanghai yinyue chubanshe, Shanghai, 1998–2005.
- [7] G. K. Koduri, V. Ishwar, J. Serrà, X. Serra, and H. Murthy: "Intonation analysis of ragas in Carnatic music," *Journal of New Music Research*, Vol. 43, No. 1, pp. 72–93.
- [8] J. Liu 刘吉典: *Jingju yinyue gailun* 京剧音乐概论 (Introduction to jingju music), Renmin yinyue chubanshe, Beijing, 1993.
- [9] A. Porter, M. Sordo, and X. Serra: "Dunya: A system for browsing audio music collections exploiting cultural context," *ISMIR 2013*, pp. 101–106, 2013.
- [10] J. Salamon and E. Gómez: "Melody Extraction From Polyphonic Music Signals Using Pitch Contour Characteristics," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No. 6, pp. 1759–1770, 2012.
- [11] S. Şentürk, A. Holzapfel, and X. Serra: "Linking Scores and Audio Recordings in Makam Music of Turkey," *JNMR*, Vol. 43, No. 1, pp. 34–52, 2014.
- [12] X. Serra: "Creating Research Corpora for the Computational Study of Music: the case of the CompMusic Project," *Proceedings of the AES 53<sup>rd</sup> International Conference*, pp. 1–9, 2014.
- [13] A. Srinivasamurthy, R. Caro Repetto, S. Harshavardhan, and X. Serra: "Transcription and Recognition of Syllable based Percussion Patterns: The Case of Beijing Opera," *ISMIR 2014*.
- [14] J. Sundberg, L. Gu, Q. Huang, and P. Huang: "Acoustical study of classical Peking Opera singing," *Journal of Voice*, Vol. 26, No. 2, pp. 137–143, 2012.
- [15] M. Tian, A. Srinivasamurthy, M. Sandler, and X. Serra: "A study of instrument-wise onset detection in Beijing opera percussion ensembles," *ICASSP 2014*, pp. 2174–2178, 2014.
- [16] E. Wichmann: *Listening to theatre: the aural dimension of Beijing opera*, University of Hawaii Press, Honolulu, 1991.
- [17] S. Zhang, R. Caro Repetto, and X. Serra: "Study of similarity between linguistic tones and melodic contours in Beijing Opera," *ISMIR 2014*.
- [18] Y. Zhang and J. Zhou: "A Study on Content-Based Music Classification," *Proceedings of the Seventh International Symposium on Signal Processing and Its Applications*, pp. 113–162, 2003.
- [19] Y. Zhang and J. Zhou: "Audio Segmentation Based on Multi-Scale Audio Classification," *ICASSP 2004*, pp. 349–352, 2004.
- [20] Y. Zhang, J. Zhou, and X. Wang: "A Study on Chinese Traditional Opera," *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics*, pp. 2476–2480, 2008.
- [21] Z. Zhang and X. Wang: "Structure Analysis of Chinese Peking Opera," *Seventh International Conference on Natural Computation*, pp. 237–241, 2011.
- [22] *Zhongguo jingju liupai jumu jicheng* 中国京剧流派剧目集成 (Collection of plays of Chinese jingju schools), 21 vols., Xueyuan chubanshe, Beijing, 2006–2010.

# MODELING TEMPORAL STRUCTURE IN MUSIC FOR EMOTION PREDICTION USING PAIRWISE COMPARISONS

Jens Madsen, Bjørn Sand Jensen, Jan Larsen

Technical University of Denmark,  
Department of Applied Mathematics and Computer Science,  
Richard Petersens Plads, Building 321,  
2800 Kongens Lyngby, Denmark  
{jenma, bjje, janla}@dtu.dk

## ABSTRACT

The temporal structure of music is essential for the cognitive processes related to the emotions expressed in music. However, such temporal information is often disregarded in typical Music Information Retrieval modeling tasks of predicting higher-level cognitive or semantic aspects of music such as emotions, genre, and similarity. This paper addresses the specific hypothesis whether temporal information is essential for predicting expressed emotions in music, as a prototypical example of a cognitive aspect of music. We propose to test this hypothesis using a novel processing pipeline: 1) Extracting audio features for each track resulting in a multivariate "feature time series". 2) Using generative models to represent these time series (acquiring a complete track representation). Specifically, we explore the Gaussian Mixture model, Vector Quantization, Autoregressive model, Markov and Hidden Markov models. 3) Utilizing the generative models in a discriminative setting by selecting the Probability Product Kernel as the natural kernel for all considered track representations. We evaluate the representations using a kernel based model specifically extended to support the robust two-alternative forced choice self-report paradigm, used for eliciting expressed emotions in music. The methods are evaluated using two data sets and show increased predictive performance using temporal information, thus supporting the overall hypothesis.

## 1. INTRODUCTION

The ability of music to represent and evoke emotions is an attractive and yet a very complex quality. This is partly a result of the dynamic temporal structures in music, which are a key aspect in understanding and creating predictive models of more complex cognitive aspects of music such as the emotions expressed in music. So far the approach

of creating predictive models of emotions expressed in music has relied on three major aspects. First, self-reported annotations (rankings, ratings, comparisons, tags, etc.) for quantifying the emotions expressed in music. Secondly, finding a suitable audio representation (using audio or lyrical features), and finally associating the two aspects using machine learning methods with the aim to create predictive models of the annotations describing the emotions expressed in music. However the audio representation has typically relied on classic audio-feature extraction, often neglecting how this audio representation is later used in the predictive models.

We propose to extend how the audio is represented by including *feature representation* as an additional aspect, which is illustrated on Figure 1. Specifically, we focus on including the temporal aspect of music using the added feature representation [10], which is often disregarded in the classic audio-representation approaches. In Music Information Retrieval (MIR), audio streams are often represented with frame-based features, where the signal is divided into frames of samples with various lengths depending on the musical aspect which is to be analyzed. Feature extraction based on the enframed signal results in multivariate time series of feature values (often vectors). In order to use these features in a discriminative setting (i.e. predicting tags, emotion, genre, etc.), they are often represented using the mean, a single or mixtures of Gaussians (GMM). This can reduce the time series to a single vector and make the features easy to use in traditional linear models or kernel machines such as the Support Vector Machine (SVM). The major problem here is that this approach disregards all temporal information in the extracted features. The frames could be randomized and would still have the same representation, however this randomization makes no sense musically.

In modeling the emotions expressed in music, the temporal aspect of emotion has been centered on how the labels are acquired and treated, not on how the musical content is treated. E.g. in [5] they used a Conditional Random Field (CRF) model to essentially smooth the predicted labels of an SVM, thus still not providing temporal information re-

This work was supported in part by the Danish Council for Strategic Research of the Danish Agency for Science Technology and Innovation under the CoSound project, case number 11-115328.



© Jens Madsen, Bjørn Sand Jensen, Jan Larsen.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Jens Madsen, Bjørn Sand Jensen, Jan Larsen. "Modeling Temporal Structure in Music for Emotion Prediction using Pairwise Comparisons", 15th International Society for Music Information Retrieval Conference, 2014.

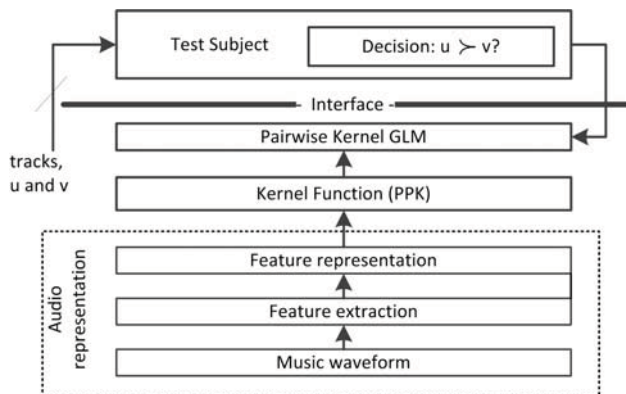


Figure 1. Modeling pipeline.

garding the features. In [12] a step to include some temporal information regarding the audio features was made, by including some first and second order Markov properties for their CRF model, however still averaging the features for one second windows. Other approaches have ranged from simple feature stacking in [13] to actually using a generative temporal model to represent features in [17]. The latter showed that using a Dynamical Texture Mixture model to represent the feature time series of MFCCs, taking temporal dynamics into account, carried a substantial amount of information about the emotional content. In the present work, in contrast to prior work, we focus on creating a common framework by using generative models to represent the multivariate feature time series for the application of modeling aspects related to the emotions expressed in music. Since very little work has been done within this field, we make a broad comparison of a multitude of generative models of time series data. We consider how the time series are modeled on two aspects: whether the observations are continuous or discrete, and whether temporal information should be taken into account or not. This results in four different combinations, which we investigate: 1) a continuous, temporal, independent representation which includes the mean, single Gaussian and GMM models; 2) a temporal, dependent, continuous representation using Autoregressive models; 3) a discretized features representation using vector quantization in a temporally independent Vector Quantization (VQ) model; and finally 4) a representation including the temporal aspect fitting Markov and Hidden Markov Models (HMM) on the discretized data. A multitude of these models have never been used in MIR as a track-based representation in this specific setting. To use these generative models in a discriminative setting, the Product Probability Kernel (PPK) is selected as the natural kernel for all the feature representations considered. We extend a kernel-generalized linear model (kGLM) model specifically for pairwise observations for use in predicting emotions expressed in music. We specifically focus on the feature representation and the modeling pipeline and therefore use simple, well-known, frequently used MFCC features. In total, eighteen different models are investigated on two datasets of pairwise comparisons evaluated on the valence and arousal dimensions.

## 2. FEATURE REPRESENTATION

In order to model higher order cognitive aspects of music, we first consider standard audio feature extraction which results in a frame-based, vector space representation of the music track. Given  $T$  frames, we obtain a collection of  $T$  vectors with each vector at time  $t$  denoted by  $\mathbf{x}_t \in \mathbb{R}^D$ , where  $D$  is the dimension of the feature space. The main concern here is how to obtain a track-level representation of the sequence of feature vectors for use in subsequent modelling steps. In the following, we will outline a number of different possibilities — and all these can be considered as probabilistic densities over either a single feature vector or a sequence of such (see also Table. 1).

**Continuous:** When considering the original feature space, i.e. the sequence of multivariate random variables, a vast number of representations have been proposed depending on whether the temporal aspects are ignored (i.e. considering each frame independently of all others) or modeling the temporal dynamics by temporal models.

In the time-independent case, we consider the feature as a bag-of-frames, and compute moments of the independent samples; namely the mean. Including higher order moments will naturally lead to the popular choice of representing the time-collapsed time series by a multivariate Gaussian distribution (or other continuous distributions). Generalizing this leads to mixtures of distributions such as the GMM (or another universal mixture of other distributions) used in an abundance of papers on music modeling and similarity (e.g. [1, 7]).

Instead of ignoring the temporal aspects, we can model the sequence of multivariate feature frames using well-known temporal models. The simplest models include AR models [10].

**Discrete:** In the discrete case, where features are naturally discrete or the original continuous feature space can be quantized using VQ with a finite set of codewords resulting in a dictionary (found e.g. using K-means). Given this dictionary each feature frame is subsequently assigned a specific codeword in a 1-of-P encoding such that a frame at time  $t$  is defined as vector  $\tilde{\mathbf{x}}_t$  with one non-zero element.

At the track level and time-independent case, each frame is encoded as a Multinomial distribution with a single draw,  $\tilde{\mathbf{x}} \sim \text{Multinomial}(\boldsymbol{\lambda}, 1)$ , where  $\boldsymbol{\lambda}$  denotes the probability of occurrence for each codeword and is computed on the basis of the histogram of codewords for the entire track. In the time-dependent case, the sequence of codewords,  $\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_T$ , can be modeled by a relatively simple (first order) Markov model, and by introducing hidden states this may be extended to the (homogeneous) Hidden Markov model with Multinomial observations ( $\text{HMM}_{\text{disc}}$ ).

### 2.1 Estimating the Representation

The probabilistic representations are all defined in terms of parametric densities which in all cases are estimated using standard maximum likelihood estimation (see e.g. [2]). Model selection, i.e. the number of mixture components, AR order, and number of hidden states, is performed using



Obs.	Time	Representation	Density Model	$\theta$	Base
Continuous	Indp.	Mean	$p(\mathbf{x} \theta) \equiv \delta(\boldsymbol{\mu})$	$\mu, \sigma$	Gaussian
		Gaussian	$p(\mathbf{x} \theta) = \mathcal{N}(\mathbf{x} \mu, \Sigma)$	$\mu, \Sigma$	Gaussian
		GMM	$p(\mathbf{x} \theta) = \sum_{i=1}^L \lambda_i \mathcal{N}(\mathbf{x} \mu_i, \Sigma_i)$	$\{\lambda_i, \mu_i, \Sigma_i\}_{i=1:L}$	Gaussian
	Temp.	AR	$p(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_P \theta) = \mathcal{N}(\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_P   \mathbf{m}, \Sigma_{ A,C})$	$\mathbf{m}, \Sigma_{ A,C}$	Gaussian
Discrete	Indp.	VQ	$p(\tilde{\mathbf{x}} \theta) = \lambda$	$\lambda$	Multinomial
	Temp.	Markov	$p(\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_T \theta) = \lambda_{\tilde{\mathbf{x}}_0} \prod_{t=1}^T \Lambda_{\tilde{\mathbf{x}}_t, \tilde{\mathbf{x}}_{t-1}}$	$\lambda, \Lambda$	Multinomial
		HMM <sub>disc</sub>	$p(\tilde{\mathbf{x}}_0, \tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_T \theta) = \sum_{\mathbf{z}_0:T} \lambda_{\mathbf{z}_0} \prod_{t=1}^T \Lambda_{\mathbf{z}_t, \mathbf{z}_{t-1}} \Phi_t$	$\lambda, \Lambda, \Phi$	Multinomial

**Table 1.** Continuous, features,  $\mathbf{x} \in \mathbb{R}^D$ ,  $L$  is the number of components in the GMM,  $P$  indicates the order of the AR model,  $\mathbf{A}$  and  $\mathbf{C}$  are the coefficients and noise covariance in the AR model respectively and  $T$  indicates the length of the sequence. Discrete, VQ:  $\tilde{\mathbf{x}} \sim \text{Multinomial}(\lambda)$ ,  $\Lambda_{\mathbf{z}_t, \mathbf{z}_{t-1}} = p(\mathbf{z}_t|\mathbf{z}_{t-1})$ ,  $\Lambda_{\tilde{\mathbf{x}}_t, \tilde{\mathbf{x}}_{t-1}} = p(\tilde{\mathbf{x}}_t|\tilde{\mathbf{x}}_{t-1})$ ,  $\Phi_t = p(\tilde{\mathbf{x}}_t|\mathbf{z}_t)$ . The basic Mean representation is often used in the MIR field in combination with a so-called squared exponential kernel [2], which is equivalent to formulating a PPK with a Gaussian with the given mean and a common, diagonal covariance matrix corresponding to the length scale which can be found by cross-validation and specifically using  $q = 1$  in the PPK.

Bayesian Information Criterion (BIC, for GMM and HMM), or in the case of the AR model, CV was used.

## 2.2 Kernel Function

The various track-level representations outlined above are all described in terms of a probability density as outlined in Table 1, for which a natural kernel function is the Probability Product Kernel [6]. The PPK forms a common ground for comparison and is defined as,

$$k(p(\mathbf{x}|\theta), p(\mathbf{x}|\theta')) = \int (p(\mathbf{x}|\theta)p(\mathbf{x}|\theta'))^q d\mathbf{x}, \quad (1)$$

where  $q > 0$  is a free model parameter. The parameters of the density model,  $\theta$ , obviously depend on the particular representation and are outlined in Tab.1. All the densities discussed previously result in (recursive) analytical computations. [6, 11].<sup>1</sup>

## 3. PAIRWISE KERNEL GLM

The pairwise paradigm is a robust elicitation method to the more traditional direct scaling approach and is reviewed extensively in [8]. This paradigm requires a non-traditional modeling approach for which we derive a relatively simple kernel version of the Bradley-Terry-Luce model [3] for pairwise comparisons. The non-kernel version was used for this particular task in [9].

In order to formulate the model, we will for now assume a standard vector representation for each of  $N$  audio excerpts collected in the set  $\mathcal{X} = \{\mathbf{x}_i | i = 1, \dots, N\}$ , where  $\mathbf{x}_i \in \mathbb{R}^D$ , denotes a standard,  $D$  dimensional audio feature vector for excerpt  $i$ . In the pairwise paradigm, any two distinct excerpts with index  $u$  and  $v$ , where  $\mathbf{x}_u \in \mathcal{X}$  and  $\mathbf{x}_v \in \mathcal{X}$ , can be compared in terms of a given aspect

<sup>1</sup> It should be noted that using the PPK does not require the same length  $T$  of the sequences (the musical excerpts). For latent variable models, such as the HMM, the number of latent states in the models can also be different. The observation space, including the dimensionality  $D$ , is the only thing that has to be the same.

(such as arousal/valence). With  $M$  such comparisons we denote the output set as  $\mathcal{Y} = \{(y_m; u_m, v_m) | m = 1, \dots, M\}$ , where  $y_m \in \{-1, +1\}$  indicates which of the two excerpts had the highest valence (or arousal).  $y_m = -1$  means that the  $u_m$ 'th excerpt is picked over the  $v_m$ 'th and visa versa when  $y_m = 1$ .

The basic assumption is that the choice,  $y_m$ , between the two distinct excerpts,  $u$  and  $v$ , can be modeled as the difference between two function values,  $f(\mathbf{x}_u)$  and  $f(\mathbf{x}_v)$ . The function  $f: \mathcal{X} \rightarrow \mathbb{R}$  hereby defines an internal, but latent, absolute reference of valence (or arousal) as a function of the excerpt (represented by the audio features,  $\mathbf{x}$ ).

Modeling such comparisons can be accomplished by the Bradley-Terry-Luce model [3, 16], here referred to more generally as the (logistic) pairwise GLM model. The choice model assumes logistically distributed noise [16] on the individual function value, and the likelihood of observing a particular choice,  $y_m$ , for a given comparison  $m$  therefore becomes

$$p(y_m | \mathbf{f}_m) \equiv \frac{1}{1 + e^{-y_m \cdot z_m}}, \quad (2)$$

with  $z_m = f(\mathbf{x}_{u_m}) - f(\mathbf{x}_{v_m})$  and  $\mathbf{f}_m = [f(\mathbf{x}_{u_m}), f(\mathbf{x}_{v_m})]^T$ . The main question is how the function,  $f(\cdot)$ , is modeled. In the following, we derive a kernel version of this model in the framework of kernel Generalized Linear Models (kGLM). We start by assuming a linear and parametric model of the form  $\mathbf{f}_i = \mathbf{x}_i \mathbf{w}^T$  and consider the likelihood defined in Eq. (2). The argument,  $z_m$ , is now redefined such that  $z_m = (\mathbf{x}_{u_m} \mathbf{w}^T - \mathbf{x}_{v_m} \mathbf{w}^T)$ . We assume that the model parameterized by  $\mathbf{w}$  is the same for the first and second input, i.e.  $\mathbf{x}_{u_m}$  and  $\mathbf{x}_{v_m}$ . This results in a projection from the audio features  $\mathbf{x}$  into the dimensions of valence (or arousal) given by  $\mathbf{w}$ , which is the same for all excerpts. Plugging this into the likelihood function we obtain:

$$p(y_m | \mathbf{x}_{u_m}, \mathbf{x}_{v_m}, \mathbf{w}) = \frac{1}{1 + e^{-y_m ((\mathbf{x}_{u_m} - \mathbf{x}_{v_m}) \mathbf{w}^T)}}. \quad (3)$$

Following a maximum likelihood approach, the effective cost function,  $\psi(\cdot)$ , defined as the negative log likelihood is:

$$\psi_{GLM}(\mathbf{w}) = -\sum_{m=1}^M \log p(y_m | \mathbf{x}_{u_m}, \mathbf{x}_{v_m}, \mathbf{w}). \quad (4)$$

Here we assume that the likelihood factorizes over the observations, i.e.  $p(\mathcal{Y}|\mathbf{f}) = \prod_{m=1}^M p(y_m|\mathbf{f}_m)$ . Furthermore, a regularized version of the model is easily formulated as

$$\psi_{GLM-L2}(\mathbf{w}) = \psi_{GLM} + \gamma \|\mathbf{w}\|_2^2, \quad (5)$$

where the regularization parameter  $\gamma$  is to be found using cross-validation, for example, as adopted here. This cost is still continuous and is solved with a L-BFGS method.

This basic pairwise GLM model has previously been used to model emotion in music [9]. In this work, the pairwise GLM model is extended to a general regularized kernel formulation allowing for both linear and non-linear models. First, consider an unknown non-linear map of an element  $\mathbf{x} \in \mathcal{X}$  into a Hilbert space,  $\mathcal{H}$ , i.e.,  $\varphi(\mathbf{x}) : \mathcal{X} \mapsto \mathcal{H}$ . Thus, the argument  $z_m$  is now given as

$$z_m = (\varphi(\mathbf{x}_{u_m}) - \varphi(\mathbf{x}_{v_m})) \mathbf{w}^T \quad (6)$$

The *representer theorem* [14] states that the weights,  $\mathbf{w}$  — despite the difference between mapped instances — can be written as a linear combination of the inputs such that

$$\mathbf{w} = \sum_{l=1}^M \alpha_l (\varphi(\mathbf{x}_{u_l}) - \varphi(\mathbf{x}_{v_l})). \quad (7)$$

Inserting this into Eq. (6) and applying the “kernel trick” [2], i.e. exploiting that  $(\varphi(\mathbf{x}) \varphi(\mathbf{x}'))_{\mathcal{H}} = k(\mathbf{x}, \mathbf{x}')$ , we obtain

$$\begin{aligned} z_m &= (\varphi(\mathbf{x}_{u_m}) - \varphi(\mathbf{x}_{v_m})) \sum_{l=1}^M \alpha_l (\varphi(\mathbf{x}_{u_l}) - \varphi(\mathbf{x}_{v_l})) \\ &= \sum_{l=1}^M \alpha_l (\varphi(\mathbf{x}_{u_m}) \varphi(\mathbf{x}_{u_l}) - \varphi(\mathbf{x}_{u_m}) \varphi(\mathbf{x}_{v_l}) \\ &\quad - \varphi(\mathbf{x}_{v_m}) \varphi(\mathbf{x}_{u_l}) + \varphi(\mathbf{x}_{v_m}) \varphi(\mathbf{x}_{v_l})) \\ &= \sum_{l=1}^M \alpha_l (k(\mathbf{x}_{u_m}, \mathbf{x}_{u_l}) - k(\mathbf{x}_{u_m}, \mathbf{x}_{v_l}) \\ &\quad - k(\mathbf{x}_{v_m}, \mathbf{x}_{u_l}) + k(\mathbf{x}_{v_m}, \mathbf{x}_{v_l})) \\ &= \sum_{l=1}^M \alpha_l k(\{\mathbf{x}_{u_m}, \mathbf{x}_{v_m}\}, \{\mathbf{x}_{u_l}, \mathbf{x}_{v_l}\}). \end{aligned} \quad (8)$$

Thus, the pairwise kernel GLM formulation leads exactly to standard kernel GLM like [19], where the only difference is the kernel function which is now a (valid) kernel between two sets of pairwise comparisons<sup>2</sup>. If the kernel function is the linear kernel, we obtain the basic pairwise logistic regression presented in Eq. (3), but the the kernel formulation easily allows for non-vectorial inputs as provided by the PPK. The general cost function for the kGLM model is

defined as,

$$\psi_{kGLM-L2}(\boldsymbol{\alpha}) = -\sum_{m=1}^M \log p(y_m | \boldsymbol{\alpha}, \mathbf{K}) + \gamma \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha},$$

i.e., dependent on the kernel matrix,  $\mathbf{K}$ , and parameters  $\boldsymbol{\alpha}$ . It is of the same form as for the basic model and we can apply standard optimization techniques. Predictions for unseen input pairs  $\{\mathbf{x}_r, \mathbf{x}_s\}$  are easily calculated as

$$\Delta f_{rs} = f(\mathbf{x}_r) - f(\mathbf{x}_s) \quad (9)$$

$$= \sum_{m=1}^M \alpha_m k(\{\mathbf{x}_{u_m}, \mathbf{x}_{v_m}\}, \{\mathbf{x}_r, \mathbf{x}_s\}). \quad (10)$$

Thus, predictions exist only as *delta* predictions. However it is easy to obtain a “true” latent (arbitrary scale) function for a single output by aggregating all the delta predictions.

## 4. DATASET & EVALUATION APPROACH

To evaluate the different feature representations, two datasets are used. The first dataset (*IMM*) consists of  $N_{\text{IMM}} = 20$  excerpts and is described in [8]. It comprises all  $M_{\text{IMM}} = 190$  unique pairwise comparisons of 20 different 15-second excerpts, chosen from the USPOP2002<sup>3</sup> dataset. 13 participants (3 female, 10 male) were compared on both the dimensions of valence and arousal. The second dataset (*YANG*) [18] consists of  $M_{\text{YANG}} = 7752$  pairwise comparisons made by multiple annotators on different parts of the  $N_{\text{YANG}} = 1240$  different Chinese 30-second excerpts on the dimension of valence. 20 MFCC features have been extracted for all excerpts by the MA toolbox<sup>4</sup>.

### 4.1 Performance Evaluation

In order to evaluate the performance of the proposed representation of the multivariate feature time series we compute learning curves. We use the so-called Leave-One-Excerpt-Out cross validation, which ensures that all comparisons with a given excerpt are left out in each fold, differing from previous work [9]. Each point on the learning curve is the result of models trained on a fraction of all available comparisons in the training set. To obtain robust learning curves, an average of 10-20 repetitions is used. Furthermore a ‘win’-based baseline (*Base<sub>low</sub>*) as suggested in [8] is used. This baseline represents a model with no information from features. We use the McNemar paired test with the *Null* hypothesis that two models are the same between each model and the baseline, if  $p < 0.05$  then the models can be rejected as equal on a 5% significance level.

## 5. RESULTS

We consider the pairwise classification error on the two outlined datasets with the kGLM-L2 model, using the outlined pairwise kernel function combined with the PPK kernel ( $q=1/2$ ). For the *YANG* dataset a single regularization parameter  $\gamma$  was estimated using 20-fold cross validation used

<sup>2</sup> In the Gaussian Process setting this kernel is also known as the Pairwise Judgment kernel [4], and can easily be applied for pairwise leaning using other kernel machines such as support vector machines

<sup>3</sup> <http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html>

<sup>4</sup> <http://www.pampalk.at/ma/>

Obs.	Time	Models	Training set size						
			1%	5%	10%	20%	40%	80%	100%
Continuous	Indp.	Mean	0.468	0.386	0.347	0.310	0.277	0.260	0.252
		$\mathcal{N}(x \mu, \sigma)$	0.464	0.394	0.358	0.328	0.297	0.279	0.274
		$\mathcal{N}(x \mu, \Sigma)$	<b>0.440</b>	0.366	0.328	0.295	0.259	0.253	0.246
		$GMM_{diag}$	0.458	0.378	0.341	0.304	0.274	0.258	0.254
		$GMM_{full}$	0.441	0.362	0.329	0.297	0.269	0.255	0.252
	Temp.	DAR <sub>CV</sub>	0.447	0.360	<b>0.316</b>	<b>0.283</b>	<b>0.251</b>	0.235	<b>0.228</b>
		VAR <sub>CV</sub>	0.457	<b>0.354</b>	0.316	0.286	0.265	0.251	0.248
Discrete	Indp.	VQ <sub>p=256</sub>	0.459	0.392	0.353	0.327	0.297	0.280	0.279*
		VQ <sub>p=512</sub>	0.459	0.394	0.353	0.322	0.290	0.272	0.269
		VQ <sub>p=1024</sub>	0.463	0.396	0.355	0.320	0.289	0.273	0.271
		Markov <sub>p=8</sub>	0.454	0.372	0.333	0.297	0.269	0.254	0.244
	Temp.	Markov <sub>p=16</sub>	0.450	0.369	0.332	0.299	0.271	0.257	0.251
		Markov <sub>p=24</sub>	0.455	0.371	0.330	0.297	0.270	0.254	0.248
		Markov <sub>p=32</sub>	0.458	0.378	0.338	0.306	0.278	0.263	0.256
		HMM <sub>p=8</sub>	0.461	0.375	0.335	0.297	0.267	0.250	0.246
		HMM <sub>p=16</sub>	0.451	0.370	0.328	0.291	0.256	<b>0.235</b>	0.228
		HMM <sub>p=24</sub>	0.441	0.366	0.328	0.293	0.263	0.245	0.240
		HMM <sub>p=32</sub>	0.460	0.373	0.337	0.299	0.268	0.251	0.247
		Baseline	0.485	0.413	0.396	0.354	0.319	0.290	0.285

**Table 2.** Classification error on the *IMM* dataset applying the pairwise kGLM-L2 model on the **valence** dimension. Results are averages of 20 folds, 13 subjects and 20 repetitions. McNemar paired tests between each model and baseline all result in  $p \ll 0.001$  except for results marked with \* which has  $p > 0.05$  with sample size of 4940.

across all folds in the CV. The quantization of the multi-variate time series, is performed using a standard online K-means algorithm [15]. Due to the inherent difficulty of estimating the number of codewords, we choose a selection specifically (8, 16, 24 and 32) for the Markov and HMM models and (256, 512 and 1024) for the VQ models. We compare results between two major categories, namely with continuous or discretized observation space and whether temporal information is included or not.

The results for the *IMM* dataset for valence are presented in Table 2. For continuous observations we see a clear increase in performance between the Diagonal AR (DAR) model of up to 0.018 and 0.024, compared to traditional Multivariate Gaussian and mean models respectively. With discretized observations, an improvement of performance when including temporal information is again observed of 0.025 comparing the Markov and VQ models. Increasing the complexity of the temporal representation with latent states in the HMM model, an increase of performance is again obtained of 0.016. Predicting the dimension of arousal shown on Table 3, the DAR is again the best performing model using all training data, outperforming the traditional temporal-independent models with 0.015. For discretized data the HMM is the best performing model where we again see that increasing the complexity of the temporal representation increases the predictive performance. Considering the *YANG* dataset, the results are shown in Table 4. Applying the Vector AR models (VAR), a performance gain is again observed compared to the standard representations like e.g. Gaussian or GMM. For discretized data, the temporal aspects again improve the performance, although we do not see a clear picture that increasing the complexity of the temporal representation increases the performance; the selection of the number of hidden states could be an issue here.

Obs.	Time	Models	Training set size						
			1%	5%	10%	20%	40%	80%	100%
Continuous	Indp.	Mean	0.368	0.258	0.230	0.215	0.202	0.190	0.190
		$\mathcal{N}(x \mu, \sigma)$	0.378	0.267	0.241	0.221	0.205	0.190	0.185
		$\mathcal{N}(x \mu, \Sigma)$	0.377	0.301	0.268	0.239	0.216	0.208	0.201
		$GMM_{diag}$	0.390	0.328	0.301	0.277	0.257	0.243	0.236
		$GMM_{full}$	0.367	0.303	0.279	0.249	0.226	0.216	0.215
	Temp.	DAR <sub>CV</sub>	0.411	0.288	0.243	0.216	<b>0.197</b>	<b>0.181</b>	<b>0.170</b>
		VAR <sub>CV</sub>	0.393	0.278	0.238	0.213	0.197	0.183	0.176
Discrete	Indp.	VQ <sub>p=256</sub>	<b>0.351</b>	<b>0.241</b>	<b>0.221</b>	<b>0.208</b>	0.197	0.186	0.183
		VQ <sub>p=512</sub>	0.356	0.253	0.226	0.211	0.199	0.190	0.189
		VQ <sub>p=1024</sub>	0.360	0.268	0.240	0.219	0.200	0.191	0.190
		Markov <sub>p=8</sub>	0.375	0.265	0.238	0.220	0.205	0.194	0.188
	Temp.	Markov <sub>p=16</sub>	0.371	0.259	0.230	0.210	0.197	0.185	0.182
		Markov <sub>p=24</sub>	0.373	0.275	0.249	0.230	0.213	0.202	0.200
		Markov <sub>p=32</sub>	0.374	0.278	0.249	0.229	0.212	0.198	0.192
		HMM <sub>p=8</sub>	0.410	0.310	0.265	0.235	0.211	0.194	0.191
		HMM <sub>p=16</sub>	0.407	0.313	0.271	0.235	0.203	0.185	0.181
		HMM <sub>p=24</sub>	0.369	0.258	0.233	0.215	0.197	0.183	0.181
		HMM <sub>p=32</sub>	0.414	0.322	0.282	0.245	0.216	0.200	0.194
		Baseline	0.483	0.417	0.401	0.355	0.303	0.278	0.269

**Table 3.** Classification error on the *IMM* dataset applying the pairwise kGLM-L2 model on the **arousal** dimension. Results are averages of 20 folds, 13 participants and 20 repetitions. McNemar paired tests between each model and baseline all result in  $p \ll 0.001$  with a sample size of 4940.

Obs.	Time	Models	Training set size						
			1%	5%	10%	20%	40%	80%	100%
Continuous	Indp.	Mean	0.331	0.300	0.283	0.266	0.248	0.235	0.233
		$\mathcal{N}(x \mu, \sigma)$	0.312	0.291	0.282	0.272	0.262	0.251	0.249
		$\mathcal{N}(x \mu, \Sigma)$	0.293	0.277	0.266	0.255	0.241	0.226	0.220
		$GMM_{diag}$	0.302	0.281	0.268	0.255	0.239	0.224	0.219
		$GMM_{full}$	0.293	0.276	0.263	0.249	0.233	0.218	0.214
	Temp.	DAR <sub>p=10</sub>	0.302	0.272	0.262	0.251	0.241	0.231	0.230
		VAR <sub>p=4</sub>	<b>0.281</b>	<b>0.260</b>	<b>0.249</b>	<b>0.236</b>	<b>0.223</b>	<b>0.210</b>	<b>0.206</b>
Discrete	Indp.	VQ <sub>p=256</sub>	0.304	0.289	0.280	0.274	0.268	0.264	0.224
		VQ <sub>p=512</sub>	0.303	0.286	0.276	0.269	0.261	0.254	0.253
		VQ <sub>p=1024</sub>	0.300	0.281	0.271	0.261	0.253	0.245	0.243
		Markov <sub>p=8</sub>	0.322	0.297	0.285	0.273	0.258	0.243	0.238
	Temp.	Markov <sub>p=16</sub>	0.317	0.287	0.272	0.257	0.239	0.224	0.219
		Markov <sub>p=24</sub>	0.314	0.287	0.270	0.252	0.235	0.221	0.217
		Markov <sub>p=32</sub>	0.317	0.292	0.275	0.255	0.238	0.223	0.217
		HMM <sub>p=8</sub>	0.359	0.320	0.306	0.295	0.282	0.267	0.255
		HMM <sub>p=16</sub>	0.354	0.324	0.316	0.307	0.297	0.289	0.233
		HMM <sub>p=24</sub>	0.344	0.308	0.290	0.273	0.254	0.236	0.234
		HMM <sub>p=32</sub>	0.344	0.307	0.290	0.272	0.254	0.235	0.231
		Baseline	0.500	0.502	0.502	0.502	0.503	0.502	0.499

**Table 4.** Classification error on the *YANG* dataset applying the pairwise kGLM-L2 model on the **valence** dimension. Results are averages of 1240 folds and 10 repetitions. McNemar paired test between each model and baseline results in  $p \ll 0.001$ . Sample size of test was 7752.

## 6. DISCUSSION

In essence we are looking for a way of representing an entire track based on the simple features extracted. That is, we are trying to find generative models that can capture meaningful information coded in the features specifically for coding aspects related to the emotions expressed in music.

Results showed that simplifying the observation space using VQ is useful when predicting the arousal data. Introducing temporal coding of VQ features by simple Markov models already provides a significant performance gain, and adding latent dimensions (i.e. complexity) a further gain is obtained. This performance gain can be attributed to the temporal changes in features and potentially hidden structures in the features not coded in each frame of the features but, by their longer term temporal structures, captured by the models.

We see the same trend with the continuous observations, i.e. including temporal information significantly increases

predictive performance. These results are specific for the features used, the complexity, and potentially the model choice might differ if other features were utilized. Future work will reveal if other structures can be found in features that describe different aspects of music; structures that are relevant for describing and predicting aspects regarding emotions expressed in music.

Another consideration when using the generative models is that the entire feature time series is replaced as such by the model, since the distances between tracks are now between the models trained on each of the tracks and not directly on the features<sup>5</sup>. These models still have to be estimated, which takes time, but this can be done offline and provide a substantial compression of the features used.

## 7. CONCLUSION

In this work we presented a general approach for evaluating various track-level representations for music emotion prediction, focusing on the benefit of modeling temporal aspects of music. Specifically, we considered datasets based on robust, pairwise paradigms for which we extended a particular kernel-based model forming a common ground for comparing different track-level representations of music using the probability product kernel. A wide range of generative models for track-level representations was considered on two datasets, focusing on evaluating both using continuous and discretized observations. Modeling both the valence and arousal dimensions of expressed emotion showed a clear gain in applying temporal modeling on both the datasets included in this work. In conclusion, we have found evidence for the hypothesis that a statistically significant gain is obtained in predictive performance by representing the temporal aspect of music for emotion prediction using MFCC's.

## 8. REFERENCES

- [1] J-J. Aucouturier and F. Pachet. Music similarity measures: What's the use? In *3rd International Conference on Music Information Retrieval (ISMIR)*, pages 157–163, 2002.
- [2] C.M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [3] R. D. Bock and J. V. Jones. *The measurement and prediction of judgment and choice*. Holden-day, 1968.
- [4] F. Huszar. A GP classification approach to preference learning. In *NIPS Workshop on Choice Models and Preference Learning*, pages 1–4, 2011.
- [5] V. Imbrasaitė, T. Baltrušaitis, and P. Robinson. Emotion tracking in music using continuous conditional random fields and relative feature representation. In *ICME AAM Workshop*, 2013.
- [6] T. Jebara and A. Howard. Probability Product Kernels. *Journal of Machine Learning Research*, 5:819–844, 2004.
- [7] J. H. Jensen, D. P. W. Ellis, M. G. Christensen, and S. Holdt Jensen. Evaluation of distance measures between gaussian mixture models of mfccs. In *8th International Conference on Music Information Retrieval (ISMIR)*, 2007.
- [8] J. Madsen, B. S. Jensen, and J. Larsen. Predictive modeling of expressed emotions in music using pairwise comparisons. *From Sounds to Music and Emotions*, Springer Berlin Heidelberg, pages 253–277, Jan 2013.
- [9] J. Madsen, B. S. Jensen, J. Larsen, and J. B. Nielsen. Towards predicting expressed emotion in music from pairwise comparisons. In *9th Sound and Music Computing Conference (SMC) Illusions*, July 2012.
- [10] A. Meng, P. Ahrendt, J. Larsen, and L. K. Hansen. Temporal feature integration for music genre classification. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1654–1664, 2007.
- [11] A. Meng and J. Shawe-Taylor. An investigation of feature models for music genre classification using the support vector classifier. In *International Conference on Music Information Retrieval*, pages 604–609, 2005.
- [12] E. M. Schmidt and Y. E. Kim. Modeling musical emotion dynamics with conditional random fields. In *12th International Conference on Music Information Retrieval (ISMIR)*, 2011.
- [13] E. M. Schmidt, J. Scott, and Y. E. Kim. Feature learning in dynamic environments: Modeling the acoustic structure of musical emotion. In *13th International Conference on Music Information Retrieval (ISMIR)*, 2012.
- [14] B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. *Computational Learning Theory*, 2111:416–426, 2001.
- [15] D. Sculley. Web-scale k-means clustering. *International World Wide Web Conference*, pages 1177–1178, 2010.
- [16] K. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, 2009.
- [17] Y. Vaizman, R. Y. Granot, and G. Lanckriet. Modeling dynamic patterns for emotional content in music. In *12th International Conference on Music Information Retrieval (ISMIR)*, pages 747–752, 2011.
- [18] Y-H. Yang and H.H. Chen. Ranking-Based Emotion Recognition for Music Organization and Retrieval. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):762–774, May 2011.
- [19] J. Zhu and T. Hastie. Kernel logistic regression and the import vector machine. In *Journal of Computational and Graphical Statistics*, pages 1081–1088. MIT Press, 2001.

<sup>5</sup> We do note that using a single model across an entire musical track could potentially be over simplifying the representation, in our case only small 15-30-second excerpts were used and for entire tracks some segmentation would be appropriate.

# MUSICAL STRUCTURAL ANALYSIS DATABASE BASED ON GTTM

**Masatoshi Hamanaka**

Kyoto University  
masatosh@kuhp.kyoto-u.ac.jp

**Keiji Hirata**

Future University Hakodate  
hirata@fun.ac.jp

**Satoshi Tojo**

JAIST  
tojo@jaist.ac.jp

## ABSTRACT

This paper, we present the publication of our analysis data and analyzing tool based on the generative theory of tonal music (GTTM). Musical databases such as score databases, instrument sound databases, and musical pieces with standard MIDI files and annotated data are key to advancements in the field of music information technology. We started implementing the GTTM on a computer in 2004 and ever since have collected and publicized test data by musicologists in a step-by-step manner. In our efforts to further advance the research on musical structure analysis, we are now publicizing 300 pieces of analysis data as well as the analyzer. Experiments showed that for 267 of 300 pieces the analysis results obtained by a new musicologist were almost the same as the original results in the GTTM database and that the other 33 pieces had different interpretations.

## 1. INTRODUCTION

For over ten years we have been constructing a musical analysis tool based on the generative theory of tonal music (GTTM) [1, 2]. The GTTM, proposed by Lerdahl and Jackendoff, is one in which the abstract structure of a musical piece is acquired from a score [3]. Of the many music analysis theories that have been proposed [4–6], we feel that the GTTM is the most promising in terms of its ability to formalize musical knowledge because it captures aspects of musical phenomena based on the Gestalt occurring in music and then presents these aspects with relatively rigid rules.

The time-span tree and prolongational trees acquired by GTTM analysis can be used for melody morphing, which generates an intermediate melody between two melodies with a systematic order [7]. It can also be used for performance rendering [8–10] and reproducing music [11] and provides a summarization of the music that can be used as a search representation in music retrieval systems [12].

In constructing a musical analyzer, test data from musical databases is very useful for evaluating and improving the performance of the analyzer. The Essen folk song collection is a database for folk-music research that contains score data on 20,000 songs along with phrase segmentation information and also provides software for processing the data [13]. The Réper-

toire International des Sources Musicales (RISM), an international, non-profit organization with the aim of comprehensively documenting extant musical sources around the world, provides an online catalogue containing over 850,000 records, mostly for music manuscripts [14]. The Variations3 project provides online access to streaming audio and scanned score images for the music community with a flexible access control framework [15], and the Real World Computing (RWC) Music Database is a copyright-cleared music database that contains the audio signals and corresponding standard MIDI files for 315 musical pieces [16,17]. The Digital Archive of Finnish Folk Tunes provides 8613 finish folk song midi files with annotated meta data and Matlab data matrix encoded by midi toolbox [18]. The Codaich contains 20,849 MP3 recordings, from 1941 artists, with high-quality annotations [19], and the Latin Music Database contains 3,227 MP3 files from different music genres [20].

When we first started constructing the GTTM analyzer, however, there was not much data that included both a score and the results of analysis by musicologists. This was due to the following reasons:

- There were no computer tools for GTTM analysis.  
Only a few paper-based analyses of GTTM data had been done because a data-saving format for computer analysis had not yet been defined. We therefore defined an XML-based format for analyzing GTTM results and developed a manual editor for the editing.
- Editing the tree was difficult.  
Musicologists using the manual editor to acquire analysis results need to perform a large number of manual operations. This is because the time-span and prolongational trees acquired by GTTM analysis are binary trees, and the number of combinations of tree structures in a score analysis increases exponentially with the number of notes. We therefore developed an automatic analyzer based on the GTTM.
- There was a lack of musicologists.  
Only a few hundred musicologists can analyze scores by using the GTTM. In order to encourage musicologists to co-operate with expanding the GTTM database, we publicized our analysis tool and analysis data based on the GTTM.
- The music analysis was ambiguous.  
A piece of music generally has more than one interpretation, and dealing with such ambiguity is a major problem when constructing a music analysis database. We performed experiments to compare the different analysis results obtained by different musicologists.



We started implementing our GTTM analyzer on a computer in 2004, immediately began collecting test data produced by musicologists, and in 2009 started publicizing the GTTM database and analysis system. We started the GTTM database with 100 pairs of scores and time-span trees comprising and then added the prolongational trees and chord progression data. At present, we have 300 data sets that are being used for researching music structural analysis [1]. The tool we use for analyzing has changed from its original form. We originally constructed a standalone application for the GTTM-based analysis system, but when we started having problems with bugs in the automatic analyzer, we changed the application to a client-server system.

In experiments we compared the analysis results of two different musicologists, one of whom was the one who provided the initial analysis data in the GTTM database. For 267 of 300 pieces of music the two results were the same, but the other 33 pieces had different interpretations. Calculating the coincidence of the time-spans in those 33 pieces revealed that 233 of the 2310 time-spans did not match.

This rest of this paper is organized as follows. In section 2 we describe the database design policy and data sets, in section 3 we explain our GTTM analysis tool, in section 4 we present the experimental results, and in section 5 we conclude with a brief summary.

## 2. GTTM DATABASE

The GTTM is composed of four modules, each of which assigns a separate structural description to a listener's understanding of a piece of music. Their output is a grouping structure, a metrical structure, a time-span tree, and a prolongational tree (Fig. 1).

The grouping structure is intended to formalize the intuitive belief that tonal music is organized into groups comprising subgroups. The metrical structure describes the rhythmical hierarchy of the piece by identifying the position of strong beats at the levels of a quarter note, half note, one measure, two measures, four measures,

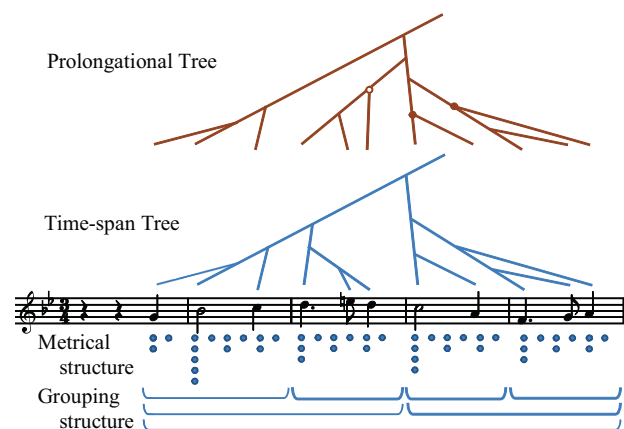


Figure 1. Grouping structure, metrical structure, time-span tree, and prolongational tree.

and so on. The time-span tree is a binary tree and is a hierarchical structure describing the relative structural importance of notes that differentiate the essential parts of the melody from the ornamentation. The prolongational tree is a binary tree that expresses the structure of tension and relaxation in a piece of music.

### 2.1 Design policy of analysis database

As at this stage several rules in the theory allow only monophony, we restrict the target analysis data to monophonic music in the GTTM database.

#### 2.1.1 Ambiguity in music analysis

We have to consider two types of ambiguity in music analysis. One involves human understanding of music and tolerates subjective interpretation, while the latter concerns the representation of music theory and is caused by the incompleteness of a formal theory like the GTTM. We therefore assume because of the former type of ambiguity that there is more than one correct result.

#### 2.1.2 XML-based data structure

We use an XML format for all analysis data. MusicXML [22] was chosen as a primary input format because it provides a common 'interlingua' for music notation, analysis, retrieval, and other applications. We designed GroupingXML, MetricalXML, TimespanXML, and ProlongationalXML as the export formats for our analyzer. We also designed HarmonicXML to express the chord progression. The XML format is suitable for expressing the hierarchical grouping structures, metrical structures, time-span trees, and prolongational trees.

### 2.2 Data sets in GTTM database

The database should contain a variety of different musical pieces, and when constructing it we cut 8-bar-long pieces from whole pieces of music because the time required for analyzing and editing would be too long if whole pieces were analyzed.

#### 2.2.1 Score data

We collected 300 8-bar-long monophonic classical music pieces that include notes, rests, slurs, accents, and articulations entered manually with music notation software called Finale [22]. We exported the MusicXML by using a plugin called Dolet. The 300 whole pieces and the eight bars were selected by a musicologist.

#### 2.2.2 Analysis data

We asked a musicology expert to manually analyze the score data faithfully with regard to the GTTM, using the manual editor in the GTTM analysis tool to assist in editing the grouping structure, metrical structure, time-span tree, and prolongational tree. She also analyzed the chord progression. Three other experts crosschecked these manually produced results.

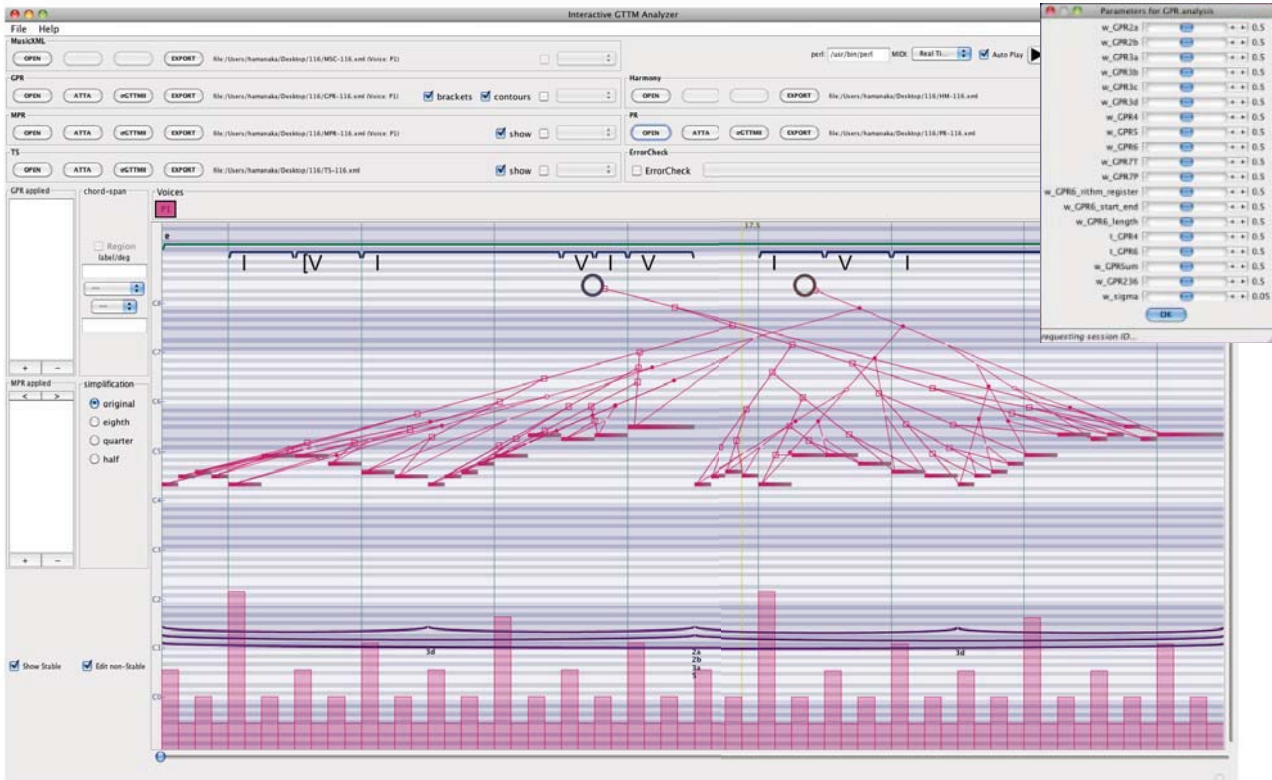


Figure 2. Interactive GTTM analyzer.

### 3. INTERACTIVE GTTM ANALYZER

Our GTTM analysis tool, called the Interactive GTTM analyzer, consists of automatic analyzers and an editor that can be used to edit the analysis results manually (Fig. 2). The graphic user interface of the tool was constructed in Java, making it usable on multiple platforms. However, some functions of the manual editor work only on MacOSX, which must use the MacOSX API.

#### 3.1 Automatic analyzer for GTTM

We have constructed four types of GTTM analyzers: ATTA, FATTA,  $\sigma$ GTTM, and  $\sigma$ GTTMII [2, 23–25]. The Interactive GTTM analyzer can use either the ATTA or the  $\sigma$ GTTMII, and there is a trade-off relationship between the automation of the analysis process and the variation of the analysis results (Fig. 3).

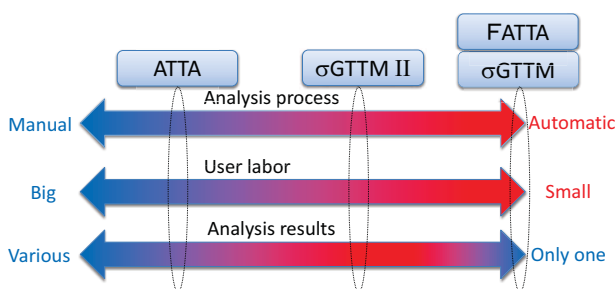


Figure 3. Trade-off between automation of analysis process and variation of analysis results.

#### 3.1.1 ATTA: Automatic Time-Span Tree Analyzer

We extended the original theory of GTTM with a full externalization and parameterization and proposed a machine-executable extension of the GTTM called exGTTM [2]. The externalization includes introducing an algorithm to generate a hierarchical structure of the time-span tree in a mixed top-down and bottom-up manner and the parameterization includes introducing a parameter for controlling the priorities of rules to avoid conflict among the rules as well as parameters for controlling the shape of the hierarchical time-span tree. We implemented the exGTTM on a computer called the ATTA, which can output multiple analysis results by configuring the parameters.

#### 3.1.2 FATTA: Full Automatic Time-Span Tree Analyzer

Although the ATTA has adjustable parameters for controlling the weight or priority of each rule, these parameters have to be set manually. This takes a long time because finding the optimal values of the settings themselves takes a long time. The FATTA can automatically estimate the optimal parameters by introducing a feedback loop from higher-level structures to lower-level structures on the basis of the stability of the time-span tree [23]. The FATTA can output only one analysis result without manual configuration. However, our experimental results showed that the performance of the FATTA is not good enough for grouping structure or time-span tree analyses.

### 3.1.3 $\sigma$ GTTM

We have developed  $\sigma$ GTTM, a system that can detect the local grouping boundaries in GTTM analysis, by combining GTTM with statistical learning [24]. The  $\sigma$ GTTM system statistically learns the priority of the GTTM rules from 100 sets of score and grouping structure data analyzed by a musicologist and does this by using a decision tree. Its performance, however, is not good enough because it can construct only one decision tree from 100 data sets and cannot output multiple results.

### 3.1.4 $\sigma$ GTTM II

The  $\sigma$ GTTM II system assumes that a piece of music has multiple interpretations and thus it constructs multiple decision trees (each corresponding to an interpretation) by iteratively clustering the training data and training the decision trees. Experimental results showed that the  $\sigma$ GTTM II system outperformed both the ATTA and  $\sigma$ GTTM systems [25].

## 3.2 Manual editor for the GTTM

In some cases the GTTM analyzer may produce an acceptable result that reflects the user's interpretation, but in other cases it may not. A user who wants to change the analysis result according to his or her interpretation can use the GTTM manual editor. This editor has numerous functions that can load and save the analysis results, call the ATTA or  $\sigma$ GTTM II analyzer, record the editing history, undo the editing, and autocorrect incorrect structures.

## 3.3 Implementation on client-server system

Our analyzer is updated frequently, and sometimes it is a little difficult for users to download an updated program. We therefore implement our Interactive GTTM analyzer on a client-server system. The graphic user interface on the client side runs as a Web application written in Java, while the analyzer on the server side runs as a program written in Perl. This enables us to update the analyzer frequently while allowing users to access the most recent version automatically.

## 4. EXPERIMENTAL RESULTS

GTTM analysis of a piece of music can produce multiple results because the interpretation of a piece of music is not unique. We compared the different analysis results obtained by different musicologists.

### 4.1 Condition of experiment

A new musicologist who had not been involved in the construction of the GTTM database was asked to manually analyze the 300 scores in the database faithfully with regard to the GTTM. We provided only the 8-bar-long monophonic pieces of music to the musicologist, but she

could refer the original score as needed. When analyzing pieces of music, she could not see the analysis results already in GTTM database. She was told to take however much time she needed, and the time needed for analyzing one song ranged from fifteen minutes to six hours.

## 4.2 Analysis results

Experiments showed that the analysis results for 267 of 300 pieces were the same as the original results in the GTTM database. The remaining 33 pieces had different interpretations, so we added the 33 new analysis results to the GTTM database after they were cross-checked by three other experts.

For those 33 pieces with different interpretations, we found the grouping structure in the database to be the same as the grouping structure obtained by the new musicologist. And for all 33 pieces, in the time-span tree the root branch and branches directly connected to the root branch in the database were the same as the ones in the new musicologist's results.

We also calculated the coincidence of time-spans in both sets of results for those 33 pieces. A time-span tree is a binary tree and each branch of a time-span tree has a time-span. In the ramification of two branches, there is a primary (salient) time-span and secondary (nonsalient) time-span in a parent time-span (Fig. 4). Two time-spans match when the start and end times of the primary and secondary time-spans are the same. We found that 233 of the 2310 time-spans in those 33 pieces of music did not match.

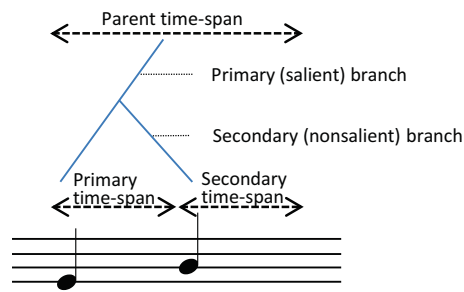


Figure 4. Parent and primary and secondary time-spans.

## 4.3 An example of analysis

"Fuga C dur" composed by Johann Pachelbel had the most unmatched time-spans when the analysis results in the GTTM database (Fig. 5a) were compared with the analysis results by the new musicologist (Fig. 5b). From another musicologist we got the following comments about different analysis results for this piece of music.

### (a) Analysis result in GTTM database

In the analysis result (a), note 2 was interpreted as the start of the subject of the fuga. Note 3 is more salient than note 2 because note 2 is a non-chord tone. Note 5 is the most salient note in the time-span tree of first bar because notes 4 to 7 are a fifth chord and note 5 is a tonic of the chord. The reason that note 2 was interpreted as



the start of the subject of the fuga is uncertain, but a musicologist who is familiar with music before the Baroque era should be able to see that note 2 is the start of the subject of the fuga.

(b) Analysis result by the musicologist

The analysis result (b) was a more simple interpretation than (a) that note 1 is the start of the subject of the fuga. However, it is curious that the trees of second and third beats of the third bar are separated, because both are the fifth chord.

The musicologist who made this comment said that it is difficult to analyze a monophonic piece of music from the contrapuntal piece of music without seeing other parts. Chord information is necessary for GTTM analysis, and a musicologist who is using only a monophonic piece of music has to imagine other parts. This imagining results in multiple interpretations.

## 5. CONCLUSION

We described the publication of our Interactive GTTM analyzer and the GTTM database. The analyzer and database can be downloaded from the following website:

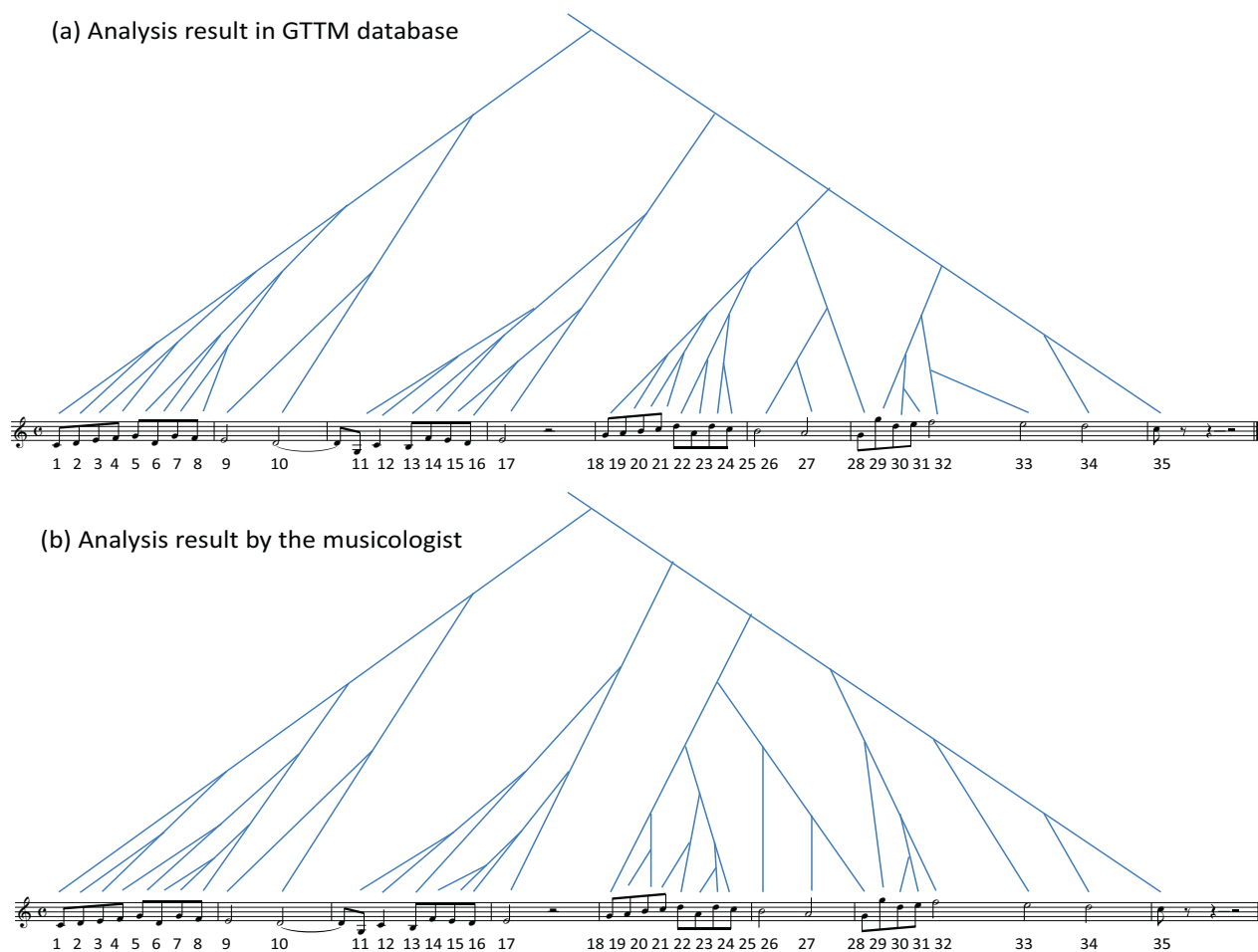
<http://www.gttm.jp/>

The GTTM database has the analysis data for the three hundred monophonic music pieces. Actually, the manual editor in our Interactive GTTM analyzer enables one to deal with polyphonic pieces. Although the analyzer itself works only on monophonic pieces, a user can analyze polyphonic pieces by using the analyzers's manual editor to divide polyphonic pieces into monophonic parts. We also attempted to extend the GTTM framework to enable the analysis of polyphonic pieces [23]. We plan to publicize a hundred pairs of polyphonic score and musicologists' analysis results.

Although the 300 pieces in the current GTTM database are only 8 bars long, we also plan to analyse whole pieces of music by using the analyzer's slide bar for zooming piano roll scores and GTTM structures.

## 6. REFERENCES

- [1] M. Hamanaka, K. Hirata, and S. Tojo: "Time-Span Tree Analyzer for Polyphonic Music," *10th International Symposium on Computer Music Multidisciplinary Research (CMMR2013)*, October 2013.
- [2] M. Hamanaka, K. Hirata, and S. Tojo: "ATTA: Automatic Time-span Tree Analyzer based on Extended GTTM," *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR2005)*, pp. 358–365, September 2005.
- [3] F. Lerdahl and R. Jackendoff: *A Generative Theory of Tonal Music*. MIT Press, Cambridge, 1983.
- [4] G. Cooper and L. Meyer: *The Rhythmic Structure of Music*. University of Chicago Press, 1960.
- [5] E. Narmour: *The Analysis and Cognition of Basic Melodic Structure*. University of Chicago Press, 1990.
- [6] D. Temperley: *The Cognition of Basic Musical Structures*. MIT Press, Cambridge, 2001.
- [7] M. Hamanaka, K. Hirata, and S. Tojo: Melody morphing method based on GTTM, *Proceedings of the 2008 International Computer Music Conference (ICMC2008)*, pp. 155–158, 2008.
- [8] N. Todd: "A Model of Expressive Timing in Tonal Music," *Musical Perception*, 3:1, 33–58, 1985.
- [9] G. Widmer: "Understanding and Learning Musical Expression," *Proceedings of 1993 International Computer Music Conference (ICMC1993)*, pp. 268–275, 1993.
- [10] K. Hirata, and R. Hiraga: "Ha-Hi-Hun plays Chopin's Etude," *Working Notes of IJCAI-03 Workshop on Methods for Automatic Music Performance and their Applications in a Public Rendering Contest*, pp. 72–73, 2003.
- [11] K. Hirata and S. Matsuda: "Annotated Music for Retrieval, Reproduction, and Sharing," *Proceedings of 2004 International Computer Music Conference (ICMC2004)*, pp. 584–587, 2004.
- [12] K. Hirata and S. Matsuda: "Interactive Music Summarization based on Generative Theory of Tonal Music," *Journal of New Music Research*, 32:2, 165–177, 2003.
- [13] H. Schaffrath: The Essen associative code: A code for folksong analysis. In E. Selfridge-Field (Ed.), *Beyond MIDI: The Handbook of Musical Codes*, Chapter 24, pp. 343–361. MIT Press, Cambridge, 1997.
- [14] RISM: International inventory of musical sources. In Series A/II Music manuscripts after 1600, K. G. Saur Verlag, 1997.
- [15] J. Riley, C. Hunter, C. Colvard, and A. Berry: Definition of a FRBR-based Metadata Model for the Indiana University Variations3 Project, <http://www.dlib.indiana.edu/projects/variations3/docs/v3FRBRreport.pdf>, 2007.
- [16] M. Goto: Development of the RWC Music Database, *Proceedings of the 18th International Congress on Acoustics (ICA 2004)*, pp. I-553–556, April 2004.



**Figure 5.** Time-span trees of "Fuga C dur" composed by Johann Pachelbel.

- [17] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka: RWC Music Database: Popular, Classical, and Jazz Music Databases, *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pp. 287–288, October 2002.
- [18] T. Eerola and, P. Toivainen: The Digital Archive of Finnish Folk Tunes, University of Jyväskylä, Available online at: <http://www.jyu.fi/musica/sks>, 2004.
- [19] C. McKay, D. McEnnis, and I. Fujinaga: A large publicly accessible prototype audio database for music research, *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, pp. 160–164, October 2006.
- [20] N. Carlos, L. Alessandro, and A. Celso: The Latin Music Database, *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR2008)*, pp. 451–456, September 2008.
- [21] E. Acotto: Toward a formalization of musical relevance, in B. Kokinov, A. Karmiloff-Smith, and J. Nersessian (Eds.), *European Perspectives on Cognitive Science*, New Bulgarian University Press, 2011.
- [22] MakeMusic Inc.: Finale, Available online at: <http://www.finalemusic.com/>, 2014.
- [23] M. Hamanaka, K. Hirata, and S. Tojo: FATTA: Full Automatic Time-span Tree Analyzer, *Proceedings of the 2007 International Computer Music Conference (ICMC2007)*, Vol. 1, pp. 153–156, August 2007.
- [24] Y. Miura, M. Hamanaka, K. Hirata, and S. Tojo: Use of Decision Tree to Detect GTTM Group Boundaries, *Proceedings of the 2009 International Computer Music Conference (ICMC2009)*, pp. 125–128, August 2009.
- [25] K. Kanamori and M. Hamanaka: Method to Detect GTTM Local Grouping Boundaries based on Clustering and Statistical Learning, *Proceedings of 2014 International Computer Music Conference (ICMC2014) joint with the 11th Sound & Music Computing Conference (SMC2014)*, September 2014 (accepted).
- [26] M. Hamanaka, K. Hirata, and S. Tojo: Time-Span Tree Analyzer for Polyphonic Music, *10th International Symposium on Computer Music Multidisciplinary Research (CMMR2013)*, October 2013.

# THEORETICAL FRAMEWORK OF A COMPUTATIONAL MODEL OF AUDITORY MEMORY FOR MUSIC EMOTION RECOGNITION

**Marcelo Caetano**

Sound and Music Computing Group  
INESC TEC, Porto, Portugal  
mcaetano@inesctec.pt

**Frans Wiering**

Dep. Information and Computing Sciences  
Utrecht University, The Netherlands  
f.wiering@uu.nl

## ABSTRACT

The bag of frames (BOF) approach commonly used in music emotion recognition (MER) has several limitations. The *semantic gap* is believed to be responsible for the *glass ceiling* on the performance of BOF MER systems. However, there are hardly any alternative proposals to address it. In this article, we introduce the theoretical framework of a computational model of auditory memory that incorporates temporal information into MER systems. We advocate that the organization of auditory memory places time at the core of the link between musical meaning and musical emotions. The main goal is to motivate MER researchers to develop an improved class of systems capable of overcoming the limitations of the BOF approach and coping with the inherent complexity of musical emotions.

## 1. INTRODUCTION

In the literature, the aim of music emotion recognition (MER) is commonly said to be the development of systems to automatically estimate listeners' emotional response to music [2, 7, 8, 11, 18, 19, 33] or simply to organize or classify music in terms of emotional content [14, 17]. Applications of MER range from managing music libraries and music recommendation systems to movies, musicals, advertising, games, and even music therapy, music education, and music composition [11]. Possibly inspired by automatic music genre classification [28, 29], a typical approach to MER categorizes emotions into a number of classes and applies machine learning techniques to train a classifier and compare the results against human annotations, considered the "ground truth" [14, 19, 28, 32]. Kim *et. al* [14] presented a thorough state-of-the-art review, exploring a wide range of research in MER systems, focusing particularly on methods that use textual information (e.g., websites, tags, and lyrics) and content-based approaches, as well as systems combining multiple feature domains (e.g., features plus text). Commonly, music features are

estimated from the audio and used to represent the music. These features are calculated independently from each other and from their temporal progression, resulting in the bag of frames (BOF) [11, 14] paradigm.

The 'Audio Mood Classification' (AMC) task in MIREX [5, 10] epitomizes the BOF approach to MER, presenting systems whose performance range from 25 % to 70 % (see Table 1). Present efforts in MER typically concentrate on the machine learning algorithm that performs the map in an attempt to break the 'glass ceiling' [1] thought to limit system performance. The perceived musical information that does not seem to be contained in the audio even though listeners agree about its existence, called 'semantic gap' [3, 31], is considered to be the cause of the 'glass ceiling.' However, the current approach to MER has been the subject of criticism [2, 11, 28, 31].

Knowledge about music cognition, music psychology, and musicology is seldom explored in MER. It is widely known that musical experience involves more than mere processing of music features. Music happens essentially in the brain [31], so we need to take the cognitive mechanisms involved in processing musical information into account if we want to be able to model people's emotional response to music. Among the cognitive processes involved in listening to music, memory plays a major role [27]. Music is intrinsically temporal, and time is experienced through memory. Studies [12, 16, 25] suggest that the temporal evolution of the musical features is intrinsically linked to listeners' emotional response to music.

In this article, we speculate that the so called 'semantic gap' [3] is a mere reflection of how the BOF approach misrepresents both the listener and musical experience. Our goal is not to review MER, but rather emphasize the limitations of the BOF approach and propose an alternative model that relies on the organization of auditory memory to exploit temporal information from the succession of musical sounds. For example, BOF MER systems typically encode temporal information in delta and delta-delta coefficients [1], capturing only local instantaneous temporal variations of the feature values. In a previous work [2], we discussed different MER systems that exploit temporal information differently. Here, we take a step further and propose the theoretical framework of a computational model of auditory memory for MER. Our aim is to motivate MER research to bridge the 'semantic gap' and break the so called 'glass ceiling' [1, 3, 31].



© Marcelo Caetano, Frans Wiering.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Marcelo Caetano, Frans Wiering. "Theoretical Framework of a Computational Model of Auditory Memory for Music Emotion Recognition", 15th International Society for Music Information Retrieval Conference, 2014.

The next section discusses the complexity of musical emotions and how this relates to the glass ceiling preventing BOF MER systems to improve their performance as a motivation for proposing a paradigm change. Then, we briefly introduce the model of auditory memory adopted, followed by the proposed framework and considerations about its implementation. Finally, we present the conclusions and discuss future directions of this theoretical work.

## 2. MACHINE LEARNING AND MER

It is generally agreed that music conveys and evokes emotions [9, 13]. In other words, listeners might feel happy listening to a piece or simply perceive it as happy [9]. Research on music and emotions usually investigates the musical factors involved in the process as well as listeners' response to music. There are many unanswered questions [13, 21], such as "which emotions does music express?", "in what context do musical emotions occur?", "how does music express emotions?", and "which factors in music are expressive of emotions?" Researchers need to address controversial issues to investigate these questions. On the one hand, the relevant musical factors, and on the other hand, the *definition* and *measurement* of emotion.

There is evidence [13] of emotional reactions to music in terms of various subcomponents, such as *subjective feeling*, *psychophysiology*, *brain activation*, *emotional expression*, *action tendency*, *emotion regulation* and these, in turn, feature different psychological mechanisms like *brain stem reflexes*, *evaluative conditioning*, *emotional contagion*, *visual imagery*, *episodic memory*, *rhythmic entrainment*, and *musical expectancy*. Each mechanism is responsive to its own combination of information in the music, the listener, and the situation. Among the causal factors that potentially affect listeners' emotional response to music are *personal*, *situational*, and *musical* [21]. Personal factors include age, gender, personality, musical training, music preference, and current mood; situational factors can be physical such as acoustic and visual conditions, time and place, or social such as type of audience, and occasion. Musical factors include genre, style, key, tuning, orchestration, among many others.

Most modern emotion theorists suggest that an emotion episode consists of coordinated changes in three major reaction components: physiological arousal, motor expression, and subjective feeling (the emotion triad). According to this *componential approach to emotion*, we would need to measure physiological changes, facial and vocal expression, as well as gestures and posture along with self-reported feelings using a rich emotion vocabulary to estimate the listeners' emotional response. In MER, the emotional response to music is commonly collected as self-reported annotations for each music track, capturing "subjective feelings" associated or experienced by the listener. Some researchers [9] speculate that musical sounds can effectively cause emotional reactions (via *brain stem reflex*, for example), suggesting that certain music dimensions and qualities communicate similar affective experiences to many listeners. The literature on the emotional effects of

music [9, 13] has accumulated evidence that listeners often agree about the emotions expressed (or elicited) by a particular piece, suggesting that there are aspects in music that can be associated with similar emotional responses across cultures, personal bias or preferences.

It is probably impractical to hope to develop a MER system that could account for all facets of this complex problem. There is no universally accepted model or explanation for the relationship between music and emotions. However, we point out that it is widely known and accepted that MER systems oversimplify the problem when adopting the BOF approach [11]. In this context, we propose a theoretical framework that uses the organization of auditory memory to incorporate temporal information into MER. We argue that time lies at the core of the complex relationship between music and emotions and that auditory memory mediates the processes involved. In what follows, we focus on the link between musical sounds and self-reported subjective feelings associated to them through music listening. In other words, the association between the audio features and perceived emotions.

### 2.1 The Glass Ceiling on System Performance

The performance of music information retrieval (MIR) systems hasn't improved satisfactorily over the years [1, 10] due to several shortcomings. Aucouturier and Pachet [1] used the term 'glass ceiling' to suggest that there is a limitation on system performance at about 65% *R*-precision when using BOF and machine learning in music similarity. Similarly, Huq *et. al* [11] examined the limitations of the BOF approach to MER. They present the results of a systematic study trying to maximize the prediction performance of an automated MER system using machine learning. They report that none of the variations they considered leads to a substantial improvement in performance, which they interpret as a limit on what is achievable with machine learning and BOF.

MIREX [10] started in 2005 with the goal of systematically evaluating state-of-the-art MIR algorithms, promoting the development of the field, and increasing system performance by competition and (possibly) cooperation. MIREX included an "Audio Mood Classification" (AMC) task for the first time in 2007 'inspired by the growing interest in classifying music by moods, and the difficulty in the evaluation of music mood classification caused by the subjective nature of mood' [10]. MIREX's AMC task uses a categorical representation of emotions divided in five classes. These five 'mood clusters' were obtained by analyzing 'mood labels' (user tags) for popular music from the All Music Guide <sup>1</sup>.

The MIREX wiki <sup>2</sup> presents the "Raw Classification Accuracy Averaged Over Three Train/Test Folds" per system. Table 1 summarizes system performance over the years for the MIREX task AMC, showing the minimum, maximum, average, and standard deviation of these values across systems. Minimum performance has steadily

<sup>1</sup> All Music Guide <http://www.allmusic.com/>

<sup>2</sup> [http://www.music-ir.org/mirex/wiki/MIREX\\_HOME](http://www.music-ir.org/mirex/wiki/MIREX_HOME)

**Table 1:** MIREX AMC performance from 2007 to 2013.

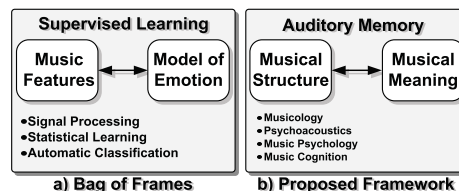
	Minimum	Maximum	Average	STD
2007	25.67%	61.50%	52.65%	11.19%
2008	30.33%	63.67%	52.39%	7.72%
2009	34.67%	65.67%	57.67%	6.17%
2010	36.12%	63.78%	56.48%	6.36%
2011	39.81%	69.48%	57.98%	9.33%
2012	46.14%	67.80%	62.67%	6.17%
2013	28.83%	67.83%	59.81%	10.29%

improved, but maximum performance presents a less significant improvement. The standard deviation of performance across systems has a general trend towards decreasing (suggesting more homogeneity over the years). Most algorithms are also tested in different classification tasks (musical genre, for example), and the best in one task are often also very good at other tasks, maybe indicating there is more machine learning than musical knowledge involved. Sturm [28] discusses the validity of the current evaluation in MER. He argues that the current paradigm of classifying music according to emotions only allows us to conclude how well an MER system can reproduce “ground truth” labels of the test data, irrespective of whether these MER systems use factors irrelevant to emotion in music.

## 2.2 Bridging the Semantic Gap

In MIR, audio processing manipulates signals generated by musical performance, whereas music is an abstract and intangible cultural construct. The sounds *per se* do not contain the essence of music because music exists in the mind of the listener. The very notion of a ‘semantic gap’ is misleading [31]. The current BOF approach to MER views music simply as data (audio signals) and therefore misrepresents musical experience. Machine learning performs a rigid map from “music features” to “emotional labels”, as illustrated in part a) of Fig. 1, treating music as a stimulus that causes a specific emotional response irrespective of personal and contextual factors which are known to affect listeners’ emotional response [12, 16, 25] such as listeners’ *previous exposure* and the impact of the *unfolding musical process*. Memory is particularly important in the recognition of patterns that are either stored in long-term memory (LTM) from previous pieces or in short-term memory (STM) from the present piece. Music seems to be one of the most powerful cues to bring emotional experiences from memory back into awareness.

Wiggins [31] suggests to look at the literature from musicology and psychology to study the cognitive mechanisms involved in human music perception as the starting point of MIR research, particularly *musical memory*, for they define music. He argues that music is not just processed by the listeners, it is defined by them. Wiggins states that “music is a cognitive model”, therefore, only cognitive models are likely to succeed in processing music in a human-like way. He writes that “to treat music in a way which is not human-like is *meaningless*, because music is *defined by humans*. Finally, he concludes that the



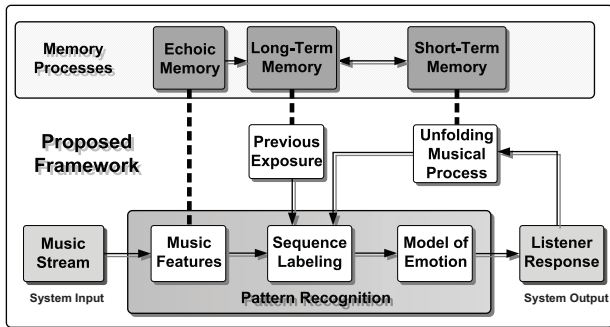
**Figure 1:** Approaches to MER. Part a) illustrates the BOF approach, which uses machine learning to map music features to a region of a model of emotion. Part b) illustrates the proposed approach, which relies on the organization of auditory memory to estimate musical emotions as a form of musical meaning emerging from musical structure.

human response to memory is key to understanding the *psychophysiological effect* of musical stimuli, and that this domain is often missing altogether from MIR research. In this work, we view perceived musical emotions as a particular form of musical meaning [12, 16, 25], which is intimately related to musical structure by the organization of auditory memory [27], as represented in part b) of Fig. 1.

## 3. AUDITORY MEMORY AND MER

Conceptually, memory can be divided into three processes [27]: sensory memory (echoic memory and early processing); short-term memory (or working memory); and long-term memory. Each of these memory processes functions on a different time scale, which can be loosely related to levels of musical experience, the “level of event fusion”, the “melodic and rhythmic level”, and the “formal level”, respectively. Echoic memory corresponds to early processing, when the inner ear converts sounds into trains of nerve impulses that represent the frequency and amplitude of individual acoustic vibrations. During feature extraction, individual acoustic features (e.g., pitch, overtone structure) are extracted and then bound together into auditory events. The events then trigger those parts of long-term memory (LTM) activated by similar events in the past, establishing a context that takes the form of expectations, or memory of the recent past. Long-term memories that are a part of this ongoing context can persist as current “short-term memory” (STM). Short-term memories disappear from consciousness unless they are brought back into the focus of awareness repeatedly (e.g. by means of the rehearsal loop). When the information is particularly striking or novel, it may be passed back to LTM and cause modifications of similar memories already established, otherwise it is lost.

The three types of processing define three basic time scales on which musical events and patterns take place, which, in turn, affect our emotional response to music. The event fusion level of experience (echoic memory) is associated with pitch perception. The main characteristic of the melodic and rhythmic level is that separate events on this time scale are grouped together in the present as melodic grouping and rhythmic grouping, associated with STM. Units on the formal level of musical experience consist of entire sections of music and are associated with



**Figure 2:** The proposed framework for MER. The blocks are system components and the arrows indicate the flow of information. In the shaded area is pattern recognition, and outside are the proposed processes, namely, the “unfolding musical process” and the listener’s “previous exposure”. The figure also illustrates how the organization of auditory memory is related to system blocks.

LTM and our previous musical exposure. Echoic memory and early processing provide our immediate experience of the present moment of music in the focus of conscious awareness, and help to segment it into manageable units; STM establishes the continuity and discontinuity of that movement with the immediate past; and LTM provides the context that gives it meaning, by relating the moment to a larger framework of ongoing experience and previous knowledge. The organization of memory and the limits of our ability to remember have a profound effect on how we perceive patterns of events and boundaries in time. Time is a key element in memory processes and should be brought to the foreground of MER [2].

#### 4. THE PROPOSED FRAMEWORK

Fig. 2 shows the framework we propose to incorporate memory processes in MER systems to illustrate how auditory memory affects musical experience. The blocks associated with the system have a white background, while memory processes have a dark background. The arrows represent the flow of information, while the dashed line represents the relationship between memory processes and system blocks. The proposed framework can be interpreted as an extension of the traditional approach (shaded area) by including two blocks, *previous exposure* and *unfolding musical process*. In the BOF approach, the music features are associated with echoic memory, related to very short temporal scales and uncorrelated with the past or predictions of future events. The framework we propose includes the “Unfolding Musical Process” and “Previous Exposure” to account for LTM and STM. The “Unfolding Musical Process” represents the listeners’ perception of time (related to musical context and expectations), while “Previous Exposure” represents the personal and cultural factors that makes listeners unique.

#### 4.1 Unfolding Musical Process

The *unfolding musical process* uses temporal information from the current music stream to account for repetitions and expectations. As Fig. 2 suggests, the unfolding musical process acts as feedback loop that affects the map between the music features and the listener response. The dynamic aspect of musical emotion relates to the cognition of musical structure [12, 16, 25]. Musical emotions change over time in intensity and quality, and these emotional changes covary with changes in psycho-physiological measures [16, 25]. The human cognitive system regulates our expectations to make predictions [12]. Music (among other stimuli) influences this principle, modulating our emotions. As the music unfolds, the model is used to generate expectations, which are implicated in the experience of listening to music. Musical meaning and emotion depend on how the actual events in the music play against this background of expectations.

#### 4.2 Previous Exposure

The framework in Fig. 2 illustrates that *previous exposure* accounts for musical events stored in LTM that affect the listeners’ emotional response to music. Musical emotions may change according to musical genre [6], cultural background, musical training and exposure, mood, physiological state, personal disposition and taste [9, 12]. This information is user specific and depends on context thus it cannot be retrieved from the current music stream, rather, it has to be supplied by the listener.

### 5. IMPLEMENTATION ISSUES

Here we address how to treat individual components of the model, which parts need human input and which are automatic, and how the different system components communicate and what information they share. The proposed framework urges for a paradigm change in MER research rather than simply a different kind of MER systems, including representing the music stream, collecting time-stamped annotations, and system validation and evaluation [28]. Thus we propose a class of dynamic MER systems that continuously estimate how the listener’s perceived emotions unfold in time from a time-varying input stream of audio features calculated from different musically related temporal levels.

#### 5.1 Music Stream as System Input

The proposed system input is a music stream unfolding in time rather than a static (BOF) representation. To incorporate time into MER, the system should monitor the temporal evolution of the music features [25] at different time scales, the “level of event fusion”, the “melodic and rhythmic level”, and the “formal level”. The feature vector should be calculated for every frame of the audio signal and kept as a time series (i.e., a time-varying vector of features). Time-series analysis techniques such as linear prediction and correlations (among many others) might be used to extract trends and model information at later stages.

## 5.2 Music Features

Eerola [6, 7] proposes to select musically relevant features that have been shown to relate to musical emotions. He presents a list of candidate features for a computational model of emotions that can be automatically estimated from the audio and that would allow meaningful annotations of the music, dividing the features into musically relevant levels related to three temporal scales. Snyder [27] describes three different temporal scales for musical events based on the limits of human perception and auditory memory. Coutinho *et. al* [4] sustain that the structure of affect elicited by music is largely dependent on dynamic temporal patterns in low-level music structural parameters. In their experiments, a significant part of the listeners' reported emotions can be predicted from a set of six psychoacoustic features, namely, loudness, pitch level, pitch contour, tempo, texture, and sharpness. Schubert [26] used loudness, tempo, melodic contour, texture, and spectral centroid as predictors in linear regression models of valence and arousal.

Fig. 1 suggests that MER systems should use the musical structure to estimate musical meaning such as emotions. Musical structure emerges from temporal patterns of music features. In other words, MER systems should include information about the rate of temporal change of music features, such as how changes in loudness correlate with the expression of emotions rather than loudness values only. These loudness variations, in turn, form patterns of repetition on a larger temporal scale related to the structure of the piece that should also be exploited. Thus the features should be hierarchically organized in a musically meaningful way according to auditory memory [27].

## 5.3 Listener Response and System Output

Recently, some authors started investigating how the emotional response evolves in time as the music unfolds. Krumhansl [16] proposes to collect listener's responses continuously while the music is played, recognizing that retrospective judgements are not sensitive to unfolding processes. Recording listener's emotional ratings over time as time-stamped annotations requires listeners to write down the emotional label and a time stamp as the music unfolds, a task that has received attention [20]. Emotions are dynamic and have distinctive temporal profiles that are not captured by traditional models (boredom is very different from astonishment in this respect, for example). In this case, the temporal profiles would be matched against prototypes stored in memory. Some musical websites allow listeners to 'tag' specific points of the waveform (for instance, SoundCloud<sup>3</sup>), a valuable source of temporal annotations for popular music.

## 5.4 Unfolding Musical Process

The *unfolding musical process* acts as feedback loop that exploits the temporal evolution of music features at the three different time scales. The temporal correlation of

each feature must be exploited and fed back to the mapping mechanism (see 'unfolding musical process') to estimate listeners' response to the repetitions and the degree of "surprise" that certain elements might have [26]. Schubert [25] studied the relationship between music features and perceived emotion using continuous response methodology and time-series analysis. Recently, MER systems started tracking temporal changes [4, 22–24, 30]. However, modeling the *unfolding musical process* describes how the time-varying emotional trajectory varies as a function of music features. Korhonen *et al.* [15] use auto-regressive models to predict current musical emotions from present and past feature values, including information about the rate of change or dynamics of the features.

## 5.5 Previous Exposure

*Previous exposure* is responsible for system customization and could use reinforcement learning to alter system response to the *unfolding musical process*. Here, the user input tunes the long-term system behavior according to external factors (independent from temporal evolution of features) such as context, mood, genre, cultural background, etc. Eerola [6] investigated the influence of musical genre on emotional expression and reported that there is a set of music features that seem to be independent of musical genre. Yang *et al.* [33] studied the role of individuality in MER by evaluating the prediction accuracy of *group-wise* and *personalized* MER systems by simply using annotations from a single user as "ground truth" to train the MER system.

## 6. CONCLUSIONS

Research on music emotion recognition (MER) commonly relies on the bag of frames (BOF) approach, which uses machine learning to train a system to map music features to a region of the emotion space. In this article, we discussed why the BOF approach misrepresents musical experience, underplays the role of memory in listeners' emotional response to music, and neglects the temporal nature of music. The organization of auditory memory plays a major role in the experience of listening to music. We proposed a framework that uses the organization of auditory memory to bring time to the foreground of MER. We prompted MER researchers to represent music as a time-varying vector of features and to investigate how the emotions evolve in time as the music develops, representing the listener's emotional response as an emotional trajectory. Finally, we discussed how to exploit the *unfolding music process* and *previous exposure* to incorporate the current musical context and personal factors into MER systems.

The incorporation of time might not be enough to account for the subjective nature of musical emotions. Culture, individual differences and the present state of the listener are factors in understanding aesthetic responses to music. Thus a probabilistic or fuzzy approach could also represent a significant step forward in understanding aesthetic responses to music. We prompt MER researchers to

<sup>3</sup> <http://soundcloud.com/>

adopt a paradigm change to cope with the complexity of human emotions in one of its canonical means of expression, music.

## 7. ACKNOWLEDGEMENTS

This work was partially supported by the Media Arts and Technologies project (MAT), NORTE-07-0124-FEDER-000061, which is financed by the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundao para a Ciéncia e a Tecnologia (FCT). Frans Wiering is supported by the FES project COMMIT/.

## 8. REFERENCES

- [1] J.J. Aucouturier, F. Pachet: "Improving timbre similarity: How high is the sky?" *Journ. Neg. Res. Speech Audio Sci.*, Vol. 1, No. 1, 2004.
- [2] M. Caetano, A. Mouchtaris, F. Wiering: *The role of time in music emotion recognition: Modeling musical emotions from time-varying music features*, LNCS, Springer-Verlag, 2013.
- [3] O. Celma, X. Serra: "FOAFing the music: Bridging the semantic gap in music recommendation," *Journ. Web Semantics* Vol. 6, No. 4, 2008.
- [4] E. Coutinho, A. Cangelosi: "Musical emotions: Predicting second-by-second subjective feelings of emotion from low-level psychoacoustic features and physiological measurements," *Emotion* Vol. 11, No. 4, pp. 921–937, 2011. 2004.
- [5] S. Cunningham, D. Bainbridge, J. Downie: "The impact of MIREX on scholarly research (2005-2010)," *Proc. ISMIR*, 2012.
- [6] T. Eerola: "Are the emotions expressed in music genre-specific? An audio-based evaluation of datasets spanning classical, film, pop and mixed genres," *Journ. New Mus. Res.*, Vol. 40, No. 4, pp. 349–366, 2011.
- [7] T. Eerola: "Modeling listeners' emotional response to music," *Topics Cog. Sci.*, Vol. 4, No. 4, pp. 1–18, 2012.
- [8] A. Friberg: "Digital audio emotions - An overview of computer analysis and synthesis of emotional expression in music," *Proc. DAFx*, 2008.
- [9] A. Gabrielsson, E. Lindström. The role of structure in the musical expression of emotions. *Handbook of Music and Emotion*, Oxford University Press, 2011.
- [10] X. Hu, J. Downie, C. Laurier, M. Bay, and A. Ehmann: "The 2007 MIREX audio mood classification task: Lessons learned," *Proc. ISMIR*, 2008.
- [11] A. Huq, J. Bello, R. Rowe: "Automated music emotion recognition: A systematic evaluation," *Journ. New Mus. Res.*, Vol. 39, No. 3, pp. 227–244, 2010.
- [12] D. Huron: *Sweet Anticipation: Music and the Psychology of Expectation*, Bradford Books, MIT Press, 2008.
- [13] P. Juslin, S. Liljeström, D. Västfjäll, L. Lundqvist: How does music evoke emotions? Exploring the underlying mechanisms. In: *Handbook of Music and Emotion*, Oxford University Press, 2011.
- [14] Y. Kim, E. Schmidt, R. Migneco, B. Morton, P. Richardson, J. Scott, J. Speck, D. Turnbull: "Music emotion recognition: A state of the art review," *Proc. ISMIR*, 2010.
- [15] M. Korhonen, D. Clausi, M. Jernigan: "Modeling Emotional Content of Music Using System Identification," *IEEE Trans. Syst., Man, Cybern.*, Vol. 36, No. 3, pp. 588–599, 2005.
- [16] C. Krumhansl: "Music: A Link Between Cognition and Emotion," *Current Direct. Psychol. Sci.*, Vol. 11, No. 2, pp. 45–50, 2002.
- [17] C. Laurier, M. Sordo, J. Serrà, P. Herrera: "Music mood representations from social tags," *Proc. ISMIR*, 2009.
- [18] L. Lu, D. Liu, H. Zhang: "Automatic mood detection and tracking of music audio signals," *IEEE Trans. Audio, Speech, Lang. Proc.*, Vol. 14, No. 1, pp. 5–18, 2006.
- [19] K. MacDorman, S. Ough, H. Chang: "Automatic emotion prediction of song excerpts: Index construction, algorithm design, and empirical comparison," *Journ. New Mus. Res.*, Vol. 36, No. 4, pp. 281–299, 2007.
- [20] F. Nagel, R. Kopiez, O. Grewe, E. Altenmüller: "EMuJoy: Software for continuous measurement of perceived emotions in music," *Behavior Res. Meth.*, Vol. 39, No. 2, pp. 283–290, 2007.
- [21] K. Scherer: "Which emotions can be induced by music? what are the underlying mechanisms? and how can we measure them?" *Journ. New Mus. Res.*, Vol. 33, No. 3, pp. 239–251, 2004.
- [22] E. Schmidt, Y. Kim: "Modeling musical emotion dynamics with conditional random fields," *Proc. ISMIR*, 2011.
- [23] E. Schmidt, Y. Kim: "Prediction of time-varying musical mood distributions from audio," *Proc. ISMIR*, 2010.
- [24] E. Schmidt, Y. Kim: "Prediction of time-varying musical mood distributions using kalman filtering," *Proc. ICMLA*, 2010.
- [25] E. Schubert: "Modeling perceived emotion with continuous musical features," *Music Percep.: An Interdiscipl. Journ.*, Vol. 21, No. 4, pp. 561–585, 2004.
- [26] E. Schubert: "Analysis of emotional dimensions in music using time series techniques," *Context: Journ. Mus. Res.*, Vol. 31, pp. 65–80, 2006.
- [27] B. Snyder: *Music and Memory: An Introduction.*, MIT Press, 2001.
- [28] B. Sturm: "Evaluating music emotion recognition: Lessons from music genre recognition?," *Proc. IEEE ICMEW*, 2013.
- [29] G. Tzanetakis, P. Cook: "Musical Genre Classification of Audio Signals," *IEEE Trans. Speech, Audio Proc.*, Vol. 10, No. 5, pp. 293–302, 2002.
- [30] Y. Vaizman, R. Granot, G. Lanckriet: "Modeling dynamic patterns for emotional content in music," *Proc. ISMIR*, 2011.
- [31] G. Wiggins: "Semantic gap?? Schematic schmap!! Methodological considerations in the scientific study of music," *Proc. Int. Symp. Mult.*, 2009.
- [32] Y. Yang, H. Chen: "Ranking-based emotion recognition for music organization and retrieval," *IEEE Trans. Audio, Speech, Lang. Proc.*, Vol. 19, No. 4, pp. 762–774, 2011.
- [33] Y. Yang, Y. Su, Y. Lin, H. Chen: "Music emotion recognition: the role of individuality," *Proc. HCM*, 2007.



# IMPROVING MUSIC STRUCTURE SEGMENTATION USING LAG-PRIORS

**Geoffroy Peeters**

STMS IRCAM-CNRS-UPMC  
geoffroy.peeters@ircam.fr

**Victor Bisot**

STMS IRCAM-CNRS-UPMC  
victor.bisot@ircam.fr

## ABSTRACT

Methods for music structure discovery usually process a music track by first detecting segments and then labeling them. Depending on the assumptions made on the signal content (repetition, homogeneity or novelty), different methods are used for these two steps. In this paper, we deal with the segmentation in the case of repetitive content. In this field, segments are usually identified by looking for sub-diagonals in a Self-Similarity-Matrix (SSM). In order to make this identification more robust, Goto proposed in 2003 to cumulate the values of the SSM over constant-lag and search only for segments in the SSM when this sum is large. Since the various repetitions of a segment start simultaneously in a self-similarity-matrix, Serra et al. proposed in 2012 to cumulate these simultaneous values (using a so-called structure feature) to enhance the novelty of the starting and ending time of a segment. In this work, we propose to combine both approaches by using Goto method locally as a prior to the lag-dimensions of Serra et al. structure features used to compute the novelty curve. Through a large experiment on RWC and Isophonics test-sets and using MIREX segmentation evaluation measure, we show that this simple combination allows a large improvement of the segmentation results.

## 1. INTRODUCTION

Music structure segmentation aims at estimating the large-scale temporal entities that compose a music track (for example the verse, chorus or bridge in popular music). This segmentation has many applications such as browsing a track by parts, a first step for music structure labeling or audio summary generation [15], music analysis, help for advanced DJ-ing.

The method used to estimate the music structure segments (and/or labels) depends on the assumptions made on the signal content. Two assumptions are commonly used [13] [14]. The first assumption considers that the audio signal can be represented as a succession of segments with homogeneous content inside each segment. This assumption is named “homogeneity assumption” and the es-

timization approach named “state approach”. It is closely related to another assumption, named “novelty”, that considers that the transition between two distinct homogeneous segments creates a large “novelty”. The second assumption considers that some segments in the audio signal are repetitions of other ones. This assumption is named “repetition assumption”. In this case the “repeated” segments can be homogeneous or not. When they are not, the approach is named “sequence approach”.

In this paper, we deal with the problem of estimating the segments (starting and ending times) in the case of repeated/ non-homogeneous segments (“sequence” approach).

### 1.1 Related works

Works related to music structure segmentation are numerous. We refer the reader to [13] or [3] for a complete overview on the topic. We only review here the most important works or the ones closely related our proposal.

**Methods relying on the homogeneity or novelty assumption.**

Because homogeneous segments form “blocks” in a time-time-Self-Similarity-Matrix (SSM) and because transitions from one homogeneous segment to the next looks like a checkerboard kernel, Foote [5] proposes in 2000 to convolve the matrix with a 2D-checkerboard-kernel. The result of the convolution along the main diagonal leads to large value at the transition times. Since, an assumption on the segment duration has to be made for the kernel of Foote, Kaiser and Peeters [9] propose in 2013 to use multiple-temporal-scale kernels. They also introduce two new kernels to represent transitions from homogeneous to non-homogeneous segments (and vice versa). Other approaches rely on information criteria (such as BIC, Akaike or GLR) applied to the sequence of audio features. Finally, labeling methods (such as k-means, hierarchical agglomerative clustering of hidden-Markov-model) also inherently allow performing time-segmentation.

**Methods relying on the repetition assumption.** Because repeated segments (when non-homogeneous) form sub-diagonals in a Self-Similarity Matrix (SSM), most methods perform the segmentation by detecting these sub-diagonals in the SSM.

If we denote by  $S(i, j) = S(t_i, t_j)$   $i, j \in [1, N]$  the time-time-SSM between the pairs of times  $t_i$  and  $t_j$ , the time-lag-SSM [1] is defined as  $L(i, l) = L(t_i, l = t_j - t_i)$ , since  $t_j - t_i \geq 0$  the matrix is upper-diagonal. The lag-matrix can be computed using  $L(i, l) = S(i, j = i + l)$  with  $i + l \leq N$ .

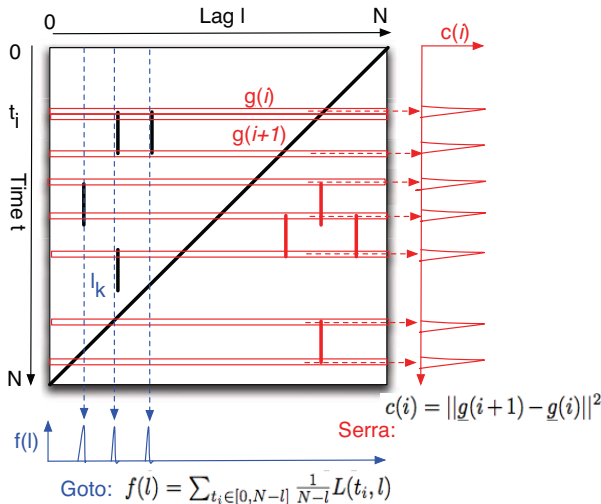


© Geoffroy Peeters, Victor Bisot.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Geoffroy Peeters, Victor Bisot. “Improving Music Structure Segmentation using lag-priors”, 15th International Society for Music Information Retrieval Conference, 2014.

In [6], Goto proposes to detect the repetitions in a time-lag-SSM using a two-step approach. He first detects the various lags  $l_k$  at which potential repetitions may occur. This is done by observing that when a repetition (at the same-speed) occurs, a vertical line (at constant lag) exists in the time-lag-SSM (see Figure 1). Therefore, the sum over the times of the time-lag-SSM for this specific lag will be large. He proposes to compute the function  $f(l) = \sum_{t_i \in [0, N-l]} \frac{1}{N-l} L(t_i, l)$ . A peak in  $f(l)$  indicates that repetitions exist at this specific lag. Then, for each detected peaks  $l_k$ , the corresponding column of  $L(t_i, l_k)$  is analyzed in order to find the starting and ending times of the segments.

Serra et al. method [16] for music structure segmentation also relies in the time-lag-SSM but works in the opposite way. In order to compute the lower-diagonal part of the matrix ( $t_j - t_i < 0$ ), They propose to apply circular permutation. The resulting matrix is named circular-time-lag-matrix (CTLM) and is computed using  $L^*(i, l) = S(i, k + 1)$ , for  $i, l \in [1, N]$  and  $k = i + l - 2$  modulo  $N$ . They then use the fact that the various repetitions of a same segment start and end at the same times in the CTLM. They therefore define a  $N$ -dimensional feature, named "structure feature"  $\underline{g}(i)$ , defined as the row of the CTLM at  $t_i$ . Start and end of the repetitions create large frame-to-frame variations of the structure feature. They therefore compute a novelty curve defined as the distance between successive structure features  $\underline{g}(i)$ :  $c(i) = \|\underline{g}(i+1) - \underline{g}(i)\|^2$  (see Figure 1). Large values in this curve indicate starts or ends times of repetitions.



**Figure 1.** Illustration of Goto method [6] on a time-lag Self-Similarity-Matrix (SSM) and Serra et al. method [16] on a circular-time-lag-matrix (CTLM).

## 1.2 Paper objective and organization

In this paper, we deal with the problem of estimating the segments (starting and ending times) in the case of repeated/ non-homogeneous segments ("sequence" approach). We propose a simple, but very efficient, method that allows using Goto method as a prior lag-probability

of segments in Serra et al. method. Indeed, Serra et al. method works efficiently when the "structure" feature  $\underline{g}(i)$  is clean, i.e. contains large values when a segment crosses  $\underline{g}(i)$  and is null otherwise. Since, this is rarely the case, we propose to create a prior assumption  $f(l)$  on the dimensions of  $\underline{g}(i)$  that may contain segments. To create this prior assumption, we use a modified version of Goto method applied locally in time to the CTLM (instead of to the time-lag-SSM).

Our proposed method for music structure segmentation is presented in part 2. We then evaluate it and compare its performance to state-of-the-art algorithms in part 3 using the RWC-Popular-Music and Isophonics/Beatles test-sets. Discussions of the results and potential extensions are discussed in part 4.

## 2. PROPOSED METHOD

### 2.1 Feature extraction

In order to represent the content of an audio signal, we use the CENS (Chroma Energy distribution Normalized Statistics) features [12] extracted using the Chroma Toolbox [11]. The CENS feature is a sort of quantized version of the chroma feature smoothed over time by convolution with a long duration Hann window. The CENS features  $x_a(t_i)$   $i \in [1, N]$  are 12-dimensional vector with a sampling rate of 2 Hz.  $x_a(t_i)$  is in the range  $[0, 1]$ . It should be noted that these features are  $l^2$ -normed<sup>1</sup>.

### 2.2 Self-Similarity-Matrix

From the sequence of CENS features we compute a time-time Self-Similarity-Matrix (SSM) [4]  $S(i, j)$  using as similarity measure the scalar-product<sup>2</sup> between the feature vector at time  $t_i$  and  $t_j$ :  $S(i, j) = \langle x_a(t_i), x_a(t_j) \rangle$ .

In order to highlight the diagonal-repetitions in the SSM while reducing the influence of noise values, we then apply the following process.

1. We apply a low-pass filter in the direction of the diagonals and high-pass filter in the orthogonal direction. For this, we use the kernel  $[-0.3, 1, -0.3]$  replicated 12 times to lead to a low-pass filter of duration 6 s.

2. We apply a threshold  $\tau \in [0, 1]$  to the resulting SSM.  $\tau$  is chosen such as to keep only  $\beta$  % of the values of the SSM. Values below  $\tau$  are set to a negative penalty-value  $\alpha$ . The interval  $[\tau, 1]$  is then mapped to the interval  $[0, 1]$ .

3. Finally, we apply a median filter over the diagonals of the matrix. For each value  $S(i, j)$ , we look in the backward and forward diagonals of  $\delta$ -points duration each  $[(i - \delta, j - \delta) \dots (i, j) \dots (i + \delta, j + \delta)]$ . If more than 50% of these points have a value of  $\alpha$ ,  $S(i, j)$  is also set to  $\alpha$ .

By experiment, we found  $\beta = 6\%$  (percentage of values kept),  $\alpha = -2$  (lower values) and  $\delta = 10$  frames (interval duration<sup>3</sup>) to be good values.

<sup>1</sup>  $\sum_{a=[1,12]} x_a^2(t_i) = 1$

<sup>2</sup> Since the vectors are  $l^2$ -normed, this is equivalent to the use of a cosine-distance.

<sup>3</sup> Since the sampling rate of  $x_a(t_i)$  is 2 Hz, this corresponds to a duration of 5 s. The median filter is then applied on a window of 10 s total duration.

### 2.3 Proposed method: introducing lag-prior

As mentioned before, Serra et al. method works efficiently when the “structure” feature  $\underline{g}(i)$  is clean, i.e. contains large values when a segment crosses  $\underline{g}(i)$  and is null otherwise. Unfortunately, this is rarely the case in practice.

If we model the structure feature  $\underline{g}(i)$  as the true contribution of the segments  $\hat{\underline{g}}(i)$  and a background noise (modeled as a centered Gaussian noise)  $\mathcal{N}_{\mu=0,\sigma}$ :  $\underline{g}(i) = \hat{\underline{g}}(i) + \mathcal{N}_{\mu=0,\sigma}$ , one can easily show that the expectation of  $c(i) = \|\underline{g}(i+1) - \underline{g}(i)\|^2$  is equal to

- $K + 2\sigma^2$  for the starting/ending of  $K$  segments at  $t_i$
- $2\sigma^2$  otherwise.

If  $\sigma$  (the amount of background noise in the CTLM) is large, then it may be difficult to discriminate between both cases for small  $K$ . In the opposite, the expectation of the values of Goto function  $f(l) = \sum_{t_i} L^*(t_i, l)$  remains independent of  $\sigma$  hence on the presence of background noise (in the case of a centered Gaussian noise).

We therefore propose to use  $f(l)$  as a prior on the lags, i.e. the dimensions of  $\underline{g}(i)$ . This will favor the discrimination provided by  $c(i)$  (in Serra et al. approach, all the lags/dimensions of  $\underline{g}(i)$  are considered equally).

For this, we consider, the circular time-lag (CMLT)  $L^*(t, l)$  as a joint probability distribution  $p(t, l)$ .

Serra et al. novelty curve  $c(i)$  can be expressed as

$$c_1(t) = \int_l \left| \frac{\partial}{\partial t} p(t, l) \right|^2 dl \quad (1)$$

In our approach, we favor the lags at which segments are more likely. This is done using a prior  $p(l)$ :

$$c_2(t) = \int_l p(l) \left| \frac{\partial}{\partial t} p(t, l) \right|^2 dl \quad (2)$$

In order to compute the prior  $p(l)$  we compute  $f(l)$  as proposed by Goto but applied to the CMLT. In other words, we compute, the marginal of  $p(t, l)$  over  $t$ :  $p(l) = \int_{t=0}^{t=N} p(t, l) dt$ .

As a variation of this method, we also propose to compute the prior  $p(l)$  locally on  $t$ :  $p_t(l) = \int_{\tau=t-\Delta}^{\tau=t+\Delta} p(\tau, l) dt$ . This leads to the novelty curve

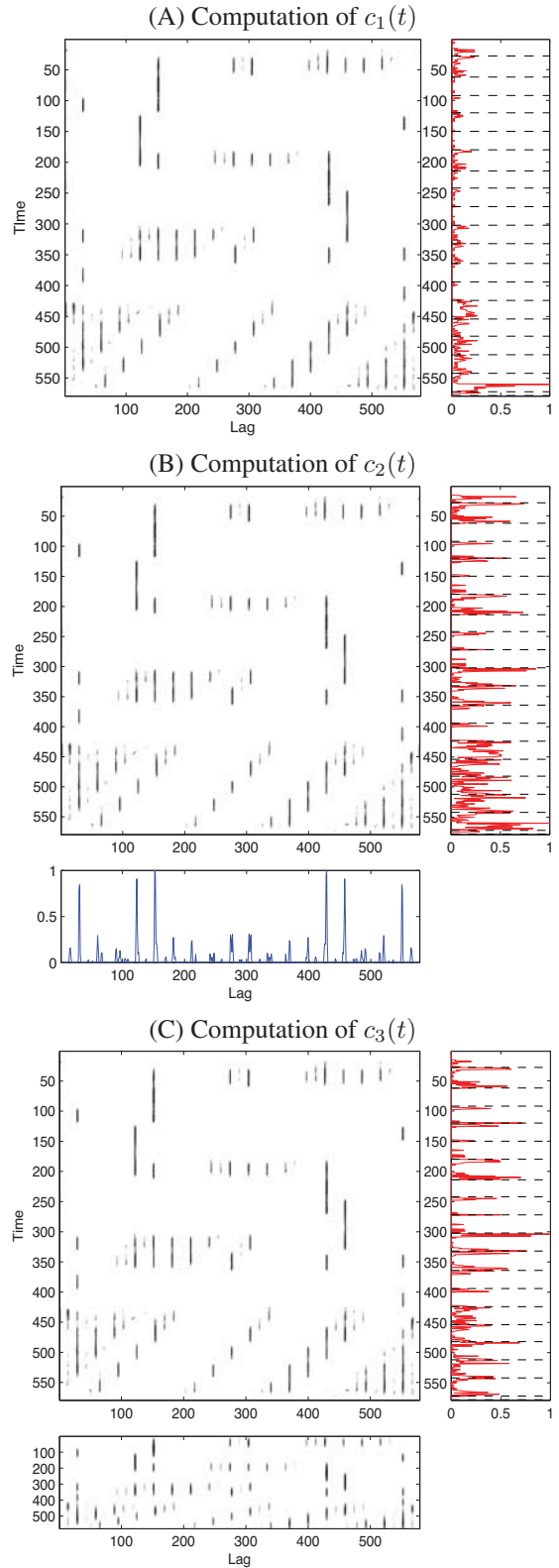
$$c_3(t) = \int_l p_t(l) \left| \frac{\partial}{\partial t} p(t, l) \right|^2 dl \quad (3)$$

By experiment, we found  $\Delta = 20$  (corresponding to 10 s), to be a good value.

### 2.4 Illustrations

In Figure 2, we illustrate the computation of  $c_1(t)$ ,  $c_2(t)$  and  $c_3(t)$  on a real signal (the track 19 from RWC Popular Music).

In Figure 2 (A) we represent Serra et al. [16] method. On the right of the time-lag-circular-matrix (CTLM), we represent the novelty curve  $c_1(t)$  (red-curve) and superimposed to it, the ground-truth segments (black dashed lines).



**Figure 2.** Illustration of the computation of  $c_1(t)$ ,  $c_2(t)$  and  $c_3(t)$  on Track 19 from RWC Popular Music. See text of Section 2.4 for explanation.

In Figure 2 (B) we represent the computation of  $c_2(t)$  (using a global lag-prior). Below the CTLM we represent the global prior  $p(l)$  (blue curve) obtained using Goto method applied to the CMLT. On the right of the CTLM

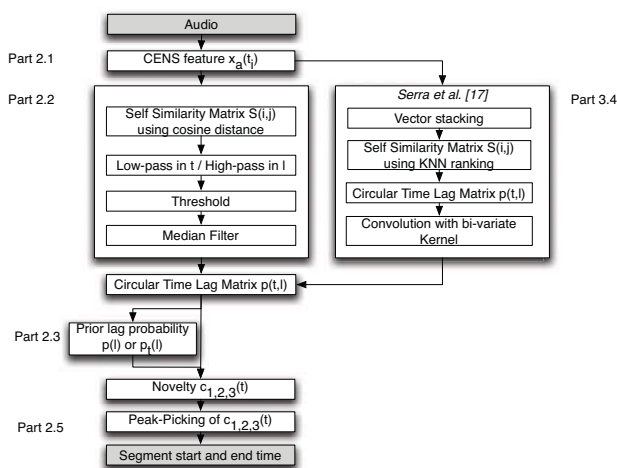
we represent  $c_2(t)$  using this global lag-prior. Compared to the above  $c_1(t)$ , we see that  $c_2(t)$  allows a larger discrimination between times that correspond to ground-truth starts and ends of segments and that do not.

In Figure 2 (C) we represent the computation of  $c_3(t)$  (using a local lag-prior). Below the CTLM we represent the local prior  $p_t(l)$  in matrix form obtained using Goto method applied locally in time to the CMLT. On the right of the CTLM we represent  $c_3(t)$  using this local lag-prior. Compared to the above  $c_1(t)$  and  $c_2(t)$ , we see that  $c_3(t)$  allows an even larger discrimination.

## 2.5 Estimation of segments start and end times

Finally, we estimate the starting and ending time of the repetitions from the novelty curves  $c_1(t)$ ,  $c_2(t)$  or  $c_3(t)$ . This is done using a peak picking process.  $c_*(t)$  is first normalized by min-max to the interval  $[0, 1]$ . Only the values above 0.1 are considered.  $t_i$  is considered as a peak if  $i = \arg \max_j c_*(t_j)$  with  $j \in [i - 10, i + 10]$ , i.e. if  $t_i$  is the maximum peak within a  $\pm 5$  s duration interval.

The flowchart of our Music Structure Segmentation method is represented in the left part of Figure 3.



**Figure 3.** Flowchart of the proposed Music Structure Segmentation method.

## 3. EVALUATION

In this part, we evaluate the performances of our proposed method for estimating the start and end times of music structure segments. We evaluate our algorithm using the three methods described in part 2.3: – without lag-prior  $c_1(t)$  (this is equivalent to the original Serra et al. algorithm although our features and the pre-processing of the CTLM differ from the ones of Serra et al.), – with global lag-prior  $c_2(t)$ , – with local lag-prior  $c_3(t)$ .

### 3.1 Test-Sets

In order to allow comparison with previously published results, we evaluate our algorithm on the following test-sets:

**RWC-Pop-A:** is the RWC-Popular-Music test-set [8], which is a collection of 100 music tracks. The annotations into structures are provided by the AIST [7].

**RWC-Pop-B** is the same test-set but with annotations provided by IRISA [2]<sup>4</sup>.

**Beatles-B** Is the Beatles test-set as part of the Isophonics test-set, which is a collection of 180 music tracks from the Beatles. The annotations into structure are provided by Isophonics [10].

### 3.2 Evaluation measures

To evaluate the quality of our segmentation we use, as it is the case in the MIREX (Music Information Retrieval Evaluation eXchange) Structure Segmentation evaluation task, the Recall (R), Precision (P) and F-Measure (F). We compute those with a tolerance window of 3 and 0.5 s.

### 3.3 Results obtained applying our lag-prior method to the SSM as computed in part 2.2.

In Table 1 we indicate the results obtained for the various configurations and test-sets. We compare our results with the ones published in Serra et al. [16] and to the best score obtained during the two last MIREX evaluation campaign: MIREX-2012 and MIREX-2013 on the same test-sets<sup>5 6</sup>.

For the three test-sets, and a 3 s tolerance window, the use of our lag-prior allows a large increase of the F-measure:

RWC-Pop-A:  $c_1(t)$  : 66.0%,  $c_2(t)$  : 72.9%,  $c_3(t)$  : 76.9%.  
 RWC-Pop-B:  $c_1(t)$  : 67.3%,  $c_2(t)$  : 72.6%,  $c_3(t)$  : 78.2%.  
 Beatles-B:  $c_1(t)$  : 65.7%,  $c_2(t)$  : 69.8%,  $c_3(t)$  : 76.1%.

For the 0.5 s tolerance window, the F-measure also increase but in smaller proportion.

The F-measure obtained by our algorithm is just below the one of [16], but our features and pre-processing of the SSM much simpler. This means that applying our lag-priors to compute  $c_{2,3}(t)$  on Serra et al. pre-processed matrix could even lead to larger results. We discuss this in the next part 3.4. We see that for the two RWC test-sets and a 3 s tolerance window, our algorithm achieves better results than the best results obtained in MIREX (even the ones obtained by Serra et al. – SMGA1). It should be noted that the comparison for the Beatles-B test-set cannot be made since MIREX use the whole Isophonics test-set and not only the Beatles sub-part.

**Statistical tests:** For a @3s tolerance window, the differences of results obtained with  $c_3(t)$  and  $c_2(t)$  are statistically significant (at 5%) for all three test-sets. They are not for a @0.5s tolerance window.

**Discussion:** For the RWC-Pop-B test-set, using  $c_3(t)$  instead of  $c_1(t)$  allows increasing the F@3s for 88/100 tracks, for the Beatles-B for 144/180 tracks. In Figure 4,

<sup>4</sup> These annotations are available at <http://musicdata.gforge.inria.fr/structureAnnotation.html>.

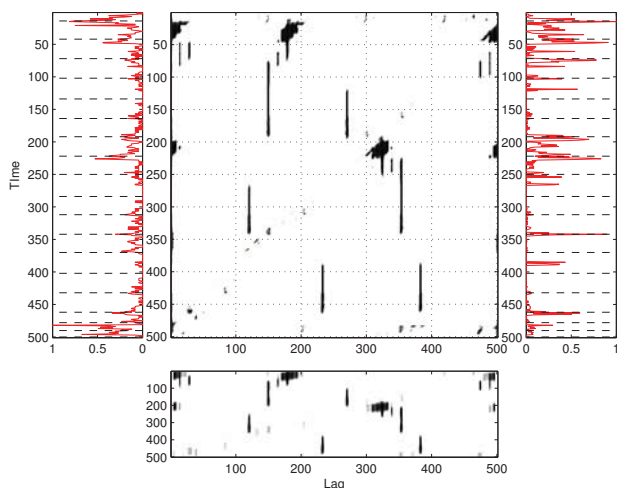
<sup>5</sup> The MIREX test-set named "M-2010 test-set Original" corresponds to RWC-Pop-A, "M-2010 test-set Quaero" to RWC-Pop-B.

<sup>6</sup> SMGA1 stands for [Joan Serra, Meinard Mueller, Peter Grosche, Josep Lluís Arcos]. FK2 stands for [Florian Kaiser and Geoffroy Peeters]. RBH1 stands [Bruno Rocha, Niels Bogaards, Aline Honingh].

**Table 1.** Results of music structure segmentation using our lag-prior method applied to the SSM as computed in part 2.2.

RWC-Pop-A						
Method	F @3s	P @3s	R @3s	F @0.5s	P @0.5s	R @0.5s
Serra et al. [16]	0.791	0.817	0.783			
MIREX-2012 (SMGA1 on M-2010 test-set Original)	0.7101	0.7411	0.7007	0.2359	0.2469	0.2319
MIREX-2013 (FK2 on M-2010 test-set Original)	0.6574	0.8160	0.5599	0.3009	0.3745	0.2562
$c_1(t)$ (without lag-prior)	0.660	0.700	0.648	0.315	0.338	0.308
$c_2(t)$ (with global lag-prior)	0.729	0.739	0.737	0.349	0.354	0.353
$c_3(t)$ (with local lag-prior)	<b>0.769</b>	0.770	0.78	<b>0.386</b>	0.392	0.390
RWC-Pop-B						
Method	F @3s	P @3s	R @3s	F @0.5s	P @0.5s	R @0.5s
Serra et al. [16]	0.8	0.81	0.805			
MIREX-2012 (SMGA1 on M-2010 test-set Quaero)	0.7657	0.8158	0.7352	0.2678	0.2867	0.2558
MIREX-2013 (RBH1 on M-2010 test-set Quaero)	0.6727	0.7003	0.6642	0.3749	0.3922	0.3682
$c_1(t)$ (without lag-prior)	0.673	0.6745	0.689	0.238	0.223	0.263
$c_2(t)$ (with global lag-prior)	0.726	0.704	0.766	0.250	0.231	0.281
$c_3(t)$ (with local lag-prior)	<b>0.782</b>	0.782	0.816	<b>0.281</b>	0.264	0.31
Beatles-B						
Method	F @3s	P @3s	R @3s	F @0.5s	P @0.5s	R @0.5s
Serra et al. [16]	0.774	0.76	0.807			
$c_1(t)$ (without lag-prior)	0.657	0.674	0.658	0.232	0.240	0.238
$c_2(t)$ (with global lag-prior)	0.698	0.696	0.718	0.254	0.258	0.265
$c_3(t)$ (with local lag-prior)	<b>0.761</b>	0.745	0.795	<b>0.262</b>	0.259	0.278

we illustrate one of the examples for which the use of  $c_3(t)$  decreases the results over  $c_1(t)$ . As before the discrimination obtained using  $c_3(t)$  (right sub-figure) is higher than the ones obtained using  $c_1(t)$  (left sub-figure). However, because of the use of the prior  $p_t(l)$  which is computed on a long duration window ( $[t - \Delta, t + \Delta]$  represents 20 s),  $c_3(t)$  favors the detection of long-duration segments. In the example of Figure 4, parts of the annotated segments (black dashed lines) are very short segments which therefore cannot be detected with the chosen duration  $\Delta$  for  $p_t(l)$ .



**Figure 4.** Illustration of a case for which  $c_3(t)$  (right sub-figure) decrease the results over  $c_1(t)$  (left sub-figure).  $F@3s(c_1(t)) = 0.93$  and  $F@3s(c_3(t)) = 0.67$  [Track 20 form RWC-Pop-B].

### 3.4 Results obtained applying our lag-prior method to the SSM as computed by Serra et al. [16]

In order to assess the use of  $c_{2,3}(t)$  as a generic process to improve the estimation of the segments on a SSM; we applied  $c_*(t)$  to the SSM computed as proposed in [16] instead of the SSM proposed in part 2.2. The SSM will be computed using the CENS features instead of the HPCP used in [16]. For recall, in [16] the recent past of the features is taken into account by stacking the feature vectors of past frames (we used a value  $m$  corresponding to 3 s). The SSM is then computed using a K nearest neighbor algorithm (we used a value of  $\kappa = 0.04$ ). Finally the SSM matrix is convolved with a long bivariate rectangular Gaussian kernel  $G = \mathbf{g}_t \mathbf{g}_l^T$  (we used  $s_l = 0.5$  s  $s_t = 30$  s and  $\sigma^2 = 0.16$ ).  $c_*(t)$  is then computed from the resulting SSM. The flowchart of this method is represented in the right part of Figure 3.

Results are given in Table 2 for the various configurations and test-sets.  $c_1(t)$  represents Serra et al. method [16]. As one can see, the use of a global prior ( $c_2(t)$ ) allows to increase the results over  $c_1(t)$  for the three test-sets and the two tolerance window (@3s and @0.5s). Surprisingly, this time, results obtained with a local prior ( $c_3(t)$ ) are lower than the ones obtained with a global prior ( $c_2(t)$ ). This can be explained by the fact that Serra et al. method applies a long duration low-pass filter ( $s_t = 30$ s) to the SSM. It significantly delays in time the maximum value of a segment in the SSM, hence delays  $p_t(l)$ , hence delays  $c_3(t)$ . In the opposite, because  $c_2(t)$  is global, it is not sensitive to Serra et al. delay.

**Statistical tests:** For a @3s tolerance window, the difference of results obtained with  $c_2(t)$  (0.805) and  $c_1(t)$  (0.772) is only statistically significant (at 5%) for the Beatles-B test-set. For a @0.5s tolerance window, the differences are statistically significant (at 5%) for all three test-sets.

**Table 2.** Results of music structure segmentation using our lag-prior method applied to the SSM as computed by [16].

RWC-Pop-A						
Method	F @3s	P @3s	R @3s	F @0.5s	P @0.5s	R @0.5s
$c_1(t)$ (without lag-prior) with Serra front-end	0.780	0.846	0.742	0.254	0.271	0.246
$c_2(t)$ (with global lag-prior) with Serra front-end	<b>0.784</b>	0.843	0.750	<b>0.289</b>	0.316	0.275
$c_3(t)$ (with local lag-prior) with Serra front-end	0.735	0.827	0.682	0.245	0.300	0.215
RWC-Pop-B						
Method	F @3s	P @3s	R @3s	F @0.5s	P @0.5s	R @0.5s
$c_1(t)$ (without lag-prior) with Serra front-end	0.799	0.795	0.818	0.338	0.326	0.359
$c_2(t)$ (with global lag-prior) with Serra front-end	<b>0.823</b>	0.846	0.820	<b>0.389</b>	0.408	0.381
$c_3(t)$ (with local lag-prior) with Serra front-end	0.797	0.856	0.765	0.336	0.369	0.318
Beatles-B						
Method	F @3s	P @3s	R @3s	F @0.5s	P @0.5s	R @0.5s
$c_1(t)$ (without lag-prior) with Serra front-end	0.772	0.792	0.773	0.371	0.365	0.394
$c_2(t)$ (with global lag-prior) with Serra front-end	<b>0.805</b>	0.813	0.817	<b>0.439</b>	0.430	0.450
$c_3(t)$ (with local lag-prior) with Serra front-end	0.799	0.790	0.827	0.422	0.416	0.442

#### 4. CONCLUSION AND FUTURE WORKS

In this paper, we have proposed a simple, but very efficient, method that allows using Goto 2003 method as a prior lag-probability on Serra et al. structure feature method. We provided the rationale for such a proposal, and proposed two versions of the method: one using a global lag prior, one using a local lag prior. We performed a large-scale experiment of our proposal in comparison to state-of-the-art algorithms using three test-sets: RWC-Popular-Music with two sets of annotations and Isophonics/Beatles. We showed that the introduction of the lag-prior allows a large improvement of the F-Measure results (with a tolerance window of 3 s) over the three sets. Also, our method improves over the best results obtained by Serra et al. or during MIREX-2012 and MIREX-2013.

Future works will concentrate on integrating this prior lag probability on an EM (Expectation-Maximization) algorithm to estimate the true  $p(t, l)$ . Also, we would like to use this segmentation as a first step to a segment labeling algorithm.

**Acknowledgements** This work was partly funded by the French government Programme Investissements d’Avenir (PIA) through the Bee Music Project.

#### 5. REFERENCES

- [1] Mark A. Bartsch and Gregory H. Wakefield. To catch a chorus: Using chroma-based representations for audio thumbnailing. In *Proc. of IEEE WASPAA (Workshop on Applications of Signal Processing to Audio and Acoustics)*, pages 15–18, New Paltz, NY, USA, 2001.
- [2] Frédéric Bimbot, Emmanuel Deruty, Sargent Gabriel, and Emmanuel Vincent. Methodology and conventions for the latent semi-otoc annotation of music structure. Technical report, IRISA, 2012.
- [3] Roger Dannenberg and Masataka Goto. Music structure analysis from acoustic signal. In *Handbook of Signal Processing in Acoustics Vol. 1*, pages 305–331. Springer Verlag, 2009.
- [4] Jonathan Foote. Visualizing music and audio using self-similarity. In *Proc. of ACM Multimedia*, pages 77–80, Orlando, Florida, USA, 1999.
- [5] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *Proc. of IEEE ICME (International Conference on Multimedia and Expo)*, pages 452–455, New York City, NY, USA, 2000.
- [6] Masataka Goto. A chorus-section detecting method for musical audio signals. In *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, pages 437–440, Hong Kong, China, 2003.
- [7] Masataka Goto. Aist annotation for the rwc music database. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, pages pp.359–360, Victoria, BC, Canada, 2006.
- [8] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. Rwc music database: Popular, classical, and jazz music databases. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, pages pp. 287–288, Paris, France, 2002.
- [9] Florian Kaiser and Geoffroy Peeters. Multiple hypotheses at multiple scales for audio novelty computation within music. In *Proc. of IEEE ICASSP (International Conference on Acoustics, Speech, and Signal Processing)*, Vancouver, British Columbia, Canada, May 2013.
- [10] Matthias Mauch, Chris Cannam, Matthew Davies, Simon Dixon, Christopher Harte, Sefki Klozali, Dan Tidhar, and Mark Sandler. Omras2 metadata project 2009. In *Proc. of ISMIR (Late-Breaking News)*, Kobe, Japan, 2009.
- [11] Meinard Müller and Sebastian Ewert. Chroma toolbox: Matlab implementations for extracting variants of chroma-based audio features. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Miami, Florida, USA, 2011.
- [12] Meinard Müller, Franz Kurth, and Michael Clausen. Audio matching via chroma-based statistical features. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, London, UK, 2005.
- [13] Jouni Paulus, Meinard Müller, and Anssi Klapuri. Audio-based music structure analysis. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, Utrecht, The Netherlands, 2010.
- [14] Geoffroy Peeters. *Deriving Musical Structures from Signal Analysis for Music Audio Summary Generation: Sequence and State Approach*, pages 142–165. Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg 2004, 2004.
- [15] Geoffroy Peeters, Amaury Laburthe, and Xavier Rodet. Toward automatic music audio summary generation from signal analysis. In *Proc. of ISMIR (International Society for Music Information Retrieval)*, pages 94–100, Paris, France, 2002.
- [16] Joan Serra, Meinard Müller, Peter Grosche, and Josep Ll. Arcos. Unsupervised detection of music boundaries by time series structure features. In *Proc. of AAAI Conference on Artificial Intelligence*, 2012.

# STUDY OF THE SIMILARITY BETWEEN LINGUISTIC TONES AND MELODIC PITCH CONTOURS IN BEIJING OPERA SINGING

Shuo Zhang, Rafael Caro Repetto, Xavier Serra

Music Technology Group, Universitat Pompeu Fabra

ssz6@georgetown.edu, {rafael.caro, xavier.serra}@upf.edu

## ABSTRACT

Features of linguistic tone contours are important factors that shape the distinct melodic characteristics of different genres of Chinese opera. In Beijing opera, the presence of a two-dialectal tone system makes the tone-melody relationship more complex. In this paper, we propose a novel data-driven approach to analyze syllable-sized tone-pitch contour similarity in a corpus of Beijing Opera (381 arias) with statistical modeling and machine learning methods. A total number of 1,993 pitch contour units and attributes were extracted from a selection of 20 arias. We then build Smoothing Spline ANOVA models to compute matrixes of average melodic contour curves by tone category and other attributes. A set of machine learning and statistical analysis methods are applied to 30-point pitch contour vectors as well as dimensionality-reduced representations using Symbolic Aggregate approxXimation(SAX). The results indicate an even mixture of shapes within all tone categories, with the absence of evidence for a predominant dialectal tone system in Beijing opera. We discuss the key methodological issues in melody-tone analysis and future work on pair-wise contour unit analysis.

## 1. INTRODUCTION

Recent development in signal processing and cognitive neuroscience, among other fields, has revived the research on the relationship between speech and musical melody [10]. Singing in tone languages offers a particularly convenient entry point to compare musical and speech melodies, allowing us to gain insight into the ways the prosody of a particular language shapes its music. In a tone language, as opposed to an intonation language, the pitch contour of a speech sound (often a syllable) can be used to distinguish lexical meaning. In singing, however, such pitch contour can be overridden by the melody of the music, making the lyrics difficult to decode by listeners.

In such consideration, musicologists have observed that features of the prosody of the local dialect often play an

important role in shaping the melodic characteristics of the regional operas in China [9, 15]. On the other hand, it is generally assumed that Beijing opera had incorporated linguistic tone systems from both the Hu-Guang (HG) dialect and Beijing (BJ) dialect [22].<sup>1</sup> Xu [19] reviewed 90 years of research on the dialect tone system in Beijing opera, and concluded that there is no agreement as to which system is predominant in shaping the melodic characteristics of the genre.

In sum, previous work indicates that the overall degree and manner of the melody-tone relationship is not entirely clear, partly due to the limitation that music scholars typically were not able to go beyond analyzing a few arias by hand [19]. In this paper, we propose a novel approach to melody-tone similarity by applying statistical modeling and machine learning methods to a set of 20 arias selected from a corpus of 381 arias of Beijing opera audio recording. The research questions are defined as follows: (1) How similar are syllable-sized melodic contours within a given tone category? (2) How similar is the "average" melodic contour to its corresponding prototype contour in speech in the same tone category? (3) Which tone system (BJ or HG) better predicts the shape of melodic contours?

Following preprocessing, we apply clustering algorithms and statistical analysis to 30-point feature vectors of pitch contours, as well as dimensionality-reduced feature vectors represented symbolically using the Symbolic Aggregate approxXimation (SAX) algorithm [8]. Special considerations are given to existing hypotheses regarding the distribution of the tone systems in Beijing opera. Lastly, we build Smoothing Spline ANOVA Models to compute matrixes of average melodic contour curves by tone category and other attributes.

## 2. KEY ISSUES IN STUDYING MELODY-TONE SIMILARITY

### 2.1 Beijing Opera: Performance Practice

Several features of Beijing opera may explain why the melody tone relationship remains challenging. First, the composition process of Beijing opera assumes no designated composer for any given opera. Rather, each opera is composed by re-arranging prototype arias from a inventory of arias according to the rhythmic type, role type, tempo,

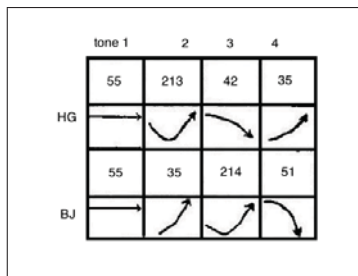
<sup>1</sup> A schematic representation of the four tones in these two systems is shown in Figure 1.



© Shuo Zhang, Rafael Caro Repetto, Xavier Serra .

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Shuo Zhang, Rafael Caro Repetto, Xavier Serra . "STUDY OF THE SIMILARITY BETWEEN LINGUISTIC TONES AND MELODIC PITCH CONTOURS IN BEIJING OPERA SINGING", 15th International Society for Music Information Retrieval Conference, 2014.

and other factors. Lyrics, taken from libretto, are relatively fixed whereas the specific melody may change according to each performer / performance. While there has been evidence [15] that the performers do take special consideration of tones in their singing, it is not clear in what manner and to what degree they are doing this. Next, we discuss several key issues and assumptions as a bridge from previous work to the novel approach proposed in this paper.



**Figure 1.** Schematic representation of the BJ and HG tone system

## 2.2 Key Issues in Studying Tone-Melodic Similarity

First, the melody-tone relationship as a problem of tone perception (and production). A key assumption underlying previous works is that speech comprehension in tone language crucially depends on the availability of tone contour information. However, recent development in tone perception and modeling has challenged this view and revealed the great capacity of human listeners in identifying words in tone languages, with the minimum amount of tone information and despite a great amount of variances in tone signals. We consider the following aspects of evidence: (1) tone contour shapes in connected speech deviates significantly from its standard (canonical) shapes due to co-articulation, limit of maximum speed of pitch change in vocal folds, and other factors [18], introducing a great amount of variances; (2) Patel et al [11] demonstrated that Mandarin speech in monotone is over 90% intelligible to native speakers in a non-noise background, pointing to a low entropy (i.e., high predicability) of the information that is carried by segmental contrast in context; (3) Gating experiments [21] have demonstrated that Mandarin speakers are able to correctly detect tone category based on only the initial fractions of second of a tone signal. From these evidence, we should use caution in making the aforementioned assumption about tone information in music. Similarly, we may also expect to find a even larger amount of variation in the syllable-sized melodic contours in a given tone category.<sup>2</sup>

<sup>2</sup> We must bear in mind also that speech tones are generated under a different mechanism than pitch contours in singing. For one thing, the latter has a more planned mechanism of design - the composition of the music. In speech, as the qTA model has demonstrated [12], speakers may have a pitch target (defined by a linear equation) in mind during articulation, but the actual F0 realization is subject to a set of much complex physiological and contextual linguistic factors, which may be modeled by a third-order critically damped system [12]. This complication does not exist in music: in singing, a singer can realize the exact F0 target as planned. Therefore, we propose that approaches that directly com-

Second, we define the hypotheses and specific goals in this work. We observe that in the review of tone systems in Beijing opera [19], one key assumption is that one of the two underlying dialectal systems must dominate. However, we also find evidence in the literature [22] that one may expect to find an even mixture of contours from both dialects.<sup>3</sup> In this work, we consider both hypotheses and find our data to be more consistent with the second hypothesis.

## 3. DATA COLLECTION

### 3.1 Beijing Opera Audio Data Collection

The music in Beijing opera is mainly structured according to two basic principles, *shengqiang* and *banshi*, which in a broad sense define respectively its melodic and rhythmic components [17]. On top of these two structural principles, the system of role-types impose particular constraints to the execution of *shengqiang* and *banshi*. The interaction of these three components, hence, offers a substantial account of Beijing opera music. Our current collection includes 48 albums, which contain 510 recordings (tracks) featuring 381 arias and over 46 hours of audio [14].

The current study focuses on a small selection of 20 arias from the corpus to serve as a manageable starting point of the melody-tone relationship analysis. This set is selected according to a number of criteria: (1) we selected only *yuanban*, a rhythmic type in which the duration of a syllable sized unit bears the most similarity to that of speech; (2) we selected both types of *shengqiang*, namely *xipi* and *erhuang*; (3) we selected five role types: D(*dan*), J(*jing*), LD(*laodan*), LS(*laosheng*), and XS(*xiaosheng*). For each combination of *shengqiang* and role types, we selected two arias, yielding a total of 20 arias for analysis.

### 3.2 Data Preprocessing

The vocal frames of the audio recordings of the 20 arias are partially-automatically segmented into syllable sized unit with boundary alignment correction by hand.<sup>4</sup> The segmentation is implemented as timestamps of a TextGrid file in the speech processing software Praat [2]. The textgrid is later integrated with the metadata labels from the annotation process.

Following segmentation, we annotate the audio with lyrics extracted from the online Beijing opera libretto database *jingju.net*. The Chinese-character lyrics files are converted into romanized pinyin form with tone marks in the end (1,2,3, or 4) using an implementation of Java library *pinyin4j*. A Praat Script is implemented

pute similarity between melodic and linguistic tone F0 contours should be ruled out.

<sup>3</sup> Some cite three dialects [22], HuGuang, Beijing, and ZhongZhou YinYun.

<sup>4</sup> Automatic segmentation using forced-alignment with machine-readable form of the score is currently being developed. For the current study, we used the result of a trained spectral-based classifier [3] that is able to separate the pure instrumental frames of the audio signal from those frames that contain both vocal and instrumental parts. The result of this segmentation is in many cases the voiced segment (vowel) of a syllable, which is precisely the unit of our analysis.



to automatically parse the romanized lyrics files and to annotate the segmented audio files. The metadata attributes (*shengqiang*, role type, artist, duration, tone category, and word) are also automatically annotated for each segmented unit.

### 3.3 Pitch Contour Extraction

We then proceed to the extraction of F0 values for each annotated pitch contours of interest. The F0 is computed using the MELODIA salience function [13] within the Essentia audio signal processing library in Python [1], in order to minimize the interference of background instrumental ensemble to the computation of F0 of the primary vocal signal. For the sake of analysis, we produce down-sampled 30-point F0 vectors by using equidistant sampling across each pitch contour<sup>5</sup>. All F0 values are normalized so that each contour has a mean F0 of 0 and sd of 1. A 5-point weighted averaging sliding window is applied to smooth the signal.

## 4. PROPOSED APPROACH

In this section we overview the methodology employed in the analysis of the extracted pitch contour dataset. As discussed above in 2.2, all methodology are boiled down to addressing the research question (1), which attempts to analyze and describe the variances and clusters found in melodic contours of each tone category and across categories. Research question (2) and (3), both of which involve comparing music with speech melody, can only be addressed indirectly by the average curves computed by the SSANOVA model for each tone category.

### 4.1 Time Series Representation

In a standard melodic similarity task, such as query-by-humming (QBH), the goal of the task is usually to match the melody as precisely as possible. However, in the current task, our goal is in a way to model the human perception of tone. An important capacity of human cognition is its capacity to abstract away the commonalities from groups of pitch contours with much different fine detail variations<sup>6</sup>. In this study, we experiment with the Symbolic Aggregate appRoXimation (SAX) [8] representation of pitch contour vectors.<sup>7</sup>

SAX offers a lower dimension coarse representation, whose distance lower-bounds true distance of time series. It transforms the pitch contour into a symbolic representation with length ( $n_{seg} = \text{desired length of the feature vector}$ )

<sup>5</sup> The unvoiced part in the beginning of the syllable is skipped in the down-sampling. In addition, the downsampling strategy is also fine tuned in order to filter out the spurious pitch values computed by MELODIA in the beginning portion of the voiced segments.

<sup>6</sup> Also known as categorical perception in cognitive sciences.

<sup>7</sup> SAX is the first symbolic representation for time series that allows for dimensionality reduction and indexing with a lower-bounding distance measure. In classic data mining tasks such as clustering, classification, index, etc., SAX is as good as well-known representations such as DWT and DFT, while requiring less storage space. [8]. Even though SAX representation is mostly used outside of MIR, it has been applied to the QBH task [16].

and alphabet size ( $m$ ) parameters, the latter being used to divide the pitch space of the contour into  $m$  equiprobable segments assuming a Gaussian distribution of F0 values<sup>8</sup>.

In this work, we rely on the SAX representation (1) as a effective and economic way to represent the shapes of time series in statistical analysis; and (2) as a coarse symbolic representation for clustering. To ensure the validity of SAX to reflect the true shape of the original 30-point vector, we experiment with different parameters and use four different ways to evaluate the effectiveness of the SAX representation (discussed below).

### 4.2 Methodology

As discussed in 2.2, we consider two different analytical approaches in this work based on the two hypotheses regarding the distribution of tone systems in Beijing opera.

In the first hypothesis (H1), we assume that there is one predominant tone system (BJ or HG) in Beijing opera. We define a time-series clustering task with the goal of clustering all tone contours into four big clusters, corresponding to four tone categories. Using dynamic time warping (DTW) as the distance measure, we perform K-means Clustering and Agglomerative Clustering (hierarchical) on the 30-point pitch vectors. Using the lower bounding mindist distance measure defined for SAX-based symbolic representation, we also perform K-means Clustering on the SAX string vectors of length 5 (alphabet size is 3).

In the second hypothesis (H2), we expect an even mixture of tone systems and tone shapes in all tone categories. In this scenario, our goal is to perform exploratory cluster analysis on the distribution of contours shapes within each tone categories. More specifically, we perform statistical and clustering analysis on the SAX-based shapes within and across tone categories. In addition, we investigate distribution of attributes associated with each sub-cluster of shape.

We infer from literature [22] that regardless of the distribution of tone systems, the first tone is expected to have the most consistent flat shape if a reasonably strong correlation is assumed between linguistic tone and melodic contour (Notice that tone 1 has the same flat shape across dialects in Figure 1). More specifically, a musicological analysis by hand reveals that the most predominant shape in tone 1 is flat or flat with a final fall (henthforce referred to as Hypothesis 3, or H3, also inferred from rules described in [22]).

Lastly, we build a Smoothing Spline ANOVA model with the goal of (1) computing average pitch contours for each tone category, and (2) quantifying the variances accounted for by each predictor variable in different tone categories. Smoothing splines are essentially a piecewise polynomial function that connects discrete data points called knots. It includes a smoothing parameter to find the

<sup>8</sup> Strictly speaking, the Gaussian assumption is not met in the pitch space musical notes. However, due to the nature of the task that does not require precise mapping, we use the original SAX implementation without revising the Gaussian assumption.

best fit when the data tend to be noisy, estimated by minimizing the following function:

$$G(x) = \frac{1}{n} \sum_{all\ i} (y_i - f(x_i))^2 + \lambda \int_a^b (f''(u))^2 du \quad (1)$$

where  $n$  is the number of data points,  $\lambda$  is the smoothing parameter, and  $a$  and  $b$  are the  $x$  coordinates of the endpoint of the spline.

The Smoothing Spline ANOVA (SSANOVA) is of the following form, each component of  $f$  is estimated with a smoothing spline:

$$f = \mu + \beta x + main\ group\ effect + smooth(x) + smooth(x; group) \quad (2)$$

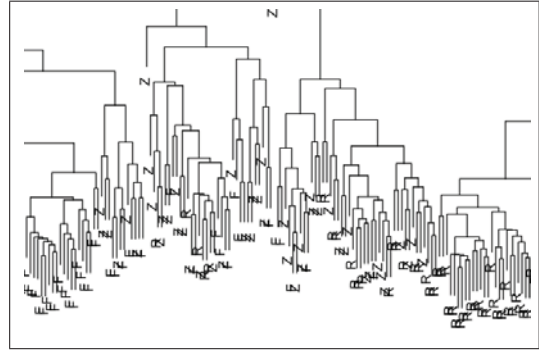
where the main group effects correspond to the smoothing splines for each dataset,  $smooth(x)$  is the single smoothing spline that would be the best fit for all of the data put together, and the interaction term  $smooth(x; group)$  is the smoothing spline representing the difference between a main effect spline and the  $smooth(x)$  spline [4]<sup>9</sup>.

## 5. RESULTS AND DISCUSSION

**Evaluation of SAX representation.** Experimentation with different values of  $nseg$  and alphabet size shows that, in order to capture the abstract nature of tone perception and to minimize the effect of large amount of noise in pitch movements, a limit of  $nseg \leq 3$  must be placed. This is a reasonable limit considering that linguists use only two or three segments to represent tone contours in any tone language<sup>10</sup>. In this work, we use  $nseg=2$  and alphabet size of 3. This choice of parameterization is evaluated as a sufficient representation for the perception of pitch contour shapes in four different ways.

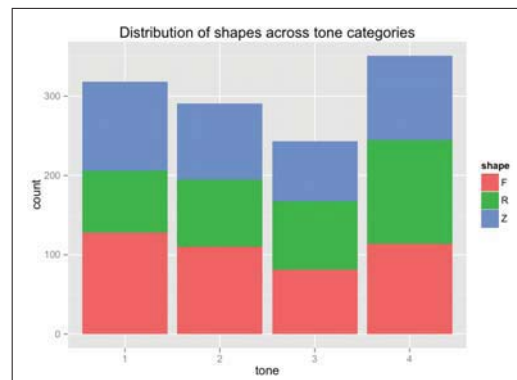
First, a perceptual evaluation is carried out by having a human listener judge the shape of the contours as flat, rising, or falling ( $n=50$ ). The result shows that the SAX representation achieves a 88% accuracy. Second, hierarchical clustering is performed on all contours in a given tone category. The result is then compared with the SAX labels. Figure 2 shows that in addition to meaningful groupings of SAX-labeled shapes, the clustering results also indicate that there are subgroups of shapes within the SAX-shape groups (especially SAX-flat group) that is more similar to falling or rising shapes. Third, we used SAX representation to select a subset of contours from all four tones<sup>11</sup>, and performed a hierarchical clustering task that showed success in separating the four tones. Finally, we performed a classification task, in which the 30-point pitch vectors

from a tone category are classified into SAX class labels with a mean accuracy of 85.2%.



**Figure 2.** Hierarchical clustering on tone 4 with SAX labels (zoomed in), F=Falling, R=Rising, Z=Flat

**Clustering of 4 tones (H1).** Unsupervised K-means Clustering with 30-point vectors cannot learn any meaningful grouping of tone categories regardless of the number of desired clusters (performed in data mining tool Weka [7] with Euclidean distance, and in  $R$  with DTW distance, numOfClust varied within [4,10], otherwise default setting). Likewise, hierarchical clustering with DTW distance cannot find any meaningful groupings of tone labels at any level. This shows that we cannot find a distinct, predominant shape for a given tone category, and failure to cluster melodic contours into meaningful groups that correspond to four tones.



**Figure 3.** Distribution of shapes across five tones, F=Falling, R=Rising, Z=Flat

**Exploratory within-category shape analysis (H2 and H3).** First, we use the validated SAX representations to compute the distribution of three shapes rising(R), falling(F), flat(Z) within each tone category. Figure 3 shows that consistent with H2, each tone category consists of an even mixture of all shapes, with the absence of a dominant shape<sup>12</sup>. To get a more fine-grained analysis of the distributions of shapes, a two-sample test on hypothesis of population proportion is performed across tones and shapes. Results show that the proportion of rising is significantly different across four tones from the proportion

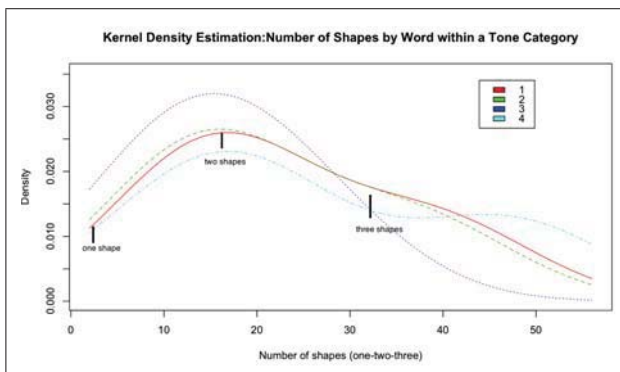
<sup>12</sup> Tone 5 is a neutral tone whose contour shape depends on the tone it precedes it. It exists in our dataset but is not under consideration in the current analysis

<sup>9</sup> The SSANOVA does not return an F value. Instead, the smoothing parameters of the components  $smooth(x)$  and  $smooth(x; group)$  are compared to determine their relative contributions to the equation [4]. In this paper, we use the implementation of  $gss$  package in statistical computing language  $R$ .

<sup>10</sup> In linguistics convention, high tone=H, low tone=L, rising=LH, falling=HL, falling rising=HLH, etc.

<sup>11</sup> Tone1-"bb", tone2-"ac", tone3-"bac", tone4-"ca",  $a < b < c$  in pitch space.

of flat ( $\chi^2 = 17.4065$ ,  $df = 4$ ,  $p = 0.002$ ) or falling ( $\chi^2 = 18.238$ ,  $df = 4$ ,  $p = 0.001$ ). The proportion of flat and falling are not significantly different ( $p = 0.96$ ). Furthermore, a one-sample test show that, only the proportion of rising shape is significantly different across four tones ( $\chi^2 = 21.9853$ ,  $df = 4$ ,  $p = 0.0002$ ), whereas the proportion of flats and fallings are not significantly different across tones ( $p = 0.23$  and  $p = 0.19$ ). Finally, a two-sample pairwise test of hypothesis of population proportion shows that the proportion of rising is significantly different between tone 1 and tone 3 ( $\chi^2 = 7.3403$ ,  $df = 1$ ,  $p = 0.007$ ), tone 1 and tone 4 ( $\chi^2 = 12.1243$ ,  $df = 1$ ,  $p = 0.0005$ ), but not between tone 2, tone 3, tone 4 (except with the difference between tone 2 and tone 4 that reached significance at  $p = 0.04$ ). Therefore, with the exception of tone 1 and tone 2 ( $p=0.22$ , tone 2 seem to behave more similarly to tone 1), the proportion of rising is highly significantly different between in tone 1 and other tones, whereas no strong significant differences are found among other tones. This result supports the H3 discussed above in asserting that tone 1 is mostly consisted of a mixture of flat and falling shapes (to be more specific, flat and flat-falling in H3).



**Figure 4.** Kernel density estimates of number of shapes by word across four tones

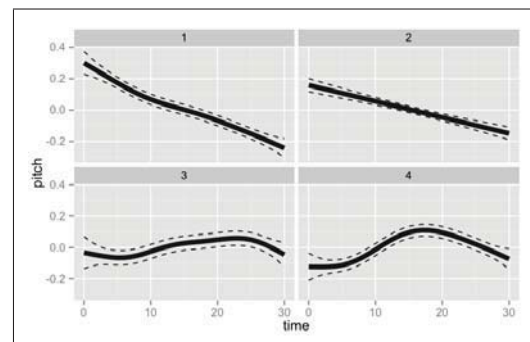
**Analysis of shapes by attributes.** We report the analysis of attributes (artist, word, duration, position, *shengqiang*, *banshi* in the current context) and its correlation with a sub-cluster of shapes within a tone category. First, we performed a classification task using the shape as class label and the listed attributes. Results show that the mean accuracy is around 42%. Second, we analyze the consistency in which a word is sung as a contour shape (a word is defined as a syllable bearing a particular tone) to probe the contour similarity at the word level. Results show that among the words that appear more than once (a mean of 58% of words in our dataset), it is most likely to take on 2 different shapes at different instances, with a lower probability of taking on the same shape or even more different shapes. Figure 4 shows a kernel density estimates of the number of shapes by word in different tones. This result indicates a strong likelihood of inconsistency in singing the same word with the same tone at different contexts.

**SSANOVA.** Results of the SSANOVA models comparison and R-squared values (Table 1) indicate that word

model parameter	levels (nominal)	R-squared (T1)	R-squared (T2)	R-squared (T3)	R-squared (T4)
word	468	0.178	0.0772	0.0566	0.0667
artist	15	0.0885	0.0606	0.0465	0.042
shengqiang	2	0.027	0.0235	0.0154	0.0123
position	4	0.028	0.0211	0.0189	0.0103
role type	5	0.029	0.0273	0.0242	0.018
all	na	0.032	0.028	0.0249	0.201

**Table 1.** SSANOVA Model comparison

and artist are the best predictors of all the predictor variables (as well as all combinations of predictor variables not shown here). However, it is noticeable that the even the best model only explains less than 20% of the variance among all pitch curves in a given tone category<sup>13</sup>. This indicates a large amount of variation in the shape of the contours. On the other hand, the consistently larger value of R-squared for tone 1 indicates positive evidence for a more consistent shape in tone 1, as stated in the H3 discussed above.



**Figure 5.** Average curves computed by the time+word SSANOVA model.

Average curves of four tones are computed based on this model (Figure 5), with confidence intervals shown in dashed lines. The interpretation of these average curves should be done with caution, because of the low R squared value and large standard error in the model. In particular, tone 1 and tone 2 has average contours that differ from both HG and BJ system; tone 3 and tone 4 show resemblance to BJ and HG system, respectively.

## 6. CONCLUSION AND FUTURE WORK

This work constitutes a preliminary step in the computational approaches to the linguistic tone-melodic contour similarity in Beijing opera singing. In this work, we focused on the single-syllable sized contours by adopting different methodologies based on competing hypothesis of tone systems. We have demonstrated the effective-

<sup>13</sup> And also notice that the R-squared value is highly correlated with the number of levels in the nominal attributes.

ness of SAX-based representations in tasks of shape analysis and time-series mining. The results indicate a even mixture of shapes within each tone category, with the absence of a dominant tone system in Beijing opera. In addition, we found evidence supporting the hypothesis that tone 1 is sung with more consistent shape than other tones. Overall, our results point to low degree of similarity in single-syllable pitch contours. Given the discussion and methodology proposed here, we expect future research on pair-wise syllable contour similarity analysis to yield more promising results.

## 7. ACKNOWLEDGEMENT

This research was funded by the European Research Council under the European Union Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583).

## 8. REFERENCES

- [1] Bogdanov, D., Wack N., Gmez E., Gulati S., Herrera P., Mayor O., et al.: ESSENTIA: an Audio Analysis Library for Music Information Retrieval. International Society for Music Information Retrieval Conference (ISMIR'13). 493-498.(2013).
- [2] Boersma, Paul.: Praat, a system for doing phonetics by computer. *Glott International* 5:9/10, 341-345.2001.
- [3] Chen,K.: Characterization of Pitch Intonation of Beijing Opera. Master Thesis, Music Technology Group, Universitat Pompeu Fabra, Barcelona, 2013.
- [4] Davidson, L.: Comparing tongue shapes from ultrasound imaging using smoothing spline analysis of variance. *Journal of the Acoustic Society of America*, 120(1).2006.
- [5] Gauthier,B., Shi,R, Xu,Y.: Learning phonetic categories by tracking movements. *Cognition*, 103, (2007),80-106.
- [6] Gu,C.: Smoothing Spline ANOVA Models. New York: Springer- Verlag.2002.
- [7] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten: The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.2009.
- [8] Lin,J., Keogh,E., Wei,L.,and Lonardi,S.: Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*. Oct.2007, Vol.15, Issue.2, pp107-144.2007.
- [9] Pian,R.C.: Text Setting with the Shipyi Animated Aria. In *Words and Music: The Scholar's View*,edited by Laurence Berman, 237-270. Cambridge: Harvard University Press,1972.
- [10] Patel,A.: *Music, Language, and the Brain*, Oxford Press, 2008.
- [11] Patel, A. D., Xu, Y. and Wang, B.: The role of F0 variation in the intelligibility of Mandarin sentences. In *Proceedings of Speech Prosody 2010*, Chicago.(2010).
- [12] Prom-on, S., Xu, Y. and Thipakorn, B.: Modeling tone and intonation in Mandarin and English as a process of target approximation. *Journal of the Acoustical Society of America* 125: 405-424.(2009).
- [13] Salamon, J and Gomez E: "Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics", *IEEE Transactions on Audio, Speech and Language Processing*, 20(6):1759-1770.2012.
- [14] Serra,X.: "Creating Research Corpora for the Computational Study of Music: the case of the CompMusic Project", AES 53rd International Conference, January 27-29th 2014, London (UK).
- [15] Stock, J: A Reassessment of the Relationship Between Text, Speech Tone, Melody, and Aria Structure in Beijing Opera. *Journal of Musicological Research* (18:3): 183-206. 1999.
- [16] Valero,J: Measuring similarity of automatically extracted melodic pitch contours for audio-based query by humming of polyphonic music collections. Master's Thesis, MTG, DTIC, UPF, 2013.
- [17] Wichmann, E.: *Listening to the theatre: the aural dimension of Beijing opera*. Honolulu: University of Hawaii Press. 1991.
- [18] Xu, Y. and Sun X.: Maximum speed of pitch change and how it may relate to speech. *Journal of the Acoustical Society of America* 111: 1399-1413.2002.
- [19] Xu,Z. 2007: *Jiu shi nian lai jingju de sheng diao yan jiu zhi hui gu*. (Overview of Ninety Years of Research of Tone Relations in Peking Opera). *Nankai Journal of Linguistics*, 10(2):39-50.
- [20] Yu,H.: *Qiang ci guan xi yan jiu*. (Study on the relationship between tone and melody). Beijing: Central Conservatory Publishing House.2009.
- [21] Lai, Yuwen and Jie Zhang. : Mandarin lexical tone recognition: the gating paradigm. In Emily Tummons and Stephanie Lux (eds.), *Proceedings of the 2007 Mid-America Linguistics Conference, Kansas Working Papers in Linguistics* 30. 183-194.2008.
- [22] Zhu, Weiyong: *Xiqu Zuoqu Jifa*.(The Composition Techniques for Chinese Operas.). Beijing: Renmin Yinyue Chubanshe.2004.

# A PROXIMITY GRID OPTIMIZATION METHOD TO IMPROVE AUDIO SEARCH FOR SOUND DESIGN

Christian Frisson, Stéphane Dupont, Willy Yvart, Nicolas Riche, Xavier Siebert, Thierry Dutoit  
numediart Institute, University of Mons, Boulevard Dolez 31, 7000 Mons, Belgium  
{christian.frisson;stephane.dupont;willy.yvart;nicolas.riche;xavier.siebert;thierry.dutoit}@umons.ac.be

## ABSTRACT

Sound designers organize their sound libraries either with dedicated applications (often featuring spreadsheet views), or with default file browsers. Content-based research applications have been favoring cloud-like similarity layouts. We propose a solution combining the advantages of these: after feature extraction and dimension reduction (Student-t Stochastic Neighbor Embedding), we apply a proximity grid, optimized to preserve nearest neighborhoods between the adjacent cells. By counting direct vertical / horizontal / diagonal neighbors, we compare this solution over a standard layout: a grid ordered by filename. Our evaluation is performed on subsets of the One Laptop Per Child sound library, either selected by thematic folders, or filtered by tag. We also compare 3 layouts (grid by filename without visual icons, with visual icons, and proximity grid) by a user evaluation through known-item search tasks. This optimization method can serve as a human-readable metric for the comparison of dimension reduction techniques.

## 1. INTRODUCTION

Sound designers source sounds in massive collections, heavily tagged by themselves and sound librarians. If a set of sounds to compose the desired sound effect is not available, a Foley artist records the missing sound and tags these recordings as accurately as possible, identifying many physical (object, source, action, material, location) and digital (effects, processing) properties. When it comes to looking for sounds in such collections, successive keywords can help the user to filter down the results. But at the end of this process, hundreds of sounds can still remain for further review. This creates an opportunity for content-based information retrieval approaches and other means for presenting the available content. From these observations, we elicited the following research question: can content-based organization complement or outperform context-based organization once a limit is reached when filtering by tag?

This work partly addresses this question and presents a solution to interactively browse collections of textural sounds after these have been filtered by tags.

We organize sounds in a two-dimensional map using content-based features extracted from their signal. These features are mapped to two visual variables. First, the position of the sample on the screen is obtained after applying dimension reduction over the features followed by a proximity grid that structures items on a grid which facilitates navigation and visualization, in particular by reducing the cluttering. The organization of the samples on the grid is optimized using a novel approach that preserves the proximity on the grid of a maximum of nearest neighbors in the original high-dimensional feature space. Second, the shape of the sample is designed to cue one important content-based feature, the perceptual sharpness (a measure the “brightness” of the sound).

This approach is evaluated through a known-item search task. Our experiments provide one of the first positive result quantitatively showing the interest of MIR-based visualization approaches for sound search, when then proper acoustic feature extraction, dimension reduction, and visualization approaches are being used.

The paper is organized as follows. First, in section 2, we examine the landscape of existing systems dedicated to browsing files in sound design. We then describe how we designed our system in Section 3. In section 4, we describe our evaluation approach, experiments and obtained results. We finish by summarizing our contributions and provide an glimpse of future research directions.

## 2. BACKGROUND

This section provides a review of the literature and empirical findings on systems for sound design, and outlines some results and gaps that motivated this work.

Systems for mining sounds, particularly for sound design, are actually rather scarce. These may however share some similarities with systems targeted to the management of music collections, in particular in the content-based processing workflow that allows to organize the audio files. A comprehensive survey on these aspects has been proposed by Casey et al. [4]. We nevertheless believe that the design of the user interface of each system class might benefit from different cues from information visualization and human-computer interaction, and that major progress is still possible in all these areas.



© Christian Frisson, Stéphane Dupont, Willy Yvart, Nicolas Riche, Xavier Siebert, Thierry Dutoit.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Christian Frisson, Stéphane Dupont, Willy Yvart, Nicolas Riche, Xavier Siebert, Thierry Dutoit. “A proximity grid optimization method to improve audio search for sound design”, 15th International Society for Music Information Retrieval Conference, 2014.

## 2.1 Research-grade systems

The work presented in [18] underlines that few published research provide accurate usability evaluations on such systems, beyond informal and heuristic ones. The author justifies that this may have occurred because complementary research communities have actually been evolving essentially in separate silos. These include the music information retrieval and the human-computer interaction communities. In that work, 20 systems with auditory display are nevertheless reviewed and compared, including 2 audio browsers that are presented hereafter.

*Sonic Browser* focused on information visualization [7], and later approached content-based organization through the Marsyas framework [3]. A 2D starfield display allows to map the metadata of audio files to visual variables. Its *HyperTree* view consists in a spring-layout hierarchical graph visualization for browsing the file tree of sound collections. They qualitatively evaluated these views with 15 students through timed tasks and a questionnaire [2]; and their system against the Microsoft Windows 2000 explorer through a think-aloud protocol with 6 students [7].

*SoundTorch*, the most recent content-based audio browser, has been designed by people aware of audio engineering practices [11]. It relies on Mel-Frequency Cepstral Coefficients (MFCCs) as features, clustered with a Self-Organizing Map (SOM) but initialized with smooth gradients rather than randomly, so that the horizontal axis corresponds to a tonal-to-noisy continuum and the vertical axis to pitch increase / dull-to-bright. In addition to cueing in the variety of content through the position of the nodes corresponding to sounds, *SoundTorch* makes use of the node shape to convey additional information: the temporal evolution of the power of the signal is mapped to a circle.

It is the only related work to provide a quantitative user evaluation. They positively evaluated known- and described-item search tasks comparatively to a list-based application. A dozen of users were involved. However, it is not clear from this comparison whether the approach outperforms the list-based application because of its content-based capabilities, or else because of its interactive abilities (particularly its instant playback of closely-located nodes in the map), or both. Moreover, it has been chosen to randomize the sound list. Sound designers either buy commercial sound libraries that are tagged properly and named accordingly, or else record their own. They also usually spend a significant amount of time to tag these libraries. Therefore, to our opinion, a more realistic baseline for comparison should be a basic ordering by filename.

*CataRT* is an application developed in the Max/MSP modular dataflow framework, that “mosaices” sounds into small fragments for concatenative synthesis. A 2D scatter plot allows to browse the sound fragments, assigning features to the axes. The authors recently applied a distribution algorithm that optimizes the spreading of the plotted sounds by means of iterative Delaunay triangulation and a mass-spring model, so as to solve the non-uniform density inherent to a scatter plot, and open new perspectives for non-rectangular interfaces such as the circular *reacTable*

and complex geometries of physical spaces to sonify. To our knowledge, no user study has yet been published for this tool. It is however claimed as future work [12].

In summary, it appears that no evaluation have been proposed previously on the specific contribution of content-based analysis to the efficiency of sound search. This is a gap we started to address in this work.

## 2.2 Commercial systems

It is worth mentioning here that some commercial systems, some making use of content-based approaches, have also been proposed, although no quantitative evaluation of those can be found in the literature. A pioneering application is *SoundFisher* by company Muscle Fish [21], start-up of scientists that graduated in the field of audio retrieval. Their application allowed to categorize sounds along several acoustic features (pitch, loudness, brightness, bandwidth, harmonicity) whose variations over time are estimated by average, variance and autocorrelation. Sounds are compared from the Euclidean distance over these features. The browser offers several views: a detail of sound attributes (filename, samplerate, file size...) in a spreadsheet, a tree of categories resulting from classification by example (the user providing a set of sounds), and a scatter plot to sort sounds along one feature per axis.

A second product, *AudioFinder* by Iced Audio<sup>1</sup> mimics personal music managers such as Apple *iTunes*: on top a textual search input widget allows to perform a query, a top pane proposes a hierarchical view similar to the “column” view of the *Finder* to browse the file tree of the collection, a central view features a spreadsheet to order the results along audio and basic file metadata, a left pane lists saved results like playlists. A bottom row offers waveform visualizations and the possibility to apply audio effect processing to quickly proof the potential variability of the sounds before dropping these into other creative applications.

A major product, *Soundminer HD*<sup>2</sup>, provides a similar interface, plus an alternative layout named *3D LaunchPad* that allows, similarly to the Apple *Finder CoverFlow* view, to browse sounds (songs) by collection (album) cover, with the difference that the former is a 2D grid and the latter a 1D rapid serial visualization technique.

Other companies facilitating creativity such as Adobe with *Bridge*<sup>3</sup> provide more general digital asset management solutions that are accessible through their entire application suite. These focus on production-required capabilities and seem to avoid content-based functionalities.

From our contextual inquiry we noticed that sound designers also make use of simple browsers, such as the default provided by the operating system, optionally associated to a spreadsheet to centralize tags.

<sup>1</sup> <http://www.icedaudio.com>

<sup>2</sup> <http://www.soundminer.com>

<sup>3</sup> <http://www.adobe.com/products/bridge.html>

### 3. OUR SOLUTION

Our system blends knowledge gained from the fields of multimedia information retrieval (content-based organization), human-computer interaction (usability evaluation) and information visualization (visual variables).

#### 3.1 A multimedia information retrieval pipeline

One first step is feature extraction. For sound and music, a large variety of temporal and/or spectral features have been proposed in the literature [4, 15]. We based our features set from [6] since their evaluation considered textural sounds. In short, we used a combination of derivatives of and statistics (standard deviation, skewness and/or kurtosis) over MFCCs and Spectral Flatness (SF). We did not perform segmentation as our test collections contain textural sounds of short length and steady homogeneity.

Another important step is dimension reduction. From our perspective, one of the most promising approach is Stochastic Neighborhood Embedding (SNE) using Student-t distributions (t-SNE) [13]. It has been previously qualitatively evaluated on sound collection visualization [6, 9]. The method has an interesting information retrieval perspective, as it actually aims at probabilistically preserving high-dimensional neighbors in a lower-dimensional projection (2D in our work), and actually maximizes continuity (a measure that can intuitively be related to recall in information retrieval) in the projected space. One emergent result is that recordings from the same sound source with only slight variations are almost always neighbors in the 2D representation, as the recall is high. Another popular but older approach for dimensionality reduction are SOMs. In [14], it has been compared with most recent techniques, and in particular the Neighbor Retrieval Visualizer (NeRV, a generalization of SNE). SOMs produced the most trustworthy (a measure that can intuitively be related to precision in information retrieval) projection but the NeRV was superior in terms of continuity and smoothed recall. As SNE is a special case of NeRV where a tradeoff is set so that only recall is maximized, we infer from those results that SNE is a better approach for our purposes than SOM. Qualitative evaluations of different approaches applied to music retrieval have been undertaken [19]: Multidimensional Scaling (MDS), NeRV and Growing SOMs (GSOM). Users described MDS to result in less positional changes, NeRV to better preserve cluster structures and GSOM to have less overlappings. NeRV and presumably t-SNE seem beneficial in handling cluster structures.

Besides, we propose in this paper an approach to reduce the possible overlappings in t-SNE. An undesirable artifact of the original t-SNE approach however comes from the optimization procedure, which relies on gradient descent with a randomly initialized low-dimensional representation. It creates a stability issue, where several runs of the algorithm may end up in different representations after convergence. This works against the human memory. We thus initialized the low-dimensional representation using the two first axes of a Principal Component Analysis (PCA) of the whole feature set.

#### 3.2 Mapping audio features to visual variables

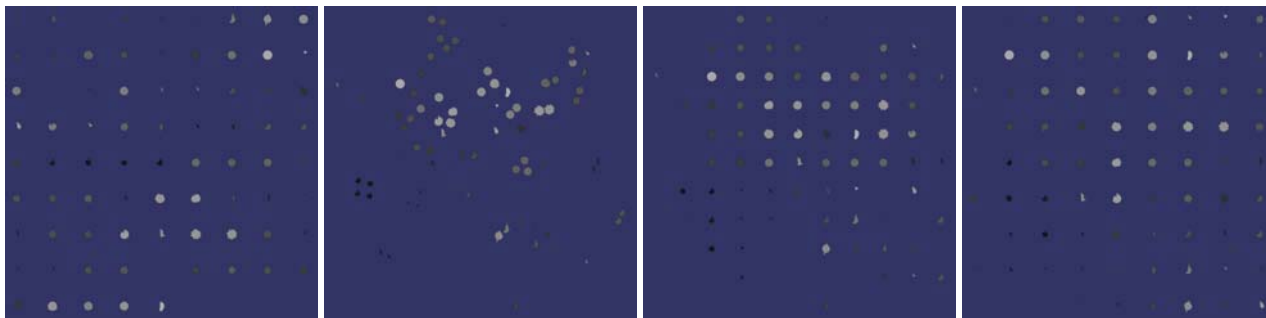
Displaying such a representation results in a scatter plot or starfield display. We address two shortcomings: 1) clusters of similar sounds might not be salient, and 2) this visualization technique may cause overlap in some areas. *SonicBrowser* [7], that we analyzed in the previous section, and the work of Thomas Grill [9], dedicated to textural sounds, approached the first issue by mapping audio features to visual variables. Ware's book [20] offer great explanations and recommendations to use visual variables to support information visualization tailored for human perception. Thomas Grill's approach was to map many perceptual audio features to many visual variables (position, color, texture, shape), in one-to-one mappings.

##### 3.2.1 Content-based glyphs as sound icons

Grill et al. designed a feature-fledged visualization technique mapping perceptual qualities in textural sounds to visual variables [9]. They chose to fully exploit the visual space by tiling textures: items are not represented by a distinct glyph, rather by a textured region. In a first attempt to discriminate the contribution of information visualization versus media information retrieval in sound browsing, we opted here for a simpler mapping. We mapped the mean over time of perceptual sharpness to the value in the Hue Saturation Value (HSV) space of the node color for each sound, normalized against the Values for all sounds in each collection. A sense of brightness is thus conveyed in both the audio and visual channels through perceptual sharpness and value. We also used the temporal evolution of perceptual sharpness to define a clockwise contour of the nodes, so that sounds of similar average brightness but different temporal evolution could be better discriminated. To compute positions, perceptual sharpness was also added to the feature selection, intuiting it would gather items that are similar visually. The choice of perceptual sharpness was motivated by another work of Grill et al. [10]: they aimed at defining features correlated to perceived characteristics of sounds that can be named or verbalized through *personal constructs*. *High-low*, or brightness of the sound, was the construct the most correlated to an existing feature: perceptual sharpness.

##### 3.2.2 A proximity grid optimizing nearest neighbors

For the removal of clutter in 2D plots, two major approaches exist: reducing the number of items to display, or readjusting the position of items. In our context, we want to display all the items resulting of search queries by tag filtering. For this purpose, we borrow a method initially designed to solve the problem of overlap for content-based image browsing [16]: a proximity grid [1]. Their work is heavily cited respectively for the evaluation of multidimensional scaling techniques [1] and as a pioneering application of usability evaluation for multimedia information retrieval [16], but almost never regarding the proximity grid. To our knowledge, no audio or music browser approached this solution.



**Figure 1.** Different layouts with glyphs for the same sound collection filtered by keyword “water”, from left to right: “album”, “cloud”, “metro”, and most dense proximity grid.

A proximity grid consists in adapting the coordinates of each item of a 2D plot to magnetize these items on an evenly-distributed grid. Basalaj proposed several variants to compute a proximity grid: greedy methods with spiral search to find empty cells and *empty/swap/bump* strategies to assign items to cells; an *improved greedy* method replacing spiral search by shortest distance estimation; a “*squeaky wheel*” optimization using simulated annealing, and a *genetic algorithm* [1]. We implemented the simplest greedy method with all strategies. To determine the order of the items to assign to cells, we used the fast minimum spanning tree algorithm implementation from the machine learning library *mlpack* of Boruvka’s dual-tree based on  $k$ -dimensional trees [5]. Applied in high dimension of the audio features, the empty strategy starts with shortest edges while it is the opposite for swap and bump strategies, according to Basalaj. We opted for a simplification: a spiral search always turning clockwise and starting above the desired cell, while it is recommended to choose the rotation and first next cell from exact distance computation between the actual coordinates of the node and the desired cell.

The minimal side of a square grid is the ceil of the square root of the collection size, providing the most space efficient density. To approximate a least distorted grid, the collection size can be taken as grid side. To come up with a tradeoff between density and neighborhood preservation, we estimate the number of high-dimensional nearest neighbors ( $k=1$ ) preserved in 2D at a given grid resolution simply by counting the number of pairs in adjacent cells. We distinguish the amounts of horizontal and vertical and diagonal neighbors since different search patterns may be opted by users: mostly horizontal or vertical for people accustomed respectively to western and non-western reading order, diagonal may be relevant for grids of light density.

For our experiments described in the next section, we prepared the collections by qualitative selection of the optimal grid resolution based on the amounts of horizontal, vertical and diagonal adjacent neighbors computed for each resolution between the minimal side and the least distorted approximate, comparing such amounts between a proximity grid applied after dimension reduction and a grid ordered by filename. Not all collections presented a proximity grid resolution that outperformed a simple grid by filename in terms of neighbor preservation.

## 4. EXPERIMENTS

### 4.1 Open dataset

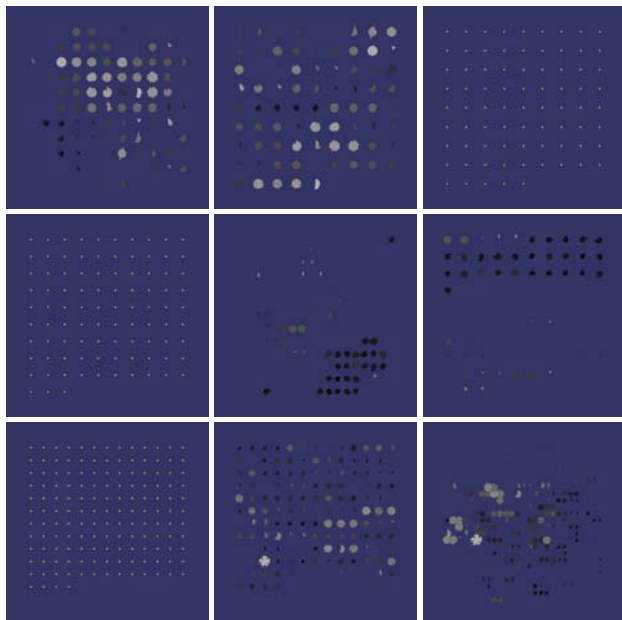
The One Laptop Per Child (OLPC) sound library<sup>4</sup> was chosen so as to make the following tests easily reproducible, for validation and comparison perspectives, and because it is not a dataset artificially generated to fit with expected results when testing machine learning algorithms. It is licensed under a Creative Commons BY license (requiring attribution). It contains 8458 sound samples, 90 sub-libraries combine diverse types of content or specialize into one type, among which: musical instruments riffs or single notes, field recordings, Foley recording, synthesized sounds, vocals, animal sounds. It is to be noted, especially for subset libraries curated by Berklee containing Foley sound design material, that within a given subset most samples seem to have been recorded, if not named, by a same author per subset. It is thus frequent to find similar sounds named incrementally, for instance *Metal on the ground [n]* with  $n$  varying from 1 to 4. These are likely to be different takes of a recording session on a same setting of sounding object and related action performed on it. Ordering search results by tag filtering in a list by path and filename, similarly to a standard file browser, will thus imprint local neighborhoods to the list.

### 4.2 Evaluation method

We chose to perform a qualitative and quantitative evaluation: qualitative through a feedback questionnaire, quantitative through known-item search tasks as popularized recently for video browsers by the Video Browser Showdown [17]. In the context of audio browsers, for each task the target sound is heard, the user has to find it back as fast as possible using a given layout. Font’s thesis compared layouts for sound browsing: *automatic* (PCA), *direct mapping* (scatter plot) and *random map* [8]. Time and speeds were deliberately not investigated, claiming that people employ different search behaviors.

<sup>4</sup>[http://wiki.laptop.org/go/Free\\_sound\\_samples](http://wiki.laptop.org/go/Free_sound_samples)





**Figure 2.** Sequence of tasks in the last experiment. In rows subsets of the One Laptop Per Child (OLPC) sound library filtered by keyword, respectively “water”, “spring”, “metal”. In columns: permutations of layouts.

### 4.3 Design

We undertook four experiments: the first comparing *grid* and glyph-less *cloud* layouts motivated us to add glyph representations (*cloud* was outperformed), the second and third confirmed that a proximity grid was to be investigated (*cloud* still outperformed), the last validated these choices. We recorded several metrics (success times, pointer distances and speeds, audio hovers) and ratings from feedback questionnaires. Here we only report the last experiment and only analyze times taken to successfully find targets.

The fourth experiment was designed as a within-subject summative evaluation. Figure 2 shows the exact sequence of tasks presented to the users. An additional collection was used for training tasks with each layout.

Each layout was given a nickname: *grid* for the simple grid ordered by filename, *album* for its upgrade with glyphs, *metro* for the proximity grid of optimal resolution for neighbors preservation. These short nicknames brought two advantages: facilitating their instant recognition when announced by the test observer at the beginning of each task, and suggesting search patterns: horizontal land mowing for *grid* and *album*, adjacent cell browsing for *metro*. The *metro* layout was described to users using the metaphor of metro maps: items (stations) can form (connect) local neighborhoods and remote “friends” (through metro lines usually identified by color).

### 4.4 Participants and apparatus

16 participants (5 female) of mean age 28 (+/- 6.3) each performed 9 tasks on 3 different collections. Besides 2 subjects, all the participants have studied or taught audiovisual communication practices (sound design, film edition).

They were asked which human sense they favored in their work (if not, daily) on a 5-point Likert scale, 1 for audition to 5 for vision: on average 3.56 (+/- 0.60). All self-rated themselves with normal audition, 10 with corrected vision.

We used an Apple Macbook Pro Late 2013 laptop with 15-inch Retina display, with a RME FireFace UCX sound card, and a pair of Genelec active loudspeakers. A 3Dconnexion Space Navigator 3D mouse was repurposed into a buzzer to submit targets hovered by the touchpad, with audio feedback of the closest node to the pointer.

### 4.5 Results

A one-way ANOVA shows that there is a quite significant difference between views within subjects on success times ( $p=.02$ ), more on self-reported ratings of efficiency ( $p<.001$ ) and pleasurability ( $p<.001$ ). Mean and standard deviations are compared in table 1. A Tukey multiple comparisons of success times means at a 95% family-wise confidence level on layouts shows that *metro* outperformed *grid* ( $p=.01$ ), but *album* was not significantly better than *grid* ( $p=.34$ ) or worse than *metro* ( $p=.26$ ).

	<i>grid</i>	<i>album</i>	<i>metro</i>
success times (s)	53.0(46.6)	43.1(38.0)	31.3(22.9)
efficiency [1-5]	1.87(1.01)	3.75(1.00)	4.12(0.96)
pleasurability [1-5]	2.25(1.18)	3.62(0.81)	4.25(0.86)

**Table 1.** Mean (standard deviations) of evaluation metrics

### 4.6 Discussion

Feature extraction is a one-shot offline process at indexing time. Dimension reduction for layout computation is a process that should be close to real-time so as not to slow down search tasks and that is likely to be performed at least once per query. Decent results can be achieved by combining only content-based icons and simple ordering by filename. A content-based layout comes at a greater computational cost but brings significant improvements.

## 5. CONCLUSION, FUTURE WORKS

We proposed a method to assist sound designers in reviewing results of queries by browsing a sound map optimized for nearest neighbors preservation in adjacent cells of a proximity grid, with content-based features cued through glyph-based representations. Through a usability evaluation of known-item search tasks, we showed that this solution was more efficient and pleasurable than a grid of sounds ordered by filenames.

An improvement to this method would require to investigate all blocks from the multimedia information retrieval data flow. First, other features tailored for sound effects should be tried. Second, we have noticed that some of the first high-dimensional nearest neighbors are positioned quite far away in 2D, already past dimension reduction. Reducing pairwise distance preservation errors may be an investigation track.

## 6. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their careful recommendations. We thank all the testers for their time and patience in performing tasks that were sometimes too difficult. This work has been partly funded by the Walloon Region of Belgium through GreenTIC grant SONIXTRIP.

## 7. REFERENCES

- [1] Wojciech Basalaj. *Proximity visualisation of abstract data*. PhD thesis, University of Cambridge, 2000.
- [2] Eoin Brazil. Investigation of multiple visualisation techniques and dynamic queries in conjunction with direct sonification to support the browsing of audio resources. Master's thesis, Interaction Design Centre, Dept. of Computer Science & Information Systems University of Limerick, 2003.
- [3] Eoin Brazil, Mikael Fernström, George Tzanetakis, and Perry Cook. Enhancing sonic browsing using audio information retrieval. In *Proceedings of the International Conference on Auditory Display (ICAD)*, 2002.
- [4] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. In *Proceedings of the IEEE*, volume 96, 2008.
- [5] Ryan R. Curtin, James R. Cline, Neil P. Slagle, William B. March, P. Ram, Nishant A. Mehta, and Alexander G. Gray. MLPACK: A scalable C++ machine learning library. *Journal of Machine Learning Research*, 14:801–805, 2013.
- [6] Stéphane Dupont, Thierry Ravet, Cécile Picard-Limpens, and Christian Frisson. Nonlinear dimensionality reduction approaches applied to music and textural sounds. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2013.
- [7] Mikael Fernström and Eoin Brazil. Sonic browsing: An auditory tool for multimedia asset management. In *Proceedings of the 2001 International Conference on Auditory Display*, 2001.
- [8] Frederic Font. Design and evaluation of a visualization interface for querying large unstructured sound databases. Master's thesis, Universitat Pompeu Fabra, Music Technology Group, 2010.
- [9] Thomas Grill and Arthur Flexer. Visualization of perceptual qualities in textural sounds. In *Proceedings of the Intl. Computer Music Conference, ICMC*, 2012.
- [10] Thomas Grill, Arthur Flexer, and Stuart Cunningham. Identification of perceptual qualities in textural sounds using the repertory grid method. In *Proceedings of the 6th Audio Mostly Conference: A Conference on Interaction with Sound*, ACM, 2011.
- [11] Sebastian Heise, Michael Hlatky, and Jörn Loviscach. Soundtorch: Quick browsing in large audio collections. In *125th Audio Engineering Society Convention*, 2008.
- [12] Ianis Lallemand and Diemo Schwarz. Interaction-optimized sound database representation. In *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, 2011.
- [13] Joshua M. Lewis, Laurens van der Maaten, and Virginia de Sa. A behavioral investigation of dimensionality reduction. In N. Miyake, D. Peebles, and R. P. Cooper, editors, *Proceedings of the 34th Annual Conference of the Cognitive Science Society*, 2012.
- [14] Kristian Nybo, Jarkko Venna, and Samuel Kaski. The self-organizing map as a visual neighbor retrieval method. In *Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM)*, 2007.
- [15] G. Peeters. Sequence representation of music structure using higher-order similarity matrix and maximum-likelihood approach. In *Proc. of the Intl. Symposium on Music Information Retrieval (ISMIR)*, 2007.
- [16] Kerry Rodden, Wojciech Basalaj, David Sinclair, and Kenneth Wood. Does organisation by similarity assist image browsing? In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, CHI. ACM, 2001.
- [17] Klaus Schoeffmann, David Ahlström, Werner Bailer, Claudiu Cobârzan, Frank Hopfgartner, Kevin McGuinness, Cathal Gurrin, Christian Frisson, Duy-Dinh Le, Manfred Fabro, Hongliang Bai, and Wolfgang Weiss. The video browser showdown: a live evaluation of interactive video search tools. *International Journal of Multimedia Information Retrieval*, pages 1–15, 2013.
- [18] Rebecca Stewart. *Spatial Auditory Display for Acoustics and Music Collections*. PhD thesis, School of Electronic Engineering and Computer Science Queen Mary, University of London, 2010.
- [19] Sebastian Stober, Thomas Low, Tatiana Gossen, and Andreas Nürnberger. Incremental visualization of growing music collections. In *Proceedings of the 14th Conference of the International Society for Music Information Retrieval (ISMIR)*, 2013.
- [20] Colin Ware. *Visual Thinking: for Design*. Interactive Technologies. Morgan Kaufmann, 2008.
- [21] E. Wold, T. Blum, D. Keislar, and J. Wheaten. Content-based classification, search, and retrieval of audio. *MultiMedia, IEEE*, 3(3):27–36, 1996.

# INTRODUCING A DATASET OF EMOTIONAL AND COLOR RESPONSES TO MUSIC

Matevž Pesek<sup>1</sup>, Primož Godec<sup>1</sup>, Mojca Poredoš<sup>1</sup>, Gregor Strle<sup>2</sup>, Jože Guna<sup>3</sup>,  
Emilija Stojmenova<sup>3</sup>, Matevž Pogačnik<sup>3</sup>, Matija Marolt<sup>1</sup>

<sup>1</sup>University of Ljubljana, Faculty of Computer and Information Science  
[matevz.pesek, primoz.godec, mojca.poredos, matija.marolt]@fri.uni-lj.si

<sup>2</sup>Scientific Research Centre of the Slovenian Academy of Sciences and Arts,  
Institute of Ethnomusicology  
gregor.strle@zrc-sazu.si

<sup>3</sup>University of Ljubljana, Faculty of Electrotechnics  
[joze.guna, emilija.stojmenova, matevz.pogacnik]@fe.uni-lj.si

## ABSTRACT

The paper presents a new dataset of mood-dependent and color responses to music. The methodology of gathering user responses is described along with two new interfaces for capturing emotional states: the MoodGraph and MoodStripe. An evaluation study showed both interfaces have significant advantage over more traditional methods in terms of intuitiveness, usability and time complexity. The preliminary analysis of current data (over 6.000 responses) gives an interesting insight into participants' emotional states and color associations, as well as relationships between musically perceived and induced emotions. We believe the size of the dataset, interfaces and multi-modal approach (connecting emotional, visual and auditory aspects of human perception) give a valuable contribution to current research.

## 1. INTRODUCTION

There is no denial that strong relationship exists between music and emotions. On one hand, music can express and induce a variety of emotional responses in listeners and can change our mood (e.g. make us happy – we consider mood to be a longer lasting state). On the other hand, our current mood strongly influences our choice of music - we listen to different music when we're sad than when we're happy.

It is therefore not surprising that this relationship has been studied within a variety of fields, such as philosophy, psychology, musicology, anthropology or sociology [1]. Within Music Information Retrieval, the focus has been on mood estimation from audio (a MIREX task since 2007), lyrics or tags and its use for music recommendation and playlist generation, e.g. [2-5].

To estimate and analyze the relationship between mood and music, several datasets were made available in the past

years. The soundtracks dataset for music and emotion contains single mean ratings of perceived emotions (labels and values in a three-dimensional model are given) for over 400 film music excerpts [6]. The MoodSwings Turk Dataset contains on average 17 valence-arousal ratings for 240 clips of popular music [7]. The Cal500 contains a set of mood labels for 500 popular songs [8], at around three annotations per song, and the MTV Music Data Set [9] a set of 5 bipolar valence-arousal ratings for 192 popular songs.

In this paper, we introduce a new dataset that captures users' mood states, their perceived and induced emotions to music and their association of colors with music. Our goals when gathering the dataset were to capture data about the user (emotional state, genre preferences, their perception of emotions) together with ratings of perceived and induced emotions on a set of unknown music excerpts representing a variety of genres. We aimed for a large number of annotations per song, to capture the variability, inherent in user ratings.

In addition, we wished to capture the relation between color and emotions, as well as color and music, as we believe that color is an important factor in music visualizations. A notable effort has been put into visualizing the music data on multiple levels: audio signal, symbolic representations and meta-data [10]. Color tone mappings can be applied onto the frequency, pitch or other spectral components [11], in order to describe the audio features of the music [12], or may represent music segments. The color set used for most visualizations is picked instinctively by the creator. To be able to provide a more informed color set based on emotional qualities of music, our goal thus was to find out whether certain uniformity exist in the perception of relations between colors, emotions and music.

The paper is structured as follows: section 2 describes the survey and its design, section 3 provides preliminary analyses of the gathered data and survey evaluation and section 4 concludes the paper and describes our future work.



© Matevž Pesek, Primož Godec, Mojca Poredoš, Gregor Strle, Jože Guna, Emilija Stojmenova, Matevž Pogačnik, Matija Marolt Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Matevž Pesek, Primož Godec, Mojca Poredoš, Gregor Strle, Jože Guna, Emilija Stojmenova, Matevž Pogačnik, Matija Marolt. "Introducing a dataset of emotional and color responses to music", 15th International Society for Music Information Retrieval Conference, 2014.

## 2. ONLINE SURVEY

We gathered the dataset with an online survey, with the intention to reach a wide audience and gather a large number of responses. We started our survey design with a preliminary questionnaire, which provided some basic guidelines for the overall design. We formed several research questions to drive the design and finally implemented the survey which captures the user's current emotional state, their perception of colors and corresponding emotions, as well as emotions perceived and induced from music, along with the corresponding color. After the first round of response gathering was completed, we performed a new survey designed to evaluate different aspects of user experience with our original survey.

### 2.1 Preliminary study

Although there exists some consent that a common set of basic emotions can be defined [13], in general there is no standard set of emotion labels that would be used in music and mood researches. Some authors choose labeled sets intuitively, with no further explanation [14]. In contrast, we performed an initial study in order to establish the relevant set of labels. For the purpose of eliminating the cultural and lingual bias on the labelling, we performed our survey in Slovenian language for Slovene-speaking participants.

The preliminary questionnaire asked the user to describe their current emotional state through a set of 48 emotion labels selected from literature [15-17], each with an intensity-scale from 1 (inactive) to 7 (active). The questionnaire was solved by 63 participants. Principal component analysis of the data revealed that first three components explain 64% of the variance in the dataset. These three components strongly correlate to 17 emotion labels chosen as emotional descriptors for our survey.

We also evaluated the effectiveness of the continuous color wheel to capture relationships between colors and emotions. Responses indicated the continuous color scale to be too complex and misleading for some users. Thus, a modified discrete-scale version with 49 colors displayed on larger tiles was chosen for the survey instead. The 49 colors have been chosen to provide a good balance between the complexity of the full continuous color wheel and the limitations of choosing a smaller subset of colors.

### 2.2 The survey

The survey is structured into three parts, and contains questions that were formulated according to our hypotheses and research goals:

- user's mood impacts their emotional and color perception of music;
- relations between colors and emotions are uniform in groups of users with similar mood and personal characteristics;

- correlation between sets of perceived and induced emotions depends both on the personal musical preferences, as well as on the user's current mood;
- identify a subset of emotionally ambiguous music excerpts and study their characteristics;
- mappings between colors and music depend on the music genre;
- perceived emotions in a music excerpt are expected to be similar across listeners, while induced emotions are expected to be correlated across groups of songs and users with similar characteristics.

We outline all parts of the survey in the following subsections, a more detailed overview can be found in [18].

#### 2.2.1 Part one – personal characteristics

The first part of the survey contains nine questions that capture personal characteristics of users. Basic demographics were captured: age, gender, area of living, native language. We also included questions regarding their music education, music listening and genre preferences. We decided not to introduce a larger set of personal questions, as the focus of our research lies in investigating the interplay of colors, music and emotions and we did not want to irritate the users with a lengthy first part. Our goal was to keep the amount of time spent for filling in the survey to under 10 minutes.

#### 2.2.2 Part two - mood, emotions and colors

The second part of our survey was designed to capture information about the user's current mood, their perception of relation between colors and emotions and their perception of emotions in terms of pleasantness and activeness.

The user's emotional state was captured in several ways. First, users had to place a point in the valence-arousal space. This is a standard mood estimation approach, also frequently used for estimation of perceived emotions in music. Users also indicated the preferred color of their current emotional state, as well as marked the presence of a set of emotion labels by using the *MoodStripe* interface (see Figure 1).



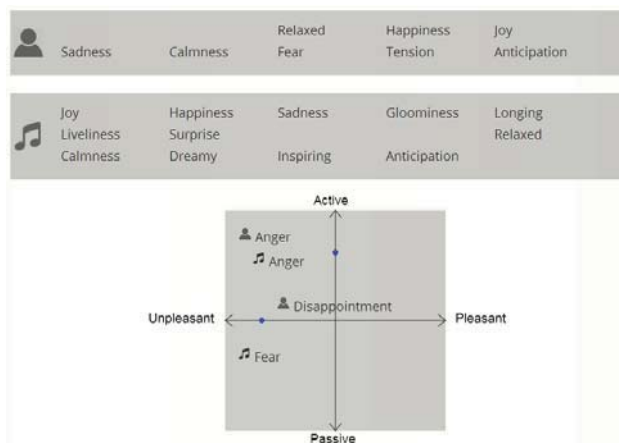
**Figure 1:** The *MoodStripe* allows users to express their emotional state by dragging emotions onto a canvas, thereby denoting their activity

To match colors with emotions, users had to pick a color in the color wheel that best matches a given emotion label (10 labels were presented to each user). Finally, users had to assess how they perceive the pleasantness and activeness of emotions by placing a set of emotion labels into the valence-arousal space using the *MoodGraph* (see Figure 2). This enables us to evaluate the variability of placement of emotion labels in terms of their activeness and pleasantness and compare it to data gathered in part three, where users described musical excerpts in a similar manner.

### 2.2.3 Part three - music in relation to colors and emotions

In the third part of our survey users were asked to complete two tasks on a set of ten 15-second long music excerpts. These were randomly selected from a database of 200 music excerpts. When compiling the database, we strived for a diverse, yet unknown set of music pieces, to avoid judgments based on familiarity with the content. The database contains 80 songs from the royalty free online music service *Jamendo*, representing a diverse variety of “standard” genres, with songs unknown to the wider audience. 80 songs were included from a dataset of film music excerpts [6], 20 from a database of folk music and 20 from a contemporary electro-acoustic music collection.

After listening to an excerpt, users were first asked to choose the color best representing the music from the color wheel. Next, users were asked to describe the music by dragging emotion labels onto the valence-arousal space using the *MoodGraph* interface (Figure 2). Two different sets of labels were used for describing induced and perceived emotions, as different emotions correspond with respective category[19], and at least one label from each category had to be placed onto the space. shown and



**Figure 2:** The *MoodGraph*: users drag emotion labels onto the valence-arousal space. Induced emotions are marked with a person icon, perceived emotions with a note icon.

## 2.3 Evaluation survey

After responses were gathered, we performed an additional evaluation survey, where we asked participants to evaluate

the original survey. Although the survey was anonymous, users had the opportunity to leave their email at the end, which we used to invite them to fill in the evaluation questionnaire. Participants were presented a set of twelve questions about different aspects of the survey: user experience, complexity of the questionnaire, and aspects of our new *MoodGraph* and *MoodStripe* interfaces. Some of the questions were drawn from the existing evaluation standard NASA load task index [20], while others were intended to evaluate different aspects of our interfaces.

## 3. RESULTS

The survey was taken by 952 users, providing 6609 mood/color-perception responses for the 200 music excerpts used. We thus obtained a large number of responses per music excerpt (each has 33 responses on average), including sets of induced and perceived emotion labels, their placement in the valence-arousal space, as well as the color describing the excerpt. To our knowledge, no currently available mood-music dataset has such a high ratio of user annotations per music excerpt. The data, as well as music excerpts will be made public as soon as the second round of response gathering, currently underway, will be finished.

In the following subsections, we provide some preliminary analyses of our data.

### 3.1 Demographic analysis

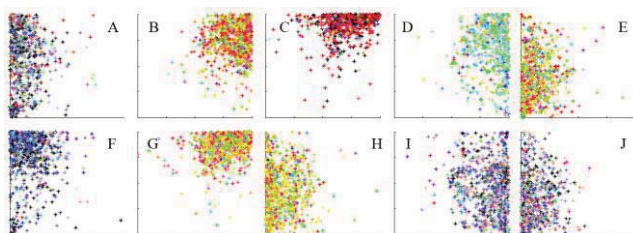
The basic demographic characteristics of the 952 participants are as follows. The average age of participants was 26.5 years, the youngest had 15, the oldest 64 years. 65% of participants are women, 66% are from urban areas. 50% have no music education, 47% do not play instruments or sing. The amount of music listening per day is evenly spread from less than 1 hour to over 4 hours. 3% claimed they were under the influence of drugs when taking the survey.

### 3.2 Colors and emotions

In the second part of the survey, participants indicated their emotional state within the valence-arousal space, as well as by choosing a color. Relations between the color hue and location in the valence-arousal space are not very consistent, but overall less active emotional states correspond more with darker blue-violet hues, while the more active ones to red-yellow-green hues. There is also a statistically significant positive correlation between color saturation and value (in a HSV color model) and activeness, as well as pleasantness of emotions: the more positive and active the user’s emotional state is, the more vivid the colors are.

Colors attributed to individual emotion labels, as well as the placement of labels in the valence-arousal space are visible in Figure 3. Associations between colors and emotions are quite consistent and in line with previous research [21-24]. Fear (A) and anger (F) are basic negative emotions and have dark blue/violet or black hues. Sadness (I)

and relaxation (J), interestingly are also very similar, although different in valence. Energetic (C) as a very active mood is mostly red, joy (B) and liveliness (G) somewhat less (more yellowy, even green). Another interesting outcome is that similar red-yellow-green hues are also prevalent for disappointment (E) and discontent (H). Happiness (D) is very distinct, in pastels of green and blue (similar to [21-24]). As these hues are often related to inner balance (peace), their choice for happiness, by some definitions a state where ones needs are satisfied, reflects the participants' notion that happiness and inner balance are related [21, 24].



**Figure 3:** position of emotions in the valence-arousal space, and their colors. *A: fear, B: joy, C: energy, D: happiness, E: disappointment, F: anger, G: liveliness, H: discontent, I: relaxation, J: sadness*

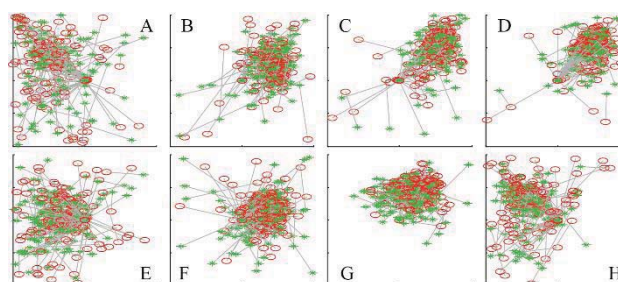
### 3.3 Relationships between induced and perceived emotions

In part three of the survey participants were asked to mark induced and perceived emotions for individual music excerpt by dragging emotion labels from the respective categories onto the valence-arousal space (see Figure 2). Here, we focus on the relationship between induced and perceived emotions.

Figure 4 shows the centroids (averages) for induced-perceived emotion pairs of participants' ratings for each music excerpt: 'anger', 'relaxed', 'happiness', 'joy', 'sadness', 'calmness', 'anticipation' and 'fear'. Positions of induced-perceived emotion pairs (Figure 4) loosely correspond to the positions of participant's emotional states in the valence-arousal space from Figure 3, with some obvious differences. For example (with respect to B, D and I on Figure 3), positive induced-perceived emotion pairs, such as relaxed, happiness and joy (B, C and D in Figure 4) occupy a more central space in the 'pleasant/active' quadrant of valence-arousal space. Similarly, negative emotion pairs (A, E and H in Figure 4) are also more central on the 'unpleasant' quadrants than corresponding emotions on Figure 3, but have significantly larger variance and spread on valence-arousal space compared to positive emotions (apart from relaxed (B)), especially along arousal dimension.

Let us compare the relationships in Figure 4. There is a noticeable variance between induced and perceived emotions for negative emotions, such as fear (H), anger (A) and sadness (E), as they spread over both arousal and valence

axes. The central position of sadness (E) along the arousal dimension is especially interesting, as it is typically associated with low arousal (compare to J in Figure 3). Furthermore, all three negative emotions (A, E and H) are in certain musical contexts experienced or perceived as pleasant. On the other hand, positive induced-perceived emotion pairs, such as joy (D) and happiness (C), tend to be more similar on both valence (positive) and arousal (relatively high) dimension and consequently have less variance. More neutral emotions, such as calmness (F) and anticipation (G), occupy the center, with relaxed (B) untypically potent on the arousal dimension.



**Figure 4:** Representation of relationships between induced-perceived emotion pairs of all music excerpts (induced centroid: green star, perceived centroid: red circle). *A: anger, B: relaxation, C: happiness, D: joy, E: sadness, F: calmness, G: anticipation, H: fear*

Discriminating between induced and perceived emotions in music is a complex task and to date there is no universally agreed upon theory, or emotional model, that would best capture emotional experiences of listeners (see e.g. [19, 25-29]). Many argue (e.g. [6, 19, 28, 30, 31]) that simple valence-arousal dimensional model (one that *MoodGraph* is based on) might be too reductionist, as it ignores the variance of emotions and results in inherently different emotions occupying similar regions of valence-arousal space (e.g., compare regions of fear (H), anger (A) and sadness (E) in Figure 4). Our preliminary results nevertheless show some interesting aspects of induction and perception of musical emotions. For example, the representations of relationships among and within induced-perceived emotion pairs shown in Figure 4 support Gabrielson's theory of four basic types of relationship between induced and perceived emotions in relation to music: positive/in agreement, negative/opposite, non-systematic/neutral and absent/no relationship [25]. Positive relationship is the most common (e.g., when music perceived to express sad emotions also evokes such emotions in the listener), resulting in the overlap (in some cases above 60%; see e.g. [19, 26, 29]) of induced-perceived emotion pairs. In one study [32], researchers found extremely strong positive correlation for induced and perceived emotions on both valence and arousal dimensions, and concluded that results show "listeners will typically feel the emotions ex-

pressed by the song” [p. 93]. However, our preliminary results do not support this claim. There is a significant variance among induced-perceived emotion pairs, particularly among negative emotions. Furthermore, while effects of positive correlation between induced and perceived emotions are evident (especially in positive emotions), other types of relationships are equally significant: from negative/opposite, non-matching, to complex and neutral. The preliminary results clearly show differential variance across induced and perceived emotions (in line with recent findings [33]).

When analyzing the induced-perceived emotion pairs in *MoodGraph*, we’ve found that: a) they do not necessarily positively correlate, b) they occupy different regions and c) even when they fall into the same region of valence-arousal space, both rotation and standard deviation within each induced-perceived emotion pair are significantly larger than reported in some of the previous studies (e.g., [32]). This shows that participants understood both concepts (i.e. induced vs. perceived emotion) and were able to differentiate emotions from both categories on the valence-arousal space.

One reason for large amount of variance in representations of induced/perceived pairs is probably due to the model itself, as participants can rate both induced and perceived emotions together and directly onto *MoodGraph* after listening to the music excerpt. Another advantage, we argue, is the construction of the *MoodGraph* itself. While bearing similarity with traditional approach to dimensional modeling (a classic example being Russell’s circumplex model of affect [15]), the *MoodGraph* has no pre-defined and categorically segmented/discrete regions of valence-arousal space, hence avoiding initial bias, while still offering an intuitive interface – the participant is free to drag emotion labels onto *MoodGraph* according to her preferences and interpretation of the valence-arousal space.

### 3.4 Evaluation survey

The online evaluation questionnaire was filled-in by 125 users, who all took part in our survey. Results were positive and indicate that the survey was properly balanced and the new interfaces were appropriate. Detailed results can be found in [34]. To summarize, responses show appropriate mental difficulty of the questionnaire, while the physical difficulty seems to be more uniformly distributed across participants. Thus, it can be speculated that the listening part of the questionnaire represents a physical challenge to a significant number of participants. The presented *MoodGraph* interface was quite intuitive; however, it was also time demanding. Considering the task load of the interface (combining three distinctive tasks), this was expected. The number of emotions in *MoodGraph* categories was slightly unbalanced and should be extended in our future work. The *MoodStripe* interface represents a significant improvement over a group of radio buttons, both in

intuitiveness and time complexity. Participants also indicated that the set of 49 colors available for labeling emotions may not be large enough, so we will consider enlarging the set of color tones in our future work.

## 4. CONCLUSIONS

We intend to make the gathered dataset available to the public, including the musical excerpts, data on users’ personal characteristics and emotional state, their placement of emotions within the valence/arousal space, their perceived and induced emotional responses to music and their perception of color in relation to emotions and music. This will open new possibilities for evaluating and re-evaluating mood estimation and music recommendation approaches on a well annotated dataset, where the ground truth lies in the statistically significant amount of responses per song, rather than relying on annotations of a small number of users.

Shortly, we will start with the second round of response gathering with an English version of the survey. We also intend to enlarge the number of music excerpts in the music dataset and provide it to the users who have already participated in this study. Thus, we hope to further extend and diversify the dataset.

Preliminary analyses already show new and interesting contributions, and next to answering the questions already posed in section 2.2, the dataset will provide grounds for our future work (and work of others), including:

- previously introduced mood estimation algorithms will be evaluated by weighting the correctness of their predictions of perceived emotion responses for music excerpts. New mood estimation algorithms will be developed, building upon the newly obtained data;
- we will explore modelling of relations between music and colors chosen by users in the survey. Results may be useful for music visualization, provided that correlations between audio and visual perception will be consistent enough;
- music recommendation interfaces will be explored, presenting recommendations in a visual manner with the intent to raise user satisfaction by reducing the textual burden placed on the user. The interface will include personal characteristics and their variability in the decision model;
- the dataset can also be used in other domains, as responses that relate colors to emotions based on the user’s emotional state can be used independently.

## 5. REFERENCES

- [1] P. Juslin and J. A. Sloboda, *Music and Emotion: Theory and Research*. USA: Oxford University Press, 2001
- [2] C. Laurier, O. Meyers, J. Serrà, M. Blech, P. Herrera, and X. Serra, "Indexing music by mood: design and integration of an automatic content-based annotator," *Multimedia Tools Appl.*, vol. 48, pp. 161-184, 2010.

- [3] M. Schedl, A. Flexer, and J. Urbano, "The neglected user in music information retrieval research," *J. Intell. Inf. Syst.*, vol. 41, pp. 523-539, 2013.
- [4] Y. Song, S. Dixon, and M. Pearce, "A survey of music recommendation systems and future perspectives," presented at the 9th Int. Symp. Computer Music Modelling and Retrieval, London, UK, 2012.
- [5] Y. E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. J. Scott, *et al.*, "State of the Art Report: Music Emotion Recognition: A State of the Art Review," presented at the ISMIR, 2010.
- [6] T. Eerola and J. K. Vuoskoski, "A comparison of the discrete and dimensional models of emotion in music," *Psychology of Music*, vol. 39, pp. 18-49, 2011.
- [7] E. M. Schmidt and Y. E. Kim, "Modeling musical emotion dynamics with conditional random fields," presented at the International Society for Music Information Retrieval Conference, Miami, Florida, 2011.
- [8] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic Annotation and Retrieval of Music and Sound Effects," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, pp. 467-476, 2008.
- [9] B. Schuller, C. Hage, D. Schuller, and G. Rigoll, "'Mister D.J., Cheer Me Up!': Musical and Textual Features for Automatic Mood Classification," *Journal of New Music Research*, vol. 39, pp. 13-34, 2014/05/09 2010.
- [10] N. Orio, "Music Retrieval: A Tutorial and Review," *Foundations and Trends in Information Retrieval*, vol. 1, pp. 1-90, 2006.
- [11] S. Sagayama and K. Takahashi, "Specmurt anasyllis: A piano-roll-visualization of polyphonic music signal by deconvolution of log-frequency spectrum," presented at the ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing, Jeju, Korea, 2004.
- [12] H.-H. Wu and J. P. Bello, "Audio-based music visualization for music structure analysis," presented at the International Conference on Music Information Retrieval, Barcelona, Spain, 2010.
- [13] P. Ekman, "Basic Emotions," in *Handbook of Cognition and Emotion*, T. Dalgleish and M. Power, Eds., ed Sussex, UK: John Wiley & Sons, 1999.
- [14] B. Wu, S. Wun, C. Lee, and A. Horner, "Spectral correlates in emotion labeling of sustained musical instrument tones" presented at the International Society for Music Information Retrieval Conference, Curitiba, Brasil, 2013.
- [15] J. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, pp. 1161-1178, 1980.
- [16] N. A. Remington, P. S. Visser, and L. R. Fabrigar, "Reexamining the Circumplex Model of Affect," *Jurnal of Personality and Social Psychology*, vol. 79, pp. 286-300, 2000.
- [17] D. Watson, C. L. A., and A. Tellegen, "Development and validation of brief measures of positive and negative affect: The PANAS scales," *Journal of Personality and Social Psychology*, vol. 54, pp. 1063-1070, 1988.
- [18] M. Pesek, P. Godec, M. Poredoš, G. Strle, J. Guna, E. Stojmenova, *et al.*, "Gathering a dataset of multi-modal mood-dependent perceptual responses to music," presented at the 2nd Workshop on "Emotions and Personality in Personalized Services", Aalborg, Denmark, 2014.
- [19] P. N. Juslin and P. Laukka, "Expression, Perception, and Induction of Musical Emotions: A Review and a Questionnaire Study of Everyday Listening," *Journal of New Music Research*, vol. 33, pp. 217-238, 2014/05/09 2004.
- [20] S. G. Hart, "Nasa-Task Load Index (NASA-TLX); 20 Years Later," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 50, pp. 904-908, 2006.
- [21] P. Valdez and A. Mehrabian, "Effects of Color on Emotions," *Journal of Experimental Psychology: General*, vol. 123, pp. 394-409, 1994.
- [22] O. L. C., M. R. Luo, A. Woodcock, and A. Wright, "A Study of Colour Emotion and Color Preference. Part I: Colour Emotions for Single Colours," *Color research and application*, vol. 29, pp. 232-240, 2004.
- [23] R. D. Norman and W. A. Scott, "Color and affect: A review and semantic evaluation," *Journal of General Psychology*, vol. 46, pp. 185-223, 1952.
- [24] L. B. Wexner "The degree to which colors (hues) are associated with mood-tones," *Journal of Applied Psychology*, vol. 38, pp. 432-435, 1954.
- [25] A. Gabrielsson, "Emotion Perceived and Emotion Felt: Same or Different?," *Musicae Scientiae*, vol. 5, pp. 123--147, 2002.
- [26] K. Kallinen and N. Ravaja, "Emotion perceived and emotion felt: Same and different," *Musicae Scientiae*, vol. 10, pp. 191-213, 2006.
- [27] P. Evans and E. Schubert, "Relationships between expressed and felt emotions in music," *Musicae Scientiae*, vol. 12, pp. 75-99, 2008.
- [28] E. Schubert, "Measuring Emotion Continuously: Validity and Reliability of the Two-Dimensional Emotion-Space," *Australian Journal of Psychology*, vol. 51, pp. 154-165, 1999.
- [29] T. Eerola and J. K. Vuoskoski, "A review of music and emotion studies: approaches, emotion models, and stimuli," vol. 30, pp. 307-340, 2013.
- [30] T. a. V. J. K. Eerola, "A comparison of the discrete and dimensional models of emotion in music," *Psychology of Music*, vol. 39, pp. 18--49, 2010.
- [31] N. Haslam, "The Discreteness of Emotion Concepts: Categorical Structure in the Affective Circumplex," *Personality and Social Psychology Bulletin*, vol. 21, pp. 1012-1019, 1995.
- [32] Y. Song, S. Dixon, M. Pearce, and A. R. Halpern, "Do Online Social Tags Predict Perceived or Induced Emotional Responses to Music?," presented at the International Society for Music Information Retrieval Conference, Curitiba, Brasil, 2013.
- [33] E. Schubert, "Emotion felt by listener and expressed by music: A literature review and theoretical investigation," *Frontiers in Psychology*, vol. 4, 2013.
- [34] M. Pesek, P. Godec, M. Poredoš, G. Strle, J. Guna, E. Stojmenova, *et al.*, "Capturing the Mood: Evaluation of the MoodStripe and MoodGraph Interfaces," in *Management Information Systems in Multimedia Art, Education, Entertainment, and Culture (MIS-MEDIA), IEEE Internation Conference on Multimedia & Expo (ICME)*, 2014, pp. 1-4.



# IN-DEPTH MOTIVIC ANALYSIS BASED ON MULTIPARAMETRIC CLOSED PATTERN AND CYCLIC SEQUENCE MINING

Olivier Lartillot

Aalborg University, Department of Architecture, Design and Media Technology, Denmark  
olartillot@gmail.com

## ABSTRACT

The paper describes a computational system for exhaustive but compact description of repeated motivic patterns in symbolic representations of music. The approach follows a method based on closed heterogeneous pattern mining in multiparametrical space with control of pattern cyclicity. This paper presents a much simpler description and justification of this general strategy, as well as significant simplifications of the model, in particular concerning the management of pattern cyclicity. A new method for automated bundling of patterns belonging to same motivic or thematic classes is also presented.

The good performance of the method is shown through the analysis of a piece from the JKUPDD database. Ground-truth motives are detected, while additional relevant information completes the ground-truth musicological analysis.

The system, implemented in Matlab, is made publicly available as part of *MiningSuite*, a new open-source framework for audio and music analysis.

## 1. INTRODUCTION

The detection of repetitions of sequential representations in symbolic music is a problem of high importance in music analysis. It enables the detection of repeated motifs and themes<sup>1</sup>, and of structural repetition of musical passages.

### 1.1 Limitation of previous approaches

Finding these patterns without knowing in advance their actual description is a difficult problem. Previous approaches have shown the difficulty of the problem related to the combinatorial explosion of possible candidate patterns [2]. Some approaches tackle this issue by generating a large set of candidate patterns and applying simple global heuristics, such as finding longest or most frequent patterns [3,8]. Similarly, other approaches base the search for patterns on

<sup>1</sup> Here motif and theme are considered as different musicological interpretations of a same pattern configuration: motifs are usually shorter than themes.



© Olivier Lartillot.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Olivier Lartillot. "IN-DEPTH MOTIVIC ANALYSIS BASED ON MULTIPARAMETRIC CLOSED PATTERN AND CYCLIC SEQUENCE MINING", 15th International Society for Music Information Retrieval Conference, 2014.

general statistical characteristics [5]. The problem is that there is no guarantee that this global filtering leads to a selection of patterns corresponding to those selected by musicologists and perceived by listeners.

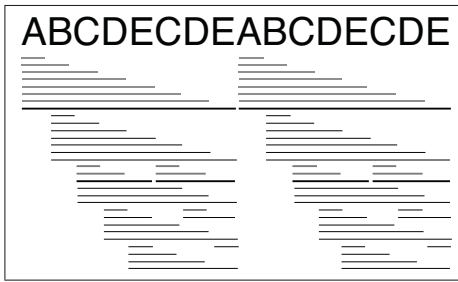
### 1.2 Exhaustive mining of closed and cyclic patterns

In our research, we endeavour to reveal the factors underlying this structural explosion of possible patterns and to formalise heuristics describing how listeners are able to consensually perceive clear pattern structures out of this apparent maze. We found that pattern redundancy is based on two core issues [6]:

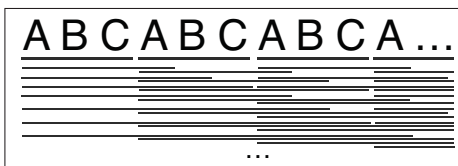
- *closed pattern* mining: When a pattern is repeated, all underlying pattern representations it encompasses are repeated as well. In simple string representation, studied in section 2<sup>2</sup>, these more *general* patterns correspond to prefixes, suffixes and prefixes of suffixes. The proliferation of general patterns, as shown in Figure 1, leads to combinatorial explosion. Restricting the search to the most specific (or "maximal") patterns is excessively selective as it filters out potentially interesting patterns (such as CDE in Figure 1), and would solely focus on large sequence repetitions. By restricting the search to *closed* patterns – i.e., patterns that have more occurrences than their more specific patterns –, all pattern redundancy is filtered out without loss of information. [6] introduces a method for exhaustive closed pattern mining.
- *pattern cyclicity*: When repetitions of a pattern are immediately successive, another combinatorial set of possible sequential repetitions can be logically inferred [2], as shown in Figure 2. This redundancy can be avoided by explicitly modelling the cyclic loop in the pattern representation, and by generalising the notion of closed pattern accordingly.

By carefully controlling these factors of combinatorial redundancy without damaging the non-redundant pattern information, the proposed approach in [6] enables to output an exhaustive description of pattern repetitions. Previous approaches did not consider those issues and performed instead global filtering techniques that broadly miss the rich pattern structure.

<sup>2</sup> The more complex multiparametric general/specific transformations are studied in section 3.



**Figure 1.** Patterns found in a sequence of symbols. Below the sequence, each row represents a different pattern class with the occurrences aligned to the sequence. Thick black lines correspond to closed patterns (the upper one is the maximal pattern), grey lines to prefixes of closed patterns, and thin lines to non-closed patterns.



**Figure 2.** Closed patterns found in a cyclic sequence of symbols. The occurrences of the pattern shown in thick lines do not overlap, whereas those shown in thin lines do.

### 1.3 New approach

In this paper, we propose a simplified description and modelling of this exhaustive pattern mining approach. In section 2, we present the problem of closed pattern mining on the simple case of monoparametric string analysis, introduce a simplified algorithmic implementation, and present a new way to simply justify the interest of the approach. In section 3, the approach is generalised to the multidimensionality of the musical parametric space. Section 4 discusses pattern cyclicity and presents a new simple model that solves this issue. In section 5, the interest of the method is shown through the analysis of a piece of music from the JKUPDD database.

## 2. CORE PRINCIPLES OF THE MODEL

### 2.1 Advantages of incremental one-pass approach

As explained in the previous section, testing the closedness of a pattern requires comparing its number of occurrences with those of all the more specific patterns. Previous computer science researches in closed pattern mining (one recent being [9]) incrementally construct the closed patterns dictionary while considering the whole document to be analysed (in our case, the piece of music). This requires the design of complex algorithms to estimate the number of occurrences of each possible pattern candidate.

We introduced in [6] a simpler approach based on an incremental single pass throughout the document (i.e., from the beginning to the end of the piece of music), during which the closed pattern dictionary is incrementally constructed: for each successive note  $n$  in the sequence, all

patterns in the subsequence ending to that note  $n$  are exhaustively searched for. The main advantage of the incremental approach is based on the following property.

**Lemma 2.1** (Closed pattern characterisation). *When following the incremental approach, for any closed pattern  $P$ , there exists a particular moment in the piece of music where an occurrence  $O$  of  $P$  can be inferred while no occurrence of any more specific pattern can be inferred.*

*Proof.* There are three alternative conditions concerning the patterns more specific than  $P$ :

- There is no pattern more specific than  $P$ . In this case, the observation is evident.
- There is only one pattern  $S$  more specific than  $P$ . For instance, in Figure 3,  $S = ABCD$  is more specific than  $P = CD$ . Since  $P$  is closed, it has more occurrences than  $S$ , so there exists an occurrence of  $P$  that is not occurrence of  $S$ .
- There are several patterns  $S_1, \dots, S_n$  more specific than  $P$ . For instance, in Figure 1,  $S_1 = ABCDE$  and  $S_2 = ABCDECDE$  are both more specific than  $P = CDE$ . As soon as two different more specific patterns  $S_1$  (one or several time) and  $S_2$  (first time) have appeared in the sequence, pattern  $P$  can be detected, since it is repeated in  $S_1$  and  $S_2$ , but  $S_2$  is not detected yet, since it has not been repeated yet.

□

As soon as we detect a new pattern repetition, such that for that particular occurrence where the repetition is detected, there is no more specific pattern repetition, we can be sure that the discovered pattern is closed.

When considering a given pattern candidate at a given point in the piece of music, we need to be already informed about the eventual existence of more specific pattern occurrences at the same place. Hence, for a given note, patterns need to be extended *in decreasing order of specificity*.

To details further the approach, let's consider in a first simple case the *monoparametric contiguous string* case, where the main document is a sequence of symbols, and where pattern occurrences are made of contiguous substrings. In this case, 'more general than' simple means 'is a subsequence of'. In other words, a more general pattern is a *prefix* or/of a *suffix* of a more specific pattern. Let's consider these two aspects separately:

- Since the approach is incremental, patterns are constructed by incrementally extending their *prefixes* (in grey in Figure 1). Patterns are therefore represented as chains of prefixes, and the pattern dictionary is represented as a prefix tree. In this paradigm, if a given pattern  $P$  is a prefix of a closed pattern  $S$ , and if both have same number of occurrences, the prefix  $P$  can still be considered as a closed pattern, in the sense that it is an intermediary state to the constitution of the closed pattern  $S$ .

**Figure 3.** Closed patterns found in a sequence of symbols. The occurrence during which a pattern is discovered is shown in black. Dashed extensions indicate two possible pattern extensions when integrating the last note.

- The closedness of a pattern depends hence solely on the patterns to which it is a *suffix*. Thanks to the incremental one-pass approach, these more specific patterns are already inferred. The only constraint to be added is that when a given note is considered, the candidate patterns should be considered in decreasing order of specificity, i.e. from the longest to the shortest (which are suffixes of the longer ones). For instance, in Figure 3, when analysing the last note, E, there are two candidate patterns for extension, ABCD and CD. Since we first extend the most specific pattern ABCDE, when considering then the more general pattern CD, extension CDE is found as non-closed and thus not inferred.

## 2.2 Algorithmic details

Following these principles, the main routine of the algorithm simply scans the musical sequence chronologically, from the first to the last note. Integrating a new note consists in checking:

- whether pattern occurrence(s) ending at the previous note can be extended with the new note,
- whether the new note initiates the start of a new pattern occurrence.

The extension of a pattern occurrence results from two alternative mechanisms:

**Recognition** the new note is recognised as a known extension of the pattern.

**Discovery** the new note continues the occurrence in the same way that a previous note continued an older occurrence of the pattern: the pattern is extended with this new common description, and the two occurrences are extended as well.

Concerning the discovery mechanism, the identification of new notes continuing older contexts can be implemented using a simple associative array, storing the note following each occurrence according to its description. This will be called a *continuation memory*. Before actually extending the pattern, we should make sure that the extended pattern is closed.

## 2.3 Specific Pattern Class

Searching for all closed patterns in a sequence, instead of all possible patterns, enables an exhaustive pattern analysis without combinatorial explosion: all non-closed patterns

can be deduced from the closed pattern analysis. Yet, the set of closed patterns can remain quite large and the exhaustive collection of their occurrences can become cumbersome. [6] proposes to limit the analysis, without any loss of information, to closed patterns' *specific classes*, which correspond to pattern occurrences that are not included in occurrences of more specific patterns. For instance, in Figure 3, the specific class of CD contains only its first occurrence, because the two other ones are superposed to occurrences of the more specific pattern ABCDE.

We propose a simpler model for the determination of specific class of closed patterns. Non-specific occurrences are regenerated whenever necessary. Because occurrences of a given pattern are not all represented, the notes following these occurrences are not memorised, although they could generate new pattern extensions. To circumvent this issue, the extension memory related to any given pattern contains the extensions not only of that pattern but also of any more specific pattern.

## 3. MULTIPARAMETRIC PATTERN MINING

The model presented in the previous section searches for sequential patterns on monoparametric sequences, composed of a succession of symbols taken from a given alphabet. Music cannot be reduced to unidimensional parametric description.

### 3.1 Parametric space

The problem needs to be generalised by taking into account three main aspects:

- Notes are defined by a hierarchically structured combination of parameters (diatonic and chromatic pitch and pitch class, metrical position, etc.).
- Notes are defined not only in terms of their absolute position on fixed scales, but also relatively to a given local context, and in particular with respect to the previous notes (defining pitch interval, gross contour, rhythmic values, etc.). These interval representations are also hierarchically structured. Gross contour, for instance, is a simple description of the inter-pitch interval between successive notes as “increasing”, “decreasing” or “unison”. Matching along gross contour enables to track intervallic augmentation and diminution. For instance, in the example in section 5, the first interval of the fugue subject is either a decreasing third or a decreasing second. The actual diatonic pitch interval representation differs, but the gross contour remains constantly “decreasing”.
- A large part of melodic transformations can be understood as repetitions of sequential patterns that do not follow strictly all the parametric descriptions, but only a subset. For instance, a rhythmical variation of a melodic motif consists in repeating the pitch sequence, while developing the rhythmical part more freely.

[6] proposes to integrate both absolute note position and relative note interval into a single parametric space. This enables to define a motive and any occurrence as a simple succession of parametric descriptions. [6] also shows the importance of heterogeneous patterns, which are made of a succession of parameters that can each be defined on different parametric dimensions. For instance, the subject of the fugue analysed in section 5 is heterogeneous, as it starts with a gross contour interval followed by more specific descriptions. In the multiparametric paradigm, a pattern  $G$  is more general than a pattern  $S$  if it is a suffix of  $S$  and/or the successive parametric descriptions of the patterns are equal or more general than the related parametric descriptions in pattern  $P$ .

### 3.2 Motivic/thematic class as “paradigmatic sheaf”

Extending the exhaustive method developed in the previous section to this heterogeneous pattern paradigm enables to describe all possible sequential repetitions along all parametric dimensions. This leads to very detailed pattern characterisation, describing in details the common sequential descriptions between any pair of similar motif. However, a more synthetic analysis requires structuring the set of discovered patterns into motivic or thematic classes. Manual motivic taxonomy of these discovered patterns has been shown in [7].

We have conceived a method for the collection of all patterns belonging to a same motivic or thematic class. Starting from one pattern seed, the method collects all other patterns that can be partially aligned to the seed, as well as those that can be aligned to any pattern thus collected. Patterns are searched along the following transformations:

- More general patterns of same length
- More specific patterns: only the suffix that have same length that the pattern seed is selected.
- Prefixes of pattern seed can be used as pattern seeds too: they might contain additional sets of more general and more specific patterns of interest.
- Pattern extensions, leading to a forking of the motivic or thematic class into several possible continuations

All the patterns contained in the bundle remain informative in the way they show particular commonalities between subset of the motivic/thematic class, as shown in the analysis in section 5.

### 3.3 Heterogeneous pattern mining

A parametric description of a given note in the musical sequence instantiates values to all fields in the parametric space. Values in the more general fields are automatically computed from their more specific fields. A parametric description of a note in a pattern instantiates values to some fields in the space, the other indeterminate fields corresponding to undefined parameters. Values can be assigned to more general fields, even if no value is assigned

to their corresponding more specific fields. Methods have been implemented that enable to compare two parametric descriptions, in order to see if they are equal, or if one is subsumed into the other, and if not, to compute the intersection of the two descriptions.

The multiparametric description is integrated in the two core mechanisms of the incremental pattern mining model as follows:

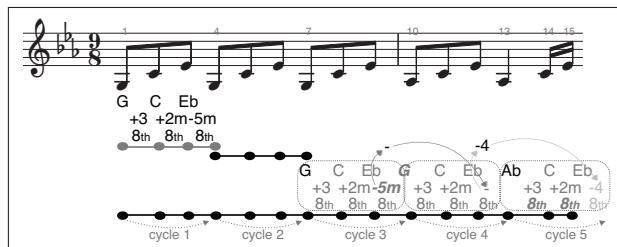
**Recognition** As before, the observed parametric description of the new note is compared to the descriptions of the patterns’ extensions. If the pattern extension’s description fits only partially, a new more general pattern extension is created (if not existing yet) related to the common description.

**Discovery** The continuation memory is structured in the same way as the parametric space: for each possible parametric field, an associative memory stores pattern continuations according to their values along that particular parametric field. As soon as a stored pattern continuation is identified with the current note along a particular parametric field, the complete parametric description common to these two contexts is computed, and the pattern extension is attempted along that common parametric description. As before, a pattern is extended only if the extended pattern is closed.

## 4. PATTERN CYCLICITY

A solution to the problem of cyclicity introduced in section 1.2 was proposed in [6] through the formalisation of *cyclic patterns*, where the last state of the chain representing the pattern is connected back to its first state, formalising this compelling expectation of the return of the periodic pattern. One limitation of the approach is that it required the explicit construction of cyclic pattern, which demanded contrived algorithmic formalisations. The problem gets even more difficult when dealing with multiparametric space, in particular when the pattern is only partially extended, i.e., when the expected parametric description is replaced by a less specific parametric matching, such as in the musical example shown in Figure 4. In this case, a more general pattern cyclic needs to be constructed, leading to the inference of a complex network of pattern cycles particularly difficult to conceptualise and implement.

We propose a simpler approach: instead of formalising *cyclic patterns*, pattern cyclicity is represented on the pattern *occurrences* directly. Once a successive repetition of a pattern has been detected, such as the 3-note pattern starting the musical example in Figure 4, the two occurrences are fused into one single chain of notes, and all the subsequent notes in the cyclic sequence are progressively added to that chain. This *cyclic chain* is first used to track the development of the new cycle (i.e., the third cycle, since there were already two cycles). The tracking of each new cycle is guided by a model describing the expected sequence of musical parameters. Initially, for the third cycle, this



**Figure 4.** Two successive repetitions of a pattern, at the beginning of the musical sequence, characterised by a pitch sequence (G, C, Eb, and back to G), a pitch interval sequence (ascending perfect fourth (+3), ascending minor third (+2m) and descending minor sixth (-5m)), and a rhythmical sequence made of a succession of 8th notes. This successive repetition leads to the inference of a cyclic chain, indicated at the bottom of the figure. When this cycle is initially inferred, at note 7, the model of the cycle, represented above “cycle 3”, corresponds to the initial pattern description. At note 10, some descriptions expected by the model (indicated in bold italics) are not fulfilled, but a more general description is inferred (descending gross contour (-)). Consequently, the next cycle (4)’s model is generalised accordingly. At note 13, a new regularity is detected, due to the repetition of pitch Ab and of descending perfect fifth (-4). Consequently, the next cycle (5)’s model is specialised accordingly.

model corresponds to the pattern that was repeated twice in the two first cycles.

- If the new cycle scrupulously follows the model, this same model will be used to guide the development of the subsequent cycle.
- If the new cycle partially follows the model (such as the modification, at the beginning of bar 2 in Figure 4, of the decreasing sixth interval, replaced by a more general decreasing contour), the model is updated accordingly by replacing the parameters that have not been matched with more general parameters.
- If the new cycle shows any new pattern identification with the previous cycle (such as the repetition of pitch Ab at the beginning of cycles 4 and 5 in Figure 4), the corresponding descriptions are added to the model.
- If at some point, the new note does not match at all the corresponding description in the model, the cyclic sequence is terminated.

This simple method enables to track the cyclic development of repeated patterns, while avoiding the combinatorial explosion inherent to this structural configuration.

## 5. TESTS

The model described in this paper is applied to the analysis of the Johannes Kepler University Patterns Develop-

ment Database (JKUPDD-Aug2013), which is the training set part of the MIREX task on Discovery of Repeated Themes & Sections initiated in 2013, and made publicly available, both symbolic representation of the scores and ground-truth musicological analyses [4].

This section details the analysis of one particular piece of music included in the JKUPDD, the 20th Fugue in the Second Book of Johann Sebastian Bach’s *Well-Tempered Clavier*. The ground truth consists of the two first bars of the third entry in the exposition part along the three voices that constitute this fugue [1]. The third entry is chosen because it is the first entry where the subject and the two countersubjects are exposed altogether. To each of these three ground-truth patterns (the subject and the two countersubjects in this two-bar entry), the ground-truth data specifies a list of occurrences in the score.

Figure 5 shows the thematic class related to ground-truth pattern #1, i.e., the fugue’s subject. This is detected by the model as one single motivic/thematic class, i.e., one complete paradigmatic sheaf, resulting from the bundling method presented in section 3.2. All occurrences indicated in the ground truth are retrieved. The patterns forming this thematic class are longer than the two-bar motif indicated in the ground truth. The limitation of all subjects and counter-subjects in the musicological analysis to two bars stems from a theoretical understanding of fugue structure that cannot be automatically inferred from a direct analysis of the score.

The analysis offered by the computational model offers much richer information than simply listing the occurrences of the subjects and countersubjects. It shows what musical descriptions characterise them, and details particular commonalities shared by occurrences of these subjects and countersubjects. For instance entries M1 and U1 belong to a same more specific pattern that describes their particular development. L1, U1 and U3 start all with a decreasing third interval, and so on.

The model presented in this paper does not yet integrate mechanisms for the reduction of ornamentation, as discussed in the next section. The only melodic ornamentation appearing in pattern #1 is the addition of a passing note after the first note of occurrences L2 and L3. This leads to a small error in the model’s results, where the first actual note is not detected.

The thematic class related to ground-truth pattern #2, which is the first countersubject, is extracted in the same way, forming a paradigmatic sheaf. The pattern class given by the model corresponds mostly to the ground truth. Here again, some occurrences present similar extensions that are inventoried by the model, although they are ignored in the ground truth. The last occurrence, which is a suffix of the pattern, is also detected accordingly. On the other hand, the second last occurrence is not properly detected, once again due to the addition of passing notes.

Pattern #3, which is the second countersubject, is more problematic, because it is only 7 notes long. Several other longer patterns are found by the model, and the specificity of pattern #3 is not grounded on characteristics purely re-

**Figure 5.** Entries of the subject in Bach's Fugue, as found by the model. The fugue has three voices: upper (U), middle (M) and lower (L). In each entry is slurred the part actually indicated in the ground-truth description of the subject. The model proposes a longer description of the subject, that is particularly developed in M1 and U1.

lated to pattern repetition. As aforementioned, the ground-truth selection of these three patterns are based on principles related to fugue rules, namely the synchronised iteration of the three patterns along the separate voices. It seems questionable to expect a general pattern mining algorithm non-specialised to a particular type of music to be able to infer this type of configuration.

## 6. CONCLUSION

The approach is incremental, progressively analysing the musical sequence through one single pass. This enables to control the structural complexity in a way similar to the way listeners perceive music.

Gross contour needs to be constrained by factors related to local saliency and short-term memory. The integration of more complex melodic transformation such as ornamentation and reduction is currently under investigation. Motivic repetition with local ornamentation is detected by reconstructing, on top of "surface-level" monodic voices, longer-term relations between non-adjacent notes related to deeper structures, and by tracking motives on the resulting syntagmatic network. More generally, the analysis of

polyphony is under study, as well as the application of the pattern mining approach to metrical analysis. The system, implemented in Matlab, is made publicly available as part of *MiningSuite*<sup>3</sup>, a new open-source framework for audio and music analysis.

## 7. ACKNOWLEDGMENTS

This work was funded by an Academy of Finland research fellowship at the Finnish Centre of Excellence in Interdisciplinary Music Research at the University of Jyväskylä. The research is continued in the context of the European project *Learning to Create (Lrn2Cre8)*, which acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859.

## 8. REFERENCES

- [1] S. Bruhn. *J.S. Bach's Well-Tempered Clavier: in-depth analysis and interpretation*, Mainer International, Hong Kong, 1993.
- [2] E. Cambouropoulos. *Towards a General Computational Theory of Musical Structure*, PhD thesis, University of Edinburgh, 1998.
- [3] E. Cambouropoulos. "Musical parallelism and melodic segmentation: A computational approach," *Music Perception*, 23(3), pp. 249–268, 2006.
- [4] T. Collins. MIREX 2013: Discovery of Repeated Themes and Sections, 2013. [http://www.music-ir.org/mirex/wiki/2013:Discovery\\_of\\_Repeated\\_Themes\\_&\\_Sections](http://www.music-ir.org/mirex/wiki/2013:Discovery_of_Repeated_Themes_&_Sections) Accessed on 14 August 2014.
- [5] D. Conklin, and C. Anagnostopoulou. "Representation and Discovery of Multiple Viewpoint Patterns," *Proceedings of the International Computer Music Conference*, 2001.
- [6] O. Lartillot. "Efficient Extraction of Closed Motivic Patterns in Multi-Dimensional Symbolic Representations of Music," *Proceedings of the International Symposium on Music Information Retrieval*, 2005.
- [7] O. Lartillot. "Taxonomic categorisation of motivic patterns," *Musicae Scientiae*, Discussion Forum 4B, pp. 25–46, 2009.
- [8] D. Meredith, K., Lemström, and G. Wiggins. "Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music," *Journal of New Music Research*, 31(4), pp. 321–345, 2002.
- [9] J. Wang, J. Han, and C. Li. "Frequent closed sequence mining without candidate maintenance," *IEEE Transactions on Knowledge and Data Engineering*, 19:8, pp. 1042–1056, 2007.

<sup>3</sup> Available at <http://code.google.com/p/miningsuite/>.

# **mir\_eval:** A TRANSPARENT IMPLEMENTATION OF COMMON MIR METRICS

**Colin Raffel<sup>1,\*</sup>, Brian McFee<sup>1,2</sup>, Eric J. Humphrey<sup>3</sup>, Justin Salamon<sup>3,4</sup>, Oriol Nieto<sup>3</sup>,  
Dawen Liang<sup>1</sup>, and Daniel P. W. Ellis<sup>1</sup>**

<sup>1</sup>LabROSA, Dept. of Electrical Engineering, Columbia University, New York

<sup>2</sup>Center for Jazz Studies, Columbia University, New York

<sup>3</sup>Music and Audio Research Lab, New York University, New York

<sup>4</sup>Center for Urban Science and Progress, New York University, New York

## ABSTRACT

Central to the field of MIR research is the evaluation of algorithms used to extract information from music data. We present `mir_eval`, an open source software library which provides a transparent and easy-to-use implementation of the most common metrics used to measure the performance of MIR algorithms. In this paper, we enumerate the metrics implemented by `mir_eval` and quantitatively compare each to existing implementations. When the scores reported by `mir_eval` differ substantially from the reference, we detail the differences in implementation. We also provide a brief overview of `mir_eval`'s architecture, design, and intended use.

## 1. EVALUATING MIR ALGORITHMS

Much of the research in Music Information Retrieval (MIR) involves the development of systems that process raw music data to produce semantic information. The goal of these systems is frequently defined as attempting to duplicate the performance of a human listener given the same task [5]. A natural way to determine a system's effectiveness might be for a human to study the output produced by the system and judge its correctness. However, this would yield only subjective ratings, and would also be extremely time-consuming when evaluating a system's output over a large corpus of music.

Instead, objective metrics are developed to provide a well-defined way of computing a score which indicates each system's output's correctness. These metrics typically involve a heuristically-motivated comparison of the system's output to a reference which is known to be correct. Over time, certain metrics have become standard for each

task, so that the performance of systems created by different researchers can be compared when they are evaluated over the same dataset [5]. Unfortunately, this comparison can be confounded by small details of the implementations or procedures that can have disproportionate impacts on the resulting scores.

For the past 10 years, the yearly Music Information Retrieval Evaluation eXchange (MIREX) has been a forum for comparing MIR algorithms over common datasets [6]. By providing a standardized shared-task setting, MIREX has become critically useful for tracking progress in MIR research. MIREX is built upon the Networked Environment for Music Analysis (NEMA) [22], a large-scale system which includes exhaustive functionality for evaluating, summarizing, and displaying evaluation results. The NEMA codebase includes multiple programming languages and dependencies (some of which, *e.g.* Matlab, are proprietary) so compiling and running it at individual sites is nontrivial. In consequence, the NEMA system is rarely used for evaluating MIR algorithms outside of the setting of MIREX [6]. Instead, researchers often create their own implementations of common metrics for evaluating their algorithms. These implementations are thus not standardized, and may contain differences in details, or even bugs, that confound comparisons.

These factors motivate the development of a standardized software package which implements the most common metrics used to evaluate MIR systems. Such a package should be straightforward to use and well-documented so that it can be easily adopted by MIR researchers. In addition, it should be community-developed and transparently implemented so that all design decisions are easily understood and open to discussion and improvement.

Following these criteria, we present `mir_eval`, a software package which intends to provide an easy and standardized way to evaluate MIR systems. This paper first discusses the architecture and design of `mir_eval` in Section 2, then, in Section 3, describes all of the tasks covered by `mir_eval` and the metrics included. In order to validate our implementation decisions, we compare `mir_eval` to existing software in Section 4. Finally, we discuss and summarize our contributions in Section 5.

\*Please direct correspondence to [craffel@gmail.com](mailto:craffel@gmail.com)



© Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel P. W. Ellis.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel P. W. Ellis.

"mir\_eval:

A Transparent Implementation of Common MIR Metrics", 15th International Society for Music Information Retrieval Conference, 2014.

## 2. `mir_eval`'S ARCHITECTURE

`mir_eval` is a Python library which currently includes metrics for the following tasks: Beat detection, chord estimation, pattern discovery, structural segmentation, melody extraction, and onset detection. Each task is given its own submodule, and each metric is defined as a separate function in each submodule. Each task submodule also includes common data pre-processing steps for the task. Every metric function includes detailed documentation, example usage, input validation, and references to the original paper which defined the metric. `mir_eval` also includes a submodule `io` which provides convenience functions for loading in task-specific data from common file formats (e.g. comma/tab separated values, `.lab` files [7], etc.). For readability, all code follows the PEP8 style guide [21]. `mir_eval`'s only dependencies outside of the Python standard library are the free and open-source SciPy/Numpy [9] and scikit-learn [15] libraries.

In order to simplify the usage of `mir_eval`, it is packaged with a set of “evaluator” scripts, one for each task. These scripts include all code necessary to load in data, pre-process it, and compute all metrics for a given task. The evaluators allow for `mir_eval` to be called directly from the command line so that no knowledge of Python is necessary. They are also distributed as executables for Windows and Mac OS X, so that `mir_eval` may be used with no dependencies installed.

## 3. TASKS INCLUDED IN `mir_eval`

In this section, we enumerate the tasks and metrics implemented in `mir_eval`. Due to space constraints, we only give high-level descriptions for each metric; for exact definitions see the references provided.

### 3.1 Beat Detection

The aim of a beat detection algorithm is to report the times at which a typical human listener might tap their foot to a piece of music. As a result, most metrics for evaluating the performance of beat tracking systems involve computing the error between the estimated beat times and some reference list of beat locations. Many metrics additionally compare the beat sequences at different metric levels in order to deal with the ambiguity of tempo [4].

`mir_eval` includes the following metrics for beat tracking, which are defined in detail in [4]: The **F-measure** of the beat sequence, where an estimated beat is considered correct if it is sufficiently close to a reference beat; **Cemgil's score**, which computes the sum of Gaussian errors for each beat; **Goto's score**, a binary score which is 1 when at least 25% of the estimated beat sequence closely matches the reference beat sequence; **McKinney's P-score**, which computes the cross-correlation of the estimated and reference beat sequences represented as impulse trains; **continuity-based scores** which compute the proportion of the beat sequence which is continuously correct; and finally the **Information Gain** of a normalized beat error histogram over a uniform distribution.

### 3.2 Chord Estimation

Despite being one of the oldest MIREX tasks, evaluation methodology and metrics for automatic chord estimation is an ongoing topic of discussion, due to issues with vocabularies, comparison semantics, and other lexicographical challenges unique to the task [14]. One source of difficulty stems from an inherent subjectivity in “spelling” a chord name and the level of detail a human observer can provide in a reference annotation [12]. As a result, a consensus has yet to be reached regarding the single best approach to comparing two sequences of chord labels, and instead are often compared over a set of rules, i.e. Root, Major-Minor, and Sevenths, with or without inversions.

To efficiently compare chords, we first separate a given chord label into its constituent parts, based on the syntax of [7]. For example, the chord label `G:maj(6)/5` is mapped to three pieces of information: the root (“G”), the root-invariant active semitones as determined by the quality shorthand (“maj”) and scale degrees (“6”), and the bass interval (“5”).

Based on this representation, we can compare an estimated chord label with a reference by the following rules as used in MIREX 2013 [2]: **Root** requires only that the roots are equivalent; **Major-Minor** includes Root, and further requires that the active semitones are equivalent subject to the reference chord quality being Maj or min; **Sevenths** follows Major-minor, but is instead subject to the reference chord quality being one of Maj, min, Maj7, min7, 7, or minmaj7; and finally, **Major-Minor-Inv** and **Sevenths-Inv** include Major-Minor and Sevenths respectively, but further require that the bass intervals are equivalent subject to the reference bass interval being an active semitone. The “subject to...” conditions above indicate that a comparison is *ignored* during evaluation if the given criteria is not satisfied.

Track-wise scores are computed by weighting each comparison by the duration of its interval, over all intervals in an audio file. This is achieved by forming the union of the boundaries in each sequence, sampling the labels, and summing the time intervals of the “correct” ranges. The cumulative score, referred to as *weighted chord symbol recall*, is tallied over a set audio files by discrete summation, where the importance of each score is weighted by the duration of each annotation [2].

### 3.3 Pattern Discovery

Pattern discovery involves the identification of musical patterns (i.e. short fragments or melodic ideas that repeat at least twice) both from audio and symbolic representations. The metrics used to evaluation pattern discovery systems attempt to quantify the ability of the algorithm to not only determine the present patterns in a piece, but also to find all of their occurrences.

Collins compiled all previously existent metrics and proposed novel ones [3] which resulted in 19 different scores, each one implemented in `mir_eval`: **Standard F-measure, Precision, and Recall**, where an estimated prototype pattern is considered correct only if it matches



(up to translation) a reference prototype pattern; **Establishment F-measure, Precision, and Recall**, which compute the number of reference patterns that were successfully found, no matter how many occurrences were found; **Occurrence F-measure, Precision, and Recall**, which measure whether an algorithm is able to retrieve all occurrences of a pattern; **Three-layer F-measure, Precision, and Recall**, which capture both the establishment of the patterns and the occurrence retrieval in a single set of scores; and the **First  $N$  patterns metrics**, which compute the target proportion establishment recall and three-layer precision for the first  $N$  patterns only in order to measure the ability of the algorithm to sort the identified patterns based on their relevance.

### 3.4 Structural Segmentation

Evaluation criteria for structural segmentation fall into two categories: boundary annotation and structural annotation. Boundary annotation is the task of predicting the times at which structural changes occur, such as when a verse transitions to a refrain. Structural annotation is the task of assigning labels to detected segments. The estimated labels may be arbitrary strings — such as  $A, B, C$ , etc. — and they need not describe functional concepts. In both tasks, we assume that annotations express a partitioning of the track into intervals.

`mir_eval` implements the following boundary detection metrics: **Boundary Detection Precision, Recall, and F-measure Scores** where an estimated boundary is considered correct if it falls within a window around a reference boundary [20]; and **Boundary Deviation** which computes median absolute time difference from a reference boundary to its nearest estimated boundary, and vice versa [20]. The following structure annotation metrics are also included: **Pairwise Classification Precision, Recall, and F-measure Scores** for classifying pairs of sampled time instants as belonging to the same structural component [10]; **Rand Index**<sup>1</sup> which clusters reference and estimated annotations and compares them by the Rand Index [17]; and the **Normalized Conditional Entropy** where sampled reference and estimated labels are interpreted as samples of random variables  $Y_R, Y_E$  from which the conditional entropy of  $Y_R$  given  $Y_E$  (**Under-Segmentation**) and  $Y_E$  given  $Y_R$  (**Over-Segmentation**) are estimated [11].

### 3.5 Melody Extraction

Melody extraction algorithms aim to produce a sequence of frequency values corresponding to the pitch of the dominant melody from a musical recording [19]. An estimated pitch series is evaluated against a reference by computing the following five measures defined in [19], first used in MIREX 2005 [16]: **Voicing Recall Rate** which computes the proportion of frames labeled as melody frames in the reference that are estimated as melody frames by the algorithm; **Voicing False Alarm Rate** which computes the proportion of frames labeled as non-melody in the reference that are

<sup>1</sup> The MIREX results page refers to Rand Index as “random clustering index”.

mistakenly estimated as melody frames by the algorithm; **Raw Pitch Accuracy** which computes the proportion of melody frames in the reference for which the frequency is considered correct (*i.e.* within half a semitone of the reference frequency); **Raw Chroma Accuracy** where the estimated and reference  $f_0$  sequences are mapped onto a single octave before computing the raw pitch accuracy; and the **Overall Accuracy**, which computes the proportion of all frames correctly estimated by the algorithm, including whether non-melody frames were labeled by the algorithm as non-melody. Prior to computing these metrics, both the estimate and reference sequences must be sampled onto the same time base.

### 3.6 Onset Detection

The goal of an onset detection algorithm is to automatically determine when notes are played in a piece of music. As is also done in beat tracking and segment boundary detection, the primary method used to evaluate onset detectors is to first determine which estimated onsets are “correct”, where correctness is defined as being within a small window of a reference onset [1]. From this, **Precision, Recall, and F-measure** scores are computed.

## 4. COMPARISON TO EXISTING IMPLEMENTATIONS

In order to validate the design choices made in `mir_eval`, it is useful to compare the scores it reports to those reported by an existing evaluation system. Beyond pinpointing intentional differences in implementation, this process can also help find and fix bugs in either `mir_eval` or the system it is being compared to.

For each task covered by `mir_eval`, we obtained a collection of reference and estimated annotations and computed or obtained a score for each metric using `mir_eval` and the evaluation system being compared to. In order to facilitate comparison, we ensured that all parameters and pre-processing used by `mir_eval` were equivalent to the reference system unless otherwise explicitly noted. Then, for each reported score, we computed the relative change between the scores as their absolute difference divided by their mean, or

$$\frac{|s_m - s_c|}{(s_m + s_c)/2}$$

where  $s_m$  is the score reported by `mir_eval` and  $s_c$  is the score being compared to. Finally, we computed the average relative change across all examples in the obtained dataset for each score.

For the beat detection, chord estimation, structural segmentation, and onset detection tasks, MIREX releases the the output of submitted algorithms, the ground truth annotations, and the reported score for each example in each data set. We therefore can directly compare `mir_eval` to MIREX for these tasks by collecting all reference and estimated annotations, computing each metric for each example using identical pre-processing and parameters as appropriate, and comparing the result to the score reported by

MIREX. We chose to compare against the results reported in MIREX 2013 for all tasks.

In contrast to the tasks listed above, MIREX does not release ground truth annotations or algorithm output for the melody extraction and pattern discovery tasks. As a result, we compared `mir_eval`'s output on smaller development datasets for these tasks. For melody extraction, the ADC2004 dataset used by MIREX is publicly available. We performed melody extraction using the "SG2" algorithm evaluated in 2011 [18] and compared `mir_eval`'s reported scores to those of MIREX. For pattern discovery, we used the development dataset released by Collins [3] and used the algorithms submitted by Nieto and Farhood [13] for MIREX 2013 to produce estimated patterns. We evaluated the estimated patterns using the MATLAB code released by Collins [3]. The number of algorithms, examples, and total number of scores for all tasks are summarized in Table 1.

Task	Algorithms	Examples	Scores
Beat Detection	20	679	13580
Segmentation	8	1397	11176
Onset Detection	11	85	935
Chord Estimation	12	217	2604
Melody	1	20	20
Pattern Discovery	4	5	20

**Table 1.** Number of scores collected for each task for comparison against `mir_eval`.

The resulting average relative change for each metric is presented in Table 2. The average relative change for all of the pattern discovery metrics was 0, so they are not included in this table. For many of the other metrics, the average relative change was less than a few tenths of a percent, indicating that `mir_eval` is equivalent up to rounding/precision errors. In the following sections, we enumerate the known implementation differences which account for the larger average relative changes.

#### 4.1 Non-greedy matching of events

In the computation of the F-measure, Precision and Recall metrics for the beat tracking, boundary detection, and onset detection tasks, an estimated event is considered correct (a "hit") if it falls within a small window of a reference event. No estimated event is counted as a hit for more than one reference event, and vice versa. In MIREX, this assignment is done in a greedy fashion, however in `mir_eval` we use an optimal matching strategy. This is accomplished by computing a maximum bipartite matching between the estimated events and the reference events (subject to the window constraint) using the Hopcroft-Karp algorithm [8]. This explains the observed discrepancy between `mir_eval` and MIREX for each of these metrics. In all cases where the metric differs, `mir_eval` reports a higher score, indicating that the greedy matching strategy was suboptimal.

#### 4.2 McKinney's P-score

When computing McKinney's P-score [4], the beat sequences are first converted to impulse trains sampled at a 10 millisecond resolution. Because this sampling involves quantizing the beat times, shifting both beat sequences by a constant offset can result in substantial changes in the P-score. As a result, in `mir_eval`, we normalize the beat sequences by subtracting from each reference and estimated beat location the minimum beat location in either series. In this way, the smallest beat in the estimated and reference beat sequences is always 0 and the metric remains the same even when both beat sequences have a constant offset applied. This is not done in MIREX (which uses the Beat Evaluation Toolbox [4]), and as a result, we observe a considerable average relative change for the P-score metric.

#### 4.3 Information Gain

The Information Gain metric [4] involves the computation of a histogram of the per-beat errors. The Beat Evaluation Toolbox (and therefore MIREX) uses a non-uniform histogram binning where the first, second and last bins are smaller than the rest of the bins while `mir_eval` uses a standard uniformly-binned histogram. As a result, the Information Gain score reported by `mir_eval` differs substantially from that reported by MIREX.

#### 4.4 Segment Boundary Deviation

When computing the median of the absolute time differences for the boundary deviation metrics, there are often an even number of reference or estimated segment boundaries, resulting in an even number of differences to compute the median over. In this case, there is no "middle" element to choose as the median. `mir_eval` follows the typical convention of computing the mean of the two middle elements in lieu of the median for even-length sequences, while MIREX chooses the larger of the two middle elements. This accounts for the discrepancy in the reference-to-estimated and estimated-to-reference boundary deviation metrics.

#### 4.5 Interval Sampling for Structure Metrics

When computing the structure annotation metrics (Pairwise Precision, Recall, and F-measure, Rand Index, and Normalized Conditional Entropy Over- and Under-Segmentation Scores), the reference and estimated labels must be sampled to a common time base. In MIREX, a fixed sampling grid is used for the Rand Index and pairwise classification metrics, but a different sampling rate is used for each, while a fixed number of samples is used for the conditional entropy scores. In `mir_eval`, the same fixed sampling rate of 100 milliseconds is used for all structure annotation metrics, as specified in [23].

Furthermore, in MIREX the start and end time over which the intervals are sampled depends on both the reference and estimated intervals while `mir_eval` always samples with respect to the reference to ensure fair comparison across multiple estimates. This additionally requires

Beat Detection									
F-measure	Cemgil	Goto	P-score	CMLc	CMLt	AMLc	AMLt	In. Gain	
0.703%	0.035%	0.054%	0.877%	0.161%	0.143%	0.137%	0.139%	9.174%	
Structural Segmentation									
NCE-Over	NCE-under	Pairwise F	Pairwise P	Pairwise R	Rand	F@.5	P@.5	R@.5	
3.182%	11.082%	0.937%	0.942%	0.785%	0.291%	0.429%	0.088%	1.021%	
Structural Segmentation (continued)					Onset Detection				
F@3	P@3	R@3	Ref-est dev.	Est-ref dev.	F-measure	Precision	Recall		
0.393%	0.094%	0.954%	0.935%	0.000%	0.165%	0.165%	0.165%		
Chord Estimation					Melody Extraction				
Root	Maj/min	Maj/min + Inv	7ths	7ths + Inv	Overall	Raw pitch	Chroma	Voicing R	Voicing FA
0.007%	0.163%	1.005%	0.483%	0.899%	0.070%	0.087%	0.114%	0.000%	10.095%

**Table 2.** Average relative change for every metric in `mir_eval` when compared to a pre-existing implementation. The average relative change for all pattern discovery metrics was 0, so they are not shown here.

that estimated intervals are adjusted to span the exact duration specified by the reference intervals. This is done by adding synthetic intervals when the estimated intervals do not span the reference intervals or otherwise trimming estimated intervals. These differences account for the average relative changes for the structure annotation metrics.

#### 4.6 Segment Normalized Conditional Entropy

When adding intervals to the estimated annotation as described above, `mir_eval` ensures that the labels do not conflict with existing labels. This has the effect of changing the normalization constant in the Normalized Conditional Entropy scores. Furthermore, when there’s only one label, the Normalized Conditional Entropy scores are not well defined. MIREX assigns a score of 1 in this case; `mir_eval` assigns a score of 0. This results in a larger average change for these two metrics.

#### 4.7 Melody Voicing False Alarm Rate

When a reference melody annotation contains no unvoiced frames, the Voicing False Alarm Rate is not well defined. MIREX assigns a score of 1 in this case, while `mir_eval` assigns a score of 0 because, intuitively, no reference unvoiced frames could be estimated, so no false alarms should be reported. In the data set over which the average relative change for the melody metrics was computed, one reference annotation contained no unvoiced frames. This discrepancy caused a large inflation of the average relative change reported for the Voicing False Alarm Rate due to the small number of examples in our dataset.

#### 4.8 Weighted Chord Symbol Recall

The non-negligible average relative changes seen in the chord metrics are caused by two main sources of ambiguity. First, we find some chord labels in the MIREX reference annotations lack well-defined, *i.e.* singular, mappings into a comparison space. One such example is `D:maj(*1)/#1`.

While the quality shorthand indicates “major”, the asterisk implies the root is omitted and thus it is unclear whether `D:maj(*1)/#1` is equivalent to `D:maj1`. Second, and more importantly, such chords are likely ignored during evaluation, and we are unable to replicate the exact exclusion logic used by MIREX. This has proven to be particularly difficult in the two inversion rules, and manifests in Table 2. For example, `Bb:maj(9)/9` was *not* excluded from the MIREX evaluation, contradicting the description provided by the task specification [2]. This chord alone causes an observable difference between `mir_eval` and MIREX’s results.

## 5. TOWARDS TRANSPARENCY AND COMMUNITY INVOLVEMENT

The results in Section 4 clearly demonstrate that differences in implementation can lead to substantial differences in reported scores. This corroborates the need for transparency and community involvement in comparative evaluation. The primary motivation behind developing `mir_eval` is to establish an open-source, publicly refined implementation of the most common MIR metrics. By encouraging MIR researchers to use the same easily understandable evaluation codebase, we can ensure that different systems are being compared fairly.

While we have given thorough consideration to the design choices made in `mir_eval`, we recognize that standards change over time, new metrics are proposed each year, and that only a subset of MIR tasks are currently implemented in `mir_eval`. Towards this end, `mir_eval` is hosted on Github,<sup>2</sup> which provides a straightforward way of proposing changes and additions to the codebase using the Pull Request feature. With active community participation, we believe that `mir_eval` can ensure that MIR research converges on a standard methodology for evaluation.

<sup>2</sup>[http://github.com/craffel/mir\\_eval](http://github.com/craffel/mir_eval)

## 6. ACKNOWLEDGEMENTS

The authors would like to thank Matthew McVicar for helpful advice on comparing chord labels and Tom Collins for sharing his MATLAB implementation to evaluate musical patterns. Support provided in part by The Andrew W. Mellon Foundation and the National Science Foundation, under grants IIS-0844654 and IIS-1117015.

## 7. REFERENCES

- [1] S. Böck, F. Krebs, and M. Schedl. Evaluating the online capabilities of onset detection methods. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 49–54, 2012.
- [2] K. Choi and J. A. Burgoyne. MIREX task: Audio chord estimation. [http://www.music-ir.org/mirex/wiki/2013:Audio\\_Chord\\_Estimation](http://www.music-ir.org/mirex/wiki/2013:Audio_Chord_Estimation), 2013. Accessed: 2014-04-30.
- [3] T. Collins. MIREX task: Discovery of repeated themes & sections. [http://www.music-ir.org/mirex/wiki/2013:Discovery\\_of\\_Repeated\\_Themes\\_&\\_Sections](http://www.music-ir.org/mirex/wiki/2013:Discovery_of_Repeated_Themes_&_Sections), 2013. Accessed: 2014-04-30.
- [4] M. E. P. Davies, N. Degara, and M. D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. Technical Report C4DM-TR-09-06, Centre for Digital Music, Queen Mary University of London, London, England, October 2009.
- [5] J. S. Downie. Toward the scientific evaluation of music information retrieval systems. In *Proceedings of the 4th International Society for Music Information Retrieval Conference (ISMIR)*, pages 25–32, 2003.
- [6] J. S. Downie. The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008.
- [7] C. Harte. *Towards Automatic Extraction of Harmony Information from Music Signals*. PhD thesis, Queen Mary University of London, August 2010.
- [8] J. E. Hopcroft and R. M. Karp. An  $n^2$  algorithm for maximum matchings in bipartite graphs. *SIAM Journal on computing*, 2(4):225–231, 1973.
- [9] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.
- [10] M. Levy and M. Sandler. Structural segmentation of musical audio by constrained clustering. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):318–326, 2008.
- [11] H. M. Lukashevich. Towards quantitative measures of evaluating song segmentation. In *Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR)*, pages 375–380, 2008.
- [12] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. De Bie. Understanding effects of subjectivity in measuring chord estimation accuracy. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(12):2607–2615, 2013.
- [13] O. Nieto and M. Farbood. Discovering musical patterns using audio structural segmentation techniques. *7th Music Information Retrieval Evaluation eXchange (MIREX)*, 2013.
- [14] J. Pauwels and G. Peeters. Evaluating automatically estimated chord sequences. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 749–753. IEEE, 2013.
- [15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [16] G. E. Poliner, D. P. W. Ellis, A. F. Ehmann, E. Gómez, S. Streich, and B. Ong. Melody transcription from music audio: Approaches and evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4):1247–1256, 2007.
- [17] W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.
- [18] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770, Aug. 2012.
- [19] J. Salamon, E. Gómez, D. P. W. Ellis, and G. Richard. Melody extraction from polyphonic music signals: Approaches, applications and challenges. *IEEE Signal Processing Magazine*, 31(2):118–134, March 2014.
- [20] D. Turnbull, G. Lanckriet, E. Pampalk, and M. Goto. A supervised approach for detecting boundaries in music using difference features and boosting. In *Proceedings of the 8th International Society for Music Information Retrieval Conference (ISMIR)*, pages 51–54, 2007.
- [21] G. van Rossum, B. Warsaw, and N. Coghlan. PEP 8—style guide for python code. <http://www.python.org/dev/peps/pep-0008>, 2001. Accessed: 2014-04-30.
- [22] K. West, A. Kumar, A. Shirk, G. Zhu, J. S. Downie, A. Ehmann, and M. Bay. The networked environment for music analysis (nema). In *IEEE 6th World Congress on Services (SERVICES 2010)*, pages 314–317. IEEE, 2010.
- [23] C. Willis. MIREX task: Structural segmentation. [http://www.music-ir.org/mirex/wiki/2013:Structural\\_Segmentation](http://www.music-ir.org/mirex/wiki/2013:Structural_Segmentation), 2013. Accessed: 2014-04-30.

# COMPUTATIONAL MODELING OF INDUCED EMOTION USING GEMS

**Anna Aljanaki**  
Utrecht University  
A.Aljanaki@uu.nl

**Frans Wiering**  
Utrecht University  
F.Wiering@uu.nl

**Remco C. Veltkamp**  
Utrecht University  
R.C.Veltkamp@uu.nl

## ABSTRACT

Most researchers in the automatic music emotion recognition field focus on the two-dimensional valence and arousal model. This model though does not account for the whole diversity of emotions expressible through music. Moreover, in many cases it might be important to model induced (felt) emotion, rather than perceived emotion. In this paper we explore a multidimensional emotional space, the Geneva Emotional Music Scales (GEMS), which addresses these two issues. We collected the data for our study using a game with a purpose. We exploit a comprehensive set of features from several state-of-the-art toolboxes and propose a new set of harmonically motivated features. The performance of these feature sets is compared. Additionally, we use expert human annotations to explore the dependency between musicologically meaningful characteristics of music and emotional categories of GEMS, demonstrating the need for algorithms that can better approximate human perception.

## 1. INTRODUCTION

Most of the effort in automatic music emotion recognition (MER) is invested into modeling two dimensions of musical emotion: valence (positive vs. negative) and arousal (quiet vs. energetic) (V-A) [16]. Regardless of the popularity of V-A, the question of which model of musical emotion is best has not yet been solved. The difficulty is, on one hand, in creating a model that reflects the complexity and subtlety of the emotions that music can demonstrate, while on the other hand providing a linguistically unambiguous framework that is convenient to use to refer to such a complex non-verbal concept as musical emotion. Categorical models, possessing few (usually 4–6, but sometimes as many as 18) [16] classes are oversimplifying the problem, while V-A has been criticized for a lack of discerning capability, for instance in the case of fear and anger. Other pitfalls of V-A model are that it was not created specifically for music, and is especially unsuited to describe induced (felt) emotion, which might be important for some MER tasks, e.g. composing a playlist using emo-

tional query and in any other cases when the music should create a certain emotion in listener. The relationship between induced and perceived emotion is not yet fully understood, but they are surely not equivalent — one may listen to angry music without feeling angry, but instead feel energetic and happy. It was demonstrated that some types of emotions (especially negative ones) are less likely to be induced by music, though music can express them [17].

In this paper we address the problem of modeling induced emotion by using GEMS. GEMS is a domain-specific categorical emotional model, developed by Zentner et al. [17] specifically for music. The model was derived via a three-stage collection and filtering of terms which are relevant to musical emotion, after which the model was verified in a music listening-context. Being based on emotional ontology which comes from listeners, it must be a more convenient tool to retrieve music than, for instance, points on a V-A plane. The full GEMS scale consists of 45 terms, with shorter versions of 25 and 9 terms. We used the 9-term version of GEMS (see Table 1) to collect data using a game with a purpose.

Emotion induced by music depends on many factors, some of which are external to music itself, such as cultural and personal associations, social listening context, the mood of the listener. Naturally, induced emotion is also highly subjective and varies a lot across listeners, depending on their musical taste and personality. In this paper we do not consider all these factors and will only deal with the question to which extent induced emotion can be modeled using acoustic features only. Such a scenario, when no input from the end-user (except for, maybe, genre preferences) is available, is plausible for a real-world application of a MER task. We employ four different feature sets: low-level features related to timbre and energy, extracted using OpenSmile,<sup>1</sup> and a more musically motivated feature set, containing high-level features, related to mode, rhythm, and harmony, from the MIRToolbox,<sup>2</sup> PsySound<sup>3</sup> and SonicAnnotator.<sup>4</sup> We also enhance the performance of the latter by designing new features that describe the harmonic content of music. As induced emotion is a highly subjective phenomenon, the performance of the model will be confounded by the amount of agreement between listeners which provide the ground-truth. As far as audio-based features are not perfect yet, we try to estimate this upper bound for our data by employing human experts, who an-



© Anna Aljanaki, Frans Wiering, Remco C. Veltkamp.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Anna Aljanaki, Frans Wiering, Remco C. Veltkamp. “COMPUTATIONAL MODELING OF INDUCED EMOTION USING GEMS”, 15th International Society for Music Information Retrieval Conference, 2014.

<sup>1</sup> opensmile.sourceforge.net

<sup>2</sup> jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox

<sup>3</sup> pssystem.wikiidot.com

<sup>4</sup> isophonics.net/SonicAnnotator

notate a subset of the data with ten musicological features.

**Contribution.** This paper explores computational approaches to modeling induced musical emotion and estimates the upper boundary for such a task, in case when no personal or contextual factors can be taken into account. It is also suggested that more than two dimensions are necessary to represent musical emotion adequately. New features for harmonic description of music are proposed.

## 2. RELATED WORK

Music emotion recognition is a young, but fast-developing field. Reviewing it in its entirety is out of scope of this paper. For such a review we are referring to [16]. In this section we will briefly summarize the commonly used methods and approaches that are relevant to this paper.

Automatic MER can be formulated both as a regression and classification problem, depending on the underlying emotional model. As such, the whole entirety of machine learning algorithms can be used for MER. In this paper we are employing Support Vector Regression (SVR), as it demonstrated good performance [7, 15] and can learn complex non-linear dependencies from the feature space. Below we describe several MER systems.

In [15], V-A is modeled with acoustic features (spectral contrast, DWCH and other low-level features from Marsyas and PsySound) using SVR, achieving performance of 0.76 for arousal and 0.53 for valence (in terms of Pearson's  $r$  here and further). In [7], five dimensions (basic emotions) were modeled with a set of timbral, rhythmic and tonal features, using SVR. The performance varied from 0.59 to 0.69. In [5], pleasure, arousal and dominance were modeled with AdaBoost.RM using features extracted from audio, MIDI and lyrics. An approach based on audio features only performed worse than multimodal features approach (0.4 for valence, 0.72 for arousal and 0.62 for dominance).

Various chord-based statistical measures have already been employed for different MIR tasks, such as music similarity or genre detection. In [3], chordal features (longest common chord sequence and histogram statistics on chords) were used to find similar songs and to estimate their emotion (in terms of valence) based on chord similarity. In [9], chordal statistics is used for MER, but the duration of chords is not taken into account, which we account for in this paper. Interval-based features, described here, to our knowledge have not been used before.

A computational approach to modeling musical emotion using GEMS has not been adopted before. In [11], GEMS was used to collect data dynamically on 36 musical excerpts. Listener agreement was very good (Cronbach's alpha ranging from 0.84 to 0.98). In [12], GEMS is compared to a three-dimensional (valence-arousal-tension) and categorical (anger, fear, happiness, sadness, tenderness) models. The consistency of responses is compared, and it is found that GEMS categories have both some of the highest (joyful activation, tension) and some of the lowest (wonder, transcendence) agreement. It was also found that GEMS categories are redundant, and valence

and arousal dimensions account for 89% of variance. That experiment, though, was performed on 16 musical excerpts only, and the excerpts were selected using criteria based on V-A model, which might have resulted in bias.

## 3. DATA DESCRIPTION

The dataset that we analyze consists of 400 musical excerpts (44100 Hz, 128 kbps). Each excerpt is 1 minute long (except for 4 classical pieces which were shorter than 1 minute). It is evenly split (100 pieces per genre) by four genres (classical, rock, pop and electronic music). In many studies, musical excerpts are specially selected for their strong emotional content that best fits the chosen emotional model, and only the excerpts that all the annotators agree upon, are left. In our dataset we maintain a good ecological validity by selecting music randomly from a Creative Commons recording label Magnatune, only making sure that the recordings are of good quality.

Based on conclusions from [11, 12], we renamed two GEMS categories by replacing them with one of their sub-categories (*wonder* was replaced with *amazement*, and *transcendence* with *solemnity*). Participants were asked to select no more than three emotional terms from a list of nine. They were instructed to describe how music made them feel, and not what it expressed, and were encouraged to do so in a game context [1]. All the songs were annotated by at least 10 players (mean = 20.8, SD = 14).

The game with a purpose was launched and advertised through social networks. The game,<sup>5</sup> as well as annotations and audio,<sup>6</sup> are accessible online. More than 1700 players have contributed. The game was streaming music for 138 hours in total. A detailed description and analysis of the data can be found in [1] or in a technical report. [2]

We are not interested in modeling irritation from non-preferred music, but rather differences in emotional perception across listeners that come from other factors. We introduce a question to report disliking the music and discard such answers. We also clean the data by computing Fleiss's kappa on all the annotations for every musical excerpt, and discarding the songs with negative kappa (this indicates that the answers are extremely inconsistent (33 songs)). Fleiss's kappa is designed to estimate agreement, when the answers are binary or categorical. We use this very loose criteria, as it is expected to find a lot of disagreement. We retain the remaining 367 songs for analysis.

The game participants were asked to choose several categories from a list, but for the purposes of modeling we translate the annotations into a continuous space by using the following equation:

$$\text{score}_{ij}^1 = \frac{1}{n} \sum_{k=1}^n a_k, \quad (1)$$

where  $\text{score}_{ij}^1$  is an estimated value of emotion  $i$  for song  $j$ ,  $a_k$  is the answer of the  $k$ -th participant on a question whether emotion  $i$  is present in song  $j$  or not (answer is

<sup>5</sup> [www.emotify.org](http://www.emotify.org)

<sup>6</sup> [www.projects.science.uu.nl/memotion/emotifydata/](http://www.projects.science.uu.nl/memotion/emotifydata/)

	C1	C2	C3
Amazement	0.01	-0.73	-0.07
Solemnity	-0.07	0.12	0.89
Tenderness	0.75	0.19	-0.22
Nostalgia	0.57	0.46	-0.41
Calmness	0.80	0.22	0.28
Power	-0.80	-0.17	-0.06
Joyful activation	-0.37	-0.74	-0.32
Tension	-0.72	0.20	0.30
Sadness	0.13	0.80	-0.05

Table 1. PCA on the GEMS categories.

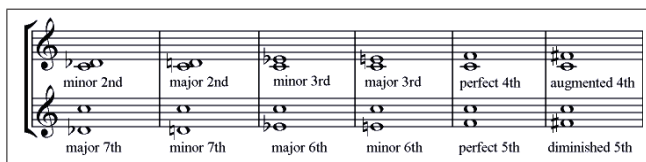


Figure 1. Intervals and their inversions.

either 0 or 1), and  $n$  is the total number of participants, who listened to song  $j$ .

The dimensions that we obtain are not orthogonal: most of them are somewhat correlated. To determine the underlying structure, we perform Principal Components Analysis. According to a Scree test, three underlying dimensions were found in the data, which together explain 69% of variance. Table 1 shows the three-component solution rotated with varimax. The first component, which accounts for 32% of variance, is mostly correlated with calmness vs. power, the second (accounts for 23% of variance) with joyful activation vs. sadness, and the third (accounts for 14% of variance) with solemnity vs. nostalgia. This suggests that the underlying dimensional space of GEMS is three-dimensional. We might suggest that it resembles valence-arousal-triviality model [13].

#### 4. HARMONIC FEATURES

It has been repeatedly shown that valence is more difficult to model than arousal. In this section we describe features, that we added to our dataset to improve prediction of modality in music.

Musical chords, as well as intervals are known to be important for affective perception of music [10], as well as other MIR tasks. Chord and melody based features have been successfully applied to genre recognition of symbolically represented music [8]. We compute statistics on the intervals and chords occurring in the piece.

##### 4.1 Interval Features

We segment audio, using local peaks in the harmonic change detection function (HCDF) [6]. HCDF describes tonal centroid fluctuations. The segments that we obtain are mostly smaller than 1 second and reflect single notes, chords or intervals. Based on the wrapped chromagrams

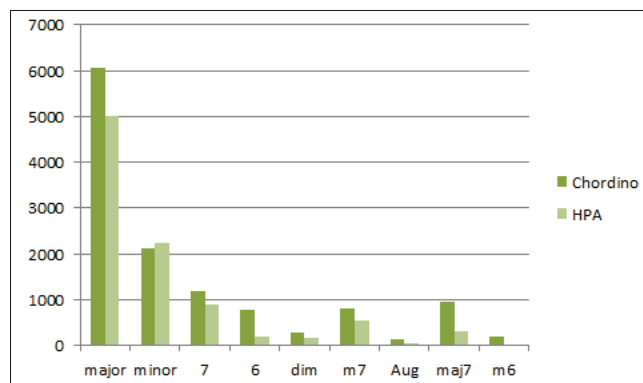


Figure 2. Distribution of chords (Chordino and HPA).

computed from the spectrum of this segments, we select two highest (energy-wise) peaks and compute the interval between them. For each interval, we compute its combined duration, weighted by its loudness (expressed by energy of the bins). Then, we sum up this statistics for intervals and their inversions. Figure 1 illustrates the concept (each bar corresponds to the musical representation of a feature that we obtain). As there are 6 distinct intervals with inversions, we obtain 6 features. We expect that augmented fourths and fifths (tritone) could reflect tension, contrary to perfect fourths and fifths. The proportion of minor thirds and major sixths, as opposed to proportion of major thirds and minor sixths, could reflect the modality. The interval-inversion pairs containing seconds are rather unrestful.

##### 4.2 Chord Features

To extract chord statistics, we used 2 chord extraction tools, HPA<sup>7</sup> (Harmonic Progression Analyzer) and Chordino<sup>8</sup> plugins for Sonic Annotator. The first plugin provides 8 types of chords: major, minor, seventh, major and minor seventh, diminished, sixth and augmented. The second plugin, in addition to these eight types, also provides minor sixth and slash chords (chords for which bass note is different from the tonic, and might as well not belong to the chord). The chords are annotated with their onsets and offsets. After experimentation, only the chords from Chordino were left, because those demonstrated more correlation with the data. We computed the proportion of each type of chord in the dataset, obtaining nine new features. The slash chords were discarded by merging them with their base chord (e.g., Am/F chord is counted as a minor chord). The distribution of chords was uneven, with major chords being in majority (for details see Figure 2). Examining the accuracy of these chord extraction tools was not our goal, but the amount of disagreement between the two tools could give an idea about that (see Figure 2). From our experiments we concluded that weighting the chords by their duration is an important step, which improves the performance of chord histograms.

<sup>7</sup> patterns.enm.bris.ac.uk/hpa-software-package

<sup>8</sup> isophonics.net/nlms-chroma

	Tempo	Articulation	Rhythmic complexity	Mode	Intensity	Tonalness	Pitch	Melody	Rhythmic clarity
Amazement	0.50	-0.37	*0.27	**0.24	*0.27				
Solemnity	-0.44	0.39		-0.45					-0.34
Tenderness	-0.48	0.56		0.30	-0.48	*0.29	0.44	0.54	
Nostalgia	-0.47	-0.57			-0.30	*0.28	*0.27	0.50	
Calmness	-0.64	0.48			-0.50		0.36		
Power	0.39	-0.35		*-0.27	0.51		-0.47	-0.43	
Joyful activation	0.76	-0.70	*0.27	**0.24	0.41				0.31
Tension		-0.36		-0.36		-0.47	-0.44	-0.66	
Sadness	-0.45	0.51	-0.38	** -0.23	** -0.24			*0.27	

**Table 2.** Correlations between manually assessed factors and emotional categories.

## 5. MANUALLY ASSESSED FEATURES

In this section we describe an additional feature set that we composed using human experts, and explain the properties of GEMS categories through perceptual musically motivated factors. Because of huge time load that manual annotation creates we only could annotate part of the data (60 pieces out of 367).

### 5.1 Procedure

Three musicians (26–61 years, over 10 years of formal musical training) annotated 60 pieces (15 pieces from each genre) from the dataset with 10 factors, on a scale from 1 to 10. The meaning of points on the scale was different for each factor (for instance, for tempo 1 would mean ‘very slow’ and 10 would mean ‘very fast’). The list of factors was taken from the study of Wedin [13]: tempo (slow—fast), articulation (staccato—legato), mode (minor—major), intensity (pp—ff), tonalness (atonal—tonal), pitch (bass—treble), melody (unmelodious—melodious), rhythmic clarity (vague—firm). We added rhythmic complexity (simple—complex) to this list, and eliminated style (date of composition) and type (serious—popular) from it.

### 5.2 Analysis

After examining correlations with the data, one of the factors was discarded as non-informative (simple or complex harmony). This factor lacked consistency between annotators as well. Table 2 shows the correlations (Spearman’s  $\rho$ ) between manually assessed factors and emotional categories. We used a non-parametric test, because distribution of emotional categories is not normal, skewed towards smaller values (emotion was more often not present than present). All the correlations are significant with p-value < 0.01, except for the ones marked with asterisk, which are significant with p-value < 0.05. The values that are absent or marked with double asterisks failed to reach statistical significance, but some of them are still listed, because they illustrate important trends which are very probable to reach significance should we have more data.

Many GEMS categories were quite correlated (*tenderness* and *nostalgia*:  $r = 0.5$ , *tenderness* and *calmness*:

$r = 0.52$ , *power* and *joyful activation*:  $r = 0.4$ ). All of these have, however, musical characteristics that allow listeners to differentiate them, as we will see below.

Both *nostalgia* and *tenderness* correlate with slow tempo and legato articulation, but *tenderness* is also correlated with higher pitch, major mode, and legato articulation (as opposed to staccato for *nostalgia*). *Calmness* is characterized by slow tempo, legato articulation and smaller intensity, similarly to *tenderness*. But *tenderness* features a correlation with melodiousness and major mode as well. Both *power* and *joyful activation* are correlated with fast tempo, and intensity, but *power* is correlated with minor mode and *joyful activation* with major mode.

As we would expect, *tension* is strongly correlated with non-melodiousness and atonality, lower pitch and minor mode. *Sadness*, strangely, is much less correlated with mode, but it more characterized by legato articulation, slow tempo and smaller rhythmic complexity.

## 6. EVALUATION

### 6.1 Features

We use four toolboxes for MIR to extract features from audio: MIRToolbox, OpenSmile, PsySound and two VAMP plugins for SonicAnnotator. We also extract harmonic features, described in Section 4. These particular tools are chosen because the features they provide were specially designed for MER. MIRToolbox was conceived as a tool for investigating a relationship between emotion and features in music. OpenSmile combines features from Speech Processing and MIR and demonstrated good performance on cross-domain emotion recognition [14]. We evaluate three following computational and one human-assessed feature sets:

1. **MIRToolbox + PsySound**: 40 features from MIRToolbox (spectral features, HCDF, mode, inharmonicity etc.) and 4 features related to loudness from PsySound (using the loudness model of Chalupper and Fastl).
2. **OpenSmile**: 6552 low-level supra-segmental features (chroma features, MFCCs or energy, and statistical



Feature set	MIRToolbox + PsySound		OpenSmile		MP + Harm		Musicological	
	$r$	RMSE	$r$	RMSE	$r$	RMSE	$r$	RMSE
Amazement	.07 ± .18	.99 ± .16	.19 ± .15	.95 ± .13	.16 ± .15	1.05 ± .11	.35 ± .30	.85 ± .24
Solemnity	.35 ± .14	.80 ± .09	.42 ± .16	.95 ± .13	.43 ± .08	.89 ± .15	.60 ± .24	.84 ± .22
Tenderness	.50 ± .10	.84 ± .10	.52 ± .12	.95 ± .07	.57 ± .12	.85 ± .18	.87 ± .09	.50 ± .19
Nostalgia	.53 ± .16	.82 ± .12	.53 ± .18	.89 ± .07	.45 ± .12	.88 ± .10	.69 ± .24	.69 ± .16
Calmness	.55 ± .14	.83 ± .09	.55 ± .16	.89 ± .07	.60 ± .11	.78 ± .09	.71 ± .17	.70 ± .16
Power	.48 ± .18	.82 ± .13	.56 ± .09	.84 ± .09	.56 ± .11	.80 ± .16	.65 ± .13	.78 ± .26
Joyful activation	.63 ± .08	.77 ± .11	.68 ± .08	.80 ± .08	.66 ± .12	.75 ± .11	.74 ± .28	.58 ± .15
Tension	.38 ± .14	.87 ± .20	.41 ± .19	.94 ± .19	.46 ± .11	.85 ± .13	.58 ± .35	.71 ± .36
Sadness	.41 ± .13	.87 ± .11	.40 ± .18	.96 ± .18	.42 ± .13	.88 ± .12	.39 ± .28	.93 ± .20

**Table 3.** Evaluation of 4 feature sets on the data. Pearson’s  $r$  and RMSE with their standard deviations (across cross-validation rounds) are shown.

functionals applied to them (such as mean, standard deviation, inter-quartile range, skewness, kurtosis etc.).

3. **MP+Harm:** to evaluate performance of harmonic features, we add them to the first feature set. It doesn’t make sense to evaluate them alone, because they only cover one aspect of music.
4. **Musicological feature set:** these are 9 factors of music described in section 5.

## 6.2 Learning Algorithm

After trying SVR, Gaussian Processes Regression and linear regression, we chose SVR (the LIBSVM implementation<sup>9</sup>) as a learning algorithm. The best performance was achieved using the RBF kernel, which is defined as follows:

$$k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \quad (2)$$

where  $\gamma$  is a parameter given to SVR. All the parameters,  $C$  (error cost),  $\epsilon$  (slack of the loss function) and  $\gamma$ , are optimized with grid-search for each feature set (but not for each emotion). To select an optimal set of features, we use recursive feature elimination (RFE). RFE assigns weights to features based on output from a model, and removes attributes until performance is no longer improved.

## 6.3 Evaluation

We evaluate the performances of the four systems using 10-fold cross-validation, splitting the dataset by artist (there are 140 distinct artists per 400 songs). If a song from artist A appears in the training set, there will be no songs from this artist in the test set. Table 3 shows evaluation results. The accuracy of the models differs greatly per category, while all the feature sets demonstrate the same pattern of success and failure (for instance, perform badly on *amazement* and well on *joyful activation*). This reflects the fact that these two categories are very different in their subjectiveness. Figure 3 illustrates the performance of the

systems ( $r$ ) for each of the categories and Cronbach’s alpha (which measures agreement) computed on listener’s answers (see [1] for more details), and shows that they are highly correlated. The low agreement between listeners results in conflicting cues, which limit model performance.

In general, the accuracy is comparable to accuracy achieved for perceived emotion by others [5, 7, 15], though it is somewhat lower. This might be explained by the fact that all the categories contain both arousal and valence components, and induced emotion annotations are less consistent. In [7], *tenderness* was predicted with  $R = 0.67$ , as compared to  $R = 0.57$  for **MP+Harm** system in our case. For *power* and *joyful activation*, the predictions from the best systems (**MP+Harm** and **OpenSmile**) demonstrated 0.56 and 0.68 correlation with the ground truth, while in [5, 15] it was 0.72 and 0.76 for arousal.

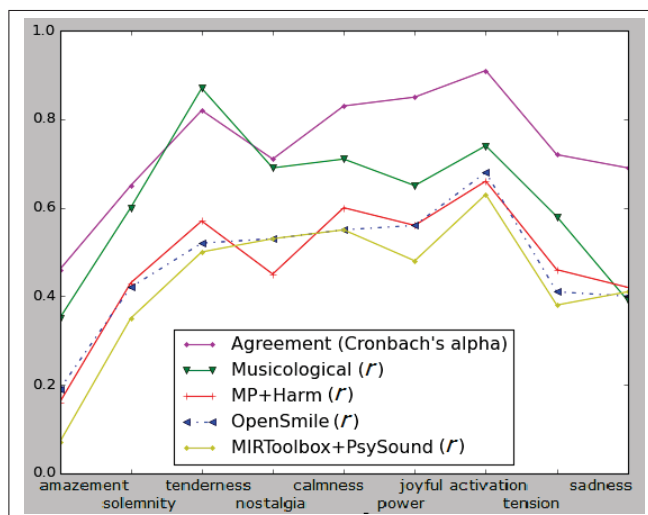
The performance of all the three computational models is comparable, though **MP+Harm** model performs slightly better in general. Adding harmonic features improves average performance from 0.43 to 0.47, and performance of the best system (**MP+Harm**) decreases to 0.35 when answers from people who disliked the music are not discarded. As we were interested in evaluating the new features, we checked which features were considered important by RFE. For *power*, the tritone proportion was important (positively correlated with power), for *sadness*, the proportion of minor chords, for *tenderness*, the proportion of seventh chords (negatively correlates), for *tension*, the proportion of tritones, for *joyful activation*, the proportion of seconds and inversions (positive correlation).

The **musicological** feature set demonstrates the best performance as compared to all the features derived from signal-processing, demonstrating that our ability to model human perception is not yet perfect.

## 7. CONCLUSION

We analyze the performance of audio features on prediction of induced musical emotion. The performance of the best system is somewhat lower than can be achieved for perceived emotion recognition. We conduct PCA and find

<sup>9</sup> www.csie.ntu.edu.tw/~cjlin/libsvm/



**Figure 3.** Comparison of systems' performance with Cronbach's alpha per category.

three dimensions in the GEMS model, which are best explained by axes spanning *calmness—power*, *joyful activation—sadness* and *solemnity—nostalgia*). This finding is supported by other studies in the field [4, 13].

We conclude that it is possible to predict induced musical emotion for some emotional categories, such as *tenderness* and *joyful activation*, but for many others it might not be possible without contextual information. We also show that despite this limitation, there is still room for improvement by developing features that can better approximate human perception of music, which can be pursued in future work on emotion recognition.<sup>10</sup>

## 8. REFERENCES

- [1] A. Aljanaki, D. Bountouridis, J.A. Burgoyne, J. van Balen, F. Wiering, H. Honing, and R. C. Veltkamp: "Designing Games with a Purpose for Data Collection in Music Research. Emotify and Hooked: Two Case Studies", *Proceedings of Games and Learning Alliance Conference*, 2013.
- [2] A. Aljanaki, F. Wiering, and R. C. Veltkamp: "Collecting annotations for induced musical emotion via online game with a purpose Emotify", [www.cs.uu.nl/research/techreps/UU-CS-2014-015.html](http://www.cs.uu.nl/research/techreps/UU-CS-2014-015.html), 2014.
- [3] H.-T. Cheng, Y.-H. Yang, Y.-C. Lin, I.-B. Liao, and H. H. Chen: "Automatic chord recognition for music classification and retrieval", *IEEE International Conference on Multimedia and Expo*, pp. 1505–1508, 2008.
- [4] J. R. J. Fontaine, K. R. Scherer, E. B. Roesch, and P. C. Ellsworth: "The World of Emotions is not Two-Dimensional", *Psychological Science*, Vol. 18, No. 12, pp. 1050–1057, 2007.
- [5] D. Guan, X. Chen, and D. Yang: "Music Emotion Regression Based on Multi-modal Features", *CMMR*, p. 70–77, 2012.
- [6] C. A. Harte, and M. B. Sandler: "Detecting harmonic change in musical audio", *Proceedings of Audio and Music Computing for Multimedia Workshop*, 2006.
- [7] C. Laurier, O. Lartillot, T. Eerola, and P. Toivainen: "Exploring Relationships between Audio Features and Emotion in Music", *Conference of European Society for the Cognitive Sciences of Music*, 2009.
- [8] C. McKay, and I. Fujinaga: "Automatic genre classification using large high-level musical feature sets", *In Int. Conf. on Music Information Retrieval*, pp. 525–530, 2004.
- [9] B. Schuller, J. Dorfner, and G. Rigoll: "Determination of Nonprototypical Valence and Arousal in Popular Music: Features and Performances", *EURASIP Journal on Audio, Speech, and Music Processing, Special Issue on Scalable Audio-Content Analysis* pp. 735–854, 2010.
- [10] B. Sollberge, R. Rebe, and D. Eckstein: "Musical Chords as Affective Priming Context in a Word-Evaluation Task", *Music Perception: An Interdisciplinary Journal*, Vol. 20, No. 3, pp. 263–282, 2003.
- [11] K. Torres-Eliard, C. Labbe, and D. Grandjean: "Towards a dynamic approach to the study of emotions expressed by music", *Proceedings of the 4th International ICST Conference on Intelligent Technologies for Interactive Entertainment*, pp. 252–259, 2011.
- [12] J. K. Vuoskoski, and T. Eerola: "Domain-specific or not? The applicability of different emotion models in the assessment of music-induced emotions", *Proceedings of the 10th International Conference on Music Perception and Cognition*, pp. 196–199, 2010.
- [13] L. Wedin: "A Multidimensional Study of Perceptual-Emotional Qualities in Music", *Scandinavian Journal of Psychology*, Vol. 13, pp. 241–257, 1972.
- [14] F. Weninger, F. Eyben, B. W. Schuller, M. Mortillaro, and K. R. Scherer: "On the Acoustics of Emotion in Audio: What Speech, Music, and Sound have in Common", *Front Psychol*, Vol. 4, p. 292, 2013.
- [15] Y.-H. Yang, Y.-C. Lin, Y.-F. Su, and H. H. Chen: "A Regression Approach to Music Emotion Recognition", *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 16, No. 2, pp. 448–457, 2008.
- [16] Y.-H. Yang, and H. H. Chen: "Machine Recognition of Music Emotion: A Review", *ACM Trans. Intell. Syst. Technol.*, Vol. 3, No. 3, pp. 1–30, 2012.
- [17] M. Zentner, D. Grandjean, and K. R. Scherer: "Emotions evoked by the sound of music: characterization, classification, and measurement", *Emotion*, Vol. 8, No. 4, pp. 494–521, 2008.

<sup>10</sup> This research was supported by COMMIT/.

# COGNITION-INSPIRED DESCRIPTORS FOR SCALABLE COVER SONG RETRIEVAL

Jan Van Balen, Dimitrios Bountouridis, Frans Wiering, Remco Veltkamp

Utrecht University, Department of Information and Computing Sciences

j.m.h.vanbalen@uu.nl, d.bountouridis@uu.nl

## ABSTRACT

Inspired by representations used in music cognition studies and computational musicology, we propose three simple and interpretable descriptors for use in mid- to high-level computational analysis of musical audio and applications in content-based retrieval. We also argue that the task of scalable cover song retrieval is very suitable for the development of descriptors that effectively capture musical structures at the song level. The performance of the proposed descriptions in a cover song problem is presented. We further demonstrate that, due to the musically-informed nature of the proposed descriptors, an independently established model of stability and variation in covers songs can be integrated to improve performance.

## 1. INTRODUCTION

This paper demonstrates the use of three new cognition-inspired music descriptors for content-based retrieval.

### 1.1 Audio Descriptors

There is a growing consensus that some of the most widely used features in Music Information Research, while very effective for engineering applications, do not serve the dialog with other branches of music research [1]. As a classic example, MFCC features can be shown to predict human ratings of various perceptual qualities of a sound. Yet, from the perspective of neuropsychology, claims that they mathematically approximate parts of auditory perception have become difficult to justify as more parts of the auditory pathway are understood.

Meanwhile, a recent analysis of evaluation practices by Sturm [18] suggests that MIR systems designed to classify songs into high-level attributes like genre, mood or instrumentation may rely on confounded factors unrelated to any high-level property of the music, even if their performance numbers approach 100%. Researchers have focused too much on the same evaluation measures and the

same datasets and as a result, today, top performing genre and mood recognition systems rely on the same low-level features that are used to classify bird sounds.<sup>1</sup>

We also observe that, despite the increasing availability of truly big audio data and the promising achievements of MIR over the last decade, studies that turn big audio data into findings about music itself seem hard to find. Notable exceptions include studies on scales and intonation, and [16]. In the latter, pitch, timbre and loudness data were analyzed for the Million Song Dataset, focusing on the distribution and transitions of discretized code words. Yet, we have also observed that this analysis sparks debate among music researchers outside the MIR field, in part because of the descriptors used. The study uses the Echo Nest audio features provided with the dataset, which are computed using undisclosed, proprietary methods and therefore objectively difficult in interpretation.

### 1.2 Towards Cognitive Audio Descriptors

On the long term we would like to model cognition-level qualities of music such as its complexity, expectedness and repetitiveness from raw audio data. Therefore we aim to design and evaluate features that describe harmony, melody and rhythm on a level that has not gained the attention it deserves in MIR's audio community, perhaps due to the 'success' of low-level features discussed above. In the long run, we believe, this will provide insights into the building blocks of music: riffs, motives, choruses, and so on.

### 1.3 Cover Song Detection

In this section, we argue that the task of scalable cover song retrieval is very suitable for developing descriptors that effectively capture mid- to high-level musical structures, such as chords, riffs and hooks.

Cover detection systems take query song and a database and aim to find other versions of the query song. Since many real-world cover versions drastically modulate multiple aspects of the original: systems must allow for deviations in key, tempo, structure, lyrics, harmonisation and phrasing, to name just a few. Most successful cover detection algorithms are built around a two-stage architecture. In the first stage, the system computes a time series representation of the harmony or pitch for each of the songs in a database. In the second stage, the time series representing



© Jan Van Balen, Dimitrios Bountouridis, Frans Wiering, Remco Veltkamp.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Jan Van Balen, Dimitrios Bountouridis, Frans Wiering, Remco Veltkamp. "Cognition-inspired descriptors for Scalable Cover Song Retrieval", 15th International Society for Music Information Retrieval Conference, 2014.

<sup>1</sup> largely MFCC and spectral moments, see [6, 18] for examples

the query is compared to each of these representations, typically by means of some kind of alignment, i.e. computing the locations of maximum local correspondence between the two documents being compared. See [15] for more on this task and an overview of cover detection strategies.

## 2. SCALABLE COVER SONG RETRIEVAL

Generally, alignment methods are computationally expensive but effective. Results achieved this way have reached mean average precision (MAP) figures of around 0.75 at the MIREX evaluation exchange.<sup>2</sup>

When it comes to large-scale cover detection (hundreds of queries and thousands of songs), however, alignment-based methods can become impractical. Imagine a musicologist whose aim is not to retrieve matches to a single query, but to study all the relations in a large, representative corpus. Alignment-based techniques are no longer an option: a full pair-wise comparison of 10,000 documents would take weeks, if not months.<sup>3</sup>

This is why some researchers have been developing scalable techniques for cover song detection. Scalable strategies are often inspired by audio fingerprinting and involve the computation of an indexable digest of (a set of) potentially stable landmarks in the time series, which can be stored and matched through a single inexpensive look-up. Examples include the ‘jumpcodes’ approach by [2], the first system to be tested using the Million Song Dataset. This study reports a recall of 9.6% on the top 1 percent of retrieved candidates. Another relevant example is the interval-gram approach by Walters [19], which computes fingerprinting-inspired histograms of local pitch intervals, designed for hashing using wavelet decomposition.

Reality shows that stable landmarks are relatively easy to find when looking for exact matches (as in fingerprinting), but hard to find in real-world cover songs. A more promising approach was presented by Bertin-Mahieux in [3], where the 2D Fourier transform of beat-synchronized chroma features is used as the primary representation. The accuracy reported is several times better than for the system based on jumpcodes. Unfortunately, exactly what the Fourier transformed features capture is difficult to explain.

The challenges laid out in the above paragraph make cover song detection an ideal test case to evaluate a special class of descriptors: harmony, melody and rhythm descriptors, global or local, which have a fixed dimensionality and some tolerance to deviations in key, tempo and global structure. If a collection of descriptors can be designed that accurately describes a song’s melody, harmony and rhythm in a way that is robust to the song’s precise structure, tempo and key, we should have a way to determine similarity between the ‘musical material’ of two songs and assess if the underlying composition is likely to be the same.

<sup>2</sup> [http://www.music-ir.org/mirex/wiki/2009:Audio\\_Cover\\_Song\\_Identification\\_Results](http://www.music-ir.org/mirex/wiki/2009:Audio_Cover_Song_Identification_Results)

<sup>3</sup> MIREX 2008 (the last to report runtimes) saw times of around 1.4 – 3.7 × 10<sup>5</sup> s for a task that involves 115,000 comparisons. The fastest of these algorithms would take 1.8 years to compute the ½10<sup>8</sup> comparisons required in the above scenario. The best performing algorithm would take 6 years.

## 3. PITCH AND HARMONY DESCRIPTORS

There is an increasing amount of evidence that the primary mechanism governing musical expectations is statistical learning [7, 12]. On a general level, this implies that the conditional probabilities of musical events play a large role in their cognitive processing. Regarding features and descriptors, it justifies opportunities of analyzing songs and corpora in terms of probably distributions. Expectations resulting from the exposure to statistical patterns have in turn been shown to affect the perception of melodic complexity and familiarity. See [7] for more on the role of expectation in preference, familiarity and recall.

We propose three new descriptors: the pitch bihistogram, the chroma correlation coefficients and the harmonization feature. The pitch bihistogram describes melody and approximates a histogram of pitch bigrams. The chroma correlation coefficients relate to harmony. They approximate the co-occurrence of chord notes in a song. The third representation, the harmonization feature, combines harmony and melody information. These three descriptors will now be presented in more detail.

### 3.1 The Pitch Bihistogram

Pitch bigrams are ordered pairs of pitches, similar to word or letter bigrams used in computational linguistics. Several authors have proposed music descriptions based on pitch bigrams, most of them from the domain of cognitive science [10, 11, 13]. Distributions of bigrams effectively encode first-degree expectations. More precisely: if the distribution of bigrams in a piece is conditioned on the first pitch in the bigram, we obtain the conditional frequency of a pitch given the one preceding it.

The first new feature we introduce will follow the bigram paradigm. Essentially, it captures how often two pitches  $p_1$  and  $p_2$  occur less than a distance  $d$  apart.

Assume that a melody time series  $P(t)$ , quantized to semitones and folded to one octave, can be obtained. If a pitch histogram is defined as:

$$h(p) = \sum_{P(t)=p} \frac{1}{n}, \quad (1)$$

with  $n$  the length of the time series and  $p \in \{1, 2, \dots, 12\}$ , the proposed feature is then defined:

$$B(p_1, p_2) = \sum_{\substack{P(t_1)=p_1 \\ P(t_2)=p_2}} w(t_2 - t_1) \quad (2)$$

where

$$w(x) = \begin{cases} \frac{1}{d}, & \text{if } 0 < x < d. \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

This will be referred to as the **pitch bihistogram**, a bigram representation that can be computed from continuous melodic pitch. Note that the use of pitch classes rather than pitch creates an inherent robustness to octave errors in the melody estimation step, making the feature insensitive to one of the most common errors encountered in pitch extraction.

Alternatively, scale degrees can be used instead of absolute pitch class. In this scenario, the pitch contour  $P(t)$  must first be aligned to an estimate of the piece’s overall tonal center. As a tonal center, the tonic can be used. However, for extra robustness to misestimating the tonic, we suggest to use the tonic for major keys and the minor third for minor keys.

### 3.2 Chroma Correlation Coefficients

The second feature representation we propose focuses on vertical rather than horizontal pitch relation. It encodes which pitches appear simultaneously in a signal.

$$C(p_1, p_2) = \text{corr}(c(t, p_1), c(t, p_2)), \quad (4)$$

where  $c(t, p)$  is a 12-dimensional chroma time series (also known as pitch class profile) computed from the song audio. From this chroma representation of the song  $c(t, p)$  we compute the correlation coefficients between each pair of chroma dimensions to obtain a  $12 \times 12$  matrix of **chroma correlation coefficients**  $C(p_1, p_2)$ . Like the pitch bihistogram, the chroma features can be transposed to the same tonal center (tonic or third) based on an estimate of the overall or local key.

### 3.3 Harmonisation Feature

Finally, the **harmonisation feature** is a set of histograms of the harmonic pitches  $p_h \in \{1, \dots, 12\}$  as they accompany each melodic pitch  $p_m \in \{1, \dots, 12\}$ . It is computed from the pitch contour  $P(t)$  and a chroma time series  $c(t, p_h)$ , which should be adjusted to have the same sampling rate and aligned to a common tonal center.

$$H(p_m, p_h) = \sum_{P(t)=p_m} c(t, p_h). \quad (5)$$

From a memory and statistical learning perspective, the chroma correlation coefficients and harmonisation feature may be used to approximate expectations that include: the expected consonant pitches given a chord note, the expected harmony given a melodic pitch, and the expected melodic pitch given a chord note. Apart from [8], where a feature resembling the chroma correlation coefficients is proposed, information of this kind has yet to be exploited in a functioning (audio) MIR system. Like the pitch bihistogram and the chroma correlation coefficients, the harmonisation feature has a dimensionality of  $12 \times 12$ .

## 4. EXPERIMENTS

To evaluate the performance of the above features for cover song retrieval, we set up a number of experiments around the *covers80* dataset by Ellis [5]. This dataset is a collection of 80 cover song pairs, divided into a fixed list of 80 queries and 80 candidates. Though *covers80* is not actually ‘large-scale’, it is often used for benchmarking<sup>4</sup> and its associated audio data are freely available. In contrast, the much larger Second Hand Songs dataset is distributed

only in the form of standard Echo Nest features. These features do not include any melody description, which is the basis for the descriptors proposed in this study.

Regarding scalability, we chose to follow the approach taken in [19], in which the scalability of the algorithm follows from the simplicity of the matching step. The proposed procedure is computationally scalable in the sense that, with the appropriate hashing strategy, matching can be performed in constant time with respect to the size of the database. Nevertheless, we acknowledge that the distinguishing power of the algorithm must be assessed in the context of much more data. A large scale evaluation of our algorithm, adapted to an appropriate dataset and extended to include hashing solutions and indexing, is planned as future work.

### 4.1 Experiment 1: Global Fingerprints

In the first experiment, the three descriptors from section 3 were extracted for all 160 complete songs. Pitch contours were computed using Melodia and chroma features using HPCP, using default settings [14].<sup>5</sup> For efficiency in computing the pitch bihistogram, the pitch contour was median-filtered and downsampled to  $\frac{1}{4}$  of the default frame rate. The bihistogram was also slightly compressed by taking its square root.

The resulting representations ( $B$ ,  $C$  and  $H$ ) were then scaled to the same range by whitening them for each song individually (subtracting the mean of their  $n$  dimensions, and dividing by the standard deviation;  $n = 144$ ). To avoid relying on key estimation, features in this experiment were not aligned to any tonal center, but transposed to all 12 possible keys. In a last step of the extraction stage, the features were scaled with a set of dedicated weights  $w = (w_1, w_2, w_3)$  and concatenated to 12 432-dimensional vectors, one for each key. We refer to these vectors as the *global fingerprints*.

In the matching stage of the experiment, the distances between all queries and candidates were computed using a cosine distance. For each query, all candidates were ranked by distance. Two evaluation metrics were computed: *recall at 1* (the proportion of covers retrieved among the top 1 result for each query;  $R_1$ ) and *recall at 5* (proportion of cover retrieved ‘top 5’;  $R_5$ ).

### 4.2 Experiment 2: Thumbnail Fingerprints

In a second experiment, the songs in the database were first segmented into structural sections using structure features as described by Serra [17]. This algorithm performed best at the 2012 MIREX evaluation exchange in the task of ‘music structure segmentation’, both for boundary recovery and for frame pair clustering. (A slight simplification was made in the stage where sections are compared: no dynamic time warping was applied in our model.) From this segmentation, two non-overlapping thumbnails are selected as follows:

<sup>4</sup> results for this dataset have been reported by at least four authors [15]

<sup>5</sup> mtg.upf.edu/technologies

1. Simplify the sequence of section labels (e.g. abab-CabCC): merge groups of section labels that consistently appear together (e.g. AACACC for the example above).
2. Compute the total number of seconds covered by each of the labels A, B, C... and find the two section labels covering most of the song.
3. Return the boundaries of the first appearance of the selected labels.

The fingerprint as described above was computed for the full song as well as for the resulting thumbnails, yielding three different fingerprints: one global and two *thumbnail fingerprints*, stored separately. As in experiment 1, we transposed these thumbnails to all keys, resulting in a total of 36 fingerprints extracted per song: 12 for the full song, 12 for the first thumbnail and 12 for the second thumbnail.

### 4.3 Experiment 3: Stability Model

In the last experiment, we introduced a model of stability in cover song melodies. This model was derived independently, through analysis of a dataset of annotated melodies of cover songs variations. Given the melody contour for a song section, the model estimates the stability at each point in the melody. Here, stability is defined as the probability of the same pitch appearing in the same place in a performed variation of that melody.

The stability estimates produced by the model are based on three components that are found to correlate with stability: the duration of notes, the position of a note inside a section, and the pitch interval. The details of the model and its implementation are described in the following section.

## 5. STABILITY MODEL

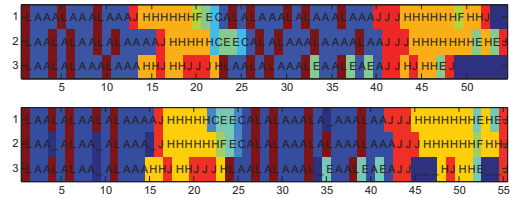
The model we apply is a quantitative model of melody stability in cover songs. As it has been established for applications broader than the current study, it is based on a unique, manually assembled collection of annotated cover songs melodies. The dataset contains four transcribed melodic variations for 45 so-called ‘cliques’ of cover songs, a subset of the Second Hand Songs dataset.<sup>6</sup> Some songs have one section transcribed, some have more, resulting in a total of 240 transcriptions.

For the case study presented here, transcriptions were analysed using multiple sequence alignment (MSA) and a probabilistic definition of stability.

### 5.1 Multiple Sequence Alignment

Multiple sequence alignment is a bioinformatics method that extends pairwise alignment of symbolic arrays to a higher number of sequences [4]. Many approaches to MSA exist, some employing hidden markov models or genetic algorithms. The most popular is progressive alignment.

<sup>6</sup> <http://labrosa.ee.columbia.edu/millionsong/secondhand>



**Figure 1.** A clique of melodies before (top) and after (bottom) multiple sequence alignment.

This technique creates an MSA by combining several pairwise alignments (PWA) starting from the most similar sequences, constructing a tree usually denoted as the ‘guide tree’. Unlike MSA, pairwise alignment has been researched extensively in the (symbolic) MIR community, see [9] for an overview.

Whenever two sequences are aligned, a *consensus* can be computed, which can be used for the alignment connecting the two sequences to the rest of the three. The consensus is a new compromise sequence formed using heuristics to resolve the ambiguity at non-matching elements. These heuristics govern how gaps propagate through the tree, or whether ‘leaf’ or ‘branch’ elements are favored. The current model favors gaps and branch elements.

When the root consensus of the tree is reached, a last iteration of PWA’s aligns each sequence to the root consensus to obtain the final MSA. Figure 1 shows two sets of melodic sequences (mapped to a one-octave alphabet {A ... L}) before and after MSA. Note that the MSA is based on a PWA strategy which maximizes an optimality criterion based on not just pitch but also duration and onset times.

### 5.2 Stability

The **stability** of a note in a melody is now defined as the probability of the same note being found in the same position in an optimally aligned variation of that melody.

Empirically, given a set of  $N$  aligned sequences

$$\{s_k(i)\} \quad i = 1 \dots n, k = 1 \dots N \quad (6)$$

we compute the stability of event  $s_k(i)$  as:

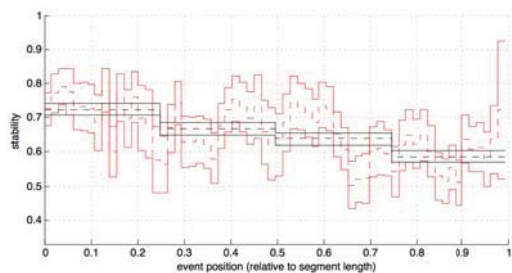
$$stab(s_k(i)) = \frac{1}{N-1} \sum_{\substack{j=1 \\ j \neq k}}^{j=N} s_j(i) == s_k(i) \quad (7)$$

As an example, in a position  $i$  with events  $s_1(i) = A$ ,  $s_2(i) = A$ ,  $s_3(i) = A$  and  $s_4(i) = B$ , the stability of A is 0.66. The stability of B is 0.

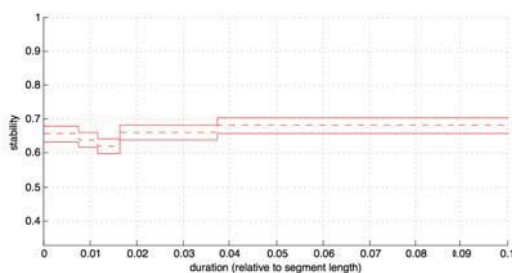
### 5.3 Findings

As described in the previous section, we drew a random sample of notes from the dataset in order to observe how stability behaves as a function of the event’s pitch, duration and position inside the song section.

The first relationship has ‘position’ as the independent variable and describes the stability as it evolves throughout



**Figure 2.** Stability of an event vs. position in the melody.



**Figure 3.** Stability of an event vs. duration.

the section. Figure 2 shows how stability changes with position. The mean and 95% CI for the mean are shown for two different binnings of the position variable. The 4-bin curve illustrates how stability generally decreases with position. The more detailed 64-bin curve shows how the first two thirds of a melody are more stable than the last, though an increased stability can be seen at the end of the section.

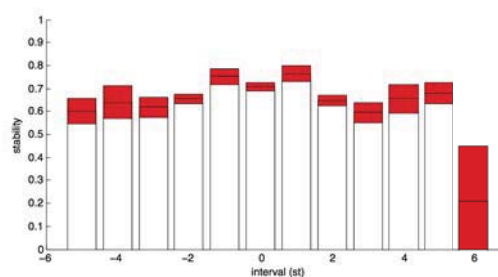
Figure 3 shows the stability of notes as a function of their duration. The distribution of note durations is centered around 1% of the segment length. Below and above this value, the stability goes slightly up. This suggests that notes with less common durations are more stable. However, the trend is weak compared with the effect of position. Note duration information will therefore not be used in the experiments in this study.

Figure 4 shows the stability (mean and 95% CI for the mean) of a note given the pitch interval that follows. Note how the relative stability of one-semitone jumps stands out compared to repetitions and two-semitone jumps, even though two-semitone jumps are far more frequent. This suggests again that less-frequent events are more stable. More analysis as to this hypothesis will be performed in a later study.

## 6. DISCUSSION

Table 1 summarizes the results of the experiments.

In the experiments where each descriptor was tested individually, the harmony descriptors (chroma correlation coefficients) performed best: we obtained an accuracy of over 30%. When looking at the top 5, there was a recall of 53.8%. The *recall at 5* evaluation measure is included to give an impression of the performance that could



**Figure 4.** Stability of an event vs. the interval that follows.

be gained if the current system were complemented with an alignment-based approach to sort the top-ranking candidates, as proposed by [19].

The next results show that, for the three features together, the global fingerprints outperform the thumbnail fingerprints (42.5% vs. 37.5%), and combining both types does not increase performance further. In other configurations, thumbnail fingerprints were observed to outperform the global fingerprints. This is possibly the result of segmentation choices: short segments produce sparse fingerprints, which are in turn farther apart in the feature space than ‘dense’ fingerprints.

In experiment 3, two components of the stability model were integrated in the cover detection system. The 4-bin stability vs. position curve (scaled to the [0, 1] range) was used as a weighting to emphasize parts of the melody before computing the thumbnails’ pitch bihistogram. The stability per interval (compressed by taking its square root) was used to weigh the pitch bihistogram directly.

With the stability information added to the model, the top 1 precision reaches 45.0%. The top 5 recall is 56.3%. This result is situated between the accuracy of the first alignment-based strategies (42.5%), and the accuracy of a recent scalable system (53.8%; [19]). We conclude that the descriptors capture enough information to discriminate between individual compositions, which we set out to show.

## 7. CONCLUSIONS

In this study, three new audio descriptors are presented. Their interpretation is discussed, and results are presented for an application in cover song retrieval. To illustrate the benefit of feature interpretability, an independent model of cover song stability is integrated into the system.

We conclude that current performance figures, though not state-of-the-art, are a strong indication that scalable cover detection can indeed be achieved using interpretable, cognition-inspired features. Second, we observe that the pitch bihistogram feature, the chroma correlation coefficients and the harmonisation feature capture enough information to discriminate between individual compositions, proving that they are at the same time meaningful and informative, a scarce resource in the MIR feature toolkit. Finally, we have demonstrated that cognition-level audio description and scalable cover detection can be successfully addressed together.

	Descriptor	$R_1$	$R_5$
Global fingerprints	$B$	0.288	0.438
	$C$	0.313	0.538
	$H$	0.200	0.375
	$w = (2, 3, 1)$	0.425	0.575
Thumbnail fingerprints	$w = (2, 3, 1)$	0.388	0.513
Global + thumbnail fingerprints	$w = (2, 3, 1)$	0.425	0.538
Both fingerprints + stability model	$w = (2, 3, 1)$	0.450	0.563

**Table 1.** Summary of experiment results.  $w$  are the feature weights. Performance measures are *recall at 1* (proportion of covers retrieved ‘top 1’;  $R_1$ ) and *recall at 5* (proportion of cover retrieved among ‘top 5’;  $R_5$ ).

As future work, tests will be carried out to assess the discriminatory power of the features when applied to a larger cover song problem.

## 8. ACKNOWLEDGEMENTS

This research is supported by the NWO CATCH project COGITCH (640.005.004), and the FES project COMMIT/.

## 9. REFERENCES

- [1] Jean-Julien Aucouturier and Emmanuel Bigand. Seven problems that keep MIR from attracting the interest of cognition and neuroscience. *Journal of Intelligent Information Systems*, 41(3):483–497, July 2013.
- [2] T Bertin-Mahieux and Daniel P W Ellis. Large-scale cover song recognition using hashed chroma landmarks. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 10–13, 2011.
- [3] T Bertin-Mahieux and Daniel P W Ellis. Large-Scale Cover Song Recognition Using The 2d Fourier Transform Magnitude. In *Proc Int Soc for Music Information Retrieval Conference*, pages 2–7, 2012.
- [4] H Carrillo and D Lipman. The Multiple Sequence Alignment Problem in Biology. *SIAM Journal on Applied Mathematics*, 1988.
- [5] Daniel P. W. Ellis and C.V. Cotton. The ‘‘covers80’’ cover song data set, 2007.
- [6] M. Graciarena, M. Delplanche, E. Shriberg, A Stolcke, and L. Ferrer. Acoustic front-end optimization for bird species recognition. In *IEEE int conf on Acoustics Speech and Signal Processing (ICASSP)*, pages 293–296, March 2010.
- [7] David Huron. Musical Expectation. In *The 1999 Ernest Bloch Lectures*. 1999.
- [8] Samuel Kim and Shrikanth Narayanan. Dynamic chroma feature vectors with applications to cover song identification. *2008 IEEE 10th Workshop on Multimedia Signal Processing*, pages 984–987, October 2008.
- [9] Peter van Kranenburg. *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*. PhD thesis, Utrecht University, 2010.
- [10] Y. Li and D. Huron. Melodic modeling: A comparison of scale degree and interval. In *Proc. of the Int. Computer Music Congerence*, 2006.
- [11] Daniel Müllensiefen and Klaus Frieler. Evaluating different approaches to measuring the similarity of melodies. *Data Science and Classification*, 2006.
- [12] Marcus T. Pearce, Mara Herrojo Ruiz, Selina Kapasi, Geraint A. Wiggins, and Joydeep Bhattacharya. Unsupervised statistical learning underpins computational, behavioural, and neural manifestations of musical expectation. *NeuroImage*, 50(1):302 – 313, 2010.
- [13] Pablo H Rodriguez Zivic, Favio Shifres, and Guillermo a Cecchi. Perceptual basis of evolving Western musical styles. *Proceedings of the National Academy of Sciences of the United States of America*, 110(24):10034–8, June 2013.
- [14] J. Salamon and E. Gomez. Melody extraction from polyphonic music signals using pitch contour characteristics. In *IEEE Trans. on Audio, Speech and Language Processing*, 2010.
- [15] Joan Serrà. *Identification of versions of the same musical composition by processing audio descriptions*. PhD thesis, Universitat Pompeu Fabra, 2011.
- [16] Joan Serrà, Alvaro Corral, Marián Bogueña, Martín Haro, and Josep Ll Arcos. Measuring the evolution of contemporary western popular music. *Scientific reports*, 2:521, January 2012.
- [17] Joan Serra, M Meinard, Peter Grosche, and Josep Ll Arcos. Unsupervised Detection of Music Boundaries by Time Series Structure Features. *Proc of the AAAI Conf on Artificial Intelligence*, pages 1613–1619, 2012.
- [18] Bob L. Sturm. Classification accuracy is not enough. *Journal of Intelligent Information Systems*, 41(3):371–406, July 2013.
- [19] Thomas C Walters, David A Ross, and Richard F Lyon. The Intervalgram : An Audio Feature for Large-scale Melody Recognition. In *Int. Symp. on Computer Music Modeling and Retrieval (CMMR)*, pages 19–22, 2012.



# A CROSS-CULTURAL STUDY OF MOOD IN K-POP SONGS

**Xiao Hu**

University of Hong Kong  
xiaoxhu@hku.hk

**Jin Ha Lee**

University of Washington  
jinhalee@uw.edu

**Kahyun Choi**

University of Illinois  
{ckahyu2, jdownie}@illinois.edu

**J. Stephen Downie**

## ABSTRACT

Prior research suggests that music mood is one of the most important criteria when people look for music—but the perception of mood may be subjective and can be influenced by many factors including the listeners' cultural background. In recent years, the number of studies of music mood perceptions by various cultural groups and of automated mood classification of music from different cultures has been increasing. However, there has yet to be a well-established testbed for evaluating cross-cultural tasks in Music Information Retrieval (MIR). Moreover, most existing datasets in MIR consist mainly of Western music and the cultural backgrounds of the annotators were mostly not taken into consideration or were limited to one cultural group. In this study, we built a collection of 1,892 K-pop (Korean Pop) songs with mood annotations collected from both Korean and American listeners, based on three different mood models. We analyze the differences and similarities between the mood judgments of the two listener groups, and propose potential MIR tasks that can be evaluated on this dataset.

## 1. INTRODUCTION

The mood of music is arguably one of the strongest factors behind people's motivation of listening to music [4]. Recognizing the importance of music mood, an increasing number of studies have been exploring the use of mood data to improve users' access to music. Recent studies in Music Information Retrieval (MIR) have indicated that people from different cultural backgrounds may perceive music mood differently ([2] [9]). In an effort toward establishing a global MIR system that can serve users from different parts of the world, researchers have developed and evaluated algorithms that can work on classifying music from different cultures and/or labeled by listeners from different countries ([17] [14]). Despite the growing interests on cultural influences on MIR ([7] [14]), we still do not have a well-established testbed for cross-cultural MIR tasks where methods proposed by interested researchers can be properly evaluated and compared. Music Information Retrieval Evaluation eXchange (MIREX), which is the primary MIR evaluation venue, has yet to add a cross-cultural evaluation task. This study aims to work toward filling this gap by 1) building a dataset<sup>1</sup> consisting of 1,892 songs from a non-

Western culture (i.e., K-pop or Korean Pop) and labels based on three music mood models annotated by listeners from two distinct cultural groups (i.e., American and Korean); 2) analyzing the differences and similarities between mood labels provided by American and Korean listeners on the same set of 1,892 K-pop songs; and 3) proposing cross-cultural MIR tasks that can be evaluated using this dataset.

## 2. RELATED WORK

Music is a medium beyond the boundary of languages, countries and cultures. As many MIR systems need to be designed to serve global users, researchers have been paying more attention to cross-cultural issues in MIR. Lee and Hu [9] compared mood labels on a set of 30 Western songs provided by American, Chinese, and Korean listeners and found that cultural background indeed influenced people's perception of music mood. Yang and Hu [17] compared mood labels on U.S. pop songs provided by Western listeners to labels on Chinese pop songs provided by Chinese listeners. The datasets were larger (nearly 500 songs) in their study, although the labels were applied to two separate datasets and thus may not be directly comparable. In this paper, we compare music mood perceptions of the same set of K-pop songs from American and Korean listeners, making the mood annotations directly comparable.

K-pop is increasingly becoming popular with international audiences, as evidenced by the launch of Billboard K-pop Hot 100 chart in 2011<sup>2</sup>, and is actively sought by people from different cultural backgrounds. K-pop has unique characteristics due its history; Korean culture has been heavily influenced by American pop culture since the 1950s, yet is deeply rooted in the long history of East Asia. A recent study by Lee et al. [7] discussed the differences in the perception of K-pop genres by American and Korean listeners based on how they applied genre labels to K-pop music. In this study, we focus on the mood aspect of K-pop, aiming to improve the understanding of how mood can be used as a descriptor for organizing and accessing music by users from different cultures.

Currently there exist several influential datasets in music mood recognition ([3] [5]). However, most of them contain primarily Western music and the cultural background of annotators was either not specified [3] or not controlled [5]. To the best of our knowledge, the dataset built in this study is the first of its kind that is composed



© Xiao Hu, Jin Ha Lee, Kahyun Choi, J. Stephen Downie. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Xiao Hu, Jin Ha Lee, Kahyun Choi, J. Stephen Downie. "A Cross-Cultural Study of Mood in K-Pop Songs", 15th International Society for Music Information Retrieval Conference, 2014.

<sup>1</sup> The tag data will be incorporated into MIREX for use in a variety of MIR evaluations and released incrementally over time when not needed for MIREX.

<sup>2</sup> <http://www.billboard.com/articles/news/467764/billboard-k-pop-hot-100-launches-sistar-is-no-1-on-new-korea-chart>

of a significant amount of non-Western music, annotated by listeners from two distinct cultures, and labeled based on three music mood models. In MIREX, there have been two mood-related (sub)-tasks: Audio Mood Classification (AMC) starting from 2007 and the mood tag subtask in Audio Tag Classification (ATC) starting from 2009<sup>1</sup>. Both tasks consist of Western songs labeled by listeners from unspecified cultural backgrounds [3]. This new dataset will enable evaluation tasks that explore the cross-cultural generalizability of automated music mood recognition systems [17].

### 3. STUDY DESIGN

#### 3.1 The K-Pop Music Dataset

The dataset consists of 1,892 K-pop songs across seven dominant music genres in K-pop, namely Ballad, Dance/Electronic, Folk, Hip-hop/Rap, Rock, R&B/Soul, and Trot [7]. 30 second music clips were extracted from each song and presented to the listeners for mood annotation. This was to mitigate the cognitive load of annotators and to minimize the effect of possible mood changes during the entire duration of some songs (which can happen for some songs but is beyond the scope of this study).

#### 3.2 Music Mood Models

In representing music mood, there are primarily two kinds of models: categorical and dimensional [5]. In categorical models, music mood is represented as a set of discrete mood categories (e.g., happy, sad, calm, angry, etc.) and each song is assigned to one or more categories. This study adopted two categorical models used in MIREX: 1) the five mood clusters (Table 1) used in the Audio Mood Classification task [3] where each song is labeled with one mood cluster exclusively; and 2) the 18 mood groups (Figure 2) used in the mood tag subtask in Audio Tag Classification where each song is labeled with up to six groups. Besides being used in MIREX, these two models were chosen due to the fact that they were developed from empirical data of user judgments and in a way that is completely independent from any dimensional models, and thus they can provide a contrast to the latter.

Unlike categorical models, dimensional models represent a “mood space” using a number of dimensions with continuous values. The most influential dimensional model in MIR is Russell’s 2-dimensional model [11], where the mood of each song is represented as a pair of numerical values indicating its degree in the *Valence* (i.e., level of pleasure) and *Arousal* (i.e., level of energy) dimensions. Both categorical and dimensional models have their advantages and disadvantages. The former uses natural language terms and thus is considered more intuitive for human users, whereas the latter can represent the degree of mood(s) a song may have (e.g., a little sad). Therefore, we used both kinds of models when annotating the mood of our K-pop song set. In addition to the 5 mood clusters and 18 mood groups, the K-pop songs were also annotated with the Valence-Arousal 2-dimensional model.

<b>Cluster1</b> <b>(C_1)</b>	passionate, rousing, confident, boisterous, rowdy
<b>Cluster2</b> <b>(C_2)</b>	rollicking, cheerful, fun, sweet, amiable/good natured
<b>Cluster3</b> <b>(C_3)</b>	literate, poignant, wistful, bittersweet, autumnal, brooding
<b>Cluster4</b> <b>(C_4)</b>	humorous, silly, campy, quirky, whimsical, witty, wry
<b>Cluster5</b> <b>(C_5)</b>	aggressive, fiery, tense/anxious, intense, volatile, visceral

**Table 1.** Five mood clusters in the MIREX AMC task.

#### 3.3 Annotation Process

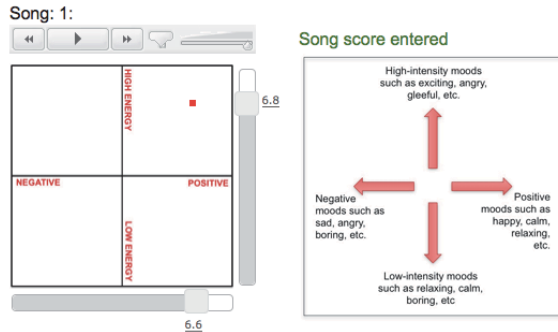
For a cross-cultural comparison, a number of American and Korean listeners were recruited to annotate the mood of the songs. The American listeners were recruited via a well-known crowdsourcing platform, Amazon Mechanical Turk (MTurk), where workers complete tasks requiring human intelligence for a small fee. MTurk has been recognized as a quick and cost-effective way of collecting human opinions and has been used successfully in previous MIR studies (e.g., [6], [8]). In total, 134 listeners who identified themselves as American participated in the annotations based on the three mood models.

For the five-mood cluster model, each “HIT” (Human Intelligence Task, the name for a task in MTurk) contained 22 clips with two duplicates for a consistency check. Answers were only accepted if the annotations on the duplicate clips were the same. Participants were paid \$2.00 for successfully completing each HIT. For the 18-group model, we paid \$1.00 for each HIT, which contained 11 clips with one duplicate song for consistency check. There were fewer clips in each HIT of this model as the cognitive load was heavier: it asked for multiple (up to six) mood labels out of 18. For the Valence-Arousal (V-A) dimensional model we designed an interface with two slide scales in the range of [-10.0, 10.0] (Figure 1). We paid \$1.00 for each HIT, which contained 11 clips with one duplicate song for a consistency check. Consistency was defined such that the difference between the two annotations of the duplicate clips in either dimension should be smaller than 2.0. The threshold was based on the findings in [16] where a number of listeners gave V-A values to the same songs in two different occasions and the differences never exceeded 10% of the entire range. For each of the three mood representation models, three annotations were collected for each music clip. The total cost was approximately \$1800.

As there was no known crowdsourcing platform for Korean people, the nine Korean listeners who participated in the annotation were recruited through professional and personal networks of the authors. The annotation was done with our in-house annotation systems, which are similar to those in MTurk. All instructions and mood labels/dimensions were translated into Korean to minimize possible misunderstanding of the terminology. Similarly, each song received three annotations in each mood model. The payments to annotators were also comparable to those in MTurk. Although the total number of annotators in the two cultural groups differs, each song had exactly

<sup>1</sup> [http://www.music-ir.org/mirex/wiki/MIREX\\_HOME](http://www.music-ir.org/mirex/wiki/MIREX_HOME)

six independent annotations on which the following analysis and comparisons are based.



**Figure 1.** Annotation interface of the VA model (horizontal dimension is Valence, vertical is Arousal).

#### 4. RESULTS

The annotations by American and Korean listeners are compared in terms of judgment distribution, agreement levels, and confusion between the two cultural groups. The Chi-square independence test is applied to estimate whether certain distributions were independent with listeners' cultural background.

##### 4.1 Distribution of Mood Judgment

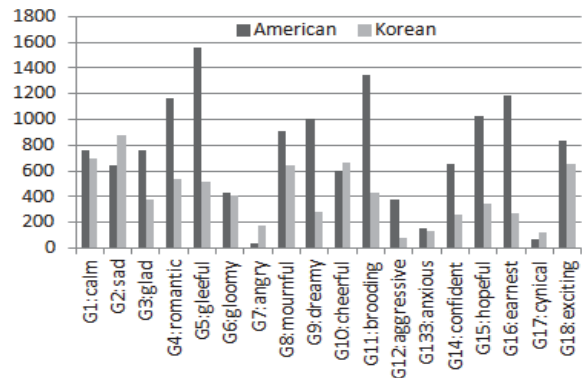
Table 2 shows the distribution of mood judgment of listeners from both cultural groups across five mood clusters. A Chi-square independence test indicates that the distribution does depend on cultural group ( $p < 0.001$ ,  $df = 4$ ,  $\chi^2=396.90$ ). American listeners chose C\_1 (*passionate*) and C\_5 (*aggressive*) more often while Korean listeners chose C\_2 (*cheerful*), C\_3 (*bittersweet*) and C\_4 (*silly/quirky*) more often. It is noteworthy that both groups chose C\_3 (*bittersweet*) most often among all five clusters. This is different from [9] where both American and Korean listeners chose C\_2 (*cheerful*) most often for American Pop songs. This difference may indicate that K-pop songs are generally more likely to express C\_3 moods than American Pop songs.

	C_1	C_2	C_3	C_4	C_5	Total
American	1768	897	2225	311	475	5676
Korean	959	1321	2598	453	345	5676

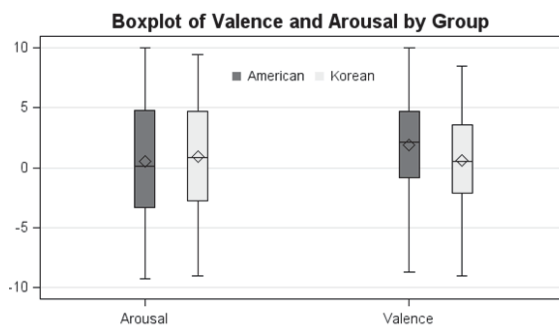
**Table 2.** Judgment distributions across 5 mood clusters.

With the 18-mood group model, a listener may label a song with up to six mood groups. The American listeners chose 13,521 groups in total, resulting in an average of 2.38 groups per song. The Korean listeners chose 7,465 groups in total, which resulted in 1.32 groups per song. The fact that American listeners assigned almost twice as many groups to each song as Korean listeners did may be related to the individualism/collectivism dichotomy found in psychology and cultural studies [13]; Americans tend to be individualistic and are more flexible in accepting a range of ideas (mood groups in this case) than people from collectivistic cultures (often represented by East Asian cultures). Future studies employing more qualitative approaches are warranted to verify this speculation.

Figure 2 shows the distribution of judgments across the 18 mood groups. A chi-square test verified that the distribution is statistically significantly dependent on cultural backgrounds ( $p < 0.001$ ,  $df = 17$ ,  $\chi^2=1664.49$ ). Americans used “gleeful”, “romantic”, “brooding”, “earnest”, “hopeful”, and “dreamy” more often than Koreans, while Koreans applied “sad” more frequently than Americans. Both groups used “angry” and “anxious” very rarely, probably due to the nature of K-pop songs. Similar observations were made in [17], where mood labels applied to Chinese and Western Pop songs were compared and radical moods such as “aggressive” and “anxious” were applied much more infrequently to Chinese songs than to Western songs. This may indicate a cultural difference in music: Chinese and Korean cultures tend to restrain and/or censor the expression of radical or destructive feelings whereas in Western cultures people are willing and free to expression of all kinds of feelings [10].



**Figure 2.** Judgment distributions across 18 mood groups (each group is represented by one representative term).



**Figure 3.** Boxplot of Valence and Arousal values.

Figure 3 shows the boxplot of the annotations based on the VA dimensional space given by the two groups of listeners. The V-A scores given by Americans are more scattered than those by Koreans, suggesting that Americans were more willing to choose extreme values. In addition, the means and medians indicate that Americans rated the songs with lower arousal values but higher valence values than Koreans ( $p < 0.001$  in non-paired  $t$ -test for both cases). In other words, Americans tended to consider the songs to be less intense and more positive than did Koreans. This may also reflect the cultural difference

that individuals from Western cultures tend to experience and/or express more positive emotions than those from Eastern cultures [12], and Asians present themselves as less aroused compared to Americans and Europeans [1].

#### 4.2 Agreements Within and Across Cultural Groups

In order to find out whether listeners from the same cultural background agree more with each other than with those from another cultural group, we examined the agreement among annotations provided by listeners in each cultural group as well as across cultural groups. The agreement measures used are the Sokal-Michener coefficient and intra-class correlation (ICC). The former is appropriate for categorical data while the latter is used for numerical data in the V-A space.

##### 4.2.1 Sokal-Michener coefficient

The Sokal-Michener (S-M) coefficient is the ratio of the number of pairs with the same values and the total number of variables [2][9], and therefore a higher value indicates a higher agreement. For instance, if two listeners  $i$  and  $j$  had the same mood judgments on 189 of the 1892 songs, the S-M coefficient between them is approximately 0.1. Table 3 shows the average S-M coefficient aggregated across all pairs of annotators within and across cultural groups on the five-cluster annotations. It is not surprising that Koreans reached a higher agreement than Americans since they are annotating songs originating from their own culture. This is consistent with the findings in [2] and [9], where American listeners reached a higher agreement on the mood of American Pop songs than did Korean and Chinese listeners. The agreement level was the lowest when annotations from American and Korean listeners (cross-cultural) were paired up. The distribution of agreed vs. disagreed judgments is significantly dependent on whether the listeners are from the same cultural group or not, evidenced by the Chi-square test results (Table 3). Listeners from the same cultural group tend to agree more with each other than with those from a different culture.

	American	Korean	$\chi^2$	df	$P$
American	0.47	0.43	25.35	1	<0.001
Korean	0.43	0.56	249.71	1	<0.001

**Table 3.** S-M coefficients of the five-cluster annotation within and across cultural groups

The analysis is more complex for the 18 group annotation, as each judgment can associate multiple labels with a song. To measure the agreement, we paired up labels applied to a song by any two annotators, and then calculated the S-M coefficient as the proportion of matched pairs among all pairs. For example, if annotator\_1 labelled a song S with g1, g2, g3 and annotator\_2 labelled it with g1, g4, then there were six annotation pairs and only one of them matched (i.e., g1 matched g1). The S-M coefficient in this case is  $1/6 = 0.17$ . Although the denominator increases when more labels are chosen, the chances they get matched also increase. All annotations from all listeners within each cultural group and across cultural groups were paired up in this way, and the result-

ant S-M coefficients are shown in Table 4. Again, the agreement level within Koreans was higher than that within Americans and also across cultural groups. However, the agreement within Americans was at the same level as the cross-cultural agreement, which is further evidenced by the statistically insignificant result of the Chi-square test.

	American	Korean	$\chi^2$	df	$p$
American	0.11	0.11	3.72	1	0.054
Korean	0.11	0.15	156.88	1	<0.001

**Table 4.** S-M coefficient of the 18-group annotation within and across cultural groups

##### 4.2.2 Intra-Class Correlation

The intra-class correlation (ICC) is a measure of agreement when ratings are given based on a continuous scale [15]. In the case of V-A annotation in this study, there is a different set of raters (listeners) for each item (song), and thus the one-way random model is used to calculate ICC within each group (3 raters) and across both groups (6 raters), for the valence and arousal dimensions. As shown in Table 5, cross-cultural agreement on valence is lower than within-cultural ones. Unlike five mood cluster annotation, both groups showed similar level of agreement on both dimensions. It is also noteworthy that the agreement on arousal annotation is much higher than valence annotation within- and cross-culturally. This is consistent with earlier MIR literature where valence has been recognized as more subjective than arousal [5].

	American	Korean	Cross-Cultural
Valence	0.27	0.28	0.23
Arousal	0.55	0.54	0.54

**Table 5.** ICC of Valence Arousal annotations within and across cultural groups

#### 4.3 Confusion Between Cultural Groups

To further our understanding on the difference and similarity of mood perceptions between the two cultural groups, we also examined the disagreement between listeners in the two groups in each of the three types of annotations. For the 5-cluster annotation, Table 6 shows the confusion matrix of the 1,438 songs with agreed labels by at least two listeners in each cultural group. Each cell shows the number of songs labeled as one mood cluster by Koreans (column) and another by Americans (row). The cells on the (highlighted) diagonal are numbers of songs agreed by the two groups, while other cells represent the disagreement between the two groups. The matrix shows that both groups agreed more on C\_3 (*bittersweet*) within themselves (661 and 842 songs respectively as shown by the "Total" cells). The bold numbers indicate major disagreements between the two groups. There are 268 songs Korean listeners judged as C\_3 (*bittersweet*) that Americans judged as C\_1 (*passionate*). The two groups only agreed on C\_5 (*aggressive*) on 18 songs, whereas 49 songs judged as C\_5 (*aggressive*) by Americans were judged by the Koreans as C\_1 (*passionate*).

Table 7 shows the confusion matrix of the seven mood groups (due to space limit) with the most agreed songs by majority vote among the Korean listeners. The biggest confusion/discrepancy is between “exciting” and “gleeful”: 135 songs perceived as “gleeful” by Americans were perceived as “exciting” by Koreans. Other major confusions are between “exciting” and “cheerful”, and “sad” and “mournful.” These moods have similar semantics in terms of valence (both “sad” and “mournful” have low valence values) and arousal (both “exciting” and “gleeful” have high arousal values), which may explain the confusion between these terms. Similarly, there are few songs with disagreement between mood labels with very distinct semantics, such as “exciting” vs. “sad/calm/mournful”; “calm” vs. “cheerful/gleeful”; and “gleeful” vs. “mournful”.

AM \ KO	C_1	C_2	C_3	C_4	C_5	Total
C 1	70	79	268	18	22	457
C 2	41	126	10	11	2	190
C 3	19	53	558	22	9	661
C 4	10	6	5	22	1	44
C 5	49	10	1	8	18	86
Total	189	274	842	81	52	1438

**Table 6.** Cross-tabulation between 5-cluster annotations across cultural groups

It is interesting to see that a number of songs perceived as “romantic” by Americans were seen as “sad” (31 songs) and “calm” (30 songs) by Koreans. On the other hand, 18 songs perceived as “romantic” by Koreans were viewed as “calm” by Americans. “Romantic” was seldom confused with other high arousal moods such as “exciting” or “cheerful” by either Koreans or Americans, suggesting that both cultures tend to associate “romantic” with low arousal music.

AM \ KO	exciting	sad	cheerful	calm	mournful	gleeful	romantic	Total
exciting	71	2	35	2	2	28	3	143
sad	0	32	0	13	13	0	4	62
cheerful	35	3	32	1	3	7	2	83
calm	0	10	0	25	4	0	18	57
mournful	0	48	0	23	27	0	6	104
gleeful	135	4	98	2	2	55	4	300
romantic	4	31	3	30	18	3	27	116
total	245	130	168	96	69	93	64	865

**Table 7.** Cross-tabulation between 18-group annotations across cultural groups

For the 2-D annotation, we show the disagreement between the two groups in the four quadrants of the 2-D space (Table 8). Both groups agreed more with listeners from their own cultural group on the first quadrant (+A+V) and the third quadrant (-A-V) (as shown by the “Total” cells). The largest discrepancy was observed between -A+V and -A-V: 116 songs were perceived as

having negative arousal and positive valence (-A+V) by Americans but negative valence (-A-V) by Koreans. Similarly, for the songs perceived as having positive arousal by both groups, 118 of them were again perceived as having positive valence (+A+V) by Americans but negative valence (+A-V) by Koreans. This is consistent with our finding that Korean listeners are more likely to label negative moods than Americans (Section 4.1).

KO \ AM	+A+V	+A-V	-A+V	-A-V	Total
+A+V	495	118	25	34	672
+A-V	8	30	1	17	56
-A+V	51	24	84	116	275
-A-V	10	19	80	178	287
Total	565	191	190	346	1290

**Table 8.** Cross-tabulation among the four quadrants in 2-D annotations across cultural groups

## 5. DISCUSSIONS

### 5.1 Differences and Similarities Between Groups

The results show that mood judgments and the level of agreement are dependent on the cultural background of the listeners. A number of differences were found between the annotations of the two groups. First, Americans assigned a larger number of labels to each song, and applied more extreme valence and arousal values than Koreans (Figure 3). We speculate that perhaps this is related to the fact that the Western culture tends to encourage individualism and divergent thinking more than the Eastern culture [13]. The difference in the number of annotators is another possible explanation. Both of these factors will be further explored in future work. Second, compared to Americans, Koreans were more likely to label songs with negative moods such as “bittersweet”, “sad,” and “mournful” (Table 2, Figure 2), give lower valence values (Figure 3), and agree with each other more often on songs with negative valence (Table 9). These observations were consistent with and supported by findings in previous cultural studies that people from Western cultures tend to experience and/or express more positive emotions than those from Eastern cultures [12]. The fact that Americans in this study could not understand the lyrics of the songs may also have contributed to these results. Sometimes song lyrics and melody may express different moods to invoke complex emotions (e.g., dark humor). In particular, a recent trend among K-pop artists to use faster tempo in Ballad songs may make the melody sound positive or neutral, although the lyrics are sad or melancholy as is the convention for Ballad songs.

It is also found that agreements of within-cultural groups are higher than that of cross-cultural groups based on the comparison of S-M coefficient, and ICC values (on valence only). For within-cultural group agreement, Koreans reached a higher agreement than Americans on 5-cluster annotation, which may be explained by the fact that Koreans were more familiar with the K-pop songs used in this study than Americans. Prior familiarity with

songs was also identified as a factor affecting the agreement level of mood perception in previous studies [2].

Some similarities were also found between the annotations of the two groups: 1) both groups applied and agreed on C\_3 (*bittersweet*) more often than other mood clusters (Tables 2 and 8); 2) both groups seldom applied radical mood labels such as “aggressive”, “angry”, “anxious” (Table 2 and Figure 2); and 3) both groups agreed more on songs with +A+V and -A-V values (Table 9). These similarities can potentially be attributed to the nature of the K-pop songs. A previous study comparing mood labels on Western and Chinese Pop songs also found that there were significantly fewer radical mood labels assigned to Chinese Pop songs than to Western songs [17]. This may reflect Eastern Asian preferences for non-aggressive music, perhaps due to their tradition of being more conservative and limiting the expression of feelings [10]. Another likely explanation would be the censorship and regulation<sup>1</sup> that still heavily affects the popular music culture in countries like South Korea and China.

## 5.2 Proposed MIR Evaluation Tasks

One of the main contributions of this study is to build a large cross-cultural dataset for MIR research. The unique characteristics of the dataset built for this study make it suitable for various evaluation tasks involving cross-cultural components. Specifically, for each of the three annotation sets (i.e., 5-clusters, 18-groups, and 2-dimensions), both within- and cross-cultural evaluations can be performed. For the former, both training and test data can be extracted from the datasets with annotations by listeners from the same cultural group (by cross-validation, for example); for the latter, models can be trained by the dataset annotated by listeners in one culture and applied to the dataset annotated by listeners in another culture. These tasks will be able to evaluate whether mood recognition models often used in Western music can be equally applied to 1) non-Western music, specifically K-Pop songs; 2) K-Pop songs annotated by American and/or Korean listeners; and 3) cross-cultural music mood recognition, for both categorical mood classification [17] and dimensional mood regression [5].

## 6. CONCLUSIONS AND FUTURE WORK

This study analyzed music mood annotations on a large set of K-Pop songs provided by listeners from two distinct cultural groups, Americans and Koreans, using three mood annotation models. By comparing annotations from the two cultural groups, differences and similarities were identified and discussed. The unique characteristics of the dataset built in this study will allow it to be used in future MIR evaluation tasks with an emphasis on cross-cultural applicability of mood recognition algorithms and systems. Future work will include detailed and qualitative investigation on the reasons behind the differences between mood judgments of these two user groups as well as listeners from other cultural groups.

<sup>1</sup> <http://freemuse.org/archives/7294>

## 7. ACKNOWLEDGEMENT

Funding support: Korean Ministry of Trade, Industry and Energy (Grant #100372) & the A.W. Mellon Foundation.

## 8. REFERENCES

- [1] E. Y. Bann and J. J. Bryson: “Measuring cultural relativity of emotional valence and arousal using semantic clustering and twitter,” *Proc. of the Annual Conf. of the Cognitive Sci. Soc.*, pp.1809-1814, 2013.
- [2] X. Hu & J. H. Lee: “A cross-cultural study of music mood perception between American and Chinese listeners,” *ISMIR*, pp.535-540. 2012.
- [3] X. Hu, J. S. Downie, C. Laurier, M., Bay and A. Ehmann: “The 2007 MIREX audio mood classification task: Lessons learned,” *ISMIR*, 2008.
- [4] P. N. Juslin and P. Laukka, P.: “Expression, perception, and induction of musical emotions: a review and a questionnaire study of everyday listening,” *J. New Music Research*, vol. 33, no. 3, pp. 217–238, 2004.
- [5] Y. E. Kim, E. M. Schmidt, R. Migenco, B. G. Morton, P. Richardson, J. J. Scott, J. A. Speck, and D. Turnbull: “Music emotion recognition: A state of the art review,” *ISMIR*, pp. 255–266, 2010.
- [6] J. H. Lee: “Crowdsourcing music similarity judgments using Mechanical Turk,” *ISMIR*, 2010.
- [7] J. H. Lee, K. Choi, X. Hu and J. S. Downie: “K-Pop genres: A cross-cultural exploration,” *ISMIR*, 2013.
- [8] J. H. Lee and X. Hu: “Generating ground truth for mood classification in Music Digital Libraries Using Mechanical Turk,” *IEEE-ACM JCDL*, 2012.
- [9] J. H. Lee and X. Hu: “Cross-cultural similarities and differences in music mood perception,” *Proceedings of the iConference*, 2014.
- [10] R. R. McCrae, P. T. Costa, and M. Yik: “Universal aspects of Chinese personality structure,” in M. H. Bond (Ed.) *The Handbook of Chinese Psychology*, pp. 189–207. Hong Kong: Oxford University Press, 1996.
- [11] J. A. Russell: “A circumscript model of affect,” *Journal of Psychology and Soc. Psychology*, vol. 39, no. 6, 1980
- [12] Y. Miyamoto and X. Ma: “Dampening or savoring positive emotions: A dialectical cultural script guides emotion regulation.” *Emotion* 11.6, pp.1346-347, 2011.
- [13] J. J. Schmidt, T. Facca and J. C. Soper: “International variations in divergent creativity and the impact on teaching entrepreneurship,” *Journal of Higher Ed. Theory & Practice*. Vol. 13 Issue 2, pp. 101-109. 2013.
- [14] X. Serra: “A multicultural approach in music information research,” *ISMIR*, 2011.
- [15] P. E. Shrout and J. L. Fleiss: “Intraclass correlations: Uses in assessing rater reliability,” *Psychological Bulletin*, Vol. 86, pp. 420-3428, 1979.
- [16] J. A. Speck, E. M. Schmidt, B. G. Morton and Y. E. Kim: “A comparative study of collaborative vs. traditional annotation methods,” *ISMIR*, 2011.
- [17] Y.-H. Yang and X. Hu: “Cross-cultural music mood classification: A comparison on English and Chinese songs,” *ISMIR*, 2012.

# CADENCE DETECTION IN WESTERN TRADITIONAL STANZAIC SONGS USING MELODIC AND TEXTUAL FEATURES

Peter van Kranenburg, Folgert Karsdorp

Meertens Institute, Amsterdam, Netherlands

{peter.van.kranenburg, folgert.karsdorp}@meertens.knaw.nl

## ABSTRACT

Many Western songs are hierarchically structured in stanzas and phrases. The melody of the song is repeated for each stanza, while the lyrics vary. Each stanza is subdivided into phrases. It is to be expected that melodic and textual formulas at the end of the phrases offer intrinsic clues of closure to a listener or singer. In the current paper we aim at a method to detect such cadences in symbolically encoded folk songs. We take a trigram approach in which we classify trigrams of notes and pitches as cadential or as non-cadential. We use pitch, contour, rhythmic, textual, and contextual features, and a group of features based on the conditions of closure as stated by Narmour [11]. We employ a random forest classification algorithm. The precision of the classifier is considerably improved by taking the class labels of adjacent trigrams into account. An ablation study shows that none of the kinds of features is sufficient to account for good classification, while some of the groups perform moderately well on their own.

## 1. INTRODUCTION

This paper presents both a method to detect cadences in Western folk-songs, particularly in folk songs from Dutch oral tradition, and a study to the importance of various musical parameters for cadence detection.

There are various reasons to focus specifically on cadence patterns. The concept of cadence has played a major role in the study of Western folk songs. In several of the most important folk song classification systems, cadence tones are among the primary features that are used to put the melodies into a linear ordering. In one of the earliest classification systems, devised by Ilmari Krohn [10], melodies are firstly ordered according to the number of phrases, and secondly according to the sequence of cadence tones. This method was adapted for Hungarian melodies by Bártok and Kodály [16], and later on for German folk songs by Suppan and Stief [17] in their monumental *Melodietypen des Deutschen Volksgesanges*. Bronson [3] introduced a number of features for the study of Anglo-American folk song melodies, of which final cadence and

mid-cadence are the most prominent ones. One of the underlying assumptions is that the sequence of cadence tones is relatively stable in the process of oral transmission. Thus, variants of the same melody are expected to end up near to each other in the resulting ordering.

From a cognitive point of view, the perception of closure is of fundamental importance for a listener or singer to understand a melody. In terms of expectation [8, 11], a final cadence implies no continuation at all. It is to be expected that specific features of the songs that are related to closure show different values for cadential patterns as compared to non-cadential patterns. We include a subset of features that are based on the conditions of closure as stated by Narmour [11, p.11].

Cadence detection is related to the problem of segmentation, which is relevant for Music Information Retrieval [21]. Most segmentation methods for symbolically represented melodies are either based on pre-defined rules [4, 18] or on statistical learning [1, 9, 12]. In the current paper, we focus on the musical properties of cadence formulas rather than on the task of segmentation as such.

Taking Dutch folk songs as case study, we investigate whether it is possible to derive a general model of the melodic patterns or formulas that specifically indicate melodic cadences using both melodic and textual features. To address this question, we take a computational approach by employing a random forest classifier (Sections 5 and 6).

To investigate which musical parameters are of importance for cadence detection, we perform an ablation study in which we subsequently remove certain types of features in order to evaluate the importance of the various kinds of features (Section 7).

## 2. DATA

We perform all our experiments on the folk song collection from the Meertens Tune Collections (MTC-FS, version 1.0), which is a set of 4,120 symbolically encoded Dutch folk songs.<sup>1</sup> Roughly half of it consists of transcriptions from field recordings that were made in the Netherlands during the 20th century. The other half is taken from song books that contain repertoire that is directly related to the recordings. Thus, we have a coherent collection of songs that reflects Dutch everyday song culture in the early 20th century. Virtually all of these songs have a stanzaic structure. Each stanza repeats the melody, and each stanza



© Peter van Kranenburg, Folgert Karsdorp.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Peter van Kranenburg, Folgert Karsdorp. “Cadence Detection in Western Traditional Stanzaic Songs using Melodic and Textual Features”, 15th International Society for Music Information Retrieval Conference, 2014.

<sup>1</sup> Available from: <http://www.liederenbank.nl/mtc>.

consists of a number of phrases. Both in the transcriptions and in the song books, phrase endings are indicated. Figure 1 shows a typical song from the collection. The language of the songs is standard Dutch with occasionally some dialect words or nonsense syllables. All songs were digitally encoded by hand at the Meertens Institute (Amsterdam) and are available in Humdrum **\*\*kern** format. The phrase endings were encoded as well and are available for computational analysis and modeling.

### 3. OUR APPROACH

Our general approach is to isolate trigrams from the melodies and to label those as either cadential or non-cadential. A cadential trigram is the last trigram in a phrase. We compare two kinds of trigrams: trigrams of successive notes (*note-trigrams*), and trigrams of successive pitches (*pitch-trigrams*), considering repeated pitches as one event. In the case of pitch-trigrams, a cadence pattern always consists of the three last unique pitches of the phrase. There are two reasons for including pitch-trigrams. First, pitch repetition is often caused by the need to place the right number of syllables to the melody. It occurs that a quarter note in one stanza corresponds to two eighth notes in another stanza because there is an extra syllable at that spot in the song text. Second, in models of closure in melody [11, 15] successions of pitches are of primary importance.

Figure 1 depicts all pitch-trigrams in the presented melody. The trigram that ends on the final note of a phrase is a cadential trigram. These are indicated in bold. Some cadential trigrams cross a phrase boundary when the next phrase starts with the same pitch.

From each trigram we extract a number of feature values that reflect both melodic and textual properties. We then perform a classification experiment using a Random Forest Classifier [2]. This approach can be regarded a ‘bag-of-trigrams’ approach, where each prediction is done independently of the others, i.e. all sequential information is lost. Therefore, as a next step we take the labels of the direct neighboring trigrams into account as well. The final classification is then based on a majority vote of the predicted labels of adjacent trigrams. These steps will be explained in detail in the next sections.

**Figure 1.** Examples of pitch-trigrams. The cadential trigrams are indicated in bold.

## 4. FEATURES

We represent each trigram as a vector of feature values. We measure several basic properties of the individual pitches and of the pattern as a whole. The code to automatically extract the feature values was written in Python, using the music21 toolbox [5]. The features are divided into groups that are related to distinct properties of the songs. Some features occur in more than one group. The following overview shows all features and in parentheses the value for the first trigram in Figure 1. Detailed explanations are provided in sections 4.1 and 4.2.

### Pitch Features

*Scale degree* Scale degrees of the first, second, and third item (5, 1, 3).

*Range* Difference between highest and lowest pitch (4).

*Has contrast third* Whether there are both even and odd scale degrees in the trigram (False).

### Contour Features

*Contains leap* Whether there is a leap in the trigram (True).

*Is ascending* Whether the first and second intervals, and both are ascending (False, True, False).

*Is descending* Whether the first and second intervals, and both are descending (True, False, False).

*Large-small* Whether the first interval is large and the second is small (True).

*Registral change* Whether there is a change in direction between the first and the second interval (True).

### Rhythmic Features

*Beat strength* The metric weights of the first, second and third item (0.25, 1.0, 0.25).

*Min beat strength* The smallest metric weight (0.25).

*Next is rest* Whether a rest follows the first, second and third item (False, False, False).

*Short-long* Whether the second item is longer than the first, and the third is longer than the second (False, False).

*Meter* The meter at the beginning of the trigram (“6/8”).

### Textual Features

*Rhymes* Whether a rhyme word ends at the first, second and third item (False, False, False).

*Word stress* Whether a stressed syllable is at the first, second and third item (True, True, True).

*Distance to last rhyme* Number of notes between the last the first, second and third item and the last rhyme word or beginning of the melody (0, 1, 2).

### Narmour Closure Features

*Beat strength* The metric weights of the first, second and third item (0.25, 1.0, 0.25).

*Next is rest* Whether a rest follows the first, second and third item (False, False, False).

*Short-long* Whether the second item is longer than the first, and the third is longer than the second (False, False).

*Large-small* Whether the first interval is large ( $\geq$  fifth) and the second is small ( $\leq$  third) (True).

*Registral change* Whether there is a change in direction between the first and the second interval (True).

### Contextual Features

*Next is rest third* Whether a rest or end of melody follows the third item (False).

*Distance to last rhyme* Number of notes between the last the first, second and third item and the last rhyme word or beginning of the melody (0, 1, 2).

### 4.1 Melodic Features

Several of the features need some explanation. In this section we describe the melodic features, while in the next section, we explain how we extracted the textual features.

HasContrastThird is based on the theory of Jos Smits-Van Waesberghe [15], the core idea of which is that a melody gets its tension and interest by alternating between



pitches with even and uneven scale degrees, which are two contrasting series of thirds.

The metric weight in the Rhythmic features is the beat-strength as implemented in music21's meter model.

The Narmour features are based on the six (preliminary) conditions of closure that Narmour states at the beginning of his first book on the Implication-Realisation theory [11, p.11]: “[...] melodic closure on some level occurs when 1. a rest, an onset of another structure, or a repetition interrupts an implied pattern; 2. metric emphasis is strong; 3. consonance resolves dissonance; 4. duration moves cumulatively (short note to long note); 5. intervallic motion moves from large interval to small interval; 6. registral direction changes (up to down, down to up, lateral to up, lateral to down, up to lateral, or down to lateral).” Because the melodies are monophonic, condition 3 has no counterpart in our feature set.

The contextual features are not features of the trigram in isolation, but are related to the position in the melody. In an initial experiment we found that the distance between the first note of the trigram and the last cadence is an important predictor for the next cadence. Since this is based on the ground-truth label, we cannot include it directly into our feature set. Since we expect rhyme in the text to have a strong relation with cadence in the melody, we include the distance to the last rhyme word in number of notes.

Figure 2 shows two lines of musical notation. The first line has lyrics: Jan Al - berts rij - en wou gaan. zag ver mooi. Below the lyrics are phonological representations: jAn Al - b@rts rE+ - j@ wA+w xan zAx vEr moj. The second line has lyrics: meis - je staan. zag ver mooi meis - je staan. Below the lyrics are phonological representations: mE+s-je@ stan zAx vEr moj mE+s-je@ stan. The third line shows whether rhyme is detected: False False False False False True False False False True.

**Figure 2.** Rhyme as detected by our method. The first line shows the original text after removing non-content words. The second line shows the phonological representations of the words (in SAMPA notation). The third line shows whether rhyme is detected (‘True’ if a rhyme word *ends* at the corresponding note).

## 4.2 Textual Features

In many poetical texts, phrase boundaries are determined by sequences of rhyme. These establish a structure in a text, both for aesthetics pleasure and memory aid [14]. In folk music, phrasal boundaries established by sequences of rhyme are likely to relate to phrases in the melody.

We developed a rhyme detection system which allows us to extract these sequences of rhyming lyrics. Because of orthographical ambiguities (e.g. *cruise*, where /u:/ is represented by *ui* whereas in *muse* it is represented by *u*), it is not as straightforward to perform rhyme detection on orthographical representations of words. Therefore, we transform each word into its phonological representation (e.g. *cruise* becomes /kru:z/ and *bike* /baik/).

-	-	-	c	r	u	i	'k
-	-	c	r	u	i	s	r
-	c	r	u	i	s	e	u:
c	r	u	i	s	e	-	0
r	u	i	s	e	-	-	z
u	i	s	e	-	-	-	0

**Figure 3.** Example sliding window for phoneme classification.

We approach the problem of phonemicization as a supervised classification task, where we try to predict for each character in a given word its corresponding phoneme. We take a sliding window-based approach where for each focus character (i.e. the character for which we want to predict its phonemic representation) we extract as features  $n$  characters to the left of the focus character,  $n$  characters to the right, and the focus character itself. Figure 3 provides a graphical representation of the feature vectors extracted for the word *cruise*. The fourth column represents the focus character with a context of three characters before and three after the focus character. The last column represents the target phonemes which we would like to predict. Note that the first target phoneme in Figure 3 is preceded by an apostrophe (‘k), which represents the stress position on the first (and only) syllable in *cruise*. This symbolic notation of stress in combination with phonology allows us to simultaneously extract a phonological representation of the input words as well as their stress patterns. For all words in the lyrics in the dataset we apply our sliding window approach with  $n = 5$ , which serves as input for the supervised classifier. In this paper we make use of a  $k = 1$  Nearest Neighbor Classifier as implemented by [6] using default settings, which was trained on the data of the e-Lex database<sup>2</sup>. In the running text of our lyrics, 89.5% of the words has a direct hit in the instance base, and for the remaining words in many cases suitable nearest neighbors were found. Therefore, we consider the phonemicization sufficiently reliable.

We assume that only content words (nouns, adjectives, verbs and adverbials) are possible candidate rhyme words. This assumption follows linguistic knowledge as phrases typically do not end with function words such as determiners, prepositions, etcetera. Function words are part of a closed category in Dutch. We extract all function words from the lexical database e-Lex and mark for each word in each lyric whether it is a function word.

We implemented rhyme detection according to the rules for Dutch rhyme as stated in [19]. The algorithm is straightforward. We compare the phoneme-representations of two words backwards, starting at the last phoneme, until we reach the first vowel, excluding schwas. If all phonemes

<sup>2</sup> <http://tst-centrale.org/en/producten/lexica/e-lex/7-25>

Class	pr	rec	$F_1$	$\sigma_{F_1}$	support
<i>note-trigrams</i>					
cadence	0.84	0.72	0.78	0.01	23,925
nocadence	0.96	0.98	0.97	0.01	183,780
<i>pitch-trigrams</i>					
cadence	0.85	0.69	0.76	0.01	23,838
nocadence	0.95	0.98	0.96	0.00	130,992

**Table 1.** Results for single labels.

and the vowel are exactly the same, the two words rhyme.

As an example we take *kinderen* ('children') and *hinderen* ('to hinder'). The phoneme representations as produced by our method are /kɪndərə/ and /hɪndərə/. The first vowel starting from the back of the word, excluding the schwas (/ə/), is /ɪ/. Starting from this vowel, the phoneme representations of both words are identical (/ɪndərə/). Therefore these words rhyme.

We also consider literal repetition of a word as 'rhyme', but not if a sequence of words is repeated literally, such as in the example in Figure 1. Such repetition of entire phrases occurs in many songs. Labeling all words as rhyme words would weaken the relation with cadence or 'end-of-sentence'. We only label the last word of repeated phrases as a rhyme word. Figure 2 shows an example.

## 5. CLASSIFICATION WITH SINGLE LABELS

As a first approach we consider the trigrams independently. A melody is represented as 'bag-of-trigrams'. Each trigram has a ground-truth label that is either 'cadence' or 'no cadence', as depicted in Figure 1 for pitch-trigrams'

We employ a Random Forest classifier [2] as implemented in the Python library scikit-learn [13]. This classifier combines  $n$  decision trees (*predictors*) that are trained on random samples extracted from the data (with replacement). The final classification is a majority vote of the predictions of the individual trees. This procedure has proven to perform more robustly than a single decision tree and is less prone to over-fitting the data. Given the relatively large size of our data set, we set the number of predictors to 50 instead of the default 10. For the other parameters, we keep the default values.

The evaluation is performed by 10-fold cross-validation. One non-trivial aspect of our procedure is that we construct the folds at the level of the songs, rather than at that of individual trigrams. Since it is quite common for folk songs to have phrases that are literally repeated, folding at the level of trigrams could result in identical trigrams in the train and test subsets, which could lead to an overfitted classifier. By ensuring that all trigrams from a song are either in the test or in the train subset, we expect better generalization. This procedure is applied throughout this paper.

The results are shown in Table 1. For both classes averages of the values for the precision, the recall and the  $F_1$ -measure over the folds are included, as well as the standard deviation of the  $F_1$  measure, which indicates the variation over the folds. The number of items in both classes (*sup-*

Class	pr	rec	$F_1$	$\sigma_{F_1}$	support
<i>note-trigrams</i>					
cadence	0.89	0.72	0.80	0.01	23,925
nocadence	0.96	0.99	0.98	0.00	183,780
<i>pitch-trigrams</i>					
cadence	0.89	0.71	0.79	0.01	23,838
nocadence	0.95	0.98	0.97	0.01	130,992

**Table 2.** Results for classification with label trigrams.

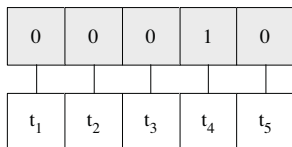
*port*) shows that cadences are clearly a minority class.

We observe that the note-trigrams lead to slightly better cadence-detection as compared to pitch-trigrams. Apparently, the repetition of pitches does not harm the discriminability. Furthermore, there is an unbalance between the precision and the recall of the cadence-trigrams. The precision is rather high, while the recall is moderate.

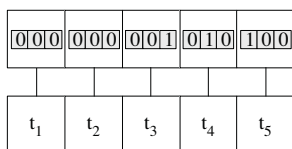
## 6. CLASSIFICATION WITH LABEL TRIGRAMS

When our cadence detection system predicts the class of a new trigram, it is oblivious of the decisions made for earlier predictions. One particularly negative effect of this near-sightedness is that the classifier frequently predicts two (or even more) cadences in a row, which, given our training material, is extremely unlikely. We attempt to circumvent this 'defect' using a method, developed by [20] that predicts trigrams of class labels instead of single, binary labels. Figure 4 depicts the standard single class classification setting, where each trigram is predicted independent of all other predictions. In the label trigram setting (see Figure 5), the original class labels are replaced with the class label of the previous trigram, the class label of the current trigram and the label of the next trigram. The learning problem is transformed into a sequential learning problem with two stages. In the first stage we predict for each trigram a label trigram  $y^{(t)} = (y_1, y_2, y_3)$  where  $y \in \{0, 1\}$ . To arrive at the final single class predictions (i.e. is it a cadence or not), in the second stage we take the majority vote over the predictions of the focus trigram and those of its immediate left and right neighboring trigrams. Take  $t_4$  in Figure 5 as an example. It predicts that the current trigram is a cadence. The next trigram and the previous trigram also predict it to be a cadence and based on this majority vote, the final prediction is that  $t_4$  is a cadence. Should  $t_3$  and  $t_5$  both have predicted the zero class (e.g.  $y^{(t_3)} = (0, 0, 0)$  and  $y^{(t_5)} = (0, 1, 0)$ ), the majority vote would be 0. The advantage of this method is that given the negligible number of neighboring cadences in our training data, we can virtually rule out the possibility to erroneously predict two or more cadences in a row.

Table 2 shows the performance of the label-trigram classifier for both classes and both for pitch and note trigrams. The values show an important improvement for the precision of cadence-detection and a slight improvement of the recall. The lower number of false positives is what we expected by observing the classification of adjacent trigrams as 'cadence' in the case of the single-label classifier.



**Figure 4.** Short example sequence of trigrams. Each trigram  $t_i$  has a binary label indicating whether the trigram is cadential (1) or non-cadential (0).



**Figure 5.** Label-trigrams for the same sequence as in Figure 1, where  $t_4$  has label 1 and the other trigrams have label 0. Each trigram  $t_i$  gets a compound label consisting of its own label and the labels of the direct neighboring trigrams.

## 7. ABLATION STUDY

To study the importance of the various kinds of features, we perform an ablation study. We successively remove each of the groups of features as defined in section 4 from the full set and do a classification experiment with the remaining features. Subsequently, we perform a similar series of classification experiments, but now with each single group of features. The first series shows the importance of the individual groups of features, and the second series shows the predictive power for each of the groups. Because the groups are assembled according to distinct properties of music and text, this will give insight in the importance of various musical and textual parameters for cadence detection. We use the label-trigram classifier with the note-trigrams, which performed best on the full set.

We expect occurrence of rests to be a very strong predictor, because according to our definition a ‘rest’ always follows after the final cadence, and we know that in our corpus rests almost exclusively occur between phrases. Therefore, we also take the three features that indicate whether a rest occurs in the trigram or directly after it, as a separate group. The performance when leaving these three features out will show whether they are crucial for cadence detection.

Table 3 shows the evaluation measures for each of the feature subsets. Precision, recall and  $F_1$  for class ‘cadence’ are reported. Again, the values are averaged over 10 folds.

We see that none of the single groups of features is crucial for the performance that was achieved with the complete set of features. The basic melodic features ( $F_{pitch}$ ,  $F_{contour}$ , and  $F_{rhythmic}$ ) all perform very bad on their own, showing low to extremely low recall values. The contour features even do not contribute at all. Only the rhythmic features yield some performance. The features on rest are

Subset	pr	rec	$F_1$	$\sigma_{F_1}$
$F_{all}$	0.89	0.72	0.80	0.01
$F_{all} \setminus F_{pitch}$	0.88	0.72	0.79	0.01
$F_{pitch}$	0.84	0.04	0.08	0.01
$F_{all} \setminus F_{contour}$	0.88	0.73	0.80	0.01
$F_{contour}$	0.00	0.00	0.00	0.00
$F_{all} \setminus F_{rhythmic}$	0.79	0.49	0.60	0.01
$F_{rhythmic}$	0.90	0.35	0.50	0.01
$F_{all} \setminus F_{textual}$	0.85	0.58	0.69	0.02
$F_{textual}$	0.70	0.40	0.51	0.01
$F_{all} \setminus F_{narmour}$	0.83	0.55	0.66	0.01
$F_{narmour}$	0.95	0.30	0.45	0.01
$F_{all} \setminus F_{contextual}$	0.87	0.67	0.76	0.01
$F_{contextual}$	0.71	0.45	0.56	0.01
$F_{all} \setminus F_{rest}$	0.87	0.67	0.76	0.01
$F_{rest}$	0.97	0.27	0.43	0.02

**Table 3.** Results for various feature subsets for class ‘cadence’.

included in the set of rhythmic features. The classification with just the features on rest,  $F_{rest}$  shows very high precision and low recall. Still, the recall with all rhythmic features is higher than only using the rest-features. Since rests are so tightly related to cadences in our corpus, the high precision for  $F_{rest}$  is what we expected. If we exclude the rest-features, the precision stays at the same level as for the entire feature set and the recall drops with 0.06, which shows that only a minority of the cadences exclusively relies on rest-features to be detected.

The set of features that is based on the conditions of closure as formulated by Narmour shows high precision and low recall. Especially the high precision is interesting, because this confirms Narmour’s conditions of closure. Apparently, most patterns that are classified as cadence based on this subset of features, are cadences indeed. Still, the low recall indicates that there are many cadences that are left undetected. One cause could be that the set of conditions as stated by Narmour is not complete, another cause could be the discrepancy between our features and Narmour’s conditions. Further investigation would be necessary to shed light on this. Removing the Narmour-based features from the full feature set does not have a big impact. The other features have enough predictive power.

The textual features on their own show moderate precision and very moderate recall. They are able to discern certain kinds of cadences to a certain extent, while missing most of the other cadences. The drop of 0.14 in recall for  $F_{all} \setminus F_{textual}$  as compared to the full set shows that text features are crucial for a considerable number of cadences to be detected. The same applies to a somewhat lesser extent to contextual features. Removing the contextual features from the full set causes a drop of 0.05 in the recall, which is considerable but not extreme. It appears that the group of cadence trigrams for which the contextual features are crucial is not very big.

## 8. CONCLUSION AND FUTURE WORK

In this paper we developed a system to detect cadences in Western folk songs. The system makes use of a Random Forest Classifier that on the basis of a number of hand-crafted features (both musical and textual) is able to accurately locate cadences in running melodies. In a follow-up experiment we employ a method, originally developed for textual sequences, that predicts label-trigrams instead of the binary labels ‘cadence’ or ‘non-cadence’. We show that incorporating the predictions of neighboring instances into the final prediction, has a strong positive effect on precision without a loss in recall.

In the ablation study we found that all groups of features, except for the contour features, contribute to the overall classification, while none of the groups is crucial for the majority of the cadences to be detected. This indicates that cadence detection is a multi-dimensional problem for which various properties of melody and text are necessary.

The current results give rise to various follow-up studies. A deeper study to the kinds of errors of our system will lead to improved features and increased knowledge about cadences. Those that were detected exclusively by textual features form a particular interesting case, possibly giving rise to new melodic features. Next, n-grams other than trigrams as well as skip-grams [7] could be used, we will compare the performance of our method with existing symbolic segmentation algorithms, and we want to make use of other features of the text such as correspondence between syntactic units in the text and melodic units in the melody.

## 9. REFERENCES

- [1] Rens Bod. Probabilistic grammars for music. In *Proceedings of BNAIC 2001*, 2001.
- [2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [3] Bertrand H Bronson. Some observations about melodic variation in british-american folk tunes. *Journal of the American Musicological Society*, 3:120–134, 1950.
- [4] Emiliós Cambouropoulos. The local boundary detection model (lbdm) and its application in the study of expressive timing. In *Proc. of the Intl. Computer Music Conf*, 2001.
- [5] Michael Scott Cuthbert and Christopher Ariza. Music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR 2010)*, pages 637–642, 2010.
- [6] Walter Daelemans, Jakub Zavrel, Ko Van der Sloot, and Antal Van den Bosch. TiMBL: Tilburg Memory Based Learner, version 6.3, Reference Guide, 2010.
- [7] David Guthrie, Ben Allison, W. Liu, Louise Guthrie, and Yorick Wilks. A closer look at skip-gram modelling. In *Proceedings of the Fifth international Conference on Language Resources and Evaluation LREC-2006*, 2006.
- [8] David Huron. *Sweet Anticipation*. MIT Press, Cambridge, Mass., 2006.
- [9] Zoltán Juhász. Segmentation of hungarian folk songs using an entropy-based learning system. *Journal of New Music Research*, 33(1):5–15, 2004.
- [10] Ilmari Krohn. Welche ist die beste Methode, um Volks- und volksmässige Lieder nach ihrer melodischen (nicht textlichen) Beschaffenheit lexikalisch zu ordnen? *Sammelbände der internationalen Musikgesellschaft*, 4(4):643–60, 1903.
- [11] Eugene Narmour. *The Analysis and Cognition of Basic Melodic Structures - The Implication-Realization Model*. The University of Chicago Press, Chicago and Londen, 1990.
- [12] Marcus Pearce, Daniel Müllensiefen, and Geraint Wiggins. The role of expectation and probabilistic learning in auditory boundary perception: A model comparison. *Perception*, 39(10):1365–1389, 2010.
- [13] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [14] David C. Rubin. *Memory in Oral Traditions*. Oxford University Press, New York, 1995.
- [15] Jos Smits van Waesberghe. *A Textbook of Melody: A course in functional melodic analysis*. American Institute of Musicology, 1955.
- [16] B. Suchoff. *Preface*, pages ix–lv. State University of New York Press, Albany, 1981.
- [17] W. Suppan and W. Stief, editors. *Melodietypen des Deutschen Volksgesanges*. Hans Schneider, Tutzing, 1976.
- [18] David Temperley. A probabilistic model of melody perception. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, Victoria, BC, 2006.
- [19] Erica van Boven and Gillis Dorleijn. *Literair Mechaniek*. Coutinho, Bussum, 2003.
- [20] Antal Van den Bosch and Walter Daelemans. Improving sequence segmentation learning by predicting tri-grams. In *Proceedings of the Ninth Conference on Natural Language Learning, CoNLL-2005*, pages 80–87, Ann Arbor, MI, 2005.
- [21] Frans Wiering and Hermi J.M. Tabachneck-Schijf. Cognition-based segmentation for music information retrieval systems. *Journal of New Music Research*, 38(2):137–154, 2009.

# DISCOVERING TYPICAL MOTIFS OF A *RĀGA* FROM ONE-LINERS OF SONGS IN CARNATIC MUSIC

**Shrey Dutta**

Dept. of Computer Sci. & Engg.  
Indian Institute of Technology Madras  
shrey@cse.iitm.ac.in

**Hema A. Murthy**

Dept. of Computer Sci. & Engg.  
Indian Institute of Technology Madras  
hema@cse.iitm.ac.in

## ABSTRACT

Typical motifs of a *rāga* can be found in the various songs that are composed in the same *rāga* by different composers. The compositions in Carnatic music have a definite structure, the one commonly seen being pallavi, anupallavi and charanam. The *tala* is also fixed for every song.

Taking lines corresponding to one or more cycles of the pallavi, anupallavi and charanam as one-liners, one-liners across different songs are compared using a dynamic programming based algorithm. The density of match between the one-liners and normalized cost along-with a new measure, which uses the stationary points in the pitch contour to reduce the false alarms, are used to determine and locate the matched pattern. The typical motifs of a *rāga* are then filtered using compositions of various *rāgas*. Motifs are considered typical if they are present in the compositions of the given *rāga* and are not found in compositions of other *rāgas*.

## 1. INTRODUCTION

Melody in Carnatic music is based on a concept called *rāga*. A *rāga* in Carnatic music is characterised by typical phrases or motifs. The phrases are not necessarily scale-based. They are primarily pitch trajectories in the time-frequency plane. Although for annotation purposes, *rāgas* in Carnatic are based on 12 srutis (or semitones), the *gamakās* associated with the same semitone can vary significantly across *rāgas*. Nevertheless, although the phrases do not occupy fixed positions in the time-frequency (t-f) plane, a listener can determine the identity of a *rāga* within few seconds of an *ālāpana*. An example, is a concert during the “music season” in Chennai, where more than 90% of the audience can figure out the *rāga*. This despite the fact that more than 80% of the audience are nonprofessionals. The objective of the presented work is to determine typical motifs of a *rāga* automatically. This is obtained by analyzing various compositions that are composed in a particular *rāga*. Unlike Hindustani music, there is a huge

repository of compositions that have been composed by a number of composers in different *rāgas*. It is often stated by musicians that the famous composers have composed such that a single line of a composition is replete with the motifs of the *rāga*. In this paper, we therefore take one-liners of different compositions and determine the typical motifs of the *rāga*.

Earlier work, [9, 10], on identifying typical motifs depended on a professional musician who sung the typical motifs for that *rāga*. These typical motifs were then spotted in *ālāpanas* which are improvisational segments. It was observed that the number of false alarms were high. High ranking false alarms were primarily due to partial matches with the given query. Many of these were considered as an instance of the queried motif by some musicians. As *alapanas* is an improvisational segment, the rendition of the same motif could be different across *alapanas* especially among different schools. On the other hand, compositions in Carnatic music are rendered more or less in a similar manner. Although the music evolved through the oral tradition and fairly significant changes have crept into the music, compositions renditions do not vary very significantly across different schools. The number of variants for each line of the song can vary quite a lot though. Nevertheless, the meter of motifs and the typical motifs will generally be preserved.

It is discussed in [15] that not all repeating patterns are interesting and relevant. In fact, the vast majority of exact repetitions within a music piece are not musically interesting. The algorithm proposed in [15] mostly generates interesting repeating patterns along with some non-interesting ones which are later filtered during post processing. The work presented in this paper is an attempt from a similar perspective. The only difference is that typical motifs of *rāgas* need not be interesting to a listener. The primary objective for discovering typical motifs, is that these typical motifs can be used to index the audio of a rendition. Typical motifs could also be used for *rāga* classification. The proposed approach in this work generates similar patterns across one-liners of a *rāga*. From these similar patterns, the typical motifs are filtered by using compositions of various *rāgas*. Motifs are considered typical of a *rāga* if they are present in the compositions of a particular *rāga* and are NOT found in other *rāgas*. This filtering approach is similar to anti-corpus approach of Conklin [6, 7] for the discovery of distinctive patterns.



© Shrey Dutta, Hema A. Murthy.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Shrey Dutta, Hema A. Murthy. “Discovering typical motifs of a *Rāga* from one-liners of songs in Carnatic Music”, 15th International Society for Music Information Retrieval Conference, 2014.

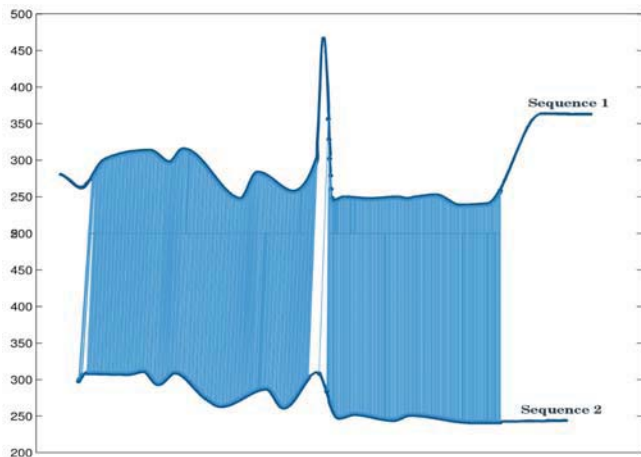


Figure 1. RLCS matching two sequences partially

Most of the previous work, regarding discovery of repeated patterns of interest in music, is on western music. In [11], B. Jansen *et al* discusses the current approaches on repeated pattern discovery. It discusses string based methods and geometric methods for pattern discovery. In [14], Lie Lu *et al* used constant Q transforms and proposed a similarity measure between musical features for doing repeated pattern discovery. In [15], Meredith *et. al.* presented Structure Induction Algorithms (SIA) using a geometric approach for discovering repeated patterns that are musically interesting to the listener. In [4, 5], Collins *et. al.* introduced improvements in Meredith’s Structure Induction Algorithms. There has also been some significant work on detecting melodic motifs in Hindustani music by Joe Cheri Ross *et. al.* [16]. In this approach, the melody is converted to a sequence of symbols and a variant of dynamic programming is used to discover the motif.

In a Carnatic music concert, many listeners from the audience are able to identify the *rāga* at the very beginning of the composition, usually during the first line itself — a line corresponds to one or more tala cycles. Thus, first lines of the compositions could contain typical motifs of a *rāga*. A pattern which is repeated within a first line could still be *not* specific to a *rāga*. Whereas, a pattern which is present in most of the first lines could be a typical motif of that *rāga*. Instead of just using first lines, we have also used other one-liners from compositions, namely, lines from the pallavi, anupallavi and charanam. In this work, an attempt is made to find repeating patterns *across* one-liners and *not* within a one-liner. Typical motifs are filtered from the generated repeating patterns during post processing. These typical motifs are available online <sup>1</sup>

The length of the typical motif to be discovered is not known a priori. Therefore there is a need for a technique which can itself determine the length of the motif at the time of discovering it. Dynamic Time Warping (DTW) based algorithms can only find a pattern of a specific length since it performs end-to-end matching of the query and test sequence. There is another version of DTW known as

<sup>1</sup><http://www.iitm.ac.in/donlab/typicalmotifs.html>

Unconstrained End Point-DTW (UE-DTW) that can match the whole query with a partial test but still the query is not partially matched. Longest Common Subsequence (LCS) algorithm on the other hand can match the partial query with partial test sequence since it looks for a longest common subsequence which need not be end-to-end. LCS by itself is not appropriate as it requires discrete symbols and does not account for local similarity. A modified version of LCS known as Rough Longest Common Subsequence takes continuous symbols and takes into account the local similarity of the longest common subsequence. The algorithm proposed in [13] to find rough longest common subsequence between two sequences fits the bill for our task of motif discovery. An example of RLCS algorithm matching two partial phrases is shown in Figure 1. The two music segments are represented by their tonic normalized smoothed pitch contours [9, 10]. The stationary points, where the first derivative is zero, of the tonic normalized pitch contour are first determined. The points are then interpolated using cubic Hermite interpolation to smooth the contour.

In previous uses of RLCS for motif spotting task [9, 10], a number of false alarms were observed. One of the most prevalent false alarms is the test phrase with a sustained note which comes in between the notes of the query. The slope of the linear trend in stationary points along with its standard deviation is used to address this issue.

The rest of the paper is organized as follows. In Section 2 the use of one-liners of compositions to find motifs is discussed. Section 3 discusses the optimization criteria to find the rough longest common subsequence. Section 4 describes the proposed approach for discovering typical motifs of *rāgas*. Section 5 describe the dataset used in this work. Experiments and results are presented in Section 6.

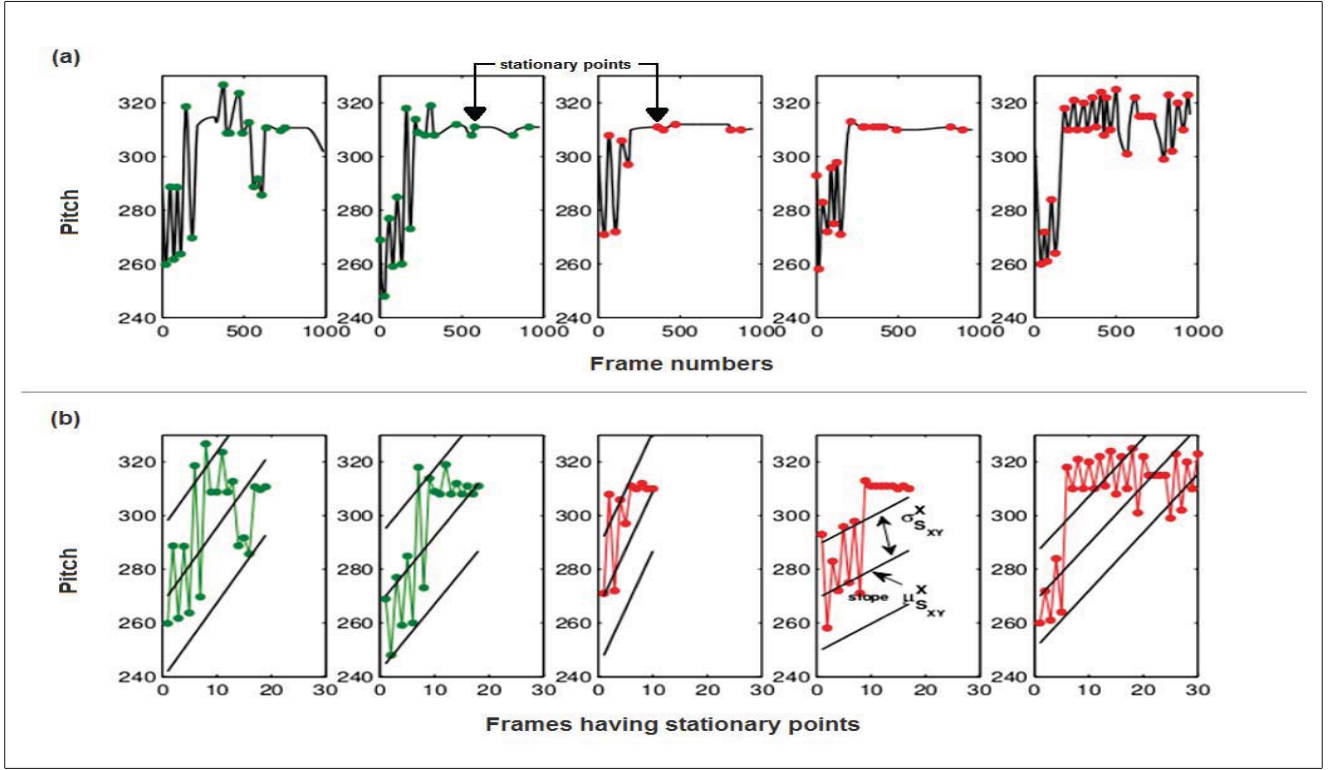
## 2. ONE-LINERS OF SONGS

As previously mentioned, first line of the composition contains the characteristic traits of a *rāga*. The importance of the first lines and the *rāga* information it holds is illustrated in great detail in the T. M. Krishna’s book on Carnatic music [12]. T. M. Krishna states that opening section called “pallavi” directs the melodic flow of the *rāga*. Through its rendition, the texture of the *rāga* can be felt. Motivated by this observation, an attempt is made to verify the conjecture that typical motifs of a *rāga* can be obtained from the first lines of compositions.

Along with the lines from pallavi, we have also selected few lines from other sections, namely, ‘anupallavi’ and ‘charanam’. Anupallavi comes after pallavi and the melodic movements in this section tend to explore the *rāga* in the higher octave [12]. These lines are referred to as one-liners for a *rāga*.

## 3. OPTIMIZATION CRITERIA TO FIND ROUGH LONGEST COMMON SUBSEQUENCE

The rough longest common subsequence (rlcs) between two sequences,  $X = \langle x_1, x_2, \dots, x_n \rangle$  and  $Y = \langle y_1, y_2, \dots$



**Figure 2.** (a) Pitch contour of the five phrases which are considered similar. Stationary points are marked in green and red for the true positives and false alarms respectively. (b) Pitch values only at the stationary points. Slope of the linear trend in stationary points along-with its standard deviation helps in reducing the false alarms.

,  $y_m$ ), of length  $n$  and  $m$  is defined as the longest common subsequence (lcs)  $Z_{XY} = \langle (x_{i_1}, y_{j_1}), (x_{i_2}, y_{j_2}), \dots, (x_{i_p}, y_{j_p}) \rangle$ ,  $1 \leq i_1 < i_2 < \dots < i_p \leq n$ ,  $1 \leq j_1 < j_2 < \dots < j_p \leq m$ ; such that the similarity between  $x_{i_k}$  and  $y_{j_k}$  is greater than a threshold,  $\tau_{sim}$ , for  $k = 1, \dots, p$ . There are no constraints on the length and on the local similarity of the rlcs. Some applications demand the rlcs to be locally similar or its length to be in a specific range. For the task of motif discovery along with these constraints, one more constraint is used to reduce false alarms. Before discussing the optimization measures used to find the rlcs in this work, a few quantities need to be defined.

$$l_{S_{XY}}^w = \sum_{k=1}^s sim(x_{i_k}, y_{j_k}) \quad (1)$$

$$g_X = i_s - i_1 + 1 - s \quad (2)$$

$$g_Y = j_s - j_1 + 1 - s \quad (3)$$

Let  $S_{XY} = \langle (x_{i_1}, y_{j_1}), (x_{i_2}, y_{j_2}), \dots, (x_{i_s}, y_{j_s}) \rangle$ ,  $1 \leq i_1 < i_2 < \dots < i_s \leq n$ ,  $1 \leq j_1 < j_2 < \dots < j_s \leq m$ ; be a rough common subsequence (rcs) of length  $s$  and  $sim(x_{i_k}, y_{j_k}) \in [0, 1]$  be the similarity between  $x_{i_k}$  and  $y_{j_k}$  for  $k = 1, \dots, s$ . Equation (1) defines the weighted length of  $S_{XY}$  as sum of similarities,  $sim(x_{i_k}, y_{j_k})$ ,  $k = 1, \dots, s$ . Thus, weighted length is less than or equal to  $s$ . The number of points in the shortest substring of sequence  $X$ , containing the rcs  $S_{XY}$ , that are not the part of the rcs  $S_{XY}$  are termed as gaps in  $S_{XY}$  with respect to sequence  $X$  as defined by Equation (2). Similarly, Equation (3) de-

fines the gaps in  $S_{XY}$  with respect to sequence  $Y$ . Small gaps indicate that the distribution of rcs is dense in that sequence.

The optimization measures to find the rlcs are described as follows.

### 3.1 Density of the match

Equation (4) represents the distribution of the rcs  $S_{XY}$  in the sequences  $X$  and  $Y$ . This is called density of match,  $\delta_{S_{XY}}$ . This quantity needs to be maximized to make sure the subsequence,  $S_{XY}$ , is locally similar.  $\beta \in [0, 1]$  weighs the individual densities in sequences  $X$  and  $Y$ .

$$\delta_{S_{XY}} = \beta \frac{l_{S_{XY}}^w}{l_{S_{XY}}^w + g_X} + (1 - \beta) \frac{l_{S_{XY}}^w}{l_{S_{XY}}^w + g_Y} \quad (4)$$

### 3.2 Normalized weighted length

The weighted length of rcs is normalized as shown in Equation (5) to restrict its range to  $[0, 1]$ .  $n$  and  $m$  are the lengths of sequences  $X$  and  $Y$ , respectively.

$$\hat{l}_{S_{XY}}^w = \frac{l_{S_{XY}}^w}{\min(m, n)} \quad (5)$$

### 3.3 Linear trend in stationary points

As observed in [9, 10], the rlcs obtained using only the above two optimization measures suffered from a large number of false alarms for the motif spotting task. The false alarms generally constituted of long and sustained notes.

This resulted in good normalised weighted lengths and density. To address this issue, the slope and standard deviation of the slope of the linear trend in stationary points of a phrase are estimated. Figure 2 shows a set of phrases. This set has five phrases which are termed as similar phrases based on their density of match and normalized weighted length. The first two phrases, shown in green, are true positives while the remaining, shown in red, are false alarms. Figure 2 also shows the linear trend in stationary points for the corresponding phrases. It is observed that the trends are similar for true positives when compared to that of the false alarms. The slope of the linear trend for the fifth phrase (false alarm) is similar to the true positives but its standard deviation is less. Therefore, a combination of the slope and the standard deviation of the linear trend is used to reduce the false alarms.

Let the stationary points in the shortest substring of sequences  $X$  and  $Y$  containing the rcs  $S_{XY}$  be  $\langle x_{q_1}, x_{q_2}, \dots, x_{q_{t_x}} \rangle$  and  $\langle y_{r_1}, y_{r_2}, \dots, y_{r_{t_y}} \rangle$  respectively, where  $t_x$  and  $t_y$  are the number of stationary points in the respective substrings. Equation (6) estimates the slope of the linear trend, of stationary points in the substring of sequence  $X$ , as the mean of the first difference of stationary points, which is same as  $\frac{x_{q_{t_x}} - x_{q_1}}{t_x - 1}$  [8]. Its standard deviation is estimated using Equation (7). Similarly,  $\mu_{S_{XY}}^Y$  and  $\sigma_{S_{XY}}^Y$  are also estimated for substring of sequence  $Y$ .

$$\mu_{S_{XY}}^X = \frac{1}{t_x - 1} \sum_{k=1}^{t_x-1} (x_{q_{k+1}} - x_{q_k}) \quad (6)$$

$$\sigma_{S_{XY}}^X = \frac{1}{t_x - 1} \sum_{k=1}^{t_x-1} ((x_{q_{k+1}} - x_{q_k}) - \mu_{S_{XY}}^X)^2 \quad (7)$$

Let  $z_1 = \mu_{S_{XY}}^X \sigma_{S_{XY}}^Y$  and  $z_2 = \mu_{S_{XY}}^Y \sigma_{S_{XY}}^X$ . For a true positive, the similarity in the linear trend should be high. Equation (8) calculates this similarity which needs to be maximized. This similarity has negative value when the two slopes are of different sign and thus, the penalization is more.

$$\rho_{S_{XY}} = \begin{cases} \frac{\max(z_1, z_2)}{\min(z_1, z_2)} & \text{if } z_1 < 0; z_2 < 0 \\ \frac{\min(z_1, z_2)}{\max(z_1, z_2)} & \text{otherwise} \end{cases} \quad (8)$$

Finally, Equation (9) combines these three optimization measures to get a score value which is maximized. Then the rcls,  $R_{XY}$ , between the sequences  $X$  and  $Y$  is defined, as an rcs with a maximum score, in Equation (10). The rcls  $R_{XY}$  can be obtained using dynamic programming based approach discussed in [9, 13].

$$Score_{S_{XY}} = \alpha \delta_{S_{XY}} \hat{l}_{S_{XY}}^w + (1 - \alpha) \rho_{S_{XY}} \quad (9)$$

$$R_{XY} = \operatorname{argmax}_{S_{XY}} (Score_{S_{XY}}) \quad (10)$$

<i>Rāga</i> Name	Number of one-liners	Average duration (secs)
Bhairavi	17	16.87
Kamboji	12	13
Kalyani	9	12.76
Shankarabharanam	12	12.55
Varali	9	9.40
Overall	59	12.91

Table 1. Database of one-liners

#### 4. DISCOVERING TYPICAL MOTIFS OF *RĀGAS*

Typical motifs of a *rāga* are discovered using one-liners of songs in that *rāga*. For each voiced part in a oneliner of a *rāga*, rcls is found with the overlapping windows in voiced parts of other one-liners of that *rāga*. Only those rcls are selected whose score values and lengths (in seconds) are greater than thresholds  $\tau_{scr}$  and  $\tau_{len}$  respectively. The voiced parts which generated no rcls are interpreted to have no motifs. The rcls generated for a voiced part are grouped and this group is interpreted as a motif found in that voiced part. This results in a number of groups (motifs) for a *rāga*. Further, filtering is performed to isolate typical motifs of that *rāga*.

##### 4.1 Filtering to get typical motifs of a *rāga*

The generated motifs are filtered to get typical motifs of a *rāga* using compositions of various *rāgas*. The most representative candidate of a motif, a candidate with highest score value, is selected to represent that motif or group. The instances of a motif are spotted in the compositions of various *rāgas* as explained in [9, 10]. Each motif is considered as a query to be searched for in a composition. The rcls is found between the query and overlapped windows in a composition. From the many generated rcls from many compositions of a *rāga*, top  $\tau_n$  rcls with highest score values are selected. The average of these score values defines the presence of this motif in that *rāga*. A motif of a *rāga* is isolated as its typical motif if the presence of this motif is more in the given *rāga* than in other *rāgas*. The value of  $\tau_n$  is selected empirically.

## 5. DATASET

The one-liners are selected from five *rāgas* as shown in Table 1. The lines are sung by a musician in isolation. This is done to ensure that the pitch estimation does not get affected due to the accompanying instruments. The average duration of the one-liners is 12.91 seconds. As mentioned earlier, these one-liners come from the various sections of the composition, primarily from the pallavi.

The compositions used for filtering also comes from the same five *rāgas* as shown in Table 2. These compositions are taken from the Charsur collection [1]. These are segments from live concerts with clean recording.



<i>Rāga</i> Name	Number of compositions	Average duration (secs)
Bhairavi	20	1133
Kamboji	10	1310.3
Kalyani	16	1204.3
Shankarabharanam	10	1300.6
Varali	18	1022
Overall	74	1194

Table 2. Database of compositions

## 6. EXPERIMENTS AND RESULTS

The pitch of the music segment is used as a basic feature in this work. This pitch is estimated from Justin Solomon’s algorithm [17] which is efficiently implemented in the *essentia* open-source C++ library [2]. This pitch is further normalized using tonic and then smoothed by interpolating the stationary points of the pitch contour using cubic spline interpolation.

The similarity,  $sim(x_{i_k}, y_{j_k})$ , between two symbols  $x_{i_k}$  and  $y_{j_k}$  is defined in the Equation (11), where  $s_t$  is the number of cent values that represent one semitone. For this work, the value of  $s_t$  is 10. The penalty is low when the two symbols are within one semitone while the penalty is significant for larger deviations. This is performed to ensure that although significant variations are possible in Carnatic music, variations larger than a semitone might result in a different *rāga*.

$$sim(x_{i_k}, y_{j_k}) = \begin{cases} 1 - \frac{|x_{i_k} - y_{j_k}|^3}{(3s_t)^3} & \text{if } |x_{i_k} - y_{j_k}| < 3s_t \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

The similarity threshold,  $\tau_{sim}$ , is empirically set to 0.45 which accepts similarities when two symbols are less than 2.5 semitones (approx.) apart, although penalty is high after a semitone. The threshold on the score of rlcs,  $\tau_{scr}$ , is empirically set to 0.6 to accept rlcs with higher score values. The threshold on the length of the rlcs,  $\tau_{len}$ , is set to 2 seconds to get longer motifs. The value of  $\beta$  is set to 0.5 to give equal importance to the individual densities in both the sequences and  $\alpha$  value is set to 0.6 which gives more importance to density of match and normalized weighted length as compared to linear trend in stationary points.  $\tau_n$  is empirically set to 3.

The similar patterns found across one-liners of a *rāga* are summarized in Table 3. Some of these similar patterns are not typical of the *rāga*. These are therefore filtered out by checking for their presence in various compositions. The summary of the resulting typical motifs is given in Table 4. The average length of all the typical motifs is sufficiently longer than what were used in [10]. The shorter motifs used in [10] also resulted in great deal of false alarms. The importance of longer motifs was discussed in [9] where the longer motifs were inspired from the *rāga* test conducted by Rama Verma [3]. Rama Verma

<i>Rāga</i> Name	Number of discovered patterns	Average duration (secs)
Bhairavi	10	3.52
Kamboji	5	3.40
Kalyani	6	4.48
Shankarabharanam	6	3.42
Varali	3	3.84
Overall	30	3.73

Table 3. Summary of discovered similar patterns across one-liners

<i>Rāga</i> Name	Number of typical motifs	Average duration (secs)
Bhairavi	5	4.52
Kamboji	0	NA
Kalyani	0	NA
Shankarabharanam	5	3.64
Varali	2	4.79
Overall	12	4.32

Table 4. Summary of typical motifs isolated after filtering

used motifs of approximately 3 seconds duration. The typical motifs discovered in our work are also of similar duration. All the patterns of Kamboji and Kalyani are filtered out resulting in no typical motifs for these *rāgas*. We have earlier discussed that the compositions in Carnatic music are composed in a way that the *rāga* information is present at the very beginning. Therefore, without a doubt we are sure that the typical motifs are present in the one-liners we have used for Kalyani and Kamboji. But, it is possible that these typical motifs are not repeating sufficient number of times across one-liners (two times in our approach) or their lengths are shorter than the threshold we have used. These could be the reasons we are not able to pick them up. All the typical patterns are verified by a musician. According to his judgment, all the filtered patterns were indeed typical motifs of the corresponding *rāgas*. Although, he noted that one typical motif in Varali is a smaller portion of the other discovered typical motif of Varali. This repetition of smaller portion is observed in Shankarabharanam as well.

## 7. CONCLUSION AND FUTURE WORK

This paper presents an approach to discover typical motifs of a *rāga* from the one-liners of the compositions in that *rāga*. The importance of one-liners is discussed in detail. A new measure is introduced, to reduce the false alarms, in the optimization criteria for finding rough longest common subsequence between two given sequences. Using the RLCS algorithm, similar patterns across one-liners of a *rāga* are found. Further, the typical motifs are isolated by a filtering technique, introduced in this paper, which uses compositions of various *rāgas*. These typical motifs

are validated by a musician. All the generated typical motifs are found to be significantly typical of their respective *rāgas*.

In this work, only one musician's viewpoint is considered on validating the characteristic nature of the discovered typical motifs. In future, we would like to conduct a MOS test, asking other experts and active listeners to determine the *rāga* from the typical motifs. We would also like to perform *rāga* classification of the compositions and alapanas using the typical motifs. In future, we would also like to do a thorough comparison of our approach with other methods. In this paper, we have only addressed one prevalent type of false alarms. Other types of false alarms also need to be identified and addressed. It should be considered that approaches taken to reduce the false alarms do not affect the true positives significantly. Further, these experiments need to be repeated for a much larger number of one-liners from many *rāgas* such that the typical motifs repeat significantly across one-liners and thus get captured. It will also be interesting to automatically detect and extract the one-liners from the available compositions. This will enable the presented approach to scale to a large number of *rāgas*.

## 8. ACKNOWLEDGMENTS

This research was partly funded by the European Research Council under the European Unions Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583). We are grateful to Vignesh Ishwar for recording the one-liners. We would also like to thank Sridharan Sankaran, Nauman Dawalatabad and Manish Jain for their invaluable and unconditional help in completing this paper.

## 9. REFERENCES

- [1] Charsur. <http://charsur.com/in/>. Accessed: 2014-07-18.
- [2] Essentia open-source c++ library. <http://essentia.upf.edu>. Accessed: 2014-07-18.
- [3] Rama verma, raga test. <http://www.youtube.com/watch?v=3nRtz9EBfeY>. Accessed: 2014-07-18.
- [4] Tom Collins, Andreas Arzt, Sebastian Flossmann, and Gerhard Widmer. Siarct-cfp: Improving precision and the discovery of inexact musical patterns in point-set representations. In Alceu de Souza Britto Jr., Fabien Gouyon, and Simon Dixon, editors, *ISMIR*, pages 549–554, 2013.
- [5] Tom Collins, Jeremy Thurlow, Robin Laney, Alistair Willis, and Paul H. Garthwaite. A comparative evaluation of algorithms for discovering translational patterns in baroque keyboard works. In J. Stephen Downie and Remco C. Veltkamp, editors, *ISMIR*, pages 3–8. International Society for Music Information Retrieval, 2010.
- [6] Darrell Conklin. Discovery of distinctive patterns in music. In *Intelligent Data Analysis*, pages 547–554, 2010.
- [7] Darrell Conklin. Distinctive patterns in the first movement of brahms string quartet in c minor. *Journal of Mathematics and Music*, 4(2):85–92, 2010.
- [8] Jonathan D. Cryer and Kung-Sik Chan. *Time Series Analysis: with Applications in R*. Springer, 2008.
- [9] Shrey Dutta and Hema A Murthy. A modified rough longest common subsequence algorithm for motif spotting in an alapana of carnatic music. In *Communications (NCC), 2014 Twentieth National Conference on*, pages 1–6, Feb 2014.
- [10] Vignesh Ishwar, Shrey Dutta, Ashwin Bellur, and Hema A. Murthy. Motif spotting in an alapana in carnatic music. In Alceu de Souza Britto Jr., Fabien Gouyon, and Simon Dixon, editors, *ISMIR*, pages 499–504, 2013.
- [11] Berit Janssen, W. Bas de Haas, Anja Volk, and Peter van Kranenburg. Discovering repeated patterns in music: state of knowledge, challenges, perspectives. *International Symposium on Computer Music Modeling and Retrieval (CMMR)*, pages 225–240, 2013.
- [12] T. M. Krishna. *A Southern Music: The Karnatic Story*, chapter 5. HarperCollins, India, 2013.
- [13] Hwei-Jen Lin, Hung-Hsuan Wu, and Chun-Wei Wang. Music matching based on rough longest common subsequence. *Journal of Information Science and Engineering*, pages 27, 95–110., 2011.
- [14] Lie Lu, Muyuan Wang, and Hong-Jiang Zhang. Repeating pattern discovery and structure analysis from acoustic music data. In *Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*, MIR '04, pages 275–282, New York, NY, USA, 2004. ACM.
- [15] David Meredith, Kjell Lemstrom, and Geraint A. Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, pages 321–345, 2002.
- [16] Joe Cheri Ross, Vinutha T. P., and Preeti Rao. Detecting melodic motifs from audio for hindustani classical music. In Fabien Gouyon, Perfecto Herrera, Luis Gustavo Martins, and Meinard Mller, editors, *ISMIR*, pages 193–198. FEUP Edies, 2012.
- [17] J. Salamon and E. Gomez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech and Language Processing*, pages 20(6):1759–1770, Aug. 2012.



## Oral Session 5

# Structure

This Page Intentionally Left Blank

## ANALYZING SONG STRUCTURE WITH SPECTRAL CLUSTERING

Brian McFee

Center for Jazz Studies  
Columbia University

brm2132@columbia.edu

Daniel P.W. Ellis

LabROSA

Columbia University

dpwe@ee.columbia.edu

## ABSTRACT

Many approaches to analyzing the structure of a musical recording involve detecting sequential patterns within a self-similarity matrix derived from time-series features. Such patterns ideally capture repeated sequences, which then form the building blocks of large-scale structure.

In this work, techniques from spectral graph theory are applied to analyze repeated patterns in musical recordings. The proposed method produces a low-dimensional encoding of repetition structure, and exposes the hierarchical relationships among structural components at differing levels of granularity. Finally, we demonstrate how to apply the proposed method to the task of music segmentation.

## 1. INTRODUCTION

Detecting repeated forms in audio is fundamental to the analysis of structure in many forms of music. While small-scale repetitions — such as instances of an individual chord — are simple to detect, accurately combining multiple small-scale repetitions into larger structures is a challenging algorithmic task. Much of the current research on this topic begins by calculating local, frame-wise similarities over acoustic features (usually harmonic), and then searching for patterns in the all-pairs self-similarity matrix [3].

In the majority of existing work on structural segmentation, the analysis is *flat*, in the sense that the representation does not explicitly encode nesting or hierarchical structure in the repeated forms. Instead, novelty curves are commonly used to detect transitions between sections.

## 1.1 Our contributions

In this paper, we formulate the structure analysis problem in the context of spectral graph theory. By combining local consistency cues with long-term repetition encodings and analyzing the eigenvectors of the resulting graph Laplacian, we produce a compact representation that effectively encodes repetition structure at multiple levels of granularity. To effectively link repeating sequences, we formulate an optimally weighted combination of local timbre consistency and long-term repetition descriptors.

To motivate the analysis technique, we demonstrate its use for the standard task of flat structural annotation. However, we emphasize that the approach itself can be applied more generally to analyze structure at multiple resolutions.

## 1.2 Related work

The structural repetition features used in this work are inspired by those of Serrà *et al.* [11], wherein structure is detected by applying filtering operators to a lag-skewed self-similarity matrix. The primary deviation in this work is the graphical interpretation and subsequent analysis of the filtered self-similarity matrix.

Recently, Kaiser *et al.* demonstrated a method to combine tonal and timbral features for structural boundary detection [6]. Whereas their method forms a novelty curve from the combination of multiple features, our feature combination differs by using local timbre consistency to build internal connections among sequences of long-range tonal repetitions.

Our general approach is similar in spirit to that of Groganz *et al.* [4], in which diagonal bands of a self-similarity matrix are expanded into block structures by spectral analysis. Their method analyzed the spectral decomposition of the self-similarity matrix directly, whereas the method proposed here operates on the graph Laplacian. Similarly, Kaiser and Sikora applied non-negative matrix factorization directly to a self-similarity matrix in order to detect blocks of repeating elements [7]. As we will demonstrate, the Laplacian provides a more direct means to expose block structure at multiple levels of detail.

## 2. GRAPHICAL REPETITION ENCODING

Our general structural analysis strategy is to construct and partition a graph over time points (samples) in the song. Let  $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times n}$  denote a  $d$ -dimensional time series feature matrix, *e.g.*, a chromagram or sequence of Mel-frequency cepstral coefficients. As a first step toward detecting and representing repetition structure, we form a *binary recurrence matrix*  $R \in \{0, 1\}^{n \times n}$ , where

$$R_{ij}(X) := \begin{cases} 1 & x_i, x_j \text{ are mutual } k\text{-nearest neighbors} \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

and  $k > 0$  parameterizes the degree of connectivity.

Ideally, repeated structures should appear as diagonal stripes in  $R$ . In practice, it is beneficial to apply a smooth-



© Brian McFee, Daniel P.W. Ellis.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Brian McFee, Daniel P.W. Ellis. “Analyzing song structure with spectral clustering”, 15th International Society for Music Information Retrieval Conference, 2014.

ing filter to suppress erroneous links and fill in gaps. We apply a windowed majority vote to each diagonal of  $R$ , resulting in the filtered matrix  $R'$ :

$$R'_{ij} := \text{maj} \{R_{i+t, j+t} \mid t \in -w, -w+1, \dots, w\}, \quad (2)$$

where  $w$  is a discrete parameter that defines the minimum length of a valid repetition sequence.

## 2.1 Internal connectivity

The filtered recurrence matrix  $R'$  can be interpreted as an unweighted, undirected graph, whose vertices correspond to samples (columns of  $X$ ), and edges correspond to equivalent position within a repeated sequence. Note, however, that successive positions  $(i, i+1)$  will not generally be connected in  $R'$ , so the constituent samples of a particular sequence may not be connected.

To facilitate discovery of repeated sections, edges between adjacent samples  $(i, i+1)$  and  $(i, i-1)$  are introduced, resulting in the *sequence-augmented graph*  $R^+$ :

$$\Delta_{ij} := \begin{cases} 1 & |i-j| = 1 \\ 0 & \text{otherwise} \end{cases}, \quad (3)$$

$$R^+_{ij} := \max(\Delta_{ij}, R'_{ij}). \quad (4)$$

With appropriate normalization,  $R^+$  characterizes a Markov process over samples, where at each step  $i$ , the process either moves to an adjacent sample  $i \pm 1$ , or a random repetition of  $i$ ; a process exemplified by the Infinite Jukebox [8].

Equation (4) combines local temporal connectivity with long-term recurrence information. Ideally, edges would exist only between pairs  $\{i, j\}$  belonging to the same structural component, but of course, this information is hidden. The added edges along the first diagonals create a fully connected graph, but due to recurrence links, repeated sections will exhibit additional internal connectivity. Let  $i$  and  $j$  denote two repetitions of the same sample at different times; then  $R^+$  should contain sequential edges  $\{i, i+1\}$ ,  $\{j, j+1\}$  and repetition edges  $\{i, j\}$ ,  $\{i+1, j+1\}$ . On the other hand, unrelated sections with no repetition edges can only connect via sequence edges.

## 2.2 Balancing local and global linkage

The construction of eq. (4) describes the intuition behind combining local sequential connections with global repetition structure, but it does not balance the two competing goals. Long tracks with many repetitions can produce recurrence links which vastly outnumber local connectivity connections. In this regime, partitioning into contiguous sections becomes difficult, and subsequent analysis of the graph may fail to detect sequential structure.

If we allow (non-negative) weights on the edges, then the combination can be parameterized by a weighting parameter  $\mu \in [0, 1]$ :

$$R^{\mu}_{ij} := \mu R'_{ij} + (1 - \mu) \Delta_{ij}. \quad (5)$$

This raises the question: how should  $\mu$  be set? Returning to the motivating example of the random walk, we opt

for a process that on average, tends to move either in sequence or across (all) repetitions with equal probability. In terms of  $\mu$ , this indicates that the combination should assign equal weight to the local and repetition edges. This suggests a balancing objective for all frames  $i$ :

$$\mu \sum_j R'_{ij} \approx (1 - \mu) \sum_j \Delta_{ij}.$$

Minimizing the average squared error between the two terms above leads to the following quadratic optimization:

$$\min_{\mu \in [0, 1]} \frac{1}{2} \sum_i (\mu d_i(R') - (1 - \mu) d_i(\Delta))^2, \quad (6)$$

where  $d_i(G) := \sum_j G_{ij}$  denotes the degree (sum of incident edge-weights) of  $i$  in  $G$ . Treating  $d(\cdot) := [d_i(\cdot)]_{i=1}^n$  as a vector in  $\mathbb{R}_+^n$  yields the optimal solution to eq. (6):

$$\mu^* = \frac{\langle d(\Delta), d(R') + d(\Delta) \rangle}{\|d(R') + d(\Delta)\|^2}. \quad (7)$$

Note that because  $\Delta$  is non-empty (contains at least one edge), it follows that  $\|d(\Delta)\|^2 > 0$ , which implies  $\mu^* > 0$ . Similarly, if  $R'$  is non-empty, then  $\mu^* < 1$ , and the resulting combination retains the full connectivity structure of the unweighted  $R^+$  (eq. (4)).

## 2.3 Edge weighting and feature fusion

The construction above relies upon a single feature representation to determine the self-similarity structure, and uses constant edge weights for the repetition and local edges. This can be generalized to support feature-weighted edges by replacing  $R'$  with a masked similarity matrix:

$$R'_{ij} \mapsto R'_{ij} S_{ij}, \quad (8)$$

where  $S_{ij}$  denotes a non-negative affinity between frames  $i$  and  $j$ , e.g., a Gaussian kernel over feature vectors  $x_i, x_j$ :

$$S_{ij}^{\text{rep}} := \exp\left(-\frac{1}{2\sigma^2} \|x_i - x_j\|^2\right)$$

Similarly,  $\Delta$  can be replaced with a weighted sequence graph. However, in doing so, care must be taken when selecting the affinity function. The same features used to detect repetition (typically harmonic in nature) may not capture local consistency, since successive frames do not generally retain harmonic similarity.

Recent work has demonstrated that local timbre differences can provide an effective cue for structural boundary detection [6]. This motivates the use of two contrasting feature descriptors: harmonic features for detecting long-range repeating forms, and timbral features for detecting local consistency. We assume that these features are respectively supplied in the form of affinity matrices  $S^{\text{rep}}$  and  $S^{\text{loc}}$ . Combining these affinities with the detected repetition structure and optimal weighting yields the *sequence-augmented affinity matrix*  $A$ :

$$A_{ij} := \mu R'_{ij} S_{ij}^{\text{rep}} + (1 - \mu) \Delta_{ij} S_{ij}^{\text{loc}}, \quad (9)$$

where  $R'$  is understood to be constructed solely from the repetition affinities  $S^{\text{rep}}$ , and  $\mu$  is optimized by solving (7) with the weighted affinity matrices.

### 3. GRAPH PARTITIONING AND STRUCTURAL ANALYSIS

The Laplacian is a fundamental tool in the field of spectral graph theory, as it can be interpreted as a discrete analog of a diffusion operator over the vertices of the graph, and its spectrum can be used to characterize vertex connectivity [2]. This section describes in detail how spectral clustering can be used to analyze and partition the repetition graph constructed in the previous section, and reveal musical structure.

#### 3.1 Background: spectral clustering

Let  $D$  denote the diagonal *degree matrix* of  $A$ :

$$D := \text{diag}(d(A)).$$

The *symmetric normalized Laplacian*  $L$  is defined as:

$$L := I - D^{-1/2}AD^{-1/2}. \quad (10)$$

The Laplacian forms the basis of spectral clustering, in which vertices are represented in terms of the eigenvectors of  $L$  [15]. More specifically, to partition a graph into  $m$  components, each point  $i$  is represented as the vector of the  $i$ th coordinates of the first  $m$  eigenvectors of  $L$ , corresponding to the  $m$  smallest eigenvalues.<sup>1</sup> The motivation for this method stems from the observation that the multiplicity of the bottom eigenvalue  $\lambda_0 = 0$  corresponds to the number of connected components in a graph, and the corresponding eigenvectors encode component memberships amongst vertices.

In the non-ideal case, the graph is fully connected, so  $\lambda_0$  has multiplicity 1, and the bottom eigenvector trivially encodes membership in the graph. However, in the case of  $A$ , we expect there to be many components with high intra-connectivity and relatively small inter-connectivity at the transition points between sections. Spectral clustering can be viewed as an approximation method for finding normalized graph cuts [15], and it is well-suited to detecting and pruning these weak links.

Figure 1 illustrates an example of the encoding produced by spectral decomposition of  $L$ . Although the first eigenvector (column) is uninformative, the remaining bases clearly encode membership in the diagonal regions depicted in the affinity matrix. The resulting pair-wise frame similarities for this example are shown in Figure 2, which clearly demonstrates the ability of this representation to iteratively reveal nested repeating structure.

To apply spectral clustering, we will use  $k$ -means clustering with the (normalized) eigenvectors  $Y \in \mathbb{R}^{n \times M}$  as features, where  $M > 0$  is a specified maximum number of structural component types. Varying  $M$  — equivalently, the dimension of the representation — directly controls the granularity of the resulting segmentation.

<sup>1</sup> An additional length-normalization is applied to each vector, to correct for scaling introduced by the symmetric normalized Laplacian [15].

---

#### Algorithm 1 Boundary detection

---

**Input:** Laplacian eigenvectors  $Y \in \mathbb{R}^{n \times m}$ ,

**Output:** Boundaries  $b$ , segment labels  $c \in [m]^n$

- 1: **function** BOUNDARY-DETECT( $Y$ )
  - 2:  $\hat{y}_i \leftarrow Y_{i,\cdot} / \|Y_{i,\cdot}\|$  // Normalize each row  $Y_{i,\cdot}$ .
  - 3: Run  $k$ -means on  $\{\hat{y}_i\}_{i=1}^n$  with  $k = m$
  - 4: Let  $c_i$  denote the cluster containing  $\hat{y}_i$
  - 5:  $b \leftarrow \{i \mid c_i \neq c_{i+1}\}$
  - 6: **return** ( $b, c$ )
- 

#### 3.2 Boundary detection

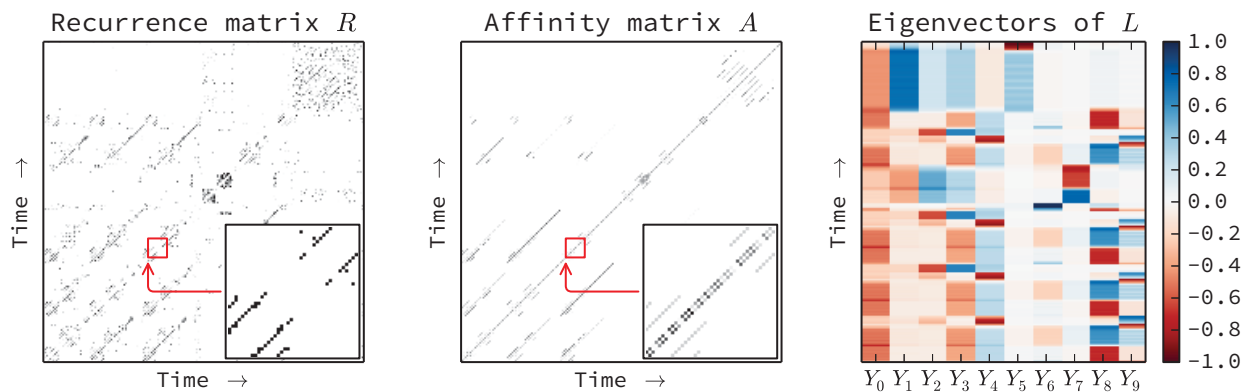
For a fixed number of segment types  $m$ , segment boundaries can be estimated by clustering the rows of  $Y$  after truncating to the first  $m$  dimensions. After clustering, segment boundaries are detected by searching for change-points in the cluster assignments. This method is formalized in Algorithm 1. Note that the number of segment *types* is distinct from the number of *segments* because a single type (e.g., verse) may repeat multiple times throughout the track.

#### 3.3 Laplacian structural decomposition

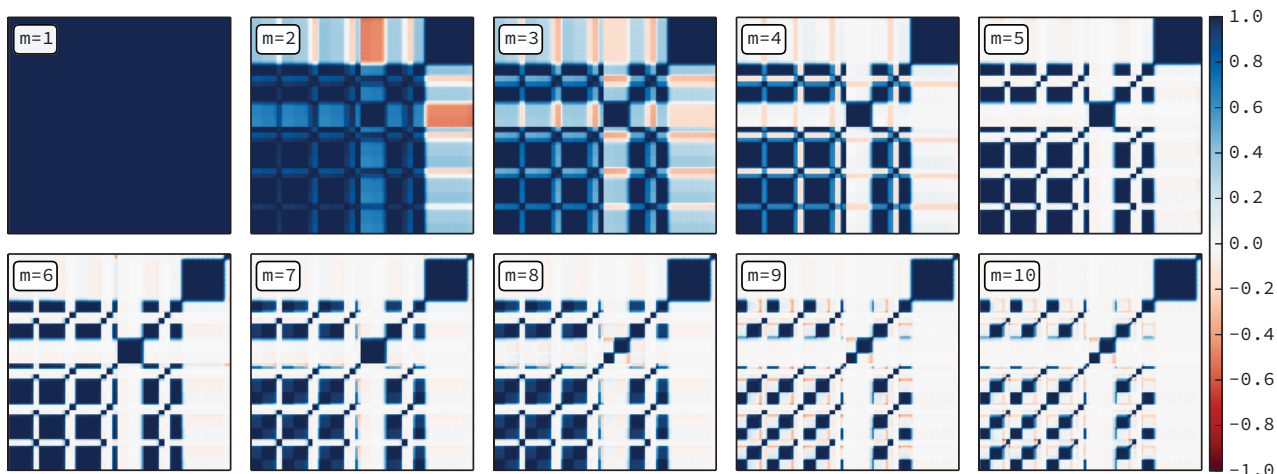
To decompose an input song into its structural components, we propose a method, listed as Algorithm 2, to find boundaries and structural annotations at multiple levels of structural complexity. Algorithm 2 first computes the Laplacian as described above, and then iteratively increases the set of eigenvectors for use in Algorithm 1. For  $m = 2$ , the first two eigenvectors — corresponding to the two smallest eigenvalues of  $L$  — are taken. In general, for  $m$  types of repeating component, the bottom  $m$  eigenvectors are used to label frames and detect boundaries. The result is a sequence of boundaries  $B^m$  and frame labels  $C^m$ , for values  $m \in 2, 3, \dots, M$ .

Note that unlike most structural analysis algorithms, Algorithm 2 does not produce a single decomposition of the song, but rather a sequence of decompositions ordered by increasing complexity. This property can be beneficial in visualization applications, where a user may be interested in the relationship between structural components at multiple levels. Similarly, in interactive display applications, a user may request more or less detailed analyses for a track. Since complexity is controlled by a single, discrete parameter  $M$ , this application is readily supported with a minimal set of interface controls (e.g., a slider).

However, for standardized evaluation, the method must produce a single, flat segmentation. Adaptively estimating the appropriate level of analysis in this context is somewhat ill-posed, as different use-cases require differing levels of detail. We apply a simple selection criterion based on the level of detail commonly observed in standard datasets [5, 12]. First, the set of candidates is reduced to those in which the mean segment duration is at least 10 seconds. Subject to this constraint, the segmentation level  $\tilde{m}$  is selected to maximize frame-level annotation entropy. This strategy tends to produce solutions with approximately balanced distributions over the set of segment types.



**Figure 1.** Left: the recurrence matrix  $R$  for *The Beatles — Come Together*. Center: the sequence-augmented affinity matrix  $A$ ; the enlarged region demonstrates the cumulative effects of recurrence filtering, sequence-augmentation, and edge weighting. Right: the first 10 basis features (columns), ordered left-to-right. The leading columns encode the primary structural components, while subsequent components encode refinements.



**Figure 2.** Pair-wise frame similarities ( $YY^T$ ) using the first 10 components for *The Beatles — Come Together*. The first (trivial) component ( $m = 1$ ) captures the entire song, and the second ( $m = 2$ ) separates the outro (final vamp) from the rest of the song. Subsequent refinements separate the solo, refrain, verse, and outro, and then individual measures.

## 4. EXPERIMENTS

To evaluate the proposed method quantitatively, we compare boundary detection and structural annotation performance on two standard datasets. We evaluate the performance of the method using the automatic complexity estimation described above, as well as performance achieved for each fixed value of  $m$  across the dataset.

Finally, to evaluate the impact of the complexity estimation method, we compare to an oracle model. For each track, a different  $m^*$  is selected to maximize the evaluation metric of interest. This can be viewed as a simulation of interactive visualization, in which the user has the freedom to dynamically adapt the level of detail until she is satisfied. Results in this setting may be interpreted as measuring the best possible decomposition within the set produced by Algorithm 2.

### 4.1 Data and evaluation

Our evaluation data is comprised of two sources:

**Beatles-TUT:** 174 structurally annotated tracks from the Beatles corpus [10]. A single annotation is provided for each track, and annotations generally correspond to functional components (e.g., *verse*, *refrain*, or *solo*).

**SALAMI:** 735 tracks from the SALAMI corpus [12]. This corpus spans a wide range of genres and instrumentation, and provides multiple annotation levels for each track. We report results on *functional* and *small-scale* annotations.

In each evaluation, we report the  $F$ -measure of boundary detection at 0.5-second and 3-second windows. To evaluate structural annotation accuracy, we report pairwise frame classification  $F$ -measure [9]. For comparison purposes, we report scores achieved by the method of Serrà *et*



**Algorithm 2** Laplacian structural decomposition

**Input:** Affinities:  $S^{\text{rep}}, S^{\text{loc}} \in \mathbb{R}_+^{n \times n}$ , maximum number of segment types  $0 < M \leq n$

**Output:** Boundaries  $B^m$  and frame labels  $C^m$  for  $m \in 2 \dots M$

```

1: function LSD( $S^{\text{rep}}, S^{\text{loc}}, M$ )
2:    $R \leftarrow$  eq. (1) on  $S^{\text{rep}}$  // Recurrence detection
3:    $R' \leftarrow$  eq. (2) on  $R$  // Recurrence filtering
4:    $A \leftarrow$  eq. (9) // Sequence augmentation
5:    $L \leftarrow I - D^{-1/2} A D^{-1/2}$ 
6:   for  $m \in 2, 3, \dots, M$  do
7:      $Y \leftarrow$  bottom  $m$  eigenvectors of  $L$ 
8:      $(B^m, C^m) \leftarrow$  BOUNDARY-DETECT( $Y$ )
9:   return  $\{(B^m, C^m)\}_{m=2}^M$ 

```

*al.*, denoted here as SMGA [11].

**4.2 Implementation details**

All input signals are sampled at 22050Hz (mono), and analyzed with a 2048-sample FFT window and 512-sample hop. Repetition similarity matrices  $S^{\text{rep}}$  were computed by first extracting log-power constant-Q spectrograms over 72 bins, ranging from  $C2$  (32.7 Hz) to  $C8$  (2093.0 Hz).

Constant-Q frames were mean-aggregated between detected beat events, and stacked using time-delay embedding with one step of history as in [11]. Similarity matrices were then computed by applying a Gaussian kernel to each pair of beat-synchronous frames  $i$  and  $j$ . The bandwidth parameter  $\sigma^2$  was estimated by computing the average squared distance between each  $x_i$  and its  $k$ th nearest neighbor, with  $k$  set to  $1 + \lceil 2 \log_2 n \rceil$  (where  $n$  denotes the number of detected beats). The same  $k$  was used to connect nearest neighbors when building the recurrence matrix  $R$ , with the additional constraint that frames cannot link to neighbors within 3 beats of each-other, which prevents self-similar connections within the same measure. The majority vote window was set to  $w = 17$ .

Local timbre similarity  $S^{\text{loc}}$  was computed by extracting the first 13 Mel frequency cepstral coefficients (MFCC), mean-aggregating between detected beats, and then applying a Gaussian kernel as done for  $S^{\text{rep}}$ .

All methods were implemented in Python with NumPy and librosa [1, 14].

**4.3 Results**

The results of the evaluation are listed in Tables 1 to 3. For each fixed  $m$ , the scores are indicated as  $L_m$ .  $L$  indicates the automatic maximum-entropy selector, and  $L^*$  indicates the best possible  $m$  for each metric independently.

As a common trend across all data sets, the automatic  $m$ -selector often achieves results comparable to the best fixed  $m$ . However, it is consistently outperformed by the oracle model  $L^*$ , indicating that the output of Algorithm 2 often contains accurate solutions, the automatic selector does not always choose them.

**Table 1.** Beatles (TUT)

Method	$F_{0.5}$	$F_3$	$F_{\text{pair}}$
$L_2$	0.307 ± 0.14	0.429 ± 0.18	0.576 ± 0.14
$L_3$	0.303 ± 0.15	0.544 ± 0.17	0.611 ± 0.13
$L_4$	0.307 ± 0.15	0.568 ± 0.17	0.616 ± 0.13
$L_5$	0.276 ± 0.14	0.553 ± 0.15	0.587 ± 0.12
$L_6$	0.259 ± 0.14	0.530 ± 0.15	0.556 ± 0.12
$L_7$	0.246 ± 0.13	0.507 ± 0.14	0.523 ± 0.12
$L_8$	0.229 ± 0.13	0.477 ± 0.15	0.495 ± 0.12
$L_9$	0.222 ± 0.12	0.446 ± 0.14	0.468 ± 0.12
$L_{10}$	0.214 ± 0.11	0.425 ± 0.13	0.443 ± 0.12
$L$	0.312 ± 0.15	0.579 ± 0.16	0.628 ± 0.13
$L^*$	0.414 ± 0.14	0.684 ± 0.13	0.694 ± 0.12
SMGA	0.293 ± 0.13	0.699 ± 0.16	0.715 ± 0.15

**Table 2.** SALAMI (Functions)

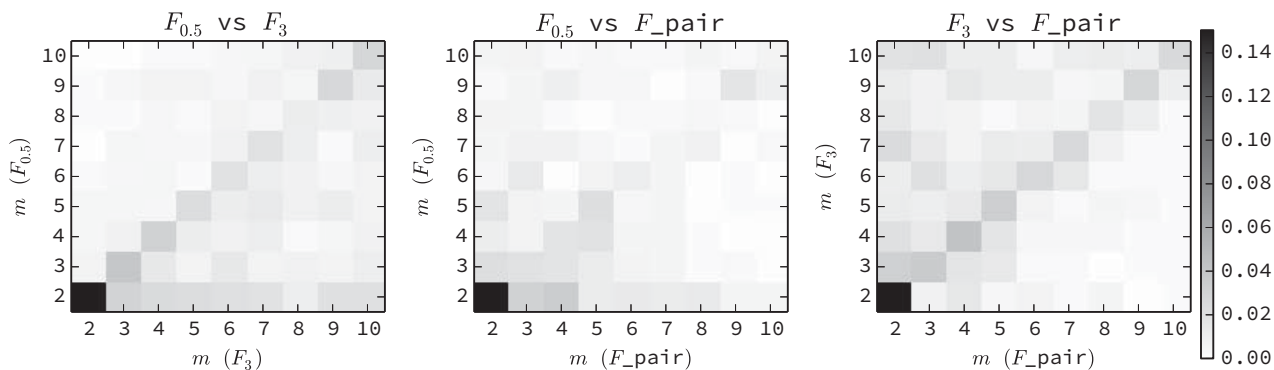
Method	$F_{0.5}$	$F_3$	$F_{\text{pair}}$
$L_2$	0.324 ± 0.13	0.383 ± 0.15	0.539 ± 0.16
$L_3$	0.314 ± 0.13	0.417 ± 0.16	0.549 ± 0.13
$L_4$	0.303 ± 0.12	0.439 ± 0.16	0.547 ± 0.13
$L_5$	0.293 ± 0.12	0.444 ± 0.16	0.535 ± 0.12
$L_6$	0.286 ± 0.12	0.452 ± 0.16	0.521 ± 0.13
$L_7$	0.273 ± 0.11	0.442 ± 0.16	0.502 ± 0.13
$L_8$	0.267 ± 0.12	0.437 ± 0.16	0.483 ± 0.13
$L_9$	0.260 ± 0.11	0.443 ± 0.16	0.464 ± 0.14
$L_{10}$	0.250 ± 0.11	0.422 ± 0.16	0.445 ± 0.14
$L$	0.304 ± 0.13	0.455 ± 0.16	0.546 ± 0.14
$L^*$	0.406 ± 0.13	0.579 ± 0.15	0.652 ± 0.13
SMGA	0.224 ± 0.11	0.550 ± 0.18	0.553 ± 0.15

In the case of SALAMI (small), the automatic selector performs dramatically worse than many of the fixed- $m$  methods, which may be explained by the relatively different statistics of segment durations and numbers of unique segment types in the small-scale annotations as compared to Beatles and SALAMI (functional).

To investigate whether a single  $m$  could simultaneously optimize multiple evaluation metrics for a given track, we plot the confusion matrices for the oracle selections on SALAMI (functional) in Figure 3. We observe that the  $m$  which optimizes  $F_3$  is frequently larger than those for  $F_{0.5}$  — as indicated by the mass in the lower triangle of the left plot — or  $F_{\text{pair}}$  — as indicated by the upper triangle of the right plot. Although this observation depends upon our particular boundary-detection strategy, it is corroborated by previous observations that the 0.5-second and 3.0-second metrics evaluate qualitatively different objectives [13]. Consequently, it may be beneficial in practice to provide segmentations at multiple resolutions when the specific choice of evaluation criterion is unknown.

**5. CONCLUSIONS**

The experimental results demonstrate that the proposed structural decomposition technique often generates solutions which achieve high scores on segmentation evaluation metrics. However, automatically selecting a single “best” segmentation without a priori knowledge of the evaluation criteria



**Figure 3.** Confusion matrices illustrating the oracle selection of the number of segment types  $m \in [2, 10]$  for different pairs of metrics on SALAMI (functional). While  $m = 2$  is most frequently selected for all metrics, the large mass off-diagonal indicates that for a given track, a single fixed  $m$  does not generally optimize all evaluation metrics.

**Table 3.** SALAMI (Small)

Method	$F_{0.5}$	$F_3$	$F_{\text{pair}}$
$L_2$	$0.151 \pm 0.11$	$0.195 \pm 0.13$	$0.451 \pm 0.19$
$L_3$	$0.171 \pm 0.12$	$0.259 \pm 0.16$	$0.459 \pm 0.17$
$L_4$	$0.186 \pm 0.12$	$0.315 \pm 0.17$	$0.461 \pm 0.15$
$L_5$	$0.195 \pm 0.12$	$0.354 \pm 0.17$	$0.455 \pm 0.14$
$L_6$	$0.207 \pm 0.12$	$0.391 \pm 0.18$	$0.452 \pm 0.13$
$L_7$	$0.214 \pm 0.12$	$0.420 \pm 0.18$	$0.445 \pm 0.13$
$L_8$	$0.224 \pm 0.12$	$0.448 \pm 0.18$	$0.435 \pm 0.13$
$L_9$	$0.229 \pm 0.12$	$0.467 \pm 0.18$	$0.425 \pm 0.13$
$L_{10}$	$0.234 \pm 0.12$	$0.486 \pm 0.18$	$0.414 \pm 0.13$
$L$	$0.192 \pm 0.11$	$0.344 \pm 0.15$	$0.448 \pm 0.16$
$L^*$	$0.292 \pm 0.15$	$0.525 \pm 0.19$	$0.561 \pm 0.16$
SMGA	$0.173 \pm 0.08$	$0.518 \pm 0.12$	$0.493 \pm 0.16$

remains a challenging practical issue.

## 6. ACKNOWLEDGMENTS

The authors acknowledge support from The Andrew W. Mellon Foundation, and NSF grant IIS-1117015.

## 7. REFERENCES

- [1] Librosa, 2014. <https://github.com/bmcfee/librosa>.
- [2] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.
- [3] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 1, pages 452–455. IEEE, 2000.
- [4] Harald Grohganz, Michael Clausen, Nanzhu Jiang, and Meinard Müller. Converting path structures into block structures using eigenvalue decomposition of self-similarity matrices. In *ISMIR*, 2013.
- [5] Christopher Harte. *Towards automatic extraction of harmony information from music signals*. PhD thesis, University of London, 2010.
- [6] Florian Kaiser and Geoffroy Peeters. A simple fusion method of state and sequence segmentation for music structure discovery. In *ISMIR*, 2013.
- [7] Florian Kaiser and Thomas Sikora. Music structure discovery in popular music using non-negative matrix factorization. In *ISMIR*, pages 429–434, 2010.
- [8] P. Lamere. The infinite jukebox, November 2012. <http://infinitejuke.com/>.
- [9] Mark Levy and Mark Sandler. Structural segmentation of musical audio by constrained clustering. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):318–326, 2008.
- [10] Jouni Paulus and Anssi Klapuri. Music structure analysis by finding repeated parts. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 59–68. ACM, 2006.
- [11] J. Serrà, M. Müller, P. Grosche, and J. Arcos. Unsupervised music structure annotation by time series structure features and segment similarity. *Multimedia, IEEE Transactions on*, PP(99):1–1, 2014.
- [12] Jordan BL Smith, John Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J Stephen Downie. Design and creation of a large-scale database of structural annotations. In *ISMIR*, pages 555–560, 2011.
- [13] Jordan BL Smith and Elaine Chew. A meta-analysis of the mirex structure segmentation task. In *ISMIR*, 2013.
- [14] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.
- [15] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

# IDENTIFYING POLYPHONIC PATTERNS FROM AUDIO RECORDINGS USING MUSIC SEGMENTATION TECHNIQUES

**Oriol Nieto and Morwaread M. Farbood**

Music and Audio Research Lab

New York University

{oriol, mfarbood}@nyu.edu

## ABSTRACT

This paper presents a method for discovering patterns of note collections that repeatedly occur in a piece of music. We assume occurrences of these patterns must appear at least twice across a musical work and that they may contain slight differences in harmony, timbre, or rhythm. We describe an algorithm that makes use of techniques from the music information retrieval task of music segmentation, which exploits repetitive features in order to automatically identify polyphonic musical patterns from audio recordings. The novel algorithm is assessed using the recently published JKU Patterns Development Dataset, and we show how it obtains state-of-the-art results employing the standard evaluation metrics.

## 1. INTRODUCTION

The task of discovering repetitive musical patterns (of which motives, themes, and repeated sections are all examples) consists of retrieving the most relevant musical ideas that repeat at least once within a specific piece [1, 8]. Besides the relevant role this task plays in musicological studies, especially with regard to intra-opus analysis, it can also yield a better understanding of how composers write and how listeners interpret the underlying structure of music. Computational approaches to this task can dramatically simplify not only the analysis of a specific piece, but of an entire corpus, potentially offering interesting explorations and relations of patterns across works. Other potential applications include the improved navigation across both large music collections and stand-alone pieces, or the development of computer-aided composition tools.

Typically the task of automatically discovering musical patterns uses symbolic representations of music [3]. Methods that assume a monophonic representation have been proposed, and operate on various musical dimensions such as chromatic/diatonic pitch, rhythm, or contour [4, 9, 10]. Other methods focusing on polyphonic music as input have

also been presented, mostly using geometric representations in Euclidean space, with a different axis assigned to each musical dimension [6, 11]. Similar techniques have also been explored [7, 11, 12] that attempt to arrive at a compressed representation of an input, multidimensional point set. Other methods using cognitively inspired rules with symbolic representations of music have also been proposed [6, 16]. Working with the score of a musical piece instead of its audio representation can indeed reduce the complexity of the problem, however this also significantly narrows the applicability of the algorithm, since it is not necessarily common to have access to symbolic representations of music, particularly when working with genres such as jazz, rock, or Western popular music.

Methods using audio recordings as input have also been explored. A good recent example is [3], where the authors first estimate the fundamental frequency (F0) from the audio in order to obtain the patterns using a symbolic-based approach. Another one uses a probabilistic approach to matrix factorization in order to learn the different parts of a western popular track in an unsupervised manner [20]. Yet another method uses a compression criterion where the most informative (i.e., repeated) parts of a piece are identified in order to automatically produce a musical “summary” [17].

In this paper, we propose a method using audio recordings as input in an attempt to broaden the applicability of pattern discovery algorithms. We make use of tools that are commonly employed in the music information retrieval task of *music segmentation* combined with a novel score-based greedy algorithm in order to identify the most repeated parts of a given audio signal. Finally, we evaluate the results using the JKU Patterns Development Dataset and the metrics proposed in the Music Information Retrieval Evaluation eXchange (MIREX) [1].

The outline of this paper is as follows: In Section 2 we review a set of music segmentation techniques that will be used in our algorithm; in Section 3 we detail our method to extract musical patterns, including the score-based greedy algorithm; in Section 4 we present the evaluation and the results; and in Section 5 we draw various conclusions and identify areas for future work.



© Oriol Nieto, Morwaread M. Farbood.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Oriol Nieto, Morwaread M. Farbood. “Identifying Polyphonic Patterns From Audio Recordings Using Music Segmentation Techniques”, 15th International Society for Music Information Retrieval Conference, 2014.

## 2. MUSIC SEGMENTATION TECHNIQUES

The task of music segmentation is well-established in the music informatics literature (see [18] for a review). Its goal is to automatically identify all the non-overlapping musical segments (or sections) of a given track, such that the concatenation of all of them reconstructs the entire piece. Once these segments are identified, they are labeled based on their similarity (e.g., verse, chorus, coda). Therefore, this task can be divided into two different subproblems: the discovery of the *boundaries* that define all the segments, and the grouping of the segments into different *labels*. In this work we will use tools that focus mainly on the former subproblem.

There is general agreement among researchers that any given boundary is defined by at least one of these three characteristics: repetition, homogeneity, and/or novelty [18]. In our case, we center the discussion on the repetition boundaries, since, as we will see in Section 3, repetition is the defining feature of the musical patterns.

### 2.1 Extracting Repetitive Boundaries

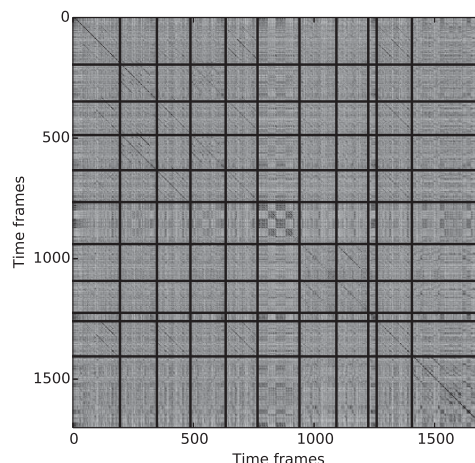
In this subsection we review a standard technique to extract boundaries characterized by repetition (also known as a sequence approach), from an input audio signal  $x$ . For a more detailed explanation, we refer the reader to [13]. The process can be summarized in three different steps:

- i The signal  $x$  is transformed into a series of feature vectors  $C = (\mathbf{c}_1, \dots, \mathbf{c}_N)$  that divide  $x$  into  $N$  frames and capture specific frame-level characteristics of the given signal. In our case, we will only focus on harmonic features, more specifically on chromagrams (or pitch class profiles).
- ii  $C$  is used in order to obtain a self-similarity matrix (SSM)  $S$ , a symmetric matrix such that  $S(n, m) = d(\mathbf{c}_n, \mathbf{c}_m)$ ,  $\forall n, m \in [1 : N]$ , where  $d$  is a distance function (e.g. Euclidean, cosine, Manhattan).
- iii The resulting matrix  $S$  will contain diagonal paths (or semi-diagonal in case of slight tempo variations) or stripes that will indicate the repetition of a specific part of the audio signal  $x$ . These paths can be extracted using greedy algorithms (e.g., as described in [13]). The final boundaries are given by the endpoints of these paths.

An example of an SSM using the Euclidean distance with the identified boundaries is shown in Figure 1. As can be seen, the annotated boundaries are visually associated with the paths of the matrix. The identification of patterns, as opposed to the task of segmentation, allows overlapping patterns and occurrences, so we base our algorithm on greedy methods to extract paths from an SSM.

### 2.2 Transposition-Invariant Self-Similarity Matrix

It is common to analyze pieces that contain key-transposed repetitions. It is therefore important for an algorithm to be invariant to these these transpositions when identifying



**Figure 1.** Self-similarity matrix for Chopin's Op. 24 No. 4, with annotated boundaries as vertical and horizontal lines.

repetitions. One effective method for solving this problem [14] involves a technique that can be described in two steps: (1) compute 12 different SSMs from harmonic representations (e.g. chromagrams), each corresponding to a transposition of the 12 pitches of the Western chromatic scale, and 2) obtain the transposition-invariant SSM by keeping the minimum distance across the 12 matrices for all the  $N \times N$  distances in the output matrix. Formally:

$$S(n, m) = \min_{k \in [0:11]} \{S_k(n, m)\}, \forall n, m \in [1 : N] \quad (1)$$

where  $S$  is the transposition-invariant SSM, and  $S_k$  is the  $k$ -th transposition of the matrix  $S$ .

## 3. IDENTIFYING MUSICAL PATTERNS

The discovery of patterns and their various occurrences involves retrieving actual note collections (which may nest and/or overlap), and so this task can be seen as more complex than structural segmentation, which involves labeling a single, temporal partition of an audio signal. We define a repeating musical pattern to be a short idea that is repeated at least once across the entire piece, even though this repetition may be transposed or contain various time shifts. Therefore, each pattern is associated with a set of occurrences that will not necessarily be exact. The patterns and their occurrences may overlap with each other, and this is perfectly acceptable in the context of pattern discovery. An optimal algorithm for this task would (1) find all the patterns contained in a piece and (2) identify all the occurrences across the piece for each pattern found. In this section we describe our algorithm, which uses audio recordings as input and finds polyphonic patterns as well as a list of all the discovered occurrences for each of the patterns. A block-diagram of the entire process is depicted in Figure 2.

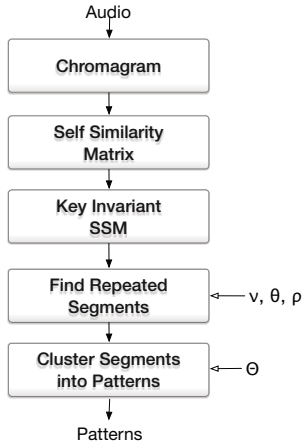


Figure 2. Block diagram of the proposed algorithm.

### 3.1 Rhythmic-Synchronous Harmonic Feature Extraction

Given a one-channel audio signal  $x$  sampled at 11025 Hz representing a piece of music, we compute the spectrogram using a Blackman window of  $N_w = 290$  milliseconds, with a hop size of  $N_w/2$ . We then compute a constant-Q transform from the spectrogram starting at 55 Hz (corresponding to the note A1 in standard tuning) comprising four octaves. Finally, we collapse each of the 12 pitches of the western scale into a single octave to obtain a chromagram, a matrix of  $12 \times N$ , which is commonly used to represent harmonic features [18]. We normalize the chromagram such that the maximum energy for a given time frame is 1. In this harmonic representation we can no longer differentiate between octaves, but its compactness and the energy of each pitch class will become convenient when identifying harmonic repetitions within a piece.

We then use a beat tracker [5] in order to average the time frames into rhythmic frames. Instead of using the traditional beat-synchronous approach, which is typically employed in a segmentation task, we divide each beat duration by 4 and aggregate accordingly, thus having  $N = 4B$  time frames, where  $B$  is the number of beats detected in the piece. The motivation behind this is that patterns may not start at the beat level, as opposed to the case for long sections. Furthermore, adding a finer level of granularity (i.e., analyzing the piece at a sixteenth-note level instead of every fourth note or at the beat level) should yield better results.

### 3.2 Finding Repeated Segments

We make use of the transposition-invariant SSM  $\mathcal{S}$  described in Section 2.2, computed from the chromagram of a given audio signal using the Euclidean distance, in order to identify repeated segments. As opposed to the task of segmentation, the goal here is to find *all* possible repeated segments in  $\mathcal{S}$ , independent of how short they are or the amount of overlap present. The other major difference is that we do not aim to find all of the segments of the piece, but rather identify all of the repeated ones.

Repeated segments appear in  $\mathcal{S}$  as diagonal “stripes”, also known as paths. If the beat-tracker results in no errors (or if the piece contains no tempo variations), these stripes will be perfectly diagonal.

#### 3.2.1 Quantifying Paths with a Score

We propose a score-based greedy algorithm to efficiently identify the most prominent paths in  $\mathcal{S}$ . Starting from  $\mathcal{S} \in \mathbb{R}^{N \times N}$ , we set half of its diagonals to zero, including the main one, due to its symmetrical properties, resulting in  $\hat{\mathcal{S}}$ , s.t.  $\hat{\mathcal{S}}(n, m) = 0$  if  $n \leq m$  and  $\hat{\mathcal{S}}(n, m) = \mathcal{S}(n, m)$  if  $n > m, \forall n, m \in [1 : N]$ . We then compute a score function  $\sigma$  for each possible path in all the non-zero diagonals of  $\hat{\mathcal{S}}$ , resulting in a search space of  $N(N-1)/2$  possible positions in which paths can start.

Before introducing the score function  $\sigma$ , we define a trace function given a square matrix  $X \in \mathbb{R}^{N_x \times N_x}$  with an offset parameter  $\omega$ :

$$\text{tr}(X, \omega) = \sum_{i=1}^{N_x - \omega} X(i, i + \omega), \omega \in \mathbb{Z} \quad (2)$$

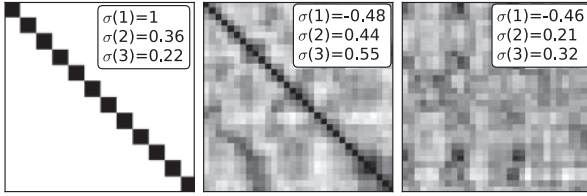
As can be seen from this equation, when  $\omega = 0$  we have the standard trace function definition.

The score function  $\sigma$  uses various traces of the matrix that comprises a possible path in order to quantify the degree of repetition of the path. If a possible path starts at indices  $n, m$  and has a duration of  $M$  time frames, then the matrix that the path defines is  $P \in \mathbb{R}^{M \times M}$ , s.t.  $P(i, j) = \hat{\mathcal{S}}(n + i - 1, m + j - 1), \forall i, j \in [1 : M]$ . We now can define the score  $\sigma$  as the sum of the closest traces to the diagonal of  $P$  (i.e., those with a small  $\omega$ ) and subtract the traces that are farther apart from the diagonal (i.e., where  $\omega$  is greater). We then normalize in order to obtain a score independent from the duration  $M$  of the possible path:

$$\sigma(\rho) = \frac{\left( \sum_{\omega=-\rho}^{\rho-1} \text{tr}(P, \omega) \right) - \text{tr}(P, \pm\rho)}{M + \sum_{i=1}^{\rho-1} 2(M-i)} \quad (3)$$

where  $\rho \in \mathbb{N}$  is the maximum offset to be taken into account when computing the traces of  $P$ . The greater the  $\rho$ , the greater the  $\sigma$  for segments that contain substantial energy around their main diagonal (e.g., paths that contain significant rhythmic variations), even though the precision decreases as  $\rho$  increases.

Examples for various  $\sigma(\rho)$  can be seen in Figure 3. For a perfectly clean path (left), we see that  $\rho = 1$  gives the maximum score of 1. However, the score decreases as  $\rho$  increases, since there is zero energy in the diagonals right next to the main diagonal. On the other hand, for matrices extracted from audio signals (middle and right), we can see that the scores  $\sigma(1)$  are low, indicating that the diagonals next to the main diagonal contain amounts of energy similar to the main diagonal. However, when  $\rho > 1$ , the score is substantially different from a matrix with a path (middle) and a matrix without one (right).



**Figure 3.** Three examples showing the behavior of the path score  $\sigma(\rho)$ . The one on the left shows a synthetic example of a perfect path. The one in the middle contains a real example of a path in which there is some noise around the diagonal of the matrix. In the example on the right, a matrix with no path is shown.

### 3.2.2 Applying the Score

For all  $N(N-1)/2$  positions in which paths can potentially start in  $\hat{S}$ , we want to extract the most prominent ones (i.e., the ones that have a high  $\sigma$ ). At the same time, we want to extract the paths from beginning to end in the most accurate way possible. The algorithm that we propose assigns a certain  $\sigma$  to an initial possible path  $\hat{z}$  of a minimum length of  $\nu$  time frames, which reduces the search space to  $(N-\nu+1)(N-\nu)/2$ . If the score  $\sigma$  is greater than a certain threshold  $\theta$ , we increase the possible path by one time frame, and recompute  $\sigma$  until  $\sigma \leq \theta$ . By then, we can store the path  $\hat{z}$  as a segment in the set of segments  $\mathcal{Z}$ . In order to avoid incorrectly identifying possible paths that are too close to the found path, we zero out the found path from  $\hat{S}$ , including all the  $\rho$  closest diagonals, and keep looking, starting from the end of the recently found path.

The pseudocode for this process can be seen in Algorithm 1, where  $|x|$  returns the length of the path  $x$ ,  $\{x\}$  returns the path in which all elements equal  $x$ , the function `ComputeScore` computes the  $\sigma(\rho)$  as described in Section 3.2.1, `OutOfBounds(x, X)` checks whether the path  $x$  is out of bounds with respect to  $X$ , `IncreasePath(x)` increases the path  $x$  by one (analogously as `DecreasePath`), and `ZeroOutPath(X, x, rho)` assigns zeros to the path  $x$  found in  $X$ , including all the closest  $\rho$  diagonals.

---

#### Algorithm 1 Find Repeated Segments

---

**Require:**  $\hat{S}, \rho, \theta, \nu$

**Ensure:**  $\mathcal{Z} = \{z_1, \dots, z_k\}$

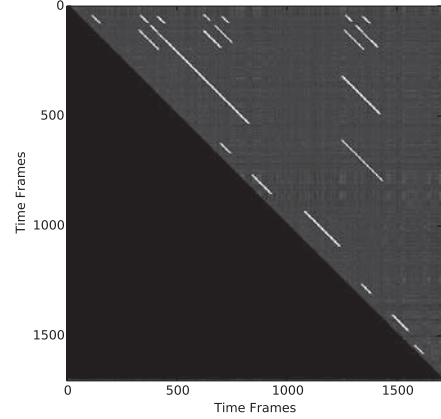
```

for  $\hat{z} \in \hat{S} \wedge |\hat{z}| = \nu \wedge \hat{z} \neq \{0\}$  do
   $b \leftarrow \text{False}$ 
   $\sigma \leftarrow \text{ComputeScore}(\hat{z}, \rho)$ 
  while  $\sigma > \theta \wedge \neg \text{OutOfBounds}(\hat{z}, \hat{S})$  do
     $b \leftarrow \text{True}$ 
     $\hat{z} \leftarrow \text{IncreasePath}(\hat{z})$ 
     $\sigma \leftarrow \text{ComputeScore}(\hat{z}, \rho)$ 
  end while
  if  $b$  then
     $\mathcal{Z}.\text{add}(\text{DecreasePath}(\hat{z}))$ 
     $\text{ZeroOutPath}(\hat{S}, \hat{z}, \rho)$ 
  end if
end for
return  $\mathcal{Z}$ 

```

---

An example of the paths found by the algorithm is shown in Figure 4. Parts of some segments are repeated as standalone segments (i.e., segments within segments), therefore allowing overlap across patterns as expected in this task. Observe how some of the segments repeat almost exactly across the piece—there is a set of patterns at the top of the matrix that appears to repeat at least three times. The next step of the algorithm is to cluster these segments together so that they represent a single pattern with various occurrences.



**Figure 4.** Paths found (marked in white) using the proposed algorithm for Chopin's Op. 24 No. 4., with  $\theta = 0.33$ ,  $\rho = 2$ .

### 3.3 Clustering the Segments

Each segment  $z \in \mathcal{Z}$ , defined by the two indices in which it starts  $(s_i, s_j)$  and ends  $(e_i, e_j)$  in  $\hat{S}$ , contains two occurrences of a pattern: the one that starts in  $s_i$  and ends in  $e_i$  and the one that occurs between the time indices  $s_j$  and  $e_j$ . In order to cluster the repeated occurrences of a single pattern, we find an occurrence for each segment  $z \in \mathcal{Z}$  if one of the other segments in  $\mathcal{Z}$  starts and ends in similar locations with respect to the second dimension of  $\hat{S}$ . Note that we set to zero the bottom left triangle of the matrix as explained in Section 3.2.1, so we cannot use the first dimension to cluster the occurrences. Formally, a segment  $\hat{z}$  is an occurrence of  $z$  if

$$(s_j^z - \Theta \leq s_j^{\hat{z}} \leq s_j^z + \Theta) \wedge (e_j^z - \Theta \leq e_j^{\hat{z}} \leq e_j^z + \Theta) \quad (4)$$

where  $s_j^z$  represents the starting point of the segment  $z$  in the second dimension of  $\hat{S}$  and analogously  $e_j^z$  is the ending point, and  $\Theta$  is a tolerance parameter.

### 3.4 Final Output

At this point, we have a set of patterns with their respective occurrences represented by their starting and ending time-frame indices. Even though the algorithm is not able to distinguish the different musical lines within the patterns, we can use the annotated score to output the exact notes that occur during the identified time indices, as suggested in the MIREX task [1]. If no score is provided, only the time

points will be presented. In order to overcome this limitation in future work, the audio should be source-separated to identify the different lines and perform an F0 estimation to correctly identify the exact melody that defines the pattern (and not just the time points at which it occurs). Progress toward this goal has been presented in [2].

### 3.5 Time Complexity Analysis

Once the rhythm-synchronous chromagram is computed, the process of calculating the transposition-invariant SSM is  $O(13N^2) = O(N^2)$ , where  $N$  is the number of time frames of the chromagram. The procedure to compute the score given a path has a time complexity of  $O(2\rho M) = O(\rho M)$ , where  $\rho$  is the required parameter for the computation of the score, and  $M$  is the length of the path from which to compute the score. The total process of identifying segments is  $O\left(\frac{(N-\nu+1)(N-\nu)}{2}\rho M\right) = O((N-\nu)^2\rho M)$ , where  $\nu$  is the minimum number of time frames that a pattern can have. Asymptotically, we can neglect the clustering of the segments, since the length of  $\mathcal{Z}$  will be much less than  $N$ . Therefore, the total time complexity of the proposed algorithm is  $O(N^2 + (N-\nu)^2\rho M)$ .

## 4. EVALUATION

We use the JKU Patterns Development Dataset<sup>1</sup> to evaluate our algorithm. This dataset is comprised of five classical pieces annotated by various musicologists and researchers [1]. This dataset is the public subset of the one employed to evaluate the Pattern Discovery task at MIREX, using the metrics described below.

### 4.1 Metrics

Two main aspects of this task are evaluated: the patterns discovered and the occurrences of the identified patterns across the piece. Collins and Meredith proposed metrics to quantify these two aspects, which are detailed in [1]; all of these metrics use the standard  $F_1$  accuracy score, defined as  $F_1 = 2PR/(P + R)$ , where  $P$  is precision (such that  $P = 1$  if all the estimated elements are correct), and  $R = 1$  is recall (such that  $R = 1$  if all the annotated elements are estimated).

**Establishment  $F_1$  Score ( $F_{\text{est}}$ ):** Determines how the annotated patterns are *established* by the estimated output. This measure returns a score of 1 if at least one occurrence of each pattern is discovered by the algorithm to be evaluated.

**Occurrence  $F_1$  Score ( $F_0$ ):** For all the patterns found, we want to estimate the ability of the algorithm to capture all of the occurrences of these patterns within the piece independently of how many different patterns the algorithm has identified. Therefore, this score would be 1 if the algorithm has only found one pattern with all the correct occurrences. A parameter  $c$  controls when a pattern is considered to have been discovered, and therefore whether it counts toward the occurrence scores. The higher the  $c$ , the

smaller the tolerance. In this evaluation, as in MIREX, we use  $c = .75$  and  $c = .5$ .

**Three-Layer  $F_1$  Score ( $F_3$ ):** This measure combines both the patterns established and the quality of their occurrences into a single score. It is computed using a three-step process that yields a score of 1 if a correct pattern has been found and all its occurrences have been correctly identified.

### 4.2 Results

The results of the proposed algorithm, computed using the open source evaluation package `mir_eval` [19], are shown in Table 1, averaged for the entire JKU Dataset, along with an earlier version of our algorithm submitted to MIREX [15], another recent algorithm called SIARCT-CFP [2] that is assessed using both audio and symbolic representations as input in [3], and ‘‘COSIATEC Segment’’, a method that only uses symbolic inputs [12]. We use this latter method for comparison because it is the only symbolic method in which we have access to all of the resulting metrics, and SIARCT-CFP since it is the most recent method that uses audio as input. The parameter values used to compute these results,  $\nu = 8, \theta = 0.33, \rho = 2$ , and  $\Theta = 4$ , were found empirically. We can see how our algorithm is better than [15] in all the metrics except running time; it also finds more correct patterns than [3] (the current state-of-the-art when using audio as input).

Our algorithm obtains state-of-the-art results when extracting patterns from audio, obtaining an  $F_{\text{est}}$  of 49.80%. This is better than the symbolic version of [2] and almost as good as the algorithm described in [12]. The fact that our results are superior or comparable to the two other algorithms using symbolic representations indicates the potential of our method.

When evaluating the occurrences of the patterns, we see that our algorithm is still better than [15], but worse than [2] (at least for  $c = .5$ , which is the only reported result). Nevertheless, the numbers are much lower than [12]. In this case, working with symbolic representations (or estimating the F0 in order to apply a symbolic algorithm as in [2]) yields significantly better results. It is interesting to note that when the tolerance increases (i.e.  $c = .5$ ), our results improve as opposed to the other algorithms. This might be due to the fact that some of the occurrences found in the SSM were actually very similar (therefore they were found in the matrix) but were slightly different in the annotated dataset. A good example of this would be an occurrence that contains only one melodic voice. Our algorithm only finds points in time in which an occurrence might be included, it does not perform any type of source separation in order to identify the different voices. If the tolerance decreases sufficiently, a polyphonic occurrence would be accepted as similar to a monophonic one corresponding to the same points in time.

Our three layer score ( $F_3$ ) is the best result when using audio recordings, with an F-measure of 31.74% (unfortunately this metric was not reported in [2]). This metric aims to evaluate the quality of the algorithm with a single

<sup>1</sup> <https://dl.dropbox.com/u/11997856/JKU/JKUPDD-Aug2013.zip>

Alg	$P_{est}$	$R_{est}$	$F_{est}$	$P_{O(.75)}$	$R_{O(.75)}$	$F_{O(.75)}$	$P_3$	$R_3$	$F_3$	$P_{O(.5)}$	$R_{O(.5)}$	$F_{O(.5)}$	Time (s)
Proposed	54.96	51.73	<b>49.80</b>	37.58	27.61	<b>31.79</b>	35.12	35.28	<b>32.01</b>	45.17	34.98	38.73	454
[3]	14.9	60.9	23.94	–	–	–	–	–	–	62.9	51.9	<b>56.87</b>	–
[15]	40.83	46.43	41.43	32.08	21.24	24.87	30.43	31.92	28.23	26.60	20.94	23.18	<b>196</b>
[3]	21.5	78.0	33.7	–	–	–	–	–	–	78.3	74.7	76.5	–
[12]	43.60	63.80	50.20	65.40	76.40	68.40	40.40	54.40	44.20	57.00	71.60	63.20	7297

**Table 1.** Results of various algorithms using the JKU Patterns Development Dataset, averaged across pieces. The top rows of the table contain algorithms that use deadpan audio as input. The bottom rows correspond to algorithms that use symbolic representations as input.

score, including both pattern establishment and occurrence retrieval. Our results are still far from perfect (32.01%), but when compared to an algorithm that uses symbolic representations [12] (44.21%), it appears our results are not far from the state-of-the-art for symbolic representations.

Finally, our algorithm takes more than twice as long as [15]. However, our method is over 16 times faster than [12], indicating it is efficient in terms of computation time. This algorithm is implemented in Python and available for public download.<sup>2</sup>

## 5. CONCLUSIONS

We presented a method to discover repeating polyphonic patterns using audio recordings as input. The method makes use of various standard techniques typically used for music segmentation. We evaluated our method using the JKU Pattern Development Dataset and showed how it obtains competent results when retrieving all the occurrences of the patterns and state-of-the-art results when finding patterns. When the algorithm is compared to others that use symbolic representations, we see that it is comparable or superior in terms of the correct patterns found. In future work, source separation might be needed to successfully identify patterns that only comprise a subset of the different musical lines.

## 6. REFERENCES

- [1] T. Collins. Discovery of Repeated Themes & Sections, 2013.
- [2] T. Collins, A. Arzt, S. Flossmann, and G. Widmer. SIARCT-CFP: Improving Precision and the Discovery of Inexact Musical Patterns in Point-set Representations. In *Proc. of the 14th International Society for Music Information Retrieval Conference*, pages 549–554, Curitiba, Brazil, 2014.
- [3] T. Collins, B. Sebastian, F. Krebs, and G. Widmer. Bridging the Audio-Symbolic Gap: The Discovery of Repeated Note Content Directly From Polyphonic Music Audio. In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*, pages 1–12, London, UK, 2014.
- [4] D. Conklin and C. Anagnostopoulou. Representation and Discovery of Multiple Viewpoint Patterns. In *Proc. of the International Computer Music Conference*, pages 479–485, La Havana, Cuba, 2001.
- [5] D. P. W. Ellis and G. E. Poliner. Identifying ‘Cover Songs’ with Chroma Features and Dynamic Programming Beat Tracking. In *Proc. of the 32nd IEEE International Conference on Acoustics Speech and Signal Processing*, pages 1429–1432, Honolulu, HI, USA, 2007.
- [6] J. C. Forth. *Cognitively-motivated Geometric Methods of Pattern Discovery and Models of Similarity in Music*. PhD thesis, Glodsmiths, University of London, 2012.
- [7] J. C. Forth and G. A. Wiggins. An Approach for Identifying Salient Repetition in Multidimensional Representations of Polyphonic Music. In J. Chan, J. W. Daykin, and M. S. Rahman, editors, *London Algorithms 2008: Theory and Practice*, pages 44–58. UK: College Publications, 2009.
- [8] B. Janssen, W. B. D. Haas, A. Volk, and P. V. Kranenburg. Discovering Repeated Patterns in Music: State of Knowledge, Challenges, Perspectives. In *Proc. of the 10th International Symposium on Computer Music Multidisciplinary Research*, Marseille, France, 2013.
- [9] O. Lartillot. Multi-Dimensional motivic pattern extraction founded on adaptive redundancy filtering. *Journal of New Music Research*, 34(4):375–393, Dec. 2005.
- [10] K. Lemström. *String Matching Techniques for Music Retrieval*. PhD thesis, University of Helsinki, Finland, 2000.
- [11] D. Meredith. Point-set Algorithms For Pattern Discovery And Pattern Matching In Music. In T. Crawford and R. C. Veltkamp, editors, *Proc. of the Dagstuhl Seminar on Content-Based Retrieval.*, Dagstuhl, Germany, 2006.
- [12] D. Meredith. COSIATEC and SIATECCompress: Pattern Discovery by Geometric Compression. In *Music Information Retrieval Evaluation eXchange*, Curitiba, Brazil, 2013.
- [13] M. Müller. *Information Retrieval for Music and Motion*. Springer, 2007.
- [14] M. Müller and M. Clausen. Transposition-Invariant Self-Similarity Matrices. In *Proc. of the 8th International Conference on Music Information Retrieval*, pages 47–50, Vienna, Austria, 2007.
- [15] O. Nieto and M. Farbood. MIREX 2013: Discovering Musical Patterns Using Audio Structural Segmentation Techniques. In *Music Information Retrieval Evaluation eXchange*, Curitiba, Brazil, 2013.
- [16] O. Nieto and M. M. Farbood. Perceptual Evaluation of Automatically Extracted Musical Motives. In *Proc. of the 12th International Conference on Music Perception and Cognition*, pages 723–727, Thessaloniki, Greece, 2012.
- [17] O. Nieto, E. J. Humphrey, and J. P. Bello. Compressing Audio Recordings into Music Summaries. In *Proc. of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012.
- [18] J. Paulus, M. Müller, and A. Klapuri. Audio-Based Music Structure Analysis. In *Proc of the 11th International Society of Music Information Retrieval*, pages 625–636, Utrecht, Netherlands, 2010.
- [19] C. Raffel, B. Mcfee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis. mir\_eval: A Transparent Implementation of Common MIR Metrics. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, 2014.
- [20] R. Weiss and J. P. Bello. Unsupervised Discovery of Temporal Structure in Music. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1240–1251, 2011.

<sup>2</sup> <https://github.com/urinieto/MotivesExtractor>



# BOUNDARY DETECTION IN MUSIC STRUCTURE ANALYSIS USING CONVOLUTIONAL NEURAL NETWORKS

Karen Ullrich    Jan Schlüter    Thomas Grill

Austrian Research Institute for Artificial Intelligence, Vienna

{karen.ullrich, jan.schluter, thomas.grill}@ofai.at

## ABSTRACT

The recognition of boundaries, e.g., between chorus and verse, is an important task in music structure analysis. The goal is to automatically detect such boundaries in audio signals so that the results are close to human annotation. In this work, we apply Convolutional Neural Networks to the task, trained directly on mel-scaled magnitude spectrograms. On a representative subset of the SALAMI structural annotation dataset, our method outperforms current techniques in terms of boundary retrieval  $F$ -measure at different temporal tolerances: We advance the state-of-the-art from 0.33 to 0.46 for tolerances of  $\pm 0.5$  seconds, and from 0.52 to 0.62 for tolerances of  $\pm 3$  seconds. As the algorithm is trained on annotated audio data without the need of expert knowledge, we expect it to be easily adaptable to changed annotation guidelines and also to related tasks such as the detection of song transitions.

## 1. INTRODUCTION

The determination of the overall structure of a piece of audio, often referred to as *musical form*, is one of the key tasks in music analysis. Knowledge of the musical structure enables a variety of real-world applications, be they commercially applicable, such as for browsing music, or educational. A large number of different techniques for automatic structure discovery have been developed, see [16] for an overview. Our contribution describes a novel approach to retrieve the boundaries between the main structural parts of a piece of music. Depending on the music under examination, the task of finding such musical boundaries can be relatively simple or difficult, in the latter case leaving ample space for ambiguity. In fact, two human annotators hardly ever annotate boundaries at the exact same positions. Instead of trying to design an algorithm that works well in all circumstances, we let a Convolutional Neural Network (CNN) learn to detect boundaries from a large corpus of human-annotated examples.

The structure of the paper is as follows: After giving an overview over related work in Section 2, we describe our

proposed method in Section 3. In Section 4, we introduce the data set used for training and testing. After presenting our main results in Section 5, we wrap up in Section 6 with a discussion and outlook.

## 2. RELATED WORK

In the overview paper to audio structure analysis by Paulus et al. [16], three fundamental approaches to segmentation are distinguished: Novelty-based, detecting transitions between contrasting parts, homogeneity-based, identifying sections that are consistent with respect to their musical properties, and repetition-based, building on the determination of recurring patterns. Many segmentation algorithms follow mixed strategies. Novelty is typically computed using Self-Similarity Matrices (SSMs) or Self-Distance Matrices (SDMs) with a sliding checkerboard kernel [4], building on audio descriptors like timbre (MFCC features), pitch, chroma vectors and rhythmic features [14]. Alternative approaches calculate difference features on more complex audio feature sets [21]. In order to achieve a higher temporal accuracy in rhythmic music, audio features can be accumulated beat-synchronously. Techniques capitalizing on homogeneity use clustering [5] or state-modelling (HMM) approaches [1], or both [9, 11]. Repeating pattern discovery is performed on SSMs or SDMs [12], and often combined with other approaches [13, 15]. Some algorithms combine all three basic approaches [18].

Almost all existing algorithms are hand-designed from end to end. To the best of our knowledge, only two methods are partly learning from human annotations: Turnbull et al. [21] compute temporal differences at three time scales over a set of standard audio features including chromagrams, MFCCs, and fluctuation patterns. Training Boosted Decision Stumps to classify the resulting vectors into boundaries and non-boundaries, they achieved significant gains over a hand-crafted boundary detector using the same features, evaluated on a set of 100 pop songs. McFee et al. [13] employ Ordinal Linear Discriminant Analysis to learn a linear transform of beat-aligned audio features (including MFCCs and chroma) that minimizes the variance within a human-annotated segment while maximizing the distance across segments. Combined with a repetition feature, their method defines the current state of the art in boundary retrieval, but still involves significant manual engineering.

For other tasks in the field of Music Information Retrieval, supervised learning with CNNs has already proven



© Karen Ullrich, Jan Schlüter, and Thomas Grill.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Karen Ullrich, Jan Schlüter, and Thomas Grill. “Boundary Detection in Music Structure Analysis using Convolutional Neural Networks”, 15th International Society for Music Information Retrieval Conference, 2014.

to outperform hand-designed algorithms, sometimes by a large margin [3, 6, 8, 10, 17]. In this work, we investigate whether CNNs are effective for structural boundary detection as well.

### 3. METHOD

We propose to train a neural network on human annotations to predict likely musical boundary locations in audio data. Our method is derived from Schlüter and Böck [17], who use CNNs for onset detection: We also train a CNN as a binary classifier on spectrogram excerpts, but we adapt their method to include a larger input context and respect the higher inaccuracy and scarcity of segment boundary annotations compared to onset annotations. In the following, we will describe the features, neural network, supervised training procedure and the post-processing of the network output to obtain boundary predictions.

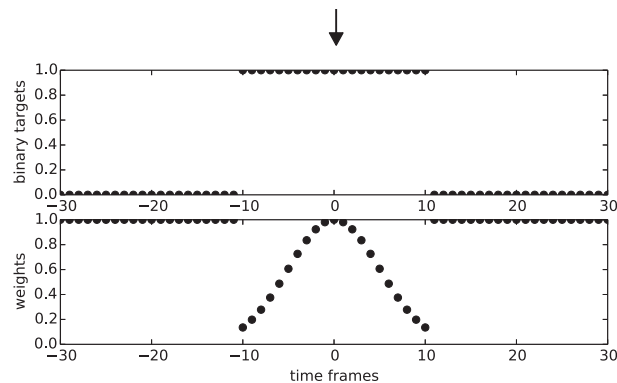
#### 3.1 Feature Extraction

For each audio file, we compute a magnitude spectrogram with a window size of 46 ms (2048 samples at 44.1 kHz) and 50% overlap, apply a mel filterbank of 80 triangular filters from 80 Hz to 16 kHz and scale magnitudes logarithmically. To be able to train and predict on spectrogram excerpts near the beginning and end of a file, we pad the spectrogram with pink noise at -70 dB as needed (padding with silence is impossible with logarithmic magnitudes, and white noise is too different from the existing background noise in natural recordings). To bring the input values to a range suitable for neural networks, we follow [17] in normalizing each frequency band to zero mean and unit variance. Finally, to allow the CNN to process larger temporal contexts while keeping the input size reasonable, we subsample the spectrogram by taking the maximum over 3, 6 or 12 adjacent time frames (without overlap), resulting in a frame rate of 14.35 fps, 7.18 fps or 3.59 fps, respectively. We will refer to these frame rates as *high*, *std* and *low*.

We also tried training on MFCCs and chroma vectors (descriptors with less continuity in the ‘vertical’ feature dimension to be exploited by convolution), as well as fluctuation patterns and self-similarity matrices derived from those. Overall, mel spectrograms proved the most suitable for the algorithm and performed best.

#### 3.2 Convolutional Neural Networks

CNNs are feed-forward neural networks usually consisting of three types of layers: Convolutional layers, pooling layers and fully-connected layers. A convolutional layer computes a convolution of its two-dimensional input with a fixed-size kernel, followed by an element-wise nonlinearity. The input may consist of multiple same-sized channels, in which case it convolves each with a separate kernel and adds up the results. Likewise, the output may consist of multiple channels computed with distinct sets of kernels. Typically the kernels are small compared to the input, allowing CNNs to process large inputs with few



**Figure 1.** The arrow at the top signifies an annotated segment boundary present within a window of feature frames. As seen in the upper panel, the target labels are set to one in the environment of this boundary, and to zero elsewhere. The lower panel shows how positive targets far from the annotation are given a lower weight in training.

learnable parameters. A pooling layer subsamples its two-dimensional input, possibly by different factors in the two dimensions, handling each input channel separately. Here, we only consider max-pooling, which introduces some translation invariance across the subsampled dimension. Finally, a fully-connected layer discards any spatial layout of its input by reshaping it into a vector, computes a dot product with a weight matrix and applies an element-wise nonlinearity to the result. Thus, unlike the other layer types, it is not restricted to local operations and can serve as the final stage integrating all information to form a decision.

In this work, we fix the network architecture to a convolutional layer of  $16 \times 8 \times 6$  kernels (8 time frames, 6 mel bands, 16 output channels), a max-pooling layer of  $3 \times 6$ , another convolution of  $32 \times 6 \times 3$  kernels, a fully-connected layer of 128 units and a fully-connected output layer of 1 unit. This architecture was determined in preliminary experiments and not further optimized for time constraints.

#### 3.3 Training

The input to the CNN is a spectrogram excerpt of  $N$  frames, and its output is a single value giving the probability of a boundary in the center of the input. The network is trained in a supervised way on pairs of spectrogram excerpts and binary labels. To account for the inaccuracy of the ground truth boundary annotations (as observable from the disagreement between two humans annotating the same piece), we employ what we will refer to as *target smearing*: All excerpts centered on a frame within  $\pm E$  frames from an annotated boundary will be presented to the network as positive examples, weighted in learning by a Gaussian kernel centered on the boundary. Figure 1 illustrates this for  $E = 10$ . We will vary both the spectrogram length  $N$  and smearing environment  $E$  in our experiments. To compensate for the scarceness of positive examples, we increase their chances of being randomly selected for a training step by a factor of 3.

Training is performed using gradient descent on cross-

entropy error with mini-batches of 64 examples, momentum of 0.95, and an initial learning rate of 0.6 multiplied by 0.85 after every mini-epoch of 2000 weight updates. We apply 50% dropout to the inputs of both fully-connected layers [7]. Training is always stopped after 20 mini-epochs, as the validation error turned out not to be robust enough for early stopping. Implemented in Theano [2], training a single CNN on an Nvidia GTX 780 Ti graphics card took 50–90 minutes.

### 3.4 Peak-picking

At test time, we apply the trained network to each position in the spectrogram of the music piece to be segmented, obtaining a boundary probability for each frame. We then employ a simple means of peak-picking on this boundary activation curve: Every output value that is not surpassed within  $\pm 6$  seconds is a boundary candidate. From each candidate value we subtract the average of the activation curve in the past 12 and future 6 seconds, to compensate for long-term trends. We end up with a list of boundary candidates along with strength values that can be thresholded at will. We found that more elaborate peak picking methods did not improve results.

## 4. DATASET

We evaluate our algorithm on a subset of the Structural Analysis of Large Amounts of Music Information (SALAMI) database [20]. In total, this dataset contains over 2400 structural annotations of nearly 1400 musical recordings of different genres and origins. About half of the annotations (779 recordings, 498 of which are doubly-annotated) are publicly available.<sup>1</sup> A part of the dataset was also used in the “Audio Structural Segmentation” task of the annual MIREX evaluation campaign in 2012 and 2013.<sup>2</sup> Along with quantitative evaluation results, the organizers published the ground truth and predictions of 17 different algorithms for each recording. By matching the ground truth to the public SALAMI annotations, we were able to identify 487 recordings. These serve as a test set to evaluate our algorithm against the 17 MIREX submissions. We had another 733 recordings at our disposal, annotated following the SALAMI guidelines, which we split into 633 items for training and 100 for validation.

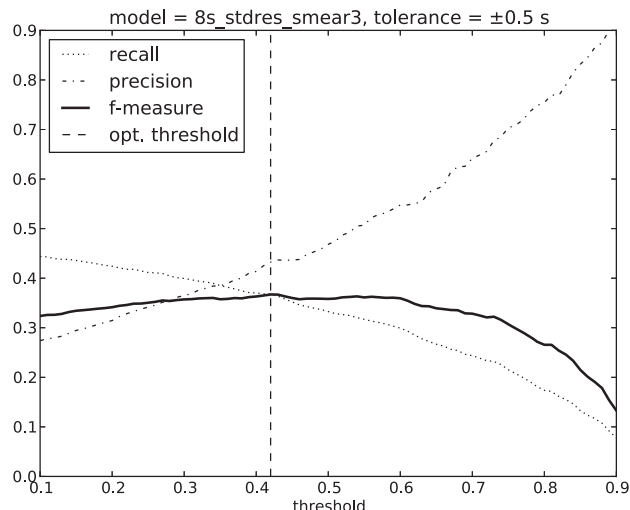
## 5. EXPERIMENTAL RESULTS

### 5.1 Evaluation

For boundary retrieval, the MIREX campaign uses two evaluation measures: *Median deviation* and *Hit rate*. The former measures the median distance between each annotated boundary and its closest predicted boundary or vice versa. The latter checks which predicted boundaries fall close enough to an unmatched annotated boundary (true

<sup>1</sup> [http://ddmal.music.mcgill.ca/datasets/salami/SALAMI\\_data\\_v1.2.zip](http://ddmal.music.mcgill.ca/datasets/salami/SALAMI_data_v1.2.zip), accessed 2014-05-02

<sup>2</sup> Music Information Retrieval Evaluation eXchange, <http://www.music-ir.org/mirex>, accessed 2014-04-29



**Figure 2.** Optimization of the threshold shown for model `8s_std_3s` at tolerance  $\pm 0.5$  seconds. Boundary retrieval precision, recall and F-measure are averaged over the 100 validation set files.

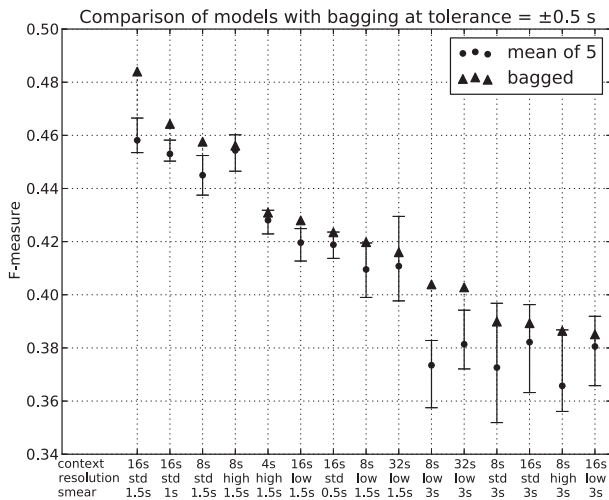
positives), records remaining unmatched predictions and annotations as false positives and negatives, respectively, then computes the precision, recall and F-measure. Since not only the temporal distance of predictions, but also the figures of precision and recall are of interest, we opted for the Hit rate at as our central measure of evaluation, computed at a temporal tolerance of  $\pm 0.5$  seconds (as in [21]) and  $\pm 3$  seconds (as in [9]). For accumulation over multiple recordings, we follow the MIREX evaluation by calculating F-measure, precision and recall per item and averaging the three measures over the items for the final result. Note that the averaged F-measure is not necessarily the harmonic mean of the averaged precision and recall. Our evaluation code is publicly available for download.<sup>3</sup>

### 5.2 Baseline and upper bound

Our focus for evaluation lies primarily on the F-measure. Theoretically, the F-measure is bounded by  $F \in [0, 1]$ , but for the given task, we can derive more useful lower and upper bounds to compare our results to. As a baseline, we use regularly spaced boundary predictions starting at time 0. Choosing an optimal spacing, we obtain an F-measure of  $F_{\text{inf},3} \approx 0.33$  for  $\pm 3$  seconds tolerance, and  $F_{\text{inf},0.5} \approx 0.13$  for a tolerance of  $\pm 0.5$  seconds. Note that it is crucial to place the first boundary at 0 seconds, where a large fraction of the music pieces has annotated segment boundaries. Many pieces have only few boundaries at all, thus the impact can be considerable. An upper bound  $F_{\text{sup}}$  can be derived from the insight that no annotation will be perfect given the fuzzy nature of the segmentation task. Even though closely following annotation guidelines,<sup>4</sup> two annotators might easily disagree on the existence or exact po-

<sup>3</sup> <http://ofai.at/research/impml/projects/audiostreams/ismir2014/>

<sup>4</sup> cf. the SALAMI Annotator’s Guide: <http://www.music.mcgill.ca/~jordan/salami/SALAMI-Annotator-Guide.pdf>, accessed 2014-04-30



**Figure 3.** Comparison of different model parameters (context length, resolution and target smearing) with respect to mean F-measure on our validation set at  $\pm 0.5$  seconds tolerance. Mean and minimum-maximum range of five individually trained models for each parameter combination are shown, as well as results for bagging the five models.

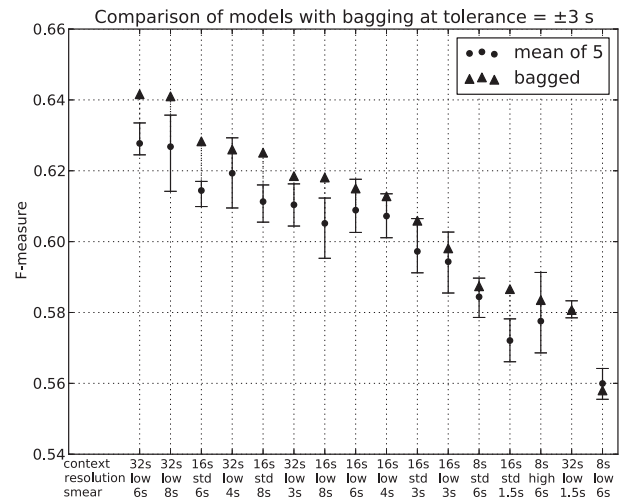
sitions of segment boundaries. By analyzing the items in the public SALAMI dataset that have been annotated twice (498 pieces in total), we calculated  $F_{\text{sup},3} \approx 0.76$  for  $\pm 3$  seconds tolerance, and  $F_{\text{sup},0.5} \approx 0.67$  for  $\pm 0.5$  seconds tolerance. Within our evaluation data subset (439 double-annotations), the results are only marginally different with  $F_{\text{sup},0.5} \approx 0.68$ .

### 5.3 Threshold optimization

Peak-picking, described in Section 3.4, delivers the positions of potential boundaries along with their probabilities, as calculated by the CNN. The application of a threshold to those probabilities rejects part of the boundaries, affecting the precision and recall rates and consequently the F-measure we use for evaluation. Figure 2 shows precision and recall rates as well as the F-measure as a function of the threshold for the example of the *8s\_std\_3s* model (8 seconds of context, standard resolution, target smearing 3 seconds) at  $\pm 0.5$  seconds tolerance, applied to the 100 files of the validation data set. By locating the maximum of the F-measure we retrieve an estimate for the optimum threshold which is specific for each individual learned model. Since the curve for the F-measure is typically flat-topped for a relatively wide range of threshold values, the choice of the actual value is not very delicate.

### 5.4 Temporal context investigation

It is intuitive to assume that the CNN needs a certain amount of temporal context to reliably judge the presence of a boundary. Furthermore, the temporal resolution of the input spectra (Section 3.1) and the applied target smearing (Section 3.3) is expected to have an impact on the temporal accuracy of the predictions. See Figure 3 and Figure 4 for comparisons of these model parameters, for tolerances  $\pm 0.5$  seconds



**Figure 4.** Comparison of different model parameters (context length, resolution and target smearing) with respect to mean F-measure on our validation set at  $\pm 3$  seconds tolerance. Mean and minimum-maximum range of five individually trained models for each parameter combination are shown, as well as results for bagging the five models.

and  $\pm 3$  seconds, respectively. Each bar in the plots represents the mean and minimum-maximum range of five individual experiments with different random initializations. For the case of only  $\pm 0.5$  seconds of acceptable error, we conclude that target smearing must also be small: A smearing width of 1 to 1.5 seconds performs best. Low temporal spectral resolution tends to diminish results, and the context length should not be shorter than 8 seconds. For  $\pm 3$  seconds tolerance, context length and target smearing are the most influential parameters, with the F-measure peaking at 32 seconds context and 4 to 6 seconds smearing. Low temporal resolution is sufficient, keeping the CNN smaller and easier to train.

### 5.5 Model bagging

As described in Section 5.4, for each set of parameters we trained five individual models. This allows us to improve the performance on the given data using a statistical approach: *Bagging*, in our case averaging the outputs of multiple identical networks trained from different initializations before the peak-picking stage, should help to reduce model uncertainty. After again applying the above described threshold optimization process on the resulting boundaries, we arrived at improvements of the F-measure of up to 0.03, indicated by arrow tips in Figures 3 and 4. Tables 1 and 2 show our final best results after model bagging for tolerances  $\pm 0.5$  seconds and  $\pm 3$  seconds, respectively. The results are set in comparison with the algorithms submitted to the MIREX campaign in 2012 and 2013, and the lower and upper bounds calculated from the annotation ground-truth (see Section 5.2).

## 6. DISCUSSION AND OUTLOOK

Algorithm	F-measure	Precision	Recall
Upper bound (est.)	0.68		
<b>16s_std_1.5s</b>	<b>0.4646</b>	0.5553	0.4583
MP2 (2013)	0.3280	0.3001	0.4108
MP1 (2013)	0.3149	0.3043	0.3605
OYZS1 (2012)	0.2899	0.4561	0.2583
<b>32s_low_6s</b>	0.2884	0.3592	0.2680
KSP2 (2012)	0.2866	0.2262	0.4622
SP1 (2012)	0.2788	0.2202	0.4497
KSP3 (2012)	0.2788	0.2202	0.4497
KSP1 (2012)	0.2788	0.2201	0.4495
RBH3 (2013)	0.2683	0.2493	0.3360
RBH1 (2013)	0.2567	0.2043	0.3936
RBH2 (2013)	0.2567	0.2043	0.3936
RBH4 (2013)	0.2567	0.2043	0.3936
CF5 (2013)	0.2128	0.1677	0.3376
CF6 (2013)	0.2101	0.2396	0.2239
SMGA1 (2012)	0.1968	0.1573	0.2943
MHRAF1 (2012)	0.1910	0.1941	0.2081
SMGA2 (2012)	0.1770	0.1425	0.2618
SBV1 (2012)	0.1546	0.1308	0.2129
Baseline (est.)	0.13		

**Table 1.** Boundary recognition results on our test set at  $\pm 0.5$  seconds tolerance. Our best result is emphasized and compared with results from the MIREX campaign in 2012 and 2013.

Algorithm	F-measure	Precision	Recall
Upper bound (est.)	0.76		
<b>32s_low_6s</b>	<b>0.6164</b>	0.5944	0.7059
<b>16s_std_1.5s</b>	0.5726	0.5648	0.6675
MP2 (2013)	0.5213	0.4793	0.6443
MP1 (2013)	0.5188	0.5040	0.5849
CF5 (2013)	0.5052	0.3990	0.7862
SMGA1 (2012)	0.4985	0.4021	0.7258
RBH1 (2013)	0.4920	0.3922	0.7482
RBH2 (2013)	0.4920	0.3922	0.7482
RBH4 (2013)	0.4920	0.3922	0.7482
SP1 (2012)	0.4891	0.3854	0.7842
KSP3 (2012)	0.4891	0.3854	0.7842
KSP1 (2012)	0.4888	0.3850	0.7838
KSP2 (2012)	0.4885	0.3846	0.7843
SMGA2 (2012)	0.4815	0.3910	0.6965
RBH3 (2013)	0.4804	0.4407	0.6076
CF6 (2013)	0.4759	0.5305	0.5102
OYZS1 (2012)	0.4401	0.6354	0.4038
SBV1 (2012)	0.4352	0.3694	0.5929
MHRAF1 (2012)	0.4192	0.4342	0.4447
Baseline (est.)	0.33		

**Table 2.** Boundary recognition results on our test set at  $\pm 3$  seconds tolerance. Our best result is emphasized and compared with results from the MIREX campaign in 2012 and 2013.

Employing Convolutional Neural Networks trained directly on mel-scaled spectrograms, we are able to achieve boundary recognition F-measures strongly outperforming any algorithm submitted to MIREX 2012 and 2013. The networks have been trained on human-annotated data, considering different context lengths, temporal target smearing and spectrogram resolutions. As we did not need any domain knowledge for training, we expect our method to be easily adaptable to different ‘foci of annotation’ such as, e.g., determined by different musical genres or annotation guidelines. In fact, our method is itself an adaption of a method for onset detection [17] to a different time focus.

There are a couple of conceivable strategies to improve the results further: With respect to the three fundamental approaches to segmentation described in Section 1, the CNNs in this work can only account for novelty and homogeneity, which can be seen as two sides of the same medal. To allow them to leverage repetition cues as well, the vectorial repetition features of McFee et al. [13] might serve as an additional input. Alternatively, the network could be extended with recurrent connections to yield a Recurrent CNN. Given suitable training data, the resulting memory might be able to account for repeating patterns. Secondly, segmentation of musical data by humans is not a trivially sequential process but inherently hierarchical. The SALAMI database actually provides annotations on two levels: A coarse one, as used in the MIREX campaign, but also a more fine-grained variant, encoding subtler details of the temporal structure. It could be helpful to feed both levels to the CNN training, weighted with respect to the significance. Thirdly, we leave much of the data preprocessing to the CNN, very likely using up a considerable part of its capacity. For example, the audio files in the SALAMI collection are of very different loudness, which could be fixed in a simple preprocessing step, either on the whole files, or using some dynamic gain control. Similarly, many of the SALAMI audio files start or end with noise or background sounds. A human annotator easily recognizes this as not belonging to the actual musical content, ignoring it in the annotations. The abrupt change from song-specific background noise to our pink noise padding may be mistaken for a boundary by the CNN, though. Therefore it could be worthwhile to apply some intelligent padding of appropriate noise or background to provide context at the beginnings and endings of the audio. And finally, we have only explored a fraction of the hyperparameter space regarding network architecture and learning, and expect further improvements by a systematic optimization of these.

Another promising direction of research is to explore the internal processing of the trained networks, e.g., by visualization of connection weights and receptive fields [19]. This may help to understand the segmentation process as well as differences to existing approaches, and to refine the network architecture.

## 7. ACKNOWLEDGMENTS

This research is funded by the Federal Ministry for Transport, Innovation & Technology (BMVIT) and the Austrian Science Fund (FWF): TRP 307-N23. Many thanks to the anonymous reviewers for your valuable feedback!

## 8. REFERENCES

- [1] J.-J. Aucouturier and M. Sandler. Segmentation of musical signals using hidden markov models. In *Proc. AES 110th Convention*, May 2001.
- [2] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proc. of the Python for Scientific Computing Conference (SciPy)*, June 2010.
- [3] S. Dieleman, P. Braken, and B. Schrauwen. Audio-based music classification with a pretrained convolutional network. In *Proc. of the 12th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, Miami, FL, USA, October 2011.
- [4] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME'00)*, volume 1, pages 452–455 vol.1, 2000.
- [5] J. T. Foote and M. L. Cooper. Media segmentation using self-similarity decomposition. In *Proc. of The SPIE Storage and Retrieval for Multimedia Databases*, volume 5021, pages 167–175, San Jose, California, USA, January 2003.
- [6] P. Hamel, S. Lemieux, Y. Bengio, and D. Eck. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. In *Proc. of the 12th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, October 2011.
- [7] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580, 2012.
- [8] E. J. Humphrey and J. P. Bello. Rethinking automatic chord recognition with convolutional neural networks. In *Proc. of the 11th Int. Conf. on Machine Learning and Applications (ICMLA)*, volume 2, Boca Raton, FL, USA, December 2012. IEEE.
- [9] M. Levy and M. Sandler. Structural segmentation of musical audio by constrained clustering. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):318–326, Feb 2008.
- [10] T. L.H. Li, A. B. Chan, and A. H.W. Chun. Automatic musical pattern feature extraction using convolutional neural network. In *Proc. of the Int. MultiConf. of Engineers and Computer Scientists (IMECS)*, Hong Kong, March 2010.
- [11] Beth Logan and Stephen Chu. Music summarization using key phrases. In *In Proc. IEEE ICASSP*, pages 749–752, 2000.
- [12] Lie Lu, Muyuan Wang, and Hong-Jiang Zhang. Repeating pattern discovery and structure analysis from acoustic music data. In *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 275–282, New York, NY, USA, 2004. ACM.
- [13] B. McFee and D. P. W. Ellis. Learning to segment songs with ordinal linear discriminant analysis. In *International conference on acoustics, speech and signal processing, ICASSP*, 2014.
- [14] Jouni Paulus and Anssi Klapuri. Acoustic features for music piece structure analysis. In *Conference: 11th International Conference on Digital Audio Effects (Espoo, Finland)*, 2008.
- [15] Jouni Paulus and Anssi Klapuri. Music structure analysis by finding repeated parts. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, AMCMM '06, pages 59–68, New York, NY, USA, 2006. ACM.
- [16] Jouni Paulus and Anssi Klapuri. Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *Trans. Audio, Speech and Lang. Proc.*, 17:12, 2009.
- [17] J. Schlüter and S. Böck. Improved musical onset detection with convolutional neural networks. *Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.
- [18] Joan Serra, Meinard Müller, Peter Grosche, and Josep Ll. Arcos. Unsupervised detection of music boundaries by time series structure features. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1613–1619. Association for the Advancement of Artificial Intelligence, 2012.
- [19] Karen Simonyan and Andrea Vedaldi and Andrew Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *CoRR*, abs/1312.6034, 2013.
- [20] Jordan Bennett Louis Smith, John Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J Stephen Downie. Design and creation of a large-scale database of structural annotations. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 555–560, 2011.
- [21] Douglas Turnbull and Gert Lanckriet. A supervised approach for detecting boundaries in music using difference features and boosting. In *In Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, pages 42–49, 2007.



Oral Session 6  
**Cultures**

This Page Intentionally Left Blank



# TRACKING THE “ODD”: METER INFERENCE IN A CULTURALLY DIVERSE MUSIC CORPUS

**Andre Holzapfel**  
New York University Abu Dhabi  
andre@rhythmos.org

**Florian Krebs**  
Johannes Kepler University  
Florian.Krebs@jku.at

**Ajay Srinivasamurthy**  
Universitat Pompeu Fabra  
ajays.murthy@upf.edu

## ABSTRACT

In this paper, we approach the tasks of beat tracking, down-beat recognition and rhythmic style classification in non-Western music. Our approach is based on a Bayesian model, which infers tempo, downbeats and rhythmic style, from an audio signal. The model can be automatically adapted to rhythmic styles and time signatures. For evaluation, we compiled and annotated a music corpus consisting of eight rhythmic styles from three cultures, containing a variety of meter types. We demonstrate that by adapting the model to specific styles, we can track beats and downbeats in odd meter types like  $9/8$  or  $7/8$  with an accuracy significantly improved over the state of the art. Even if the rhythmic style is not known in advance, a unified model is able to recognize the meter and track the beat with comparable results, providing a novel method for inferring the metrical structure in culturally diverse datasets.

## 1. INTRODUCTION

Musical rhythm subordinated to a meter is a common feature in many music cultures around the world. Meter provides a hierarchical time structure for the rendition and repetition of rhythmic patterns. Though these metrical structures vary considerably across cultures, metrical hierarchies can often be stratified into levels of differing time spans. Two of these levels are, in terminology of Eurogenetic music, referred to as beats, and measures. The *beats* are the pulsation at the perceptually most salient metrical level, and are further grouped into *measures*. The first beat of each measure is called the *downbeat*. Determining the type of the underlying meter, and the alignment between the pulsations at the levels of its hierarchy with music performance recordings – a process we refer to as meter inference – is fundamental to computational rhythm analysis and supports many further tasks, such as music transcription, structural analysis, or similarity estimation.

The automatic annotation of music with different aspects of rhythm is at the focus of numerous studies in Music Information Retrieval (MIR). Müller et al [5] discussed

the estimation of the beat (called beat tracking), and the estimation of higher-level metrical structures such as the measure length. Approaches such as the one presented by Klapuri et al [3] aim at estimating structures at several metrical levels, while being able to differentiate between certain time signatures. In [7] beats and downbeats are estimated simultaneously, given information about the tempo and the meter of a piece. Most of these approaches assume the presence of a regular metrical grid, and work reasonably well for Eurogenetic popular music. However, their adaptation to different rhythmic styles and metrical structures is not straight-forward.

Recently, a Bayesian approach referred to as *bar pointer model* has been presented [11]. It aims at the joint estimation of rhythmic pattern, the tempo, and the exact position in a metrical cycle, by expressing them as hidden variables in a Hidden Markov Model (HMM) [8]. Krebs et al. [4] applied the model to music signals and showed that explicitly modelling rhythmic patterns is useful for meter inference for a dataset of Ballroom dance music.

In this paper, we adapt the observation model of the approach presented in [4] to a collection of music from different cultures: Makam music from Turkey, Cretan music from Greece, and Carnatic music from the south of India. The adaption of observation models was shown to be of advantage in [4, 6], however restricted to the context of Ballroom dance music. Here, we extract rhythmic patterns from culturally more diverse data, and investigate if their inclusion into the model improves the performance of meter inference. Furthermore, we investigate if a unified model can be derived that covers all rhythmic styles and time signatures that are present in the training data.

## 2. MOTIVATION

The music cultures considered in this paper are based on traditions that can be traced back for centuries until the present, and were documented by research in ethnomusicology for decades. Rhythm in two of these cultures, Carnatic and Turkish Makam music, is organized based on potentially long metrical cycles. All three make use of rhythmic styles that deviate audibly from the stylistic paradigms of Eurogenetic popular music. Previous studies on music collections of these styles have shown that the current state of the art performs poorly in beat tracking [2, 9] and the recognition of rhythm class [9]. As suggested in [9], we explore a unified approach for meter inference that can rec-



ognize the rhythmic style of the piece and track the meter at the same time.

The bar pointer model, as described in Section 4, can be adapted to rhythmic styles by extracting possible patterns using small representative downbeat annotated datasets. This way, we can obtain an adapted system for a specific style without recoding and parameter tweaking. We believe that this is an important characteristic for algorithms applied in music discovery and distribution systems for a large and global audience. Through this study, we aim to answer crucial questions: Do we need to differentiate between rhythmic styles in order to track the meter, or is a universal approach sufficient? For instance, can we track a rhythmic style in Indian music using rhythmic patterns derived from Turkish music? Do we need to learn patterns at all? If a particular style description for each style is needed, this has some serious consequences for the scalability of rhythmic similarity and meter inference methods; while we should ideally aim at music discovery systems without an ethnocentric bias, the needed universal analysis methods might come at a high cost given the high diversity in the musics of the world.

### 3. MUSIC CORPORA

In this paper we use a collection of three music corpora which are described in the following.

The corpus of Cretan music consists of 42 full length pieces of Cretan leaping dances. While there are several dances that differ in terms of their steps, the differences in the sound are most noticeable in the melodic content, and we consider all pieces to belong to one rhythmic style. All these dances are usually notated using a 2/4 time signature, and the accompanying rhythmical patterns are usually played on a Cretan lute. While a variety of rhythmic patterns exist, they do not relate to a specific dance and can be assumed to occur in all of the 42 songs in this corpus.

The Turkish corpus is an extended version of the annotated data used in [9]. It includes 82 excerpts of one minute length each, and each piece belongs to one of three rhythm classes that are referred to as *usul* in Turkish Art music. 32 pieces are in the 9/8-usul *Aksak*, 20 pieces in the 10/8-usul *Curcuna*, and 30 samples in the 8/8-usul *Düyek*.

The Carnatic music corpus is a subset of the annotated dataset used in [10]. It includes 118 two minute long excerpts spanning four *tālas* (the rhythmic framework of Carnatic music, consisting of time cycles). There are 30 examples in each of *ādi tāla* (8 beats/cycle), *rūpaka tāla* (3 beats/cycle) and *mishra chāpu tāla* (7 beats/cycle), and 28 examples in *khanda chāpu tāla* (5 beats/cycle).

All excerpts described above were manually annotated with beats and downbeats. Note that for both Indian and Turkish music the cultural definition of the rhythms contain irregular beats. Since the irregular beat sequence is a subset of the (annotated) equidistant pulses, it can be derived easily from the result of a correct meter inference. For further details on meter in Carnatic and Turkish makam music, please refer to [9].

## 4. METER INFERENCE METHOD

### 4.1 Model description

To infer the metrical structure from an audio signal we use the *bar pointer model*, originally proposed in [11] and refined in [4]. In this model we assume that a bar pointer traverses a bar and describe the state of this bar pointer at each audio frame  $k$  by three (*hidden*) variables: tempo, rhythmic pattern, and position inside a bar. These hidden variables can be inferred from an (*observed*) audio signal by using an HMM. An HMM is defined by three quantities: A *transition model* which describes the transitions between the hidden variables, an *observation model* which describes the relation between the hidden states and the observations (i.e., the audio signal), and an *initial distribution* which represents our prior knowledge about the hidden states.

#### 4.1.1 Hidden states

The three hidden variables of the bar pointer model are:

- Rhythm pattern index  $r_k \in \{r_1, r_2, \dots, r_R\}$ , where  $R$  is the number of different rhythmic patterns that we consider to be present in our data. Further, we denote the time signature of each rhythmic pattern by  $\theta(r_k)$  (e.g., 9/8 for *Aksak* patterns). In this paper, we assume that each rhythmic pattern belongs to a rhythmic class, and a rhythm class (e.g., *Aksak*, *Düyek*) can hold several rhythmic patterns. We investigate the optimal number of rhythmic patterns per rhythm class in Section 5.
- Position within a bar  $m_k \in \{1, 2, \dots, M(r_k)\}$ : We subdivide a whole note duration into 1600 discrete, equidistant bar positions and compute the number of positions within a bar with rhythm  $r_k$  by  $M(r_k) = 1600 \cdot \theta(r_k)$  (e.g., a bar with 9/8 meter has  $1600 \cdot 9/8 = 1800$  bar positions).
- Tempo  $n_k \in \{n_{min}(r_k), \dots, n_{max}(r_k)\}$ : The tempo can take on positive integer values, and quantifies the number of bar positions per audio frame. Since we use an audio frame length of  $0.02s$ , this translates to a tempo resolution of  $7.5 (= \frac{60s}{1/4 \cdot 1600 \cdot 0.02s})$  beats per minute (BPM) at the quarter note level. We set the minimum tempo  $n_{min}(r_k)$  and the maximum tempo  $n_{max}(r_k)$  according to the rhythmic pattern  $r_k$ .

#### 4.1.2 Transition model

We use the transition model proposed in [4, 11] with the difference that we allow transitions between rhythmic pattern states within a song as shown in Equation 3. In the following we list the transition probabilities for each of the three variables:

- $P(m_k | m_{k-1}, n_{k-1}, r_{k-1})$  : At time frame  $k$  the bar pointer moves from position  $m_{k-1}$  to  $m_k$  as defined by
 
$$m_k = [(m_{k-1} + n_{k-1} - 1) \bmod (M(r_{k-1}))] + 1. \quad (1)$$
 Whenever the bar pointer crosses a bar border it is reset to 1 (as modeled by the modulo operator).
- $P(n_k | n_{k-1}, r_{k-1})$  : If the tempo  $n_{k-1}$  is inside the allowed tempo range  $\{n_{min}(r_{k-1}), \dots, n_{max}(r_{k-1})\}$ ,

there are three possible transitions: the bar pointer remains at the same tempo, accelerates, or decelerates:

$$P(n_k | n_{k-1}) = \begin{cases} 1 - p_n, & n_k = n_{k-1} \\ \frac{p_n}{2}, & n_k = n_{k-1} + 1 \\ \frac{p_n}{2}, & n_k = n_{k-1} - 1 \end{cases} \quad (2)$$

Transitions to tempi outside the allowed range are assigned a zero probability.  $p_n$  is the probability of a change in tempo per audio frame, and was set to  $p_n = 0.02$ , the tempo ranges ( $n_{min}(r)$ ,  $n_{max}(r)$ ) for each rhythmic pattern are learned from the data (Section 4.2).

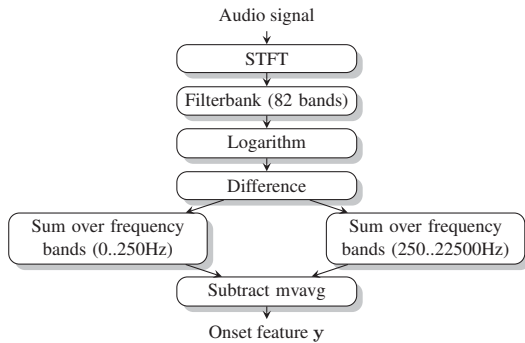
- $P(r_k | r_{k-1})$ : Finally, the rhythmic pattern state is assumed to change only at bar boundaries:

$$P(r_k | r_{k-1}, m_k < m_{k-1}) = p_r(r_{k-1}, r_k) \quad (3)$$

$p_r(r_{k-1}, r_k)$  denotes the probability of a transition from pattern  $r_{k-1}$  to pattern  $r_k$  and will be learned from the training data as described in Section 4.2. In this paper we allow transitions only between patterns of the same rhythm class, which will force the system to assign a piece of music to one of the learned rhythm classes.

#### 4.1.3 Observation model

In this paper, we use the observation model proposed in [4]. As summarized in Figure 1, a Spectral Flux-like onset feature,  $\mathbf{y}$ , is extracted from the audio signal (sampled with 44100 Hz) using the same parameters as in [4]. It summarizes the energy changes that are likely to be related to instrument onsets in two dimensions related to two frequency bands, above and below 250 Hz. In contrast to [4] we removed the normalizing step at the end of the feature computations, which we observed not to influence the results.



**Figure 1:** Computing the onset feature  $\mathbf{y}$  from the audio signal

As described in [4], the observation probabilities  $P(\mathbf{y}_k | m_k, n_k, r_k)$  are modeled by a set of Mixture of Gaussian distributions (GMM). As it is infeasible to specify a GMM for each state (this would result in  $N \times M \times R$  GMMs), we make two assumptions: First, we assume that the observation probabilities are independent of the tempo and second, we assume that the observation probabilities only change each 64th note (which corresponds to  $1600/64=25$  bar positions). Hence, for each rhythmic pattern, we have to specify  $64 \times \theta(r)$  GMMs.

#### 4.1.4 Initial distribution

For each rhythmic pattern, we assume a uniform state distribution within the tempo limits and over all bar positions.

## 4.2 Learning parameters

The parameters of the observation GMMs, the transition probabilities of the rhythm pattern states, and the tempo ranges for each rhythmic style are learned from the data described in Section 3. In our experiments we perform a two-fold cross-validation, excluding those files from the evaluation that were used for parameter learning.

#### 4.2.1 Observation model

The parameters of the observation model consist of the mean values, covariance matrix and the component weights of the GMM for each 64th note of a rhythmic pattern. We determine these as follows:

1. The two-dimensional onset feature  $\mathbf{y}$  (see Section 4.1.3) is computed from the training data.
2. The features are grouped by bar and bar position within the 64th note grid. If there are several feature values for the same bar and 64th note grid point, we compute the average, if there is no feature we interpolate between neighbors. E.g., for a rhythm class which spans a whole note (e.g., *Düyek* (8/8 meter)) this yields a matrix of size  $B \times 128$ , where  $B$  is the number of bars with *Düyek* rhythm class in the dataset.
3. Each dimension of the features is normalized to zero mean and unit variance.
4. For each of the eight rhythm classes in the corpus described in Section 3, a k-means clustering algorithm assigns each bar of the dataset (represented by a point in a 128-dimensional space) to one rhythmic pattern. The influence of the number of clusters  $k$  on the accuracy of the metrical inference will be evaluated in the experiments.
5. For each rhythmic pattern, at all 64th grid points, we compute the parameters of the GMM by maximum likelihood estimation.

#### 4.2.2 Tempo ranges and transition probabilities

For each rhythmic pattern, we compute the minimum and maximum tempo of all bars of the training fold that were assigned to this pattern by the procedure described in Section 4.2.1. In the same way, we determine the transition probabilities  $p_r$  between rhythmic patterns.

## 4.3 Inference

In order to obtain beat-, downbeat-, and rhythmic class estimations, we compute the optimal state sequence  $\{m_{1:K}^*, n_{1:K}^*, r_{1:K}^*\}$  that maximizes the posterior probability of the hidden states given the observations  $y_{1:K}$  and hence fits best to our model and the observations. This is done using the well-known Viterbi algorithm [8].

## 5. EXPERIMENTS

### 5.1 Evaluation metrics

A variety of measures for evaluating beat and downbeat tracking performance are available (see [1] for a detailed overview and descriptions of the metrics listed below)<sup>1</sup>. We chose five metrics that are characterized by a set of diverse properties and are widely used in beat tracking evaluation.

*Fmeas* (F-measure): The F-measure is computed from correctly detected beats within a window of  $\pm 70$  ms by

$$\text{F-measure} = \frac{2pr}{p+r} \quad (4)$$

where  $p$  (*precision*) denotes the ratio between correctly detected beats and all detected beats, and  $r$  (*recall*) denotes the ratio between correctly detected beats and the total number of annotated beats. The range of this measure is from 0% to 100%.

*AMLt* (Allowed Metrical Levels with no continuity required): In this method an estimated beat is counted as correct, if it lies within a small tolerance window around an annotated pulse, and the previous estimated beat lies within the tolerance window around the previous annotated beat. The value of this measure is then the ratio between the number of correctly estimated beats divided by the number of annotated beats (as percentage between 0% and 100%). Beat sequences are also considered as correct if the beats occur on the off-beat, or are double or half of the annotated tempo.

*CMLt* (Correct Metrical Level with no continuity required): The same as *AMLt*, without the tolerance for off-beat, or doubling/halving errors.

*infGain* (Information Gain): Timing errors are calculated between an annotation and all beat estimations within a one-beat length window around the annotation. Then, a beat error histogram is formed from the resulting timing error sequence. A numerical score is derived by measuring the K-L divergence between the observed error histogram and the uniform case. This method gives a measure of how much information the beats provide about the annotations. The range of values for the Information Gain is 0 bits to approximately 5.3 bits in the applied default settings.

*Db-Fmeas* (Downbeat F-measure): For measuring the downbeat tracking performance, we use the same F-measure as defined for beat tracking (using a  $\pm 70$  ms tolerance window).

### 5.2 Results

In Experiment 1, we learned the observation model described in Section 4.2 for various numbers of clusters, separately for each of the eight rhythm classes. Then, we inferred the meter using the HMM described in Section 4.1, again separately for each rhythm class. The results of this experiment indicate how many rhythmic patterns are needed for each class in order to achieve an optimal beat and downbeat tracking with the proposed model.

<sup>1</sup>We used the MATLAB code available at <http://code.soundsoftware.ac.uk/projects/beat-evaluation/> with standard settings.

Tables (1a) to (1h) show the performance with all the evaluation measures for each of the eight styles separately. For Experiment 1 (Ex-1), all significant increases compared to the previous row are emphasized using bold numbers (according to paired-sample t-tests with 5% significance level). In our experiments, increasing the number  $R$  of considered patterns from one to two leads to a statistically significant increase in most cases. Therefore, we can conclude that for tracking these individual styles, more than one pattern is always needed. Further increase to three patterns leads to significant improvement only in the exceptional case of  $\bar{A}$ di tāla, where measure cycles with long durations and rich rhythmic improvisation apparently demand higher number of patterns and cause the system to perform worse than for other classes. Higher numbers than  $R = 3$  patterns never increased any of the metrics significantly. It is important to point out again that a test song was never used to train the rhythmic patterns in the observation model in Experiment 1.

The interesting question we address in Experiment 2 is if the rhythm class of a test song is a necessary information for an accurate meter inference. To this end, we performed meter inference for a test song combining all the determined rhythmic patterns for all classes in one large HMM. This means that in this experiment the HMM can be used to determine the rhythm class of a song, as well as for the tracking of beats and downbeats. We use two patterns from each rhythm class (except  $\bar{a}$ di tāla), the optimally performing number of patterns in Experiment 1, to construct the HMM. For  $\bar{a}$ di tāla, we use three patterns since using 3 patterns improved performance in Experiment 1, to give a total of  $R = 17$  different patterns for the large HMM. The results of Experiment 2 are depicted in the rows labeled Ex-2 in Tables (1a) to (1h), significant change over the optimal setting in Experiment 1 are emphasized using bold numbers. The general conclusion is that the system is capable of a combined task of classification into a rhythm class and the inference of the metrical structure of the signal. The largest and, with the exception of  $\bar{a}$ di tāla, only significant decrease between the Experiment 1 and Experiment 2 can be observed for the downbeat recognition (*Db-Fmeas*). The reason for this is that a confusion of a test song into a wrong class may still lead to a proper tracking of the beat level, but the tracking of the higher metrical level of the downbeat will suffer severely from assigning a piece to a class with a different length of the meter than the test piece.

As described in Section 4.1, we do not allow transitions between different rhythm classes. Therefore, we can classify a piece of music into a rhythm class by evaluating to which rhythmic pattern states  $r_k$  the piece was assigned. The confusion matrix is depicted in Table 2, and it shows that the highest confusion can be observed within certain classes of Carnatic music, while the Cretan leaping dances and the Turkish classes are generally recognized with higher recall rate. The accent patterns in *mishra chāpu* and *khanda chāpu* can be indefinite, non-characteristic and non-indicative in some songs, and hence there is a possi-

	R	Fmeas	CMLt	AMLt	infGain	Db-Fmeas
Ex-1	1	65.9	45.0	57.6	0.89	46.6
	2	<b>91.0</b>	<b>76.6</b>	<b>90.0</b>	<b>1.62</b>	<b>88.6</b>
	3	90.6	77.2	91.1	1.59	86.5
Ex-2	17	85.7	68.7	89.3	1.57	<b>65.1</b>
KL		69.38	41.24	64.60	<u>1.46</u>	-

(a) Turkish Music: Aksak (9/8)

	R	Fmeas	CMLt	AMLt	infGain	Db-Fmeas
Ex-1	1	57.2	33.5	42.2	0.68	37.3
	2	<b>85.2</b>	<b>70.1</b>	<b>82.7</b>	<b>1.51</b>	<b>75.4</b>
	3	83.4	63.3	81.9	1.45	73.7
Ex-2	17	86.6	75.8	87.2	1.64	72.6
KL		70.25	49.52	71.79	<u>1.53</u>	-

(c) Turkish Music: Düyek (8/8)

	R	Fmeas	CMLt	AMLt	infGain	Db-Fmeas
Ex-1	1	49.6	38.9	47.0	0.93	16.5
	2	56.7	44.0	<b>59.5</b>	<b>1.21</b>	<b>32.5</b>
	3	61.6	49.5	<b>65.9</b>	<b>1.40</b>	32.8
Ex-2	17	62.4	40.6	<b>76.7</b>	<b>1.73</b>	<b>21.4</b>
KL		<u>59.42</u>	<u>45.90</u>	64.91	<u>1.53</u>	-

(e) Carnatic music: Ādi (8/8)

	R	Fmeas	CMLt	AMLt	infGain	Db-Fmeas
Ex-1	1	84.1	79.0	79.0	1.54	71.0
	2	<b>93.7</b>	<b>92.2</b>	<b>92.2</b>	<b>2.00</b>	<b>86.4</b>
	3	93.4	91.6	91.6	1.99	89.9
Ex-2	17	90.0	81.6	86.3	1.83	<b>55.0</b>
KL		74.61	44.99	68.71	1.25	-

(g) Carnatic music: Mishra chāpu (7/8)

	R	Fmeas	CMLt	AMLt	infGain	Db-Fmeas
Ex-1	1	71.4	47.8	50.3	0.68	38.9
	2	<b>89.1</b>	<b>75.6</b>	<b>75.6</b>	<b>1.04</b>	48.6
	3	87.7	73.0	73.0	0.99	54.4
Ex-2	17	89.3	74.8	77.5	1.16	41.1
KL		52.77	5.90	59.04	0.77	-

(b) Turkish Music: Curcuna (10/8)

	R	Fmeas	CMLt	AMLt	infGain	Db-Fmeas
Ex-1	1	68.1	60.7	60.8	1.33	59.1
	2	<b>93.0</b>	<b>91.3</b>	<b>91.3</b>	<b>2.25</b>	<b>86.2</b>
	3	92.9	91.0	91.0	2.25	85.8
Ex-2	17	88.8	<b>74.3</b>	92.5	2.24	<b>72.2</b>
KL		35.87	34.42	72.07	1.57	-

(d) Cretan leaping dances (2/4)

	R	Fmeas	CMLt	AMLt	infGain	Db-Fmeas
Ex-1	1	68.2	65.8	71.4	2.04	60.8
	2	<b>82.8</b>	<b>82.5</b>	<b>90.2</b>	<b>2.77</b>	<b>81.9</b>
	3	83.0	82.9	89.5	2.73	80.5
Ex-2	17	77.2	<b>60.6</b>	88.9	<b>2.39</b>	<b>62.0</b>
KL		53.42	29.17	60.37	1.30	-

(f) Carnatic music: Rūpaka (3/4)

	R	Fmeas	CMLt	AMLt	infGain	Db-Fmeas
Ex-1	1	58.9	38.0	41.5	0.70	27.7
	2	<b>94.3</b>	<b>88.9</b>	<b>94.9</b>	<b>2.00</b>	<b>77.3</b>
	3	93.7	88.1	94.3	1.95	78.2
Ex-2	17	<b>90.3</b>	<b>76.0</b>	93.2	2.01	70.6
KL		76.16	57.76	66.34	1.18	-

(h) Carnatic music: Khanda chāpu (5/8)

**Table 1:** Evaluation results for each rhythm class, for Experiment 1 (separate evaluation per style, shown as Ex-1), and Experiment 2 (combined evaluation using one large HMM, shown as Ex-2). The last row in each Table, with row header as KL, shows the beat tracking performance using Klapuri beat tracker. For Ex-1, bold numbers indicate significant change compared to the row above, for Ex-2, bold numbers indicate significant change over the best parameter setting in Ex-1 (bold R parameter), and for KL the only differences to Ex-2 that are not statistically significant are underlined.

bility of confusion between the two styles. Confusion between the three cultures, especially between Turkish and Carnatic is extremely rare, which makes sense due to differences in meter types, performance styles, instrumental timbres, and other aspects which influence the observation model. The recall rates of the rhythm class averaged for each culture are 69.6% for Turkish music, 69.1% for the Cretan music, and 61.02% for Carnatic music. While the datasets are not exactly the same, these numbers represent a clear improvement over the cycle length recognition results depicted in [9] for Carnatic and Turkish music.

Finally, we would like to put the beat tracking accuracies achieved with our model into relation with results obtained with state of the art approaches that do not include an adaption to the rhythm classes. In Table 1, results of the algorithm proposed in [3], which performed generally best among several other approaches, are depicted in the last rows (KL) of each subtable. We underline those results that do not differ significantly from those obtained in Experiment 2. In all other cases the proposed bar pointer model

performs significantly better. The only rhythm class, for which our approach does not achieve an improvement in most metrics is the ādi tāla. As mentioned earlier, this can be attributed to the large variety of patterns and the long cycle durations in ādi tāla.

## 6. CONCLUSIONS

In this paper we adapted the observation model of a Bayesian approach for the inference of meter in music of cultures in Greece, India, and Turkey. It combines the task of determining the type of meter with the alignment of the downbeats and beats to the audio signal. The model is capable of performing the meter recognition with an accuracy that improves over the state of the art, and is at the same time able to achieve for the first time high beat and downbeat tracking accuracies in additive meters like the Turkish Aksak and Carnatic mishra chāpu.

Our results show that increasing the diversity of a corpus means increasing the number of the patterns, *i.e.* a larger

	Turkish			Greek	Carnatic				Recall
	Aksak	Düyek	Curcuna	Cretan	Ādi	Rūpaka	M.chāpu	K.chāpu	
Aksak	21	7	2	2					66
Düyek		23	2	5					77
Curcuna	1	3	13	2				1	65
Cretan	3	5		29	3	2			69
Ādi					14	8	1	7	47
Rūpaka					3	19	1	7	63
M.chāpu					2	1	16	11	53
K.chāpu						4	1	23	82
Precision	84	61	76	76	64	56	84	47	

**Table 2:** Confusion matrix of the style classification of the large HMM (Ex-2). The rows refer to the true style and the columns to the predicted style. The empty blocks are zeros (omitted for clarity of presentation).

amount of model parameters. In the context of the HMM inference scheme applied in this paper this implies an increasingly large hidden-parameter state-space. However, we believe that this large parameter space can be handled by using more efficient inference schemes such as Monte Carlo methods.

Finally, we believe that the adaptability of a music processing system to new, unseen material is an important design aspect. Our results imply that in order to extend meter inference to new styles, at least some amount of human annotation is needed. If there exist music styles where adaptation can be achieved without human input remains an important point for future discussions.

#### Acknowledgments

This work is supported by the Austrian Science Fund (FWF) project Z159, by a Marie Curie Intra-European Fellowship (grant number 328379), and by the European Research Council (grant number 267583).

## 7. REFERENCES

- [1] M. Davies, N. Degara, and M. D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. *Queen Mary University of London, Tech. Rep. C4DM-09-06*, 2009.
- [2] A. Holzapfel and Y. Stylianou. Beat tracking using group delay based onset detection. In *Proceedings of ISMIR - International Conference on Music Information Retrieval*, pages 653–658, 2008.
- [3] A. P. Klapuri, A. J. Eronen, and J. T. Astola. Analysis of the Meter of Acoustic Musical Signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, 2006.
- [4] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat- and downbeat tracking in musical audio. In *Proc. of the 14th International Society for Music Information Retrieval Conference (ISMIR-2013)*, Curitiba, Brazil, nov 2013.
- [5] M. Müller, D. P. W. Ellis, A. Klapuri, G. Richard, and S. Sagayama. Introduction to the Special Issue on Music Signal Processing. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1085–1087, 2011.
- [6] G. Peeters. Template-based estimation of tempo: using unsupervised or supervised learning to create better spectral templates. In *Proc. of the 13th International Conference on Digital Audio Effects (DAFX 2010)*, Graz, Austria, 2010.
- [7] G. Peeters and H. Papadopoulos. Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation. *IEEE Transactions on Audio, Speech and Language Processing*, 19(6):1754–1769, 2011.
- [8] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.
- [9] A. Srinivasamurthy, A. Holzapfel, and X. Serra. In search of automatic rhythm analysis methods for Turkish and Indian art music. *Journal for New Music Research*, 43(1):94–114, 2014.
- [10] A. Srinivasamurthy and X. Serra. A supervised approach to hierarchical metrical cycle tracking from audio music recordings. In *Proc. of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP-2014)*, pages 5237–5241, Florence, Italy, May 2014.
- [11] N. Whiteley, A. Cemgil, and S. Godsill. Bayesian modelling of temporal structure in musical audio. In *Proc. of the 7th International Conference on Music Information Retrieval (ISMIR-2006)*, Victoria, 2006.

# TRANSCRIPTION AND RECOGNITION OF SYLLABLE BASED PERCUSSION PATTERNS: THE CASE OF BEIJING OPERA

Ajay Srinivasamurthy\*   Rafael Caro Repetto\*   Harshavardhan Sundar†   Xavier Serra\*  
 ajays.murthy@upf.edu   rafael.caro@upf.edu   harsha@ece.iisc.ernet.in   xavier.serra@upf.edu

\*Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

†Speech and Audio Group, Indian Institute of Science, Bangalore, India

## ABSTRACT

In many cultures of the world, traditional percussion music uses mnemonic syllables that are representative of the timbres of instruments. These syllables are orally transmitted and often provide a language for percussion in those music cultures. Percussion patterns in these cultures thus have a well defined representation in the form of these syllables, which can be utilized in several computational percussion pattern analysis tasks. We explore a connected word speech recognition based framework that can effectively utilize the syllabic representation for automatic transcription and recognition of audio percussion patterns. In particular, we consider the case of Beijing opera and present a syllable level hidden markov model (HMM) based system for transcription and classification of percussion patterns. The encouraging classification results on a representative dataset of Beijing opera percussion patterns supports our approach and provides further insights on the utility of these syllables for computational description of percussion patterns.

## 1. INTRODUCTION

One common feature in traditional musics is the development of sets of predefined, identifiable melodic and rhythmic patterns. These patterns form a repository of structural elements for the composition or performance of the traditional repertoire. Certain entities of traditional music theory, like melodic modes, rhythmic cycles or musical forms, are instantiated by means of these patterns. The patterns function as key elements for the coordination of different musical elements, like instrumental and vocal sections, and the relationship with other art forms, like dance, theatrical acting, story-telling, etc. For the transmission of such patterns in these mostly oral traditions, particular systems of oral mnemonics have been developed. These systems often share common features across different cultures, so that general principles can be established. Computational analysis of these patterns is an important aspect in Music Information Research (MIR) for such music cultures. Further, their own traditional systems of transmission can offer a solid basis for their modeling.

## 1.1 Syllable based Percussion

Many music traditions around the world have developed particular systems of oral mnemonics for transmission of the repertoire and the technique. David Hughes [7] coined the term *acoustic-iconic mnemonic* systems for these phenomena, and described their use in different genres of traditional Japanese music. As he points out, the core aspect of these systems is that the syllables are chosen for the similarity of their phonetic features with the acoustic properties of the sounds they are representing, establishing an iconic relationship with them. Therefore, these systems are essentially different from those of solmization [6], like for instance the syllables of solfège, of the Indian svaras or the Chinese gongche notation, which are nonsensical in relation to the acoustic phenomena they represent. In this paper, we focus on the oral syllabic systems of mnemonics developed for percussion traditions.

The use of the aforementioned systems for the transmission of percussion is wide extended among traditional musics. David Hughes mentions in his paper, the *shōga* used for the set of drums of *Noh* theatre. In Korea, the young genre of *samul nori*, a percussion quartet of drums and gongs, draws on traditional syllabic mnemonics for the transmission of the repertoire. In the Indian subcontinent, both Hindustani and Carnatic music cultures have developed such oral syllabic systems of mnemonics for the percussion instruments, respectively the *bōls* in the Hindustani tradition, employed mainly by tabla players, and the *solkaṭṭu* in the Carnatic tradition, where the main percussion instrument is the mridangam. The degree of sophistication that these systems have reached in India is such that the rhythmic recitation of the syllables, which requires high skills, are commonly inserted in concerts for musical appreciation. In Carnatic music, this practice has even been consolidated into a specific music form, called *konnakōl*. Furthermore, these systems are also known to be used in Turkish traditional music and Javanese music. In this paper, we explore the use of oral syllabic system developed in the Beijing opera tradition for the computational analysis of its percussion patterns.

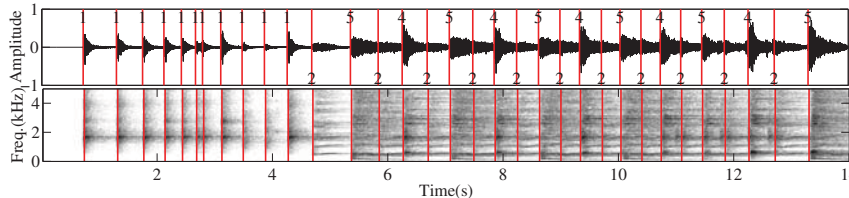
The benefits of using oral syllabic systems from an MIR perspective are both the cultural specificity of the approach and the accuracy of the representation of timbre, articulation and dynamics. The characterization of these percussion traditions need to consider elements that are essential to them such as the richness of their palettes of timbres, subtleties of articulation, and the different degrees and transitions of dynamics, all of which is accurately transmitted



© Ajay Srinivasamurthy, Rafael Caro Repetto, Harshavardhan Sundar, Xavier Serra.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Ajay Srinivasamurthy, Rafael Caro Repetto, Harshavardhan Sundar, Xavier Serra. "Transcription and Recognition of Syllable Based Percussion Patterns: The Case of Beijing Opera", 15th International Society for Music Information Retrieval Conference, 2014.

**Figure 1:** The percussion pattern *shanchui*. The score for each percussion instrument is shown. Shown at the bottom of the score is the syllable sequence for the score (top) and the transcription with the reduced group of syllables used in this paper (bottom). The part of the pattern enclosed between  $\|$  can be repeated many times. The music score is only indicative, with significant variations allowed.



**Figure 2:** An audio example of the pattern *shanchui*. The top panel shows the waveform and the bottom panel is the spectrogram. The vertical lines (in red) mark the onsets of the syllables. The onsets are labeled to indicate the specific syllable group: DA-1, TAI-2, QI-3, QIE-4, and CANG-5 (QI is not present in this pattern).

by the oral syllables.

We explore the use of oral syllables as a means of representation in the MIR tasks of percussion pattern transcription and classification in syllabic percussion systems, considering Beijing opera percussion patterns as a study case. The well defined oral syllabic system and the limited set of percussion patterns make it an ideal choice for a first exploration. Since these syllables have a clear analogy to speech and language, we present a speech recognition based approach to transcribe a percussion pattern into a sequence of syllables. We then use this transcription to classify the sequence into one of the predefined set of patterns that occur in Beijing Opera. We first provide an introduction to percussion patterns in Beijing Opera.

## 1.2 Percussion patterns in Beijing opera

Beijing opera (Jīngjù, 京剧), also called Peking Opera, is one of the most representative genres of Chinese traditional performing arts, integrating theatrical acting with singing and instrumental accompaniment. It is an active art form and exists in the current social and cultural contexts, with a large audience and significant musicological literature. One of the main characteristics of Beijing opera aesthetics is the remarkable rhythmicity that governs the acting overall. From the stylized recitatives to the performers' movements on stage and the sequence of scenes, every element presented is integrated into an overall rhythmic flow. The main element that keeps this rhythmicity is the percussion ensemble, and the main means to fulfill this task is a set of predefined and labeled percussion patterns. Along with the main purpose of keeping the overall rhythmicity of the performance, these patterns have different functions. They signal important structural points in the play. A performance starts and ends with percussion patterns, they generally introduce and conclude arias, and mark transition points within them. They accompany the actors' movements on stage and set the mood of the play, the scene, the aria or a section of the aria. Therefore, the detection and characterization of percussion patterns is a fundamental task for the description of the music dimension in Bei-

jing opera.

The percussion patterns in Beijing Opera music can be defined as sequences of strokes played by different combinations of the percussion instruments, and the resulting variety of timbres are transmitted using oral syllables as mnemonics. The percussion ensemble is formed mainly by five instruments played by four musicians. The *ban* clappers and the *danpigu* drum are played by one single performer, and are therefore known by a conjoint name, *bangu*. The other three instruments are the *xiaoluo* (small gong), the *daluo* (big gong) and the *naobo* (cymbals). *Bangu* has a high pitched drum-like sound while the rest of three instruments are metallophones with distinct timbres<sup>1</sup>. Each of the different sounds that these instruments can produce individually, either through different playing techniques or through different dynamics, as well as the sounds that are produced by a combination of different instruments have an associated syllable that represent them [9]. The syllables and their associated instrument combinations are shown in Table 1. Thus, each percussion pattern is a sequence of syllables in their pre-established order, along with their specific rhythmic structure and dynamic features. A particular feature of the oral syllabic system for Beijing opera percussion that makes it especially interesting is that the syllables that form a pattern refer to the ensemble as a whole, and not to particular instruments. Each particular pattern thus has a single unique syllabic representation shared by all the performers.

In practice, there is a library of limited set of named patterns (called *luógǔ jīng*, 锣鼓经) that are played in a performance, with each of these having a specific role in the arias. These named patterns in the library can be referred to as “pattern classes” for the purpose of classification, and classifying an instance of a pattern occurring in the audio recording of an aria into one of these pattern classes is thus a primary task. Although a definite agreed number for the total number of these patterns is lacking, some estimations,

<sup>1</sup> A few annotated audio examples of these instruments can be found at <http://compmusic.upf.edu/examples-percussion-bo>



like in [9], suggest the existence of around ninety of them.

Figure 1 shows the score for an example pattern *shanchui*. It also shows how a possible transcription in staff notation (adapted from [9]) can be simplified in a single line by the oral syllabic system. Hence, the use of these oral syllabic sequences simplify and unify the representation of these patterns played by an ensemble, making them optimal for the transcription and automatic classification of the patterns. Further, Figure 2 shows an audio example, along with time aligned markers to indicate the syllable onsets. The spectrogram shows the timbral characteristics of the percussion instruments *xiaoluo* (increasing pitch) and *daluo* (decreasing pitch). Some variation to the notated score can also be seen, such as expressive timing and additional insertion of syllables.

Though the patterns are limited in number and predefined, there are several challenges to the problem of percussion pattern transcription and classification. Being an oral tradition, the syllables used for the representation of the patterns lacks full consistency and general agreement. The result being that one particular timbre might be represented by more than one syllable. Furthermore, the syllabic representation conveys information for the conjoint timbre of the ensemble, so only the main structural sounds are represented. In an actual performance, a particular syllable might be performed by different combinations of instruments - e.g. in Figure 1, the first occurrence of the syllable *tái* is played just by the *xiaoluo*, but in the rest of the pattern is played by *xiaoluo* and the *bangu* together. In fact, generally speaking, the strokes of the *bangu* are seldom conveyed in the syllabic sequence (as can be seen in the third measure in Figure 1 for the second sixteenth-note of the *bangu*), except for the introductions and other structural points played by the drum alone. As indicated in Table 1, *cāng* is mostly a combination of all the three metallophones, but in some cases, *cāng* can be played with just the *daluo*, or just the *daluo+naobo* combination. A detailed description and scores for various patterns is available at <http://compmusic.upf.edu/bo-perc-patterns>

Since one of the main functions of the patterns is to accompany the movements of actors on stage, the overall length and the relative duration of each stroke can vary notably, which makes it difficult to set a stable pulse or a definite meter. The time signature and the measure bars used in Figure 1, as suggested in [9], are only indicative and fail to convey the rhythmic flexibility of the pattern. Furthermore, many patterns (such as *shanchui*) accompany scenic movements of undefined duration. In these cases, certain syllable sub-sequences in the pattern are repeated indefinitely, e.g. the audio example in Figure 2 has two additional repetitions of the sub-sequence *cāng-tái-qiē-tái* in the pattern. This causes the same pattern in different performances to have variable lengths, and these repetitions need to be explicitly handled. Finally, although the patterns are usually played in isolation, in many cases the string instruments or even the vocals can start playing before the patterns end, presenting challenges in identification and classification.

### 1.3 Previous work

There is significant MIR literature on percussion transcription [4]. Nakano et al. [10] explored drum pattern retrieval

Syllables	Instruments	Symbol
bā (巴, 八), běn (本), dā (答), dà (大), dōng (冬, 咚), duō (哆), lóng (龙), yī (衣)	bangu	DA
lái (来), tái (台), líng (另)	xiaoluo	TAI
qī (七), pū (扑)	naobo	QI
qiē (切)	naobo+xiaoluo	QIE
cāng (仓), kuāng (匡), kōng (空)	daluo+<naobo> +<xiaoluo>	CANG

**Table 1:** Syllables used in Beijing opera percussion and their grouping used in this paper. Column 2 shows the instrument combination used to produce the syllable, instrument shown between <> is optional. Column 3 shows the symbol we use for the syllable group in this paper.

using vocal percussion, using an HMM based approach. They used onomatopoeia as the internal representation for drum patterns, with a focus on retrieving known fixed sequences from a library of drum patterns with snare and bass drums. Kapur et al. explored query by BeatBoxing [8], aiming to map the BeatBoxing sounds into the corresponding drum sounds. A distinction to be noted here is that in vocal percussion systems such as BeatBoxing, the vocalizations form the music itself, and not a means for transmission as in the case of oral syllables. More recently, Paulus et al. proposed the use of connected HMMs for drum transcription in polyphonic music [12]. This approach is different from what we present in the sense that it aimed to transcribe individual drums (bass, snare, hi-hat) and not overall timbres due to combinations, and no reference to syllabic percussion was made. However all these approaches have indirectly and implicitly used some form of syllabic representations for drum patterns.

Chordia [2] explored the use of tabla *bōls* in transcription of solo tabla sequences. Recently, tabla syllables were used for a predictive model for tabla stroke sequences [3]. Anantapadmanabhan et al. [1] used the syllables of the mridangam in a stroke transcription task. Unlike these works, we address a syllabic system that conveys information for a whole ensemble instead of individual instruments.

Despite the rich musical heritage and the size of audience, little work has been done for computational analysis of Beijing opera from an MIR perspective. It has been studied as a target in some genre classification works [17] and the acoustical properties of Beijing opera singing has been studied [14]. Apart from a recent study [15] that explored the use of Non-negative matrix factorization for onset detection and onset classification into the different percussion instrument classes, no significant work has studied Beijing opera percussion from a computational perspective.

Similar to Nakano et al. [10], we explore a speech recognition based framework in this study. This approach is different to ours in the sense that these onomatopoeic representations were created by the authors, while we are relying on already existing oral traditions. Speech recognition is a well explored research area with many state of the art algorithms and systems [5]. Hence we can apply several available tools and knowledge for computational analysis of syllabic percussion patterns. To the best of our knowl-

edge, this is the first work to explore transcription and classification of syllable based percussion patterns, as applied to Beijing opera.

## 2. PROBLEM FORMULATION

In Beijing opera, several syllables can be mapped to a single timbre. This many-syllable to one-timbre mapping is useful to reduce the syllable space for computational analysis of percussion patterns. We first mapped each syllable to one or several of the instrument categories considered for analysis, as explained in [15], without considering differences in playing technique or dynamics. Based on inputs from expert musicologists, we then grouped the syllables with similar timbres into five syllable groups - DA, TAI, QI, QIE, and CANG, as shown in Table 1. Every individual stroke of the *bangu*, both drum and clappers, have been grouped as DA. In the rest of the syllable groups, the *bangu* can be played simultaneously or not. The single strokes of the *xiaoluo* and the *naobo* are called TAI and QI respectively, and the combined stroke of these two instruments together is the syllable QIE. Finally, any stroke of the *dalu* or any combination that includes *dalu* has been notated as CANG. This mapping to a reduced set of syllable groups is only for the purpose of computational analysis. For the remainder of the paper, we limit ourselves to the reduced set of syllable groups and use them to represent the patterns. For convenience, when it is clear from the context, we call the syllable groups as just syllables, and denote them by the common symbol in column 3 of Table 1. Hence, in the current task, there are five syllable groups. Further, in Beijing opera, the recognition of the pattern as a whole is more important than an accurate syllabic transcription of the pattern. Due to the limited set of pattern classes and owing to all the variations possible in a pattern, we are primarily interested in classifying an audio pattern into one of the possible pattern classes. Syllabic transcription is only considered as an intermediate step towards pattern classification.

We now present a formulation for transcription and recognition of syllable based audio percussion patterns. There is a significant analogy of this task to connected word speech recognition using word models. Syllables are analogous to words and a percussion pattern to a sentence - a sequence of words. There are language rules to form a sentence using a vocabulary, just as each percussion pattern is formed with a defined sequence of syllables from a vocabulary. However unlike in the case of speech recognition where infinitely many sentences are possible, in our case we have a small number of percussion patterns to be recognized.

Consider a set of  $N$  pattern classes  $\mathcal{P} = \{P_1, P_2, \dots, P_N\}$ , each of which is a sequence of syllables from the set of  $M$  syllables  $\mathcal{S} = \{S_1, S_2, \dots, S_M\}$ . So,  $P_k = [s_1, s_2, \dots, s_{L_k}]$  where  $s_i \in \mathcal{S}$  and  $L_k$  is the length of  $P_k$ . Given a test audio pattern  $x[n]$ , the transcription task aims to obtain a syllable sequence  $P^* = [s_1, s_2, \dots, s_{L^*}]$  and the classification task aims to assign  $P^*$  into one of the patterns in the set  $\mathcal{P}$ .

## 3. DATASET

Since there was no available dataset of Beijing opera percussion patterns, we built a representative dataset of patterns from the audio recordings of arias in the CompMusic

Pattern Class	ID	Instances	$\overline{LEN} (\sigma)$
daobantou 【导板头】	1	66	8.70 (1.73)
man changchui 【慢长锤】	2	33	13.99 (4.47)
duotou 【夺头】	3	19	7.18 (1.49)
xiaoluo duotou 【小锣夺头】	4	11	8.16 (2.15)
shanchui 【闪锤】	5	8	10.31 (3.26)
<b>Total</b>		<b>133</b>	<b>9.85 (3.69)</b>

**Table 2:** The Beijing Opera Percussion Pattern (BOPP) dataset. The last column is the mean pattern length and standard deviation in seconds.

Beijing opera research corpus [13], which is a curated collection of arias from commercially available releases spanning many different artists and recording conditions. For this study, we chose only the patterns at the beginning of the aria, which are characteristic and important. From all the pattern classes existing in the corpus, we chose five ( $N = 5$ ) most frequently used ones. These five patterns are also the most widely used and hence hold a high degree of representativeness. The patterns were extracted from the audio recording and assigned to a pattern class by a musicologist. The dataset is described in Table 2 and comprises about 22 minutes of audio with over 2200 syllables in total. The audio samples are stereo recordings sampled at 44.1kHz. The syllabic transcription of each audio pattern is obtained directly from the score of the pattern class it belonged to. Hence the ground truth transcriptions available in the dataset are not time aligned. Since it is a significant effort to obtain time aligned transcriptions, we aim to develop algorithms which do not require the use of time-aligned transcriptions for training. This also ensures that the approaches scale when we add more pattern classes to the dataset. In case of patterns where a sub-sequence of the pattern can be repeated (e.g. *man changchui* and *shanchui*), the additional syllables that occur due to repetitions were manually added by listening to the pattern. Though most of the dataset consists of isolated percussion patterns, there are many audio examples that contain a melodic background apart from the percussion pattern. The dataset is available for research purposes through a central online repository<sup>2</sup>.

## 4. THE APPROACH

The syllables are non-stationary signals and to model their timbral dynamics, we build an HMM for each syllable (analogous to a word-HMM). Using these syllable HMMs and a language model, an input audio pattern is transcribed into a sequence of syllables using Viterbi decoding, and then classified to a pattern class in the library using a measure of distance.

A block diagram of the approach is shown in Figure 3. We first build syllable level HMMs  $\{\lambda_m\}$ ,  $1 \leq m \leq M (= 5)$ , for each syllable  $S_m$  using features extracted from the training audio patterns. We use the MFCC features to model the timbre of the syllables. To capture the temporal dynamics of syllables, we add the velocity and the acceleration coefficients of the MFCC. The stereo audio is converted to mono, since there is no additional information in stereo channels. The 13 dimensional (including the 0<sup>th</sup>

<sup>2</sup> More details at <http://compmusic.upf.edu/bopp-dataset>

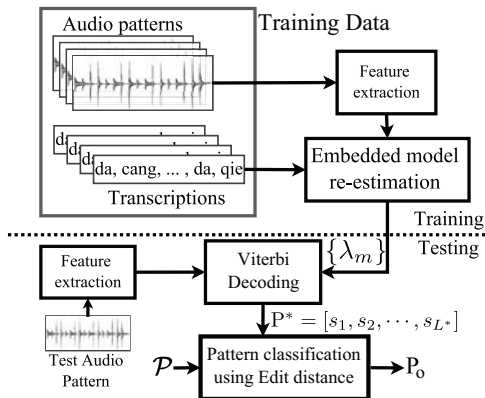


Figure 3: The block diagram of the approach

coefficient) MFCC features are computed from audio patterns with a frame size of 23.2 ms and a shift of 5.8 ms. We also explore the use of energy (as measured by the 0<sup>th</sup> MFCC coefficient) in classification performance. Hence we have two sets of features, MFCC\_0\_D\_A, the 39 dimensional feature including the 0<sup>th</sup>, delta and double-delta coefficients, and MFCC\_D\_A, the 36 dimensional vector without the 0<sup>th</sup> coefficient.

We model each syllable using a 5-state left-to-right HMM including an entry and an exit non-emitting states. The emission densities for each state is modeled with a four component Gaussian Mixture Model (GMM) to capture the timbral variability in syllables. We experimented with eight and sixteen component GMMs, but with little performance improvement. Since we do not have time aligned transcriptions, an isolated HMM training for each syllable is not possible. Hence we use an embedded model Baum-Welch re-estimation to train the HMMs using just the syllable sequence corresponding to each feature sequence. The HMMs are initialized with a flat start using all of the training data. All the experiments were done using the HMM Toolkit (HTK) [16].

For testing, since we only need a rough syllabic transcription independent of the pattern class, we treat the test pattern as a first order time-homogenous discrete Markov chain, which can consist of any finite length sequence of syllables, with uniform unigram and bi-gram (transition) probabilities, i.e.  $p(s_1 = S_i) = 1/M$  and  $p(s_{k+1} = S_j / s_k = S_i) = 1/M$ ,  $1 \leq i, j \leq M$  and with  $k$  being the sequence index. This also forms the language model for forming the percussion patterns using syllables. Given the feature sequence extracted from test audio pattern, we use the HMMs  $\{\lambda_m\}$  to do a Viterbi (forced) alignment, which aims to provide the best sequence of syllables  $P^*$ , given a syllable network constructed from the language model.

Given the decoded syllable sequence  $P^*$ , we compute the string edit distance [11] between  $P^*$  and elements in the set  $\mathcal{P}$ . The use of edit distance is motivated by two factors. First, due to errors in Viterbi alignment,  $P^*$  can have insertions (I), deletions (D), substitutions (B), and transposition (T) of syllables compared to the ground truth. Secondly, to handle the allowed variations in patterns, an edit distance is preferred over an exact match to the sequences in  $\mathcal{P}$ . We explore the use of two different string edit distance measures, Levenshtein distance ( $d_1$ ) that considers I, D, B errors and the Damerau–Levenshtein distance ( $d_2$ )

Feature	Syllable		Pattern	
	C	A	$d_1$	$d_2$
MFCC_D_A	78.14	26.32	93.23	89.47
MFCC_0_D_A	84.98	39.63	91.73	89.47

Table 3: Syllable transcription and Pattern classification performance, with Correctness (C) and Accuracy (A) measures for syllable transcription. Pattern classification results are shown for both distance measures  $d_1$  and  $d_2$ . All values are in percentage.

that considers I, D, B, T errors.

As discussed earlier, there can be repetitions of a sub-sequence in some patterns. Though the number of repetitions is indefinite, we observed in the dataset that there are at most two repetitions in a majority of pattern instances. Hence for the pattern classes that allow repetition of a sub-sequence, we compute the edit distance for the cases of zero, one and two repetitions and then take the minimum distance obtained among the three cases. This way, we can handle repeated parts in a pattern. Finally, the  $P^*$  is assigned to the pattern class  $P_o \in \mathcal{P}$  for which the edit distance  $d$  (either  $d_1$  or  $d_2$ ) is minimum, as in Eqn 1.

$$P_o = \operatorname{argmin}_{1 \leq k \leq N} d(P^*, P_k) \quad (1)$$

## 5. RESULTS AND DISCUSSION

We present the syllable transcription and pattern classification results on the dataset described in Section 3. The results shown in Table 3 are the mean values in a leave-one-out cross validation. We report the syllable transcription performance using the measures of Correctness (C) and Accuracy (A). If  $L$  is the length of the ground truth sequence,  $C = (L - D - B) / L$  and  $A = (L - D - B - I) / L$ . The Correctness measure penalizes deletions and substitutions, while Accuracy measure additionally penalizes insertions too. The pattern classification performance is shown for both edit distance measures  $d_1$  and  $d_2$  in Table 3. All the results are reported for both the features, MFCC\_0\_D\_A and MFCC\_D\_A. The difference in performance between the two features was found to be statistically significant for both Correctness and Accuracy measures in a Mann-Whitney U test at  $p = 0.05$ , assuming an asymptotic normal distribution.

In general, we see a good pattern classification performance while syllable transcription accuracy is poor. We see that MFCC\_0\_D\_A has a better performance with syllable transcription, while both kinds of features provide a comparable performance for pattern classification. Though syllable transcription is not the primary task we focus on, an analysis of its performance provides several insights. The set of percussion instruments in Beijing Opera is fixed, but there can be slight variations across different instruments of the same kind. The training examples are varied and representative, and models built can be presumed to be source independent. Nevertheless, there can be unrepresented syllable timbres in test data leading to a poorer transcription performance. A bigger training dataset can improve the performance in such a case. The energy co-efficient provides significant information about the kind of syllables

ID	1	2	3	4	5	Total
1	100					62
2		93.9			6.1	33
3	10.5		68.4		21.1	19
4			18.2	81.8		11
5		12.5			87.5	8

**Table 4:** The confusion matrix for pattern classification, using the feature MFCC\_0\_D\_A with  $d_1$  distance measure. The rows and column headers represent the True Class and Assigned Class, respectively. Class labels correspond to the ID in Table 2. The last column shows the total examples in each class. All other values are in percentage and the empty blocks are zeros (omitted for clarity).

and hence gives a better syllable transcription performance.

We see that the Correctness is higher than Accuracy showing that the exact sequence of syllables, as indicated in the score was never achieved in a majority of the cases, with several insertion errors. This is due to the combined effect of errors in decoding and allowed variations in patterns. An edit distance based distance measure for classification is quite robust in the present five class problem and provides a good classification performance, despite the low transcription accuracy. Both distance measures provide comparable performance, indicating that the number of transposition errors are low. To see if there are any systematic classification errors, we build a confusion matrix (Table 4) with one of the well performing configurations: MFCC\_0\_D\_A with  $d_1$  distance. We see that *duotou* has a low recall, and gets confused with *shanchui* (ID=5) often. A close examination of the scores showed that a part of the pattern *duotou* is contained within *shanchui*, which explains source of confusion. Such confusions can be handled with better language models, which need further exploration.

## 6. CONCLUSIONS AND SUMMARY

We presented a formulation based on connected-word speech recognition for transcription and classification of syllabic percussion patterns. On a representative collection of Beijing opera percussion patterns, the presented approach provides a good classification performance, despite a simplistic language model and inadequate syllabic transcription accuracy. Though the approach is promising, the evaluation using a small dataset necessitates a further assessment of the generalization capabilities of the proposed approach. We intend to explore better language models that use sequence and rhythmic information more effectively, and extend the task to a much larger dataset spanning more pattern classes. We used isolated patterns in this study, but an automatic segmentation of patterns from audio is a good direction for future work. We also plan to extend this formulation for computational description of percussion patterns in other music cultures such as Hindustani and Carnatic music, which have more complex syllabic percussion systems.

## Acknowledgments

This work is partly supported by the European Research Council under the European Union's Seventh Framework Program, as a part of the CompMusic project (ERC grant agreement 267583)

## 7. REFERENCES

- [1] A. Anantapadmanabhan, A. Bellur, and H. A. Murthy. Modal analysis and transcription of strokes of the mridangam using non-negative matrix factorization. In *Proc. 38th IEEE Proc. Int'l Conf. on Acoust., Speech, and Signal Processing*, pages 181–185, Vancouver, Canada, May 2013.
- [2] P. Chordia. *Automatic Transcription of Solo Tabla Music*. PhD thesis, Stanford University, 2005.
- [3] P. Chordia, A. Sastry, and S. Senturk. Predictive Tabla Modelling Using Variable length Markov and Hidden Markov Models. *Journal of New Music Research*, 40(2):105–118, 2011.
- [4] D. FitzGerald and J. Paulus. Unpitched Percussion Transcription. In A. Klapuri and M. Davy, editors, *Signal Processing Methods for Music Transcription*, pages 131–162. Springer US, 2006.
- [5] X. Huang and L. Deng. An overview of modern speech recognition. In N. Indurkha and F. J. Damerau, editors, *Handbook of Natural Language Processing*, Chapman & Hall/CRC Machine Learning & Pattern Recognition, pages 339–366. Chapman and Hall/CRC, 2nd edition, February 2010.
- [6] A. Hughes and E. Gerson-Kiwi. Solmization. In *Grove music online. Oxford Music Online*. Oxford University Press, accessed July 18, 2014.
- [7] D. Hughes. No nonsense: the logic and power of acoustic-ionic mnemonic systems. *British Journal of Ethnomusicology*, 9(2):93–120, 2000.
- [8] A. Kapur, M. Benning, and G. Tzanetakis. Query by beat-boxing: Music information retrieval for the dj. In *Proc. of the 5th Int'l Conf. on Music Information Retrieval (ISMIR)*, Barcelona, Spain, October 2004.
- [9] W. Mu(穆文义). *Jingju dajiyue jiqiao yu lianxi: yanzou jiaocheng 京剧打击乐技巧与练习: 演奏教程 (Technique and practice of Beijing opera percussion music: a performance course)*. Renmin yinyue chubanshe, Beijing, 2007.
- [10] T. Nakano, J. Ogata, M. Goto, and Y. Hiraga. A drum pattern retrieval method by voice percussion. In *Proc. of the 5th Int'l Conf. on Music Information Retrieval (ISMIR)*, pages 550–553, October 2004.
- [11] G. Navarro. A Guided Tour to Approximate String Matching. *ACM Computing Surveys*, 33(1):31–88, March 2001.
- [12] J. Paulus and A. Klapuri. Drum Sound Detection in Polyphonic Music with Hidden Markov Models. *EURASIP Journal on Audio, Speech, and Music Processing*, 2009(497292):1–9, 2009.
- [13] X. Serra. Creating research corpora for the computational study of music: the case of the compmusic project. In *Proc. of the 53rd AES Int'l Conf. on Semantic Audio*, London, January 2014.
- [14] J. Sundberg, L. Gu, Q. Huang, and P. Huang. Acoustical study of classical Peking Opera singing. *Journal of Voice*, 26(2):137–143, March 2012.
- [15] M. Tian, A. Srinivasamurthy, M. Sandler, and X. Serra. A study of instrument-wise onset detection in beijing opera percussion ensembles. In *Proc. 39th IEEE Proc. Int'l Conf. on Acoust., Speech, and Signal Processing*, pages 2174–2178, Florence, Italy, May 2014.
- [16] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. *The HTK Book, version 3.4*. Cambridge University Engineering Department, Cambridge, UK, 2006.
- [17] Y. Zhang and J. Zhou. A study on content-based music classification. In *Proc. of the Seventh Int'l Symp. on Signal Processing and its Applications*, volume 2, pages 113–116, Paris, France, July 2003.



Oral Session 7

## **Recommendation & Listeners**

This Page Intentionally Left Blank

# TASTE SPACE VERSUS THE WORLD: AN EMBEDDING ANALYSIS OF LISTENING HABITS AND GEOGRAPHY

**Joshua L. Moore, Thorsten Joachims**  
Cornell University, Dept. of Computer Science  
{j1mo|tj}@cs.cornell.edu

**Douglas Turnbull**  
Ithaca College, Dept. of Computer Science  
dturnbull@ithaca.edu

## ABSTRACT

Probabilistic embedding methods provide a principled way of deriving new spatial representations of discrete objects from human interaction data. The resulting assignment of objects to positions in a continuous, low-dimensional space not only provides a compact and accurate predictive model, but also a compact and flexible representation for understanding the data. In this paper, we demonstrate how probabilistic embedding methods reveal the “taste space” in the recently released Million Musical Tweets Dataset (MMTD), and how it transcends geographic space. In particular, by embedding cities around the world along with preferred artists, we are able to distill information about cultural and geographical differences in listening patterns into spatial representations. These representations yield a similarity metric among city pairs, artist pairs, and city-artist pairs, which can then be used to draw conclusions about the similarities and contrasts between taste space and geographic location.

## 1. INTRODUCTION

Embedding methods are a type of machine learning algorithm for distilling large amounts of data about discrete objects into a continuous and semantically meaningful representation. These methods can be applied even when only contextual information about the objects, such as co-occurrence statistics or usage data, is available. For this reason and due to the easy interpretability of the resulting models, embeddings have become popular for tasks in many fields, including natural language processing, information retrieval, and music information retrieval. Recently, embeddings have been shown to be a useful tool for analyzing trends in music listening histories [6].

In this paper, we learn embeddings that give insight into how music preferences relate to geographic and cultural boundaries. Our input data is the *Million Musical Tweets Dataset* (MMTD), which was recently collected and curated by Hauger et al. [3]. This dataset consists of over a million tweets containing track plays and rich geographical information in the form of globe coordinates, which

Hauger et al. have matched to cities and other geographic descriptors as well. Our goal in this work is to use embedding methods to enable a more thorough analysis of geographic and cultural patterns in this data by embedding cities and the artists from track plays in those cities into a joint space. The resulting *taste space* gives us a way to directly measure city/city, city/artist, and artist/artist affinities. After verifying the predictive fidelity of the learned taste space, we explore the surprisingly clear segmentations in taste space across geographic, cultural, and linguistic borders. In particular, we find that the taste space of cities gives us a remarkably clear image of some cultural and linguistic phenomena that transcend geography.

## 2. RELATED WORK

Embeddings methods have been applied to many different modeling and information retrieval tasks. In the field of music IR, these models have been used for tag prediction and song similarity metrics, as in the work of Weston et al. [7]. However, instead of a prediction task such as this, we intend to focus on data analysis tasks. Therefore, we rely on generative models like those proposed in our previous work [5, 6] and by Aizenberg et al [1]. Our prior work uses models which rely on sequences of songs augmented with social tags [5] or per-user song sequences with temporal dynamics [6]. The aim of this work differs from that of our previous work in that we are interested in aggregate global patterns and not in any particular playlist-related task, so we do not adopt the notion of song sequences. We also are concerned with geographic differences in listening patterns, and so we ignore individual users in favor of embedding entire cities into the space.

Aizenberg et al. utilize generative models like those in our work for purposes of building a recommendation engine for music from Internet radio data on the web. However, their work focuses on building a powerful recommendation system using freely available data, and does not focus on the use of the resulting models for data analysis, nor do they concern themselves with geographic data.

The data set which we will use throughout this work was published by Hauger et al. [3]. The authors of this work crawled Twitter for 17 months, looking for tweets which carried certain key words, phrases, or hashtags in order to find posts which signal that a user is listening to a track and for which the text of the tweet could be matched to a particular artist and track. In addition, the data was selected for only tweets with geographical tags (in the form



© Joshua L. Moore, Thorsten Joachims, Douglas Turnbull. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Joshua L. Moore, Thorsten Joachims, Douglas Turnbull. “Taste Space Versus the World: an Embedding Analysis of Listening Habits and Geography”, 15th International Society for Music Information Retrieval Conference, 2014.

of GPS coordinates), and temporal data was retained. The final product is a large data set of geographically and temporally tagged music plays. In their work, the authors emphasize the collection of this impressive data set and a thorough description of the properties of the data set. The authors do add some analyses of the data, but the geographic analysis is limited to only a few examples of coarse patterns found in the data. The primary contribution of our work over the work presented in that paper is to greatly extend the scope of the geographic analysis, presenting a much clearer and more exhaustive view of the differences in musical taste across regions, countries, and languages.

Finally, we describe how geographic information can be useful for various music IR tasks. Knopke [4] also discusses how geospatial data can be exploited for music marketing and musicological research. We use embedding as a tool to further explore these topics. Others, such as Lamere’s *Roadtrip Mixtape*<sup>1</sup> app, have developed systems that use a listener’s location to generate a playlist of relevant music by local artists.

### 3. PROBABILISTIC EMBEDDING MODEL

The embedding model used in this paper is similar to the one used in our previous work [6]. However, the following analysis focuses on geographical patterns instead of temporal dynamics and trends. In particular, we focus on the relationships among cities and artists, and so we elect to condense the geographical information in a tweet down to the city from which it came. Similarly, we discard the track name from each tweet and use only the artist for the song. This leads to a joint embedding of cities and artists.

At the core of the embedding model lies a probabilistic link function that connects the observed data to the underlying semantic space. Intuitively, the link function we use states that the probability  $\Pr(a|c)$  of a given city  $c$  playing a given artist  $a$  is proportional to the distance  $\|X(c) - Y(a)\|_2^2$  between that city and that artist in a Euclidean embedding space of a chosen dimension  $d$ .  $X(c)$  and  $Y(a)$  are the embedding locations of city  $c$  and artist  $a$  respectively. Similar to previous works, we also incorporate a popularity bias term  $p_a$  for each artist to model global popularity. More formally, the probability for a city  $c$  to play an artist  $a$  is:

$$\Pr(a|c) = \frac{\exp(-\|X(c) - Y(a)\|_2^2 + p_a)}{\sum_{a' \in A} \exp(-\|X(c) - Y(a')\|_2^2 + p_{a'})}$$

The sum in the denominator is over the set  $A$  of artists. This sum is known as the *partition function*, denoted  $Z(\cdot)$ , and serves to normalize the distribution over artists.

Determining the embedding locations  $X(c)$  and  $Y(a)$  for all cities and artists (and the popularity terms  $p_a$ ) is the learning problem the embedding method must solve. To fit a model to the data, we maximize the log-likelihood formed by the sum of log-probabilities  $\log(\Pr(a_i|c_i))$ :

$$\begin{aligned} (X, Y, p) &= \max_{X, Y, p} \sum_{(c_i, a_i) \in D} \log(\Pr(a_i|c_i)) \\ &= \max_{X, Y, p} \sum_{(c_i, a_i) \in D} -\|X(c_i) - Y(a_i)\|_2^2 + p_{a_i} - \log(Z(a_i)). \end{aligned}$$

We solve this optimization problem using a Stochastic Gradient Descent approach. First, each embedding vector  $X(\cdot)$  and  $Y(\cdot)$  is randomly initialized to a point in the unit ball in  $\mathbb{R}^d$  (for the chosen dimension  $d$ ). Then, the model parameters are updated in sequential stochastic gradient steps until convergence. The partition function  $Z(\cdot)$  presents an optimization challenge, in that a naïve optimization strategy requires  $O(|A|^2)$  time for each pass over the data. For this work, we used our C++ implementation of the efficient training method employed in [6], an approximate method that estimates the partition function for efficient training. This implementation is available by request, and will later be available on the project website, <http://lme.joachims.org>.

#### 3.1 Interpretation of Embedding Space

As defined above, the model gives us a joint space in which both cities and artists are represented through their respective embedding vectors  $X(\cdot)$  and  $Y(\cdot)$ . Related works have found such embedding spaces to be rich with semantic significance, compactly condensing the patterns present in the training data. Distances in embedding space reveal relationships between objects, and visual or spatial inspection of the resulting models quickly reveals a great deal of segmentation in the space. In particular, joint embeddings yield similarity metrics among the various types of embedded objects, even though individual dimensions in the embedding space have no explicit meaning (e.g. the embeddings are rotation invariant). In our case, this specifically entails the following three measures of similarity:

**City to Artist:** this is the only similarity metric explicitly formulated in the model, and it reflects the distribution  $\Pr(a|c)$  that we directly observe data for. In particular, we directly optimize the positions of cities and artists so that cities have a high probability of listening to artists which they were observed playing in the dataset. This requires placing the city and artist nearby in the embedding space, so proximity in the embedding space can be interpreted as an affinity between a city and an artist.

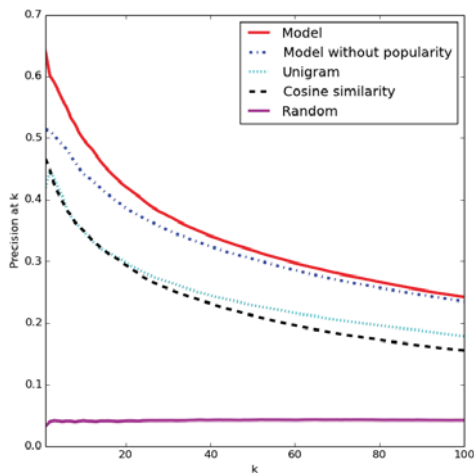
**Artist to Artist:** due to the learned conditional probability distributions’ being constrained by the metric space, two artists which are placed near each other in the space will have a similar probability mass in each city’s distribution. This implies a kind of exchangeability or similarity, since any city which is likely to listen to one artist is likely to listen to the other in the model distribution.

**City to City:** finally, the form of similarity on which we will most rely in this work is that among cities. Again due to the metric space, two nearby cities will assign similar masses to each artist, and so will have very similar distributions over artists in the model. This implies a similarity in musical taste or preferred artists between two cities.

The third type of similarity will form the basis for most of the analyses in this paper. In particular, we are interested

<sup>1</sup> <http://labs.echonest.com/CityServer/roadtrip.html>





**Figure 1:** Precision at  $k$  of our model, a cosine similarity baseline, a tweet count ranking baseline, and a random baseline on a city/artist tweet prediction task.

in the connection between the metric space of cities in the embedding space and another metric space: the one formed by the geographic distribution of cities on the Earth’s surface. As we will see, these two spaces differ greatly, and the taste space of cities gives us a clear image of some cultural and linguistic phenomena that transcend geography.

## 4. EXPERIMENTS

We use the MMTD data set presented by Hauger et al. [3]. This data set contains nearly 1.1 million tweets with geographical data. We pre-process the data by condensing each tweet to a city/artist pair, which results in a city/artist affinity matrix used to train the model. Next, we discard all cities and artists which have not appeared at least 100 times in the data, as well as all cities for which fewer than 30 distinct users tweeted from that city. The post-processed data contains 1,017 distinct cities and 1,499 distinct artists.

For choosing model parameters, we randomly selected 80% of the tweets for the training set, and the remaining 20% for the validation set. This resulted in a training set of 390,077 tweets and a validation set of 97,592 tweets. We used the validation set both to determine stopping criteria for the optimization as well as to choose the initial stochastic gradient step size  $\eta_0$  from the set  $\{0.25, 0.1, 0.05, 0.01\}$  and to evaluate the quality of models of dimension  $\{2, 50, 100\}$ . The optimal step size varied from model to model, but the 100-dimensional model consistently out-performed the others (although the difference between it and the 50-dimensional model was small).

We will analyze the data through the trained embedding models, both through spatial analyses (i.e. nearest neighbor queries and clusterings) and through visual inspection. In general, the high-dimensional model better captures the data, and so we will use it when direct visual inspection is not required. But first, we evaluate the quality of the model through quantitative means.

### 4.1 Quantitative Evaluation of the Model

Before we inspect our model in order to make qualitative claims about the patterns in the data, we first wish to evaluate it on a quantitative basis. This is essential in order to confirm that the model accurately captures the relations among cities and artists, which will offer validation for the conclusions we draw later in the work.

#### 4.1.1 Evaluating Model Fidelity

First, we considered the performance of the model in terms of perplexity, which is a reformulation of the log-likelihood objective outside of a log scale. This is a commonly used measure of performance in other areas of research where models similar to ours are used, such as natural language processing [2]. The perplexity  $p$  is related to the average log-likelihood  $L$  by the transformation  $p = \exp(-L)$ .

Our baseline is the unigram distribution, which assumes that  $\Pr(a|c)$  is directly proportional to the number of tweets artist  $a$  received in the entire data set independent of the city. Estimating the unigram distribution from the training set and using it to calculate the perplexity on the validation set yielded a perplexity of 589 (very similar to the perplexity attained when estimating this distribution from the train set and calculating the perplexity on the train set itself). Our model offered a great improvement over this – the 100-dimensional model yielded a perplexity on the validation set of 290, while the 2-dimensional model reached a perplexity of 357. This improvement suggests that our model has captured a significant amount of useful information from the data.

#### 4.1.2 Evaluating Predictive Accuracy

Second, we created a task to evaluate the predictive power of our model. To this end, we split the data chronologically into two halves, and further divided the first half into a training set and a validation set. Using the first half of the data, we trained a 100-dimensional model. Our goal is to use this model to predict which new artists various cities will begin listening to in the second half of the data.

We accomplish this by considering, for each city, the set of artists which had no observed tweets in that city in the first half of the data. We then sorted these artists by their score in the model – namely, for city  $c$  and artist  $a$ , the function  $-||X(c) - Y(a)||_2^2 + p_a$ . Using this ordering as a ranking function, we calculated the precision at  $k$  of our ranking for various values of  $k$ , where an artist is considered to be relevant if that artist receives at least one tweet from that city in the second half of the data. We average the results of each city’s ranking.

We compare the performance of our model on this task to three baselines. First, we consider a *random* ranking of all the artists which a city has not yet tweeted. Second, we sort the yet untweeted artists by their raw global tweet count in the first half of the data – which we label the *unigram* baseline. Third, we use the raw artist tweet counts for a city’s nearest neighbor city in the first half of data to rank untweeted artists for that city. In this case, the nearest

neighbor is not determined using our embedding but rather based on the maximum *cosine similarity* between the vector of artist tweet counts for the city and the vectors of tweet count for all other cities.

The results can be seen in Figure 1. At  $k = 1$ , our model correctly guesses an artist that a city will later tweet with 64% accuracy, compared to 46% for the cosine similarity, 42% for unigram and around 5% for the random baseline. This advantage is consistent as  $k$  increases, with our method attaining about 24% precision at 100, compared to 18% for unigram and 14% for cosine similarity. We also show the performance of the same model at this task when popularity terms are excluded from the scoring function at ranking time. Interestingly, the performance in this case is still quite good. We see precision at 1 of about 51% in this case, with the gap between this method and the method with popularity terms growing smaller as  $k$  increases. This suggests that proximity in the space is very meaningful, which is an important validation of the analyses to follow. Finally, the good performance on this task invites an application of the space to making marketing predictions – which cities are prone to pick up on which artists in the near future? – but we leave this for future work.

#### 4.2 Visual Inspection of the Embedding Space

In Figure 2 we present plots of the two-dimensional embedding space, with labels for some key cities (left) and artists (right). Note that the two plots are separated by city and artists only for readability, and that all points lie in the same space. In this figure, we can already see a striking segmentation in city space, with extreme distinction between, e.g., Brazilian cities, Southeast Asian cities, and American cities. We can also already see distinct regional and cultural groupings in some ways – the U.S. cities largely form a gradient, with Chicago, Atlanta, Washington, D.C., and Philadelphia in the middle, Cleveland and Detroit on one edge of the cluster, and New York and Los Angeles on the opposite edge. Interestingly, Toronto is also on the edge of the U.S. cluster, and on the same edge where New York and Los Angeles – arguably the most “international” of the U.S. cities shown here – end up.

It is also interesting to note that the space has a very clear segmentation in terms of genre – just as clear as embeddings produced in previous work from songs alone [5] or songs and individual users [6]. Of course, this does not translate into an effective user model – surely there are many users in Recife, Brazil that would quickly tire of a radio station inspired by Linkin Park – but we believe it is still a meaningful phenomenon. Specifically, this suggests that the taste of the average listener can vary dramatically from one city to the next, even within the same country. More surprisingly, this variation in the average user is so dramatic that cities themselves can form nearly as coherent a taste space as individual users, as the genre segmentation is barely any less clear than in other authors’ work with user modeling.

#### 4.3 Higher-dimensional Models

Directly visualizing two-dimensional models can give us striking images from which rough patterns can be easily

gleaned. However, higher dimensional models are able to achieve perplexities on the validation set which far exceed those of lower dimensional models. For example, as mentioned before, our best performing 2-dimensional model attains a validation perplexity of 357, while our best performing 100-dimensional model attains a perplexity of 290 on the validation set. This suggests that higher dimensional models capture more of the nuanced patterns present in the data. On the other hand, simple plotting is no longer sufficient to inspect high-dimensional data – we must resort to alternative methods, for example, clustering and nearest neighbor queries. First, in Figure 3, we present the results of using  $k$ -means clustering in the city space of the 100-dimensional model. The common algorithm for solving the  $k$ -means clustering problem is known to be prone to getting stuck in local optima, and in fact can be difficult to validate properly. We attempted to overcome these problems by using cross validation and repeated random restarts. Specifically, we used 10-fold cross-validation on the set of all cities in order to find a validation objective for each candidate value of  $k$  from 2 to 20. Then, we selected the parameter  $k$  by choosing the largest value for which no larger value offers more than a 5% improvement over the immediately previous value.

Once the value of  $k$  was chosen, we tried to overcome the problem of local optima by running the clustering algorithm 10 times on the entire set of cities with that value of  $k$  and different random initializations, finally choosing the trial with the best objective value. This process resulted in optimal  $k$  values ranging from 6 to 13. Smaller values resulted in some clusterings with granularity too coarse to see interesting patterns, while larger values were noisy and produced unstable clusterings. Ultimately, we found that  $k = 9$  was a good trade-off.

Additionally, in Table 1, we obtain a complementary view of the 100-dimensional embedding by listing the results of nearest-neighbor queries for some well-known, hand-selected cities. These queries give us an alternative perspective of the city space, pointing out similarities that may not be apparent from the clustering alone. By combining these views, we can start to see many interesting patterns arise:

**The French-speaking supercluster:** French-speaking cities form an extremely tight cluster, as can also be seen in the 2-dimensional embedding in Figure 2. Virtually every French city is part of this cluster, as well as French-speaking cities in nearby European countries, such as Brussels and Geneva. Indeed even beyond the top 10 listed in Table 1, almost all of the top 100 nearest neighbors for Paris are French-speaking. Language is almost certainly the biggest factor in this effect, but if we consider the countries near France, we see that despite linguistic divides, in the clustering, many cities in the U.K. still group closely with Dutch cities and even Spanish cities. Furthermore, this grouping can be seen in every view of the data – in the two-dimensional space, the clustering, and the nearest neighbor queries. It should be noted that in our own trials clustering the data, the French cluster is one of the first

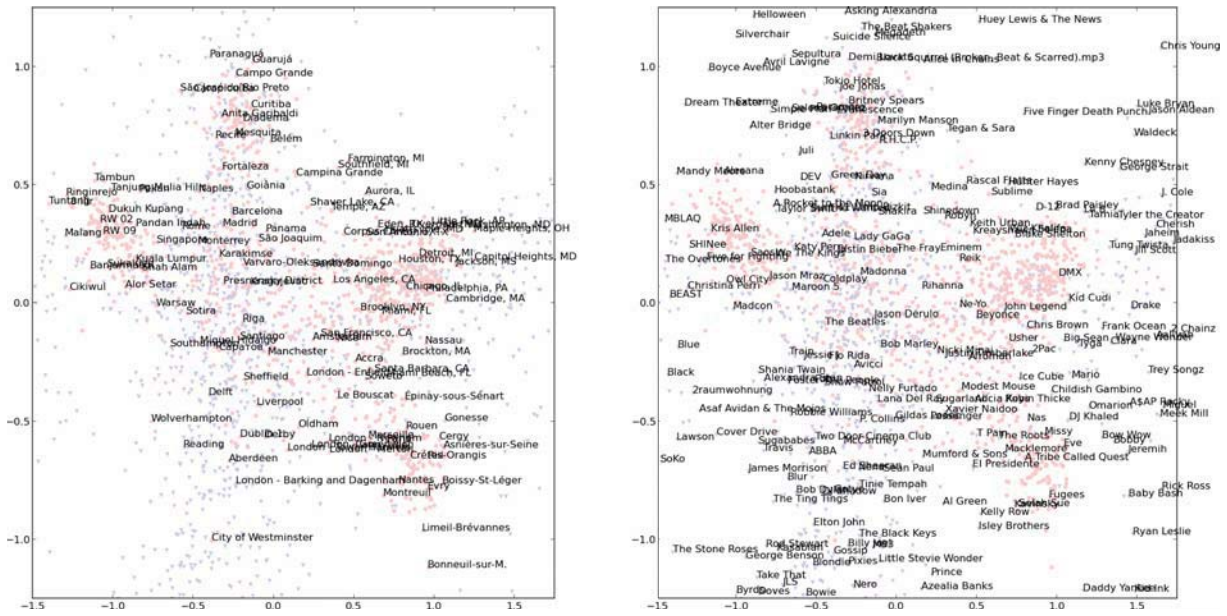


Figure 2: The joint city/artist space with some key cities and artists labeled.

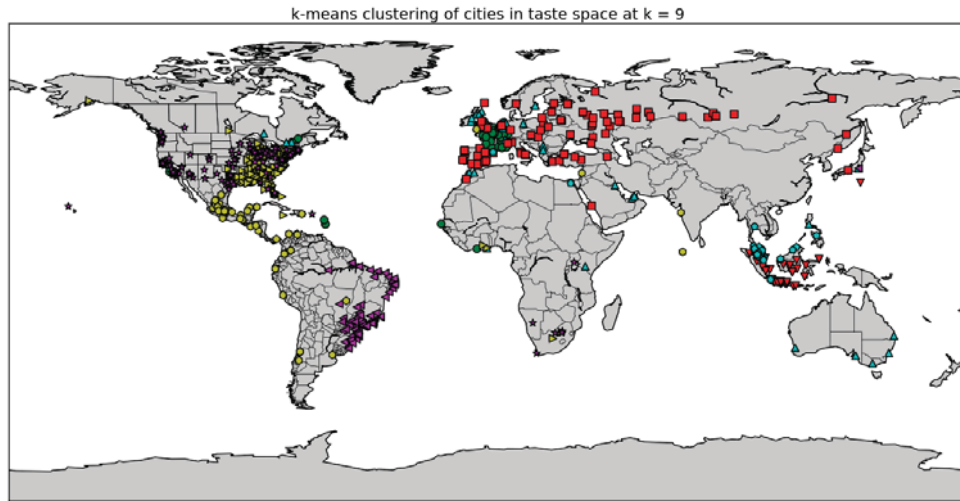


Figure 3: A  $k$ -means clustering of cities around the world with  $k = 9$ .

<b>Kuala Lumpur</b> Kulim Sungai Lembing Ipoh Kuching Sunway City Seremban Seri Kembangan Taman Cheras Hartamas Kuantan Selayang	<b>Paris</b> Boulogne-Billancourt Brussels Rennes Lille Aix-en-Provence Limoges Amiens Marseille Geneva Grenoble	<b>Singapore</b> Hougang Seng Kang USJ9 Subang Kota Bahru Bangkok Alam Damai Kota Padawan Glenmarie Budapest	<b>Los Angeles, CA</b> Grand Prairie, TX Ontario, CA Riverside, CA Sacramento, CA Salinas, CA Paterson, NJ San Bernardino, CA Inglewood, CA Modesto, CA Pomona, CA	<b>Chicago, IL</b> Buffalo, NY Clarksville, TN Cleveland, OH Durham, NC Birmingham, AL Flint, MI Montgomery, AL Nashville, TN Jackson, MS Paterson, NJ	<b>São Paulo</b> Osasco Jundiaí Carapicuíba Ribeirão Pires Shinjuku Vargem Grande Paulista Santa Maria Itapevi Cascavel Embu das Artes
<b>Brooklyn, NY</b> Minneapolis, MN Winston-Salem, NC Arlington, VA Waterbury, CT Washington, DC Syracuse, NY Jersey City, NJ Louisville, KY Tallahassee, FL Ontario, CA	<b>Atlanta, GA</b> Savannah, GA Tallahassee, FL Cleveland, OH Washington, DC Memphis, TN Flint, MI Huntsville, AL Montgomery, AL Jackson, MS Lafayette, LA	<b>Madrid</b> Sevilla Granada Barcelona Murcia Sorocaba Ponta Grossa Huntington Beach, CA Istanbul Vigo Oxford	<b>Amsterdam</b> Eindhoven Tilburg Emmen Nijmegen Enschede Zwolle Amersfoort Maastricht Antwerp Coventry	<b>Sydney</b> Toronto Denver, CO Windhoek Angers Rialto, CA Hamilton Rotterdam Ottawa London - Tower Hamlets London - Southwark	<b>Montréal</b> Montpellier Geneva Raleigh, NC Limoges Angers Ontario, CA Anchorage, AK Nice Lyon Rennes

Table 1: Nearest neighbor query results in 100-dimensional city space. Brooklyn was chosen over New York, NY due to having more tweets in the data set. In addition, only result cities with population at least 100,000 are displayed.

Country	Least typical	Most typical
Brazil	Criciúma, Santa Catarina	Itapevi, São Paulo
Canada	Surrey, BC	Toronto, ON
Netherlands	Leiden	Emmen
Mexico	Campeche, CM	Cuahtémoc, DF
Indonesia	Panunggan Barat	RW 02
France	Bordeaux	Mantes-la-Jolie, Île-de-France
United States	Huntington Beach, CA	Jackson, MS
Malaysia	Kota Damansara	Kuala Lumpur
United Kingdom	Wolverhampton, England	London Borough of Camden
Russia	Ufa	Podgory
Spain	Álora, Andalusia	Barcelona

**Table 2:** Most and least typical cities in taste profile for various countries.

clusters to become apparent, as well as one of the most consistent to appear. We can also see that the French cluster is indeed a linguistic and cultural one which is not just due to geographic proximity: although Montreal has several nearest neighbors in North America, it is present in the French group in the  $k$ -means clustering (as is Quebec City) and is also very close to many French-speaking cities in Europe, such as Geneva and Lyon. We can also see that Abidjan, Ivory Coast joins the French  $k$ -means cluster, as do Dakar in Senegal, Les Abymes in Guadeloupe and Le Lamentin and Fort-de-France in Martinique – all cities in countries which are members of the Francophonie.

**Australia:** Here again, despite the relatively tight geographical proximity of Australia and Southeast Asia, and the geographic isolation of Australia from North America, Australian cities tend to group closely with Canadian cities and some cities in the United Kingdom. One way of seeing this is the fact that Sydney’s nearest neighbors include Toronto, Hamilton, Ontario, Ottawa, and two of London’s boroughs. In addition, other cities in Australia also belong to a cluster that mainly includes cities in the Commonwealth (e.g., U.K., Canada).

**Cultural divides in the United States:** the cities in the U.S. tend to form at least two distinct subgroups in terms of listening patterns. One group contains many cities in the Southeast and Midwest, as well as a few cities on the southern edge of what some might call the Northeast (Philadelphia, for example). The other group consists primarily of cities in the Northeast, on the West Coast, and in the Southwest of the country, including most of the cities in Texas. Intuitively, there are two results that might be surprising to some here. The first is that the listening patterns of Chicago tend to cluster with listening patterns in the South and the rest of the Midwest, and not those of very large cities on the coasts (after all, Chicago is the third-largest city in the country). The second is that Texas groups with the West Coast and Northeast, and not with the Southeast, which would be considered by many to be more culturally similar in many ways.

#### 4.4 Most and least typical cities

We can also consider the relation of individual cities to their member countries. For this analysis, we considered all the countries which have at least 10 cities represented in the data. Then for each country we calculated the average position in embedding space of cities in that country. With this average city position, we can then measure the distance of individual cities from the mean of cities in their country and find the cities which have the most and least

typical taste profiles for that country.

The results are shown in Table 2. We can see a few interesting patterns here. First, in Brazil, the most typical city is an outlying city near São Paulo city, while the least typical is a city in Santa Catarina, the second southernmost state in Brazil, which is also less populous than the southernmost, Rio Grande do Sul, which was also well-represented in the data. In Canada, the least typical city is an edge city on Vancouver’s east side, while the most typical is the largest city, Toronto. In France, the most typical city is in Île-de-France, not too far from Paris. We also see in England that the least typical city is Wolverhampton, and edge city of Birmingham towards England’s industrial north, while the most typical is a borough of London.

## 5. CONCLUSIONS

In this work, we learned probabilistic embeddings of the Million Musical Tweets Dataset, a large corpus of tweets containing track plays which has rich geographical information for each play. Through the use of embeddings, we were able to easily process a large amount of data and sift through it visually and with spatial analysis in order to uncover examples of how musical taste conforms to or transcends geography, language, and culture. Our findings reflect that differences in culture and language, as well as historical affinities among countries otherwise separated by vast distances, can be seen very clearly in the differences in taste among average listeners from one region to the next. More generally, this paper shows how nuanced patterns in large collections of preference data can be condensed into a taste space, which provides a powerful tool for discovering complex relationships. *Acknowledgments:* This work was supported by NSF grants IIS-1217485, IIS-1217686, IIS-1247696, and an NSF Graduate Research Fellowship.

## 6. REFERENCES

- [1] N. Aizenberg, Y. Koren, and O. Somekh. Build your own music recommender by modeling internet radio streams. In *WWW*, pages 1–10. ACM, 2012.
- [2] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *JMLR*, 3:1137–1155, 2003.
- [3] D. Hauger, M. Schedl, A. Košir, and M. Tkalčič. The million musical tweets dataset - what we can learn from microblogs. In *ISMIR*, 2013.
- [4] I. Knopke. Geospatial location of music and sound files for music information retrieval. *ISMIR*, 2005.
- [5] J. L. Moore, S. Chen, T. Joachims, and D. Turnbull. Learning to embed songs and tags for playlist prediction. In *ISMIR*, 2012.
- [6] J. L. Moore, Shuo Chen, T. Joachims, and D. Turnbull. Taste over time: the temporal dynamics of user preferences. In *ISMIR*, 2013.
- [7] J. Weston, S. Bengio, and P. Hamel. Multi-tasking with joint semantic spaces for large-scale music annotation and retrieval. *JNMR*, 40(4):337–348, 2011.

# ENHANCING COLLABORATIVE FILTERING MUSIC RECOMMENDATION BY BALANCING EXPLORATION AND EXPLOITATION

Zhe Xing, Xinxi Wang, Ye Wang

School of Computing, National University of Singapore

{xing-zhe, wangxinxi, wangye}@comp.nus.edu.sg

## ABSTRACT

Collaborative filtering (CF) techniques have shown great success in music recommendation applications. However, traditional collaborative-filtering music recommendation algorithms work in a *greedy* way, invariably recommending songs with the highest predicted user ratings. Such a purely *exploitative* strategy may result in suboptimal performance over the long term. Using a novel reinforcement learning approach, we introduce *exploration* into CF and try to balance between exploration and exploitation. In order to learn users' musical tastes, we use a Bayesian graphical model that takes account of both CF latent factors and recommendation novelty. Moreover, we designed a Bayesian inference algorithm to efficiently estimate the posterior rating distributions. In music recommendation, this is the first attempt to remedy the greedy nature of CF approaches. Results from both simulation experiments and user study show that our proposed approach significantly improves recommendation performance.

## 1. INTRODUCTION

In the field of music recommendation, content-based approaches and collaborative filtering (CF) approaches have been the prevailing recommendation strategies. Content-based algorithms [1, 9] analyze acoustic features of the songs that the user has rated highly in the past and recommend only songs that have high degrees of acoustic similarity. On the other hand, collaborative filtering (CF) algorithms [7, 13] assume that people tend to get good recommendations from someone with similar preferences, and the user's ratings are predicted according to his neighbors' ratings. These two traditional recommendation approaches, however, share a weakness.

Working in a greedy way, they always generate "safe" recommendations by selecting songs with the highest predicted user ratings. Such a purely *exploitative* strategy may result in suboptimal performance over the long term due to the lack of *exploration*. The reason is that user preference

is only estimated based on the current knowledge available in the recommender system. As a result, uncertainty always exists in the predicted user ratings and may give rise to a situation where some of the non-greedy options deemed almost as good as the greedy ones are actually better than them. Without exploration, however, we will never know which ones are better. With the appropriate amount of *exploration*, the recommender system could gain more knowledge about the user's true preferences before *exploiting* them.

Our previous work [12] tried to mitigate the greedy problem in content-based music recommendation, but no work has addressed this problem in the CF context. We thus aim to develop a CF-based music recommendation algorithm that can strike a balance between exploration and exploitation and enhance long-term recommendation performance. To do so, we introduce exploration into collaborative filtering by formulating the music recommendation problem as a reinforcement learning task called *n-armed bandit* problem. A Bayesian graphical model taking account of both collaborative filtering latent factors and recommendation novelty is proposed to learn the user preferences. The lack of efficiency becomes a major challenge, however, when we adopt an off-the-shelf Markov Chain Monte Carlo (MCMC) sampling algorithm for the Bayesian posterior estimation. We are thus prompted to design a much faster sampling algorithm for Bayesian inference. We carried out both simulation experiments and a user study to show the efficiency and effectiveness of the proposed approach. Contributions of this paper are summarized as follows:

- To the best of our knowledge, this is the first work in music recommendation to temper CF's greedy nature by investigating the exploration-exploitation trade-off using a reinforcement learning approach.
- Compared to an off-the-shelf MCMC algorithm, a much more efficient sampling algorithm is proposed to speed up Bayesian posterior estimation.
- Experimental results show that our proposed approach enhances the performance of CF-based music recommendation significantly.

## 2. RELATED WORK

Based on the assumption that people tend to receive good recommendations from others with similar preferences, col-



© Zhe Xing, Xinxi Wang, Ye Wang.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Zhe Xing, Xinxi Wang, Ye Wang. "Enhancing collaborative filtering music recommendation by balancing exploration and exploitation", 15th International Society for Music Information Retrieval Conference, 2014.

laborative filtering (CF) techniques come in two categories: memory-based CF and model-based CF. Memory-based CF algorithms [3, 8] first search for neighbors who have similar rating histories to the target user. Then the target user's ratings can be predicted according to his neighbors' ratings. Model-based CF algorithms [7, 14] use various models and machine learning techniques to discover latent factors that account for the observed ratings.

Our previous work [12] proposed a reinforcement learning approach to balance exploration and exploitation in music recommendation. However, this work is based on a content-based approach. One major drawback of the personalized user rating model is that low-level audio features are used to represent the content of songs. This purely content-based approach is not satisfactory due to the semantic gap between low-level audio features and high-level user preferences. Moreover, it is difficult to determine which underlying acoustic features are effective in music recommendation scenarios, as these features were not originally designed for music recommendation. Another shortcoming is that songs recommended by content-based methods often lack variety, because they are all acoustically similar to each other. Ideally, users should be provided with a range of genres rather than a homogeneous set.

While no work has attempted to address the greedy problem of CF approaches in the music recommendation context, Karimi et al. tried to investigate it in other recommendation applications [4, 5]. However, their active learning approach merely explores items to optimize the prediction accuracy on a pre-determined test set [4]. No attention is paid to the exploration-exploitation trade-off problem. In their other work, the recommendation process is split into two steps [5]. In the exploration step, they select an item that brings maximum change to the user parameters, and then in the exploitation step, they pick the item based on the current parameters. The work takes balancing exploration and exploitation into consideration, but only in an ad hoc way. In addition, their approach is evaluated using only an offline and pre-determined dataset. In the end, their algorithm is not practical for deployment in online recommender systems due to its low efficiency.

### 3. PROPOSED APPROACH

We first present a simple matrix factorization model for collaborative filtering (CF) music recommendation. Then, we point out major limitations of this traditional CF algorithm and describe our proposed approach in detail.

#### 3.1 Matrix Factorization for Collaborative Filtering

Suppose we have  $m$  users and  $n$  songs in the music recommender system. Let  $\mathbf{R} = \{r_{ij}\}_{m \times n}$  denote the user-song rating matrix, where each element  $r_{ij}$  represents the rating of song  $j$  given by user  $i$ .

Matrix factorization characterizes users and songs by vectors of latent factors. Every user is associated with a user feature vector  $\mathbf{u}_i \in \mathbb{R}^f, i = 1, 2, \dots, m$ , and every

song a song feature vector  $\mathbf{v}_j \in \mathbb{R}^f, j = 1, 2, \dots, n$ . For a given song  $j$ ,  $\mathbf{v}_j$  measures the extent to which the song contains the latent factors. For a given user  $i$ ,  $\mathbf{u}_i$  measures the extent to which he likes these latent factors. The user rating can thus be approximated by the inner product of the two vectors:

$$\hat{r}_{ij} = \mathbf{u}_i^T \mathbf{v}_j \quad (1)$$

To learn the latent feature vectors, the system minimizes the following regularized squared error on the training set:

$$\sum_{(i,j) \in I} (r_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2 + \lambda \left( \sum_{i=1}^m n_{u_i} \|\mathbf{u}_i\|^2 + \sum_{j=1}^n n_{v_j} \|\mathbf{v}_j\|^2 \right) \quad (2)$$

where  $I$  is the index set of all known ratings,  $\lambda$  a regularization parameter,  $n_{u_i}$  the number of ratings by user  $i$ , and  $n_{v_j}$  the number of ratings of song  $j$ . We use the *alternating least squares* (ALS) [14] technique to minimize Eq. (2).

However, this traditional CF recommendation approach has two major drawbacks. (I) It fails to take recommendation *novelty* into consideration. For a user, the novelty of a song changes with each listening. (II) It works *greedily*, always recommending songs with the highest predicted mean ratings, while a better approach may be to actively explore a user's preferences rather than to merely exploit available rating information [12]. To address these drawbacks, we propose a reinforcement learning approach to CF-based music recommendation.

#### 3.2 A Reinforcement Learning Approach

Music recommendation is an interactive process. The system repeatedly choose among  $n$  different songs to recommend. After each recommendation, it receives a rating feedback (or *reward*) chosen from an unknown probability distribution, and its goal is to maximize user satisfaction, i.e., the expected total reward, in the long run. Similarly, reinforcement learning explores an environment and takes actions to maximize the cumulative reward. It is thus fitting to treat music recommendation as a well-studied reinforcement learning task called *n-armed bandit*.

The *n-armed bandit* problem assumes a slot machine with  $n$  levers. Pulling a lever generates a payoff from the unknown probability distribution of the lever. The objective is to maximize the expected total payoff over a given number of action selections, say, over 1000 plays.

##### 3.2.1 Modeling User Rating

To address drawback (I) in Section 3.1, we assume that a song's rating is affected by two factors: CF score, the extent to which the user likes the song in terms of each CF latent factor, and novelty score, the dynamically changing novelty of the song.

From Eq. (1), we define the CF score as:

$$U_{CF} = \boldsymbol{\theta}^T \mathbf{v} \quad (3)$$

where vector  $\boldsymbol{\theta}$  indicates the user's preferences for different CF latent factors and  $\mathbf{v}$  is the song feature vector

learned by the ALS CF algorithm. For the novelty score, we adopt the formula used in [12]:

$$U_N = 1 - e^{-t/s} \quad (4)$$

where  $t$  is the time elapsed since when the song was last heard,  $s$  the relative strength of the user's memory, and  $e^{-t/s}$  the well-known forgetting curve. The formula assumes that a song's novelty decreases immediately when listened and gradually recovers with time. (For more details on the novelty definition, please refer to [12].) We thus model the final user rating by combining these two scores:

$$U = U_{CF}U_N = (\boldsymbol{\theta}^T \mathbf{v})(1 - e^{-t/s}) \quad (5)$$

Given the variability in musical taste and memory strength, each user is associated with a pair of parameters  $\boldsymbol{\Omega} = (\boldsymbol{\theta}, s)$ , to be learned from the user's rating history. More technical details will be explained in Section 3.2.2.

Since the predicted user ratings always carry uncertainty, we assume them to be random variables rather than fixed numbers. Let  $R_j$  denote the rating of song  $j$  given by the target user, and  $R_j$  follows an unknown probability distribution. We assume that the expectation of  $R_j$  is the  $U_j$  defined in Eq. (5). Thus, the expected rating of song  $j$  can be estimated as:

$$\mathbb{E}[R_j] = U_j = (\boldsymbol{\theta}^T \mathbf{v}_j)(1 - e^{-t_j/s}) \quad (6)$$

Traditional recommendation strategy will first obtain the  $\mathbf{v}_j$  and  $t_j$  of each song in the system to compute the expected rating using Eq. (6) and then recommend the song with the highest expected rating. We call this a *greedy* recommendation as the system is *exploiting* its current knowledge of the user ratings. By selecting one of the non-greedy recommendations and gathering more user feedback, the system *explores* further and gains more knowledge about the user preferences. A greedy recommendation may maximize the expected reward in the current iteration but would result in suboptimal performance over the long term. This is because several non-greedy recommendations may be deemed nearly as good but come with substantial *variance* (or uncertainty), and it is thus possible that some of them are actually better than the greedy recommendation. Without exploration, however, we will never know which ones they are.

Therefore, to counter the greedy nature of CF (drawback II), we introduce exploration into music recommendation to balance exploitation. To do so, we adopt one of the state-of-the-art algorithms called Bayesian Upper Confidence Bounds (Bayes-UCB) [6]. In Bayes-UCB, the expected reward  $U_j$  is a random variable rather than a fixed value. Given the target user's rating history  $\mathcal{D}$ , the posterior distribution of  $U_j$ , denoted as  $p(U_j|\mathcal{D})$ , needs to be estimated. Then the song with the highest fixed-level quantile value of  $p(U_j|\mathcal{D})$  will be recommended to the target user.

### 3.2.2 Bayesian Graphical Model

To estimate the posterior distribution of  $U$ , we adopt the Bayesian model (Figure 1) used in [12]. The correspond-

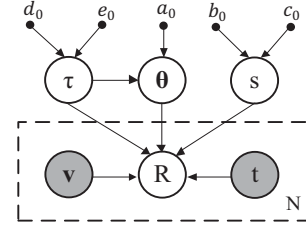


Figure 1: Bayesian Graphical Model.

ing probability dependency is defined as follows:

$$R|\mathbf{v}, t, \boldsymbol{\theta}, s, \sigma^2 \sim \mathcal{N}(\boldsymbol{\theta}^T \mathbf{v}(1 - e^{-t/s}), \sigma^2) \quad (7)$$

$$\boldsymbol{\theta}|\sigma^2 \sim \mathcal{N}(\mathbf{0}, a_0\sigma^2\mathbf{I}) \quad (8)$$

$$s \sim \text{Gamma}(b_0, c_0) \quad (9)$$

$$\tau = 1/\sigma^2 \sim \text{Gamma}(d_0, e_0) \quad (10)$$

$\mathbf{I}$  is the  $f \times f$  identity matrix.  $\mathcal{N}$  represents Gaussian distribution with parameters mean and variance. *Gamma* represents Gamma distribution with parameters shape and rate.  $\boldsymbol{\theta}$ ,  $s$ , and  $\tau$  are parameters.  $a_0$ ,  $b_0$ ,  $c_0$ ,  $d_0$ , and  $e_0$  are hyperparameters of the priors.

At current iteration  $h+1$ , we have gathered  $h$  observed recommendation history  $\mathcal{D}_h = \{(\mathbf{v}_i, t_i, r_i)\}_{i=1}^h$ . Given that each user in our model is described as  $\boldsymbol{\Omega} = (\boldsymbol{\theta}, s)$ , we have according to the Bayes theorem:

$$p(\boldsymbol{\Omega} | \mathcal{D}_h) \propto p(\boldsymbol{\Omega})p(\mathcal{D}_h | \boldsymbol{\Omega}) \quad (11)$$

Then the posterior probability density function (PDF) of the expected rating  $U_j$  of song  $j$  can be estimated as:

$$p(U_j|\mathcal{D}_h) = \int p(U_j|\boldsymbol{\Omega})p(\boldsymbol{\Omega}|\mathcal{D}_h)d\boldsymbol{\Omega} \quad (12)$$

Since Eq. (11) has no closed form solution, we are unable to directly estimate the posterior PDF in Eq. (12). We thus turn to a Markov Chain Monte Carlo (MCMC) algorithm to adequately sample the parameters  $\boldsymbol{\Omega} = (\boldsymbol{\theta}, s)$ . We then substitute every parameter sample into Eq. (6) to obtain a sample of  $U_j$ . Finally, the posterior PDF in Eq. (12) can be approximated by the histogram of the samples of  $U_j$ .

After estimating the posterior PDF of each song's expected rating, we follow the Bayes-UCB approach [6] to recommend song  $j^*$  that maximizes the quantile function:

$$j^* = \arg \max_{j=1, \dots, |S|} Q(\alpha, p(U_j|\mathcal{D}_h)) \quad (13)$$

where  $\alpha = 1 - \frac{1}{h+1}$ ,  $|S|$  is the total number of songs in the recommender system, and the quantile function  $Q$  returns the value  $x$  such that  $\Pr(U_j \leq x|\mathcal{D}_h) = \alpha$ . The pseudo code of our algorithm is presented in Algorithm 1.

### 3.3 Efficient Sampling Algorithm

Bayesian inference is very slow with an off-the-shelf MCMC sampling algorithm because it takes a long time for the Markov chain to converge. In response, we previously proposed an approximate Bayesian model using piecewise linear approximation [12]. However, not only is the original

**Algorithm 1** Exploration-Exploitation Balanced Music Recommendation

---

```

for  $h = 1 \rightarrow N$  do
  if  $h == 1$  then
    Recommend a song randomly;
  else
    Draw samples of  $\theta$  and  $s$  based on  $p(\Omega | \mathcal{D}_{h-1})$ ;
    for song  $j = 1 \rightarrow |S|$  do
      Obtain  $\mathbf{v}_j$  and  $t_j$  of song  $j$  and compute samples of  $U_j$  using Eq. (6);
      Estimate  $p(U_j | \mathcal{D}_{h-1})$  using histogram of the samples of  $U_j$ ;
      Compute quantile  $q_j^h = Q(1 - \frac{1}{h}, p(U_j | \mathcal{D}_{h-1}))$ ;
    end for
    Recommend song  $j^* = \arg \max_{j=1, \dots, |S|} q_j^h$ ;
    Collect user rating  $r_h$  and update  $p(\Omega | \mathcal{D}_h)$ ;
  end if
end for

```

---

Bayesian model altered, tuning the numerous (hyper)parameters is also tedious. In this paper, we present a better way to improve efficiency. Since it is simple to sample from a conditional distribution, we develop a specific Gibbs sampling algorithm to hasten convergence.

Given  $N$  training samples  $\mathcal{D} = \{\mathbf{v}_i, t_i, r_i\}_{i=1}^N$ , the conditional distribution  $p(\theta | \mathcal{D}, \tau, s)$  is still a Gaussian distribution and can be obtained as follows:

$$\begin{aligned}
p(\theta | \mathcal{D}, \tau, s) &\propto p(\tau) p(\theta | \tau) p(s) \prod_{i=1}^N p(r_i | \mathbf{v}_i, t_i, \theta, s, \tau) \\
&\propto p(\theta | \tau) \prod_{i=1}^N p(r_i | \mathbf{v}_i, t_i, \theta, s, \tau) \propto \exp\left(-\frac{1}{2} \theta^T (a_0 \sigma^2 \mathbf{I})^{-1} \theta\right) \\
&\quad \times \exp\left(\sum_{i=1}^N -\frac{1}{2\sigma^2} \left(r_i - \theta^T \mathbf{v}_i (1 - e^{-t_i/s})\right)^2\right) \\
&\propto \exp\left(-\frac{1}{2} \theta^T \Lambda \theta + \boldsymbol{\eta}^T \theta\right) \propto \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad (14)
\end{aligned}$$

where  $\boldsymbol{\mu}$  and  $\boldsymbol{\Sigma}$ , respectively the mean and covariance of the multivariate Gaussian distribution, satisfy:

$$\boldsymbol{\Sigma}^{-1} = \Lambda = \tau \left( \frac{1}{a_0} \mathbf{I} + \sum_{i=1}^N (1 - e^{-t_i/s})^2 \mathbf{v}_i \mathbf{v}_i^T \right) \quad (15)$$

$$\boldsymbol{\mu}^T \boldsymbol{\Sigma}^{-1} = \boldsymbol{\eta}^T = \tau \left( \sum_{i=1}^N r_i (1 - e^{-t_i/s}) \mathbf{v}_i^T \right) \quad (16)$$

Similarly, the conditional distribution  $p(\tau | \mathcal{D}, \theta, s)$  remains a Gamma distribution and can be derived as:

$$\begin{aligned}
p(\tau | \mathcal{D}, \theta, s) &\propto p(\tau) p(\theta | \tau) p(s) \prod_{i=1}^N p(r_i | \mathbf{v}_i, t_i, \theta, s, \tau) \\
&\propto p(\tau) p(\theta | \tau) \prod_{i=1}^N p(r_i | \mathbf{v}_i, t_i, \theta, s, \tau) \\
&\propto \tau^{d_0-1} \exp(-e_0 \tau) \times \exp\left(-\frac{1}{2} \theta^T (a_0 \sigma^2 \mathbf{I})^{-1} \theta\right) \times \\
&\quad \left(\sigma \sqrt{2\pi}\right)^{-N} \exp\left(\sum_{i=1}^N -\frac{1}{2\sigma^2} \left(r_i - \theta^T \mathbf{v}_i (1 - e^{-t_i/s})\right)^2\right) \\
&\propto \tau^{\alpha-1} \exp(-\beta \tau) \propto \text{Gamma}(\alpha, \beta) \quad (17)
\end{aligned}$$

# Users	# Songs	# Observations	% Density
100,000	20,000	20,699,820	1.035%

**Table 1:** Size of the dataset. *Density* is the percentage of entries in the user-song matrix that have observations.

where  $\alpha$  and  $\beta$  are respectively the shape and rate of the Gamma distribution and satisfy:

$$\alpha = d_0 + \frac{f + N}{2} \quad (18)$$

$$\beta = e_0 + \frac{\boldsymbol{\theta}^T \boldsymbol{\theta}}{2a_0} + \frac{1}{2} \sum_{i=1}^N \left( r_i - \boldsymbol{\theta}^T \mathbf{v}_i (1 - e^{-t_i/s}) \right)^2 \quad (19)$$

The conditional distribution  $p(s | \mathcal{D}, \theta, \tau)$  has no closed form expression. We thus adopt the Metropolis-Hastings (MH) algorithm [2] with a proposal distribution  $q(s_{t+1} | s_t) = \mathcal{N}(s_t, 1)$  to draw samples of  $s$ . Our detailed Gibbs sampling process is presented in Algorithm 2.

**Algorithm 2** Gibbs Sampling for Bayesian Inference

---

```

Initialize  $\theta, s, \tau$ ;
for  $t = 1 \rightarrow \text{MaxIteration}$  do
  Sample  $\theta^{(t+1)} \sim p(\theta | \mathcal{D}, \tau^{(t)}, s^{(t)})$ ;
  Sample  $\tau^{(t+1)} \sim p(\tau | \mathcal{D}, \theta^{(t+1)}, s^{(t)})$ ;
   $s_{tmp} = s^{(t)}$ ;
  for  $i = 1 \rightarrow K$  do                                     # MH Step
    Draw  $y \sim \mathcal{N}(s_{tmp}, 1)$ ;
     $\alpha = \min\left(\frac{p(y | \mathcal{D}, \theta^{(t+1)}, \tau^{(t+1)})}{p(s_{tmp} | \mathcal{D}, \theta^{(t+1)}, \tau^{(t+1)})}, 1\right)$ ;
    Draw  $u \sim \text{Uniform}(0, 1)$ ;
    if  $u < \alpha$  then
       $s_{tmp} = y$ ;
    end if
  end for
   $s^{(t+1)} = s_{tmp}$ ;
end for

```

---

## 4. EVALUATION

### 4.1 Dataset

The Taste Profile Subset<sup>1</sup> used in the Million Song Dataset Challenge [10] has over 48 million triplets (user, song, count) describing the listening history of over 1 million users and 380,000 songs. We select 20,000 songs with top listening counts and 100,000 users who have listened to the most songs. Since this collection of listening history is a form of implicit feedback data, we use the approach proposed in [11] to perform negative sampling. The detailed statistics of the final dataset are shown in Table 1.

### 4.2 Learning CF Latent Factors

First, we determine the optimal value of  $\lambda$ , the regularization parameter, and  $f$ , the dimensionality of the latent feature vectors. We randomly split the dataset into three disjoint parts: training set (80%), validation set (10%), and test set (10%). Training set is used to learn the CF latent factors, and the convergence criteria of the ALS algorithm is achieved when the change in root mean square

<sup>1</sup> <http://labrosa.ee.columbia.edu/millionsong/tasteprofile>



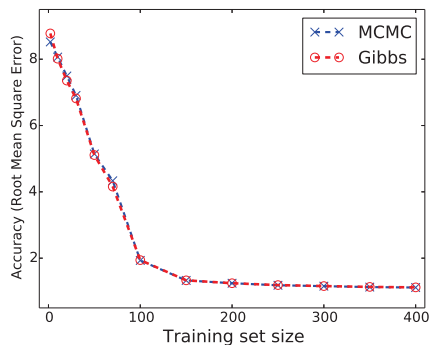


Figure 2: Prediction accuracy of sampling algorithms.

error (RMSE) on the validation set is less than  $10^{-4}$ . Then we use the learned latent factors to predict the ratings on the test set. We first fix  $f = 55$  and vary  $\lambda$  from 0.005 to 0.1; minimal RMSE is achieved at  $\lambda = 0.025$ . We then fix  $\lambda = 0.025$  and vary  $f$  from 10 to 80, and  $f = 75$  yields minimal RMSE. Therefore, we adopt the optimal value  $\lambda = 0.025$  and  $f = 75$  to perform the final ALS CF algorithm and obtain the learned latent feature vector of each song in our dataset. These vectors will later be used for reinforcement learning.

### 4.3 Efficiency Study

To show that our Gibbs sampling algorithm makes Bayesian inference significantly more efficient, we conduct simulation experiments to compare it with an off-the-shelf MCMC algorithm developed in JAGS<sup>2</sup>. We implemented the Gibbs algorithm in C++, which JAGS uses, for a fair comparison.

For each data point  $d_i \in \{(\mathbf{v}_i, t_i, r_i)\}_{i=1}^n$  in the simulation experiments,  $\mathbf{v}_i$  is randomly chosen from the latent feature vectors learned by the ALS CF algorithm.  $t_i$  is randomly sampled from  $uniform(50, 2592000)$ , i.e. between a time gap of 50 seconds and one month.  $r_i$  is calculated using Eq. (6) where elements of  $\theta$  are sampled from  $\mathcal{N}(0, 1)$  and  $s$  from  $uniform(100, 1000)$ .

To determine the burn-in and sample size of the two algorithms and to ensure they draw samples equally effectively, we first check to see if they converge to a similar level. We generate a test set of 300 data points and vary the size of the training set to gauge the prediction accuracy. We set  $K = 5$  in the MH step of our Gibbs algorithm. While our Gibbs algorithm achieves reasonable accuracy with burn-in = 20 and sample size = 100, the MCMC algorithm gives comparable results only when both parameters are 10000. Figure 2 shows their prediction accuracies averaged over 10 trials. With burn-in and sample size determined, we then conduct an efficiency study of the two algorithms. We vary the training set size from 1 to 1000 and record the time they take to finish the sampling process. We use a computer with Intel Core i7-2600 CPU @ 3.40Ghz and 8GB RAM. The efficiency comparison result is shown in Figure 3. We can see that computation time of both two sampling algorithms grows linearly with the training set size. However, our proposed Gibbs sampling algorithm is hundreds of times faster than MCMC,

<sup>2</sup> <http://mcmc-jags.sourceforge.net/>

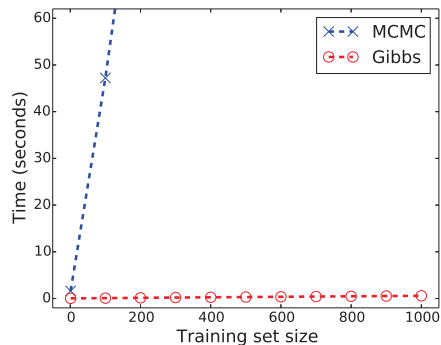


Figure 3: Efficiency comparison of sampling algorithms. ( $Time_{MCMC} = 538.762s$  and  $Time_{Gibbs} = 0.579s$  when  $TrainingSetSize = 1000$ ).

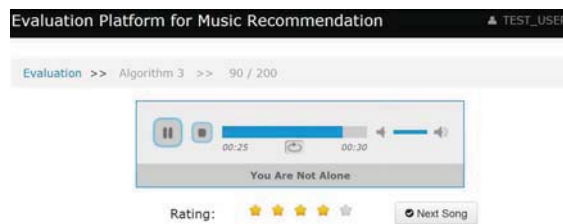


Figure 4: Online evaluation platform.

suggesting that our proposed approach is practical for deployment in online recommender systems.

### 4.4 User Study

In an online user study, we compare the effectiveness of our proposed recommendation algorithm, Bayes-UCB-CF, with that of two baseline algorithms: (1) Greedy algorithm, representing the traditional recommendation strategy without exploration-exploitation trade-off. (2) Bayes-UCB-Content algorithm [12], which also adopts the Bayes-UCB technique but is content-based instead of CF-based. We do not perform offline evaluation because it cannot capture the effect of the elapsed time  $t$  in our rating model and the interactivity of our approach.

Eighteen undergraduate and graduate students (9 females and 9 males, age 19 to 29) are invited to participate in the user study. The subject pool covers a variety of majors of study and nationalities, including American, Chinese, Korean, Malaysian, Singaporean and Iranian. Subjects receive a small payment for their participation. The user study takes place over the course of two weeks in April 2014 on a user evaluation website we constructed (Figure 4). The three algorithms evaluated are randomly assigned to numbers 1-3 to avoid bias. For each algorithm, 200 recommendations are evaluated using a rating scale from 1 to 5. Subjects are reminded to take breaks frequently to avoid fatigue. To minimize the carryover effect, subjects cannot evaluate two different algorithms in one day. For the user study, Bayes-UCB-CF's hyperparameters are set as:  $a_0 = 10$ ,  $b_0 = 3$ ,  $c_0 = 0.01$ ,  $d_0 = 0.001$  and  $e_0 = 0.001$ .

Since maximizing the total expected rating is the main objective of a music recommender system, we thus compare the cumulative average rating of the three algorithms. Figure 5 shows the average rating and standard error of

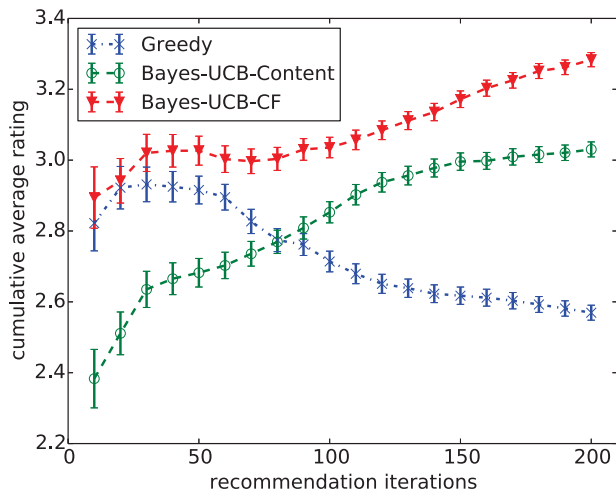


Figure 5: Recommendation performance comparison.

each algorithm from the beginning till the  $n$ -th recommendation iteration. We can see that our proposed Bayes-UCB-CF algorithm significantly outperforms Bayes-UCB-Content, suggesting that the latter still fails to bridge the semantic gap between high-level user preferences and low-level audio features.

T-tests show that Bayes-UCB-CF starts to significantly outperform the Greedy baseline after the 46th iteration ( $p$ -value  $< 0.0472$ ). In fact, Greedy’s performance decays rapidly after the 60th iteration while others continue to improve. Because Greedy solely exploits, it is quickly trapped at a local optima, repeatedly recommending the few songs with initial good ratings. As a result, the novelty of those songs plummets, and users become bored. Greedy will introduce new songs after collecting many low ratings, only to be soon trapped into a new local optima. By contrast, our Bayes-UCB-CF algorithm balances exploration and exploitation and thus significantly improves the recommendation performance.

## 5. CONCLUSION

We present a novel reinforcement learning approach to music recommendation that remedies the greedy nature of the collaborative filtering approaches by balancing exploitation with exploration. A Bayesian graphical model incorporating both the CF latent factors and novelty is used to learn user preferences. We also develop an efficient sampling algorithm to speed up Bayesian inference. In music recommendation, our work is the first attempt to investigate the exploration-exploitation trade-off and to address the greedy problem in CF-based approaches. Results from simulation experiments and user study have shown that our proposed algorithm significantly improves recommendation performance over the long term. To further improve recommendation performance, we plan to deploy a hybrid model that combines content-based and CF-based approaches in the proposed framework.

## 6. ACKNOWLEDGEMENT

We thank the subjects in our user study for their participation. We are also grateful to Haotian “Sam” Fang for proofreading

the manuscript. This project is funded by the National Research Foundation (NRF) and managed through the multi-agency Interactive & Digital Media Programme Office (IDMPO) hosted by the Media Development Authority of Singapore (MDA) under Centre of Social Media Innovations for Communities (COSMIC).

## 7. REFERENCES

- [1] P. Cano, M. Koppenberger, and N. Wack. Content-based music audio recommendation. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 211–212. ACM, 2005.
- [2] S. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.
- [3] J. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual ACM international conference on SIGIR*, pages 230–237. ACM, 1999.
- [4] R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thieme. Active learning for aspect model in recommender systems. In *Symposium on Computational Intelligence and Data Mining*, pages 162–167. IEEE, 2011.
- [5] R. Karimi, C. Freudenthaler, A. Nanopoulos, and L. Schmidt-Thieme. Non-myopic active learning for recommender systems based on matrix factorization. In *International Conference on Information Reuse and Integration*, pages 299–303. IEEE, 2011.
- [6] E. Kaufmann, O. Cappé, and A. Garivier. On bayesian upper confidence bounds for bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pages 592–600, 2012.
- [7] N. Koenigstein, G. Dror, and Y. Koren. Yahoo! music recommendations: modeling music ratings with temporal dynamics and item taxonomy. In *Proceedings of the fifth ACM conference on Recommender systems*, pages 165–172. ACM, 2011.
- [8] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, 1997.
- [9] B. Logan. Music recommendation from song sets. In *ISMIR*, 2004.
- [10] B. McFee, T. Bertin-Mahieux, D. P.W. Ellis, and G. R.G. Lanckriet. The million song dataset challenge. In *Proceedings of international conference companion on World Wide Web*, pages 909–916. ACM, 2012.
- [11] R. Pan, Y. Zhou, B. Cao, N. N. Liu, R. Lukose, M. Scholz, and Q. Yang. One-class collaborative filtering. In *Eighth IEEE International Conference on Data Mining*, pages 502–511. IEEE, 2008.
- [12] X. Wang, Y. Wang, D. Hsu, and Y. Wang. Exploration in interactive personalized music recommendation: A reinforcement learning approach. *arXiv preprint arXiv:1311.6355*, 2013.
- [13] K. Yoshii and M. Goto. Continuous plsi and smoothing techniques for hybrid music recommendation. In *ISMIR*, pages 339–344, 2009.
- [14] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Algorithmic Aspects in Information and Management*, pages 337–348. Springer, 2008.

# IMPROVING MUSIC RECOMMENDER SYSTEMS: WHAT CAN WE LEARN FROM RESEARCH ON MUSIC TASTES?

Audrey Laplante

École de bibliothéconomie et des sciences de l'information, Université de Montréal  
audrey.laplante@umontreal.ca

## ABSTRACT

The success of a music recommender system depends on its ability to predict how much a particular user will like or dislike each item in its catalogue. However, such predictions are difficult to make accurately due to the complex nature of music tastes. In this paper, we review the literature on music tastes from social psychology and sociology of music to identify the correlates of music tastes and to understand how music tastes are formed and evolve through time. Research shows associations between music preferences and a wide variety of sociodemographic and individual characteristics, including personality traits, values, ethnicity, gender, social class, and political orientation. It also reveals the importance of social influences on music tastes, more specifically from family and peers, as well as the central role of music tastes in the construction of personal and social identities. Suggestions for the design of music recommender systems are made based on this literature review.

## 1. INTRODUCTION

The success of a music recommender system (RS) depends on its ability to propose the right music, to the right user, at the right moment. This, however, is an extremely complex task. A wide variety of factors influence the development of music preferences, thus making it difficult for systems to predict how likely a particular user is to like or dislike a piece of music. This probably explains why music RS are often based on collaborative filtering (CF): it allows systems to uncover complex patterns in preferences that would be difficult to model based on musical attributes [1]. However, in order to make those predictions as accurate as possible, these systems need to collect a considerable amount of information about the music preferences of each user. To do so, they elicit explicit feedback from users, inviting them to rate, ban, or love songs, albums, or artists. They also collect implicit feedback, most often in the form of purchase or listening history data (including songs skipped) of individual users. These pieces of information are combined to form the user's music taste profile, which allows the systems to identify like-minded users and to recommend music based on

the taste profiles of these users. One of the principal limitations of RS based on CF is that, before they could gather sufficient information about the preferences of a user, they perform poorly. This corresponds to the well-documented new user cold-start problem.

One way to ease this problem would be to try to enrich the taste profile of a new user by relying on other types of information that are known to be correlated with music preferences. More recently, it has become increasingly common for music RS to encourage users to create a personal profile, or to allow them to connect to the system with a general social network site account (for instance, *Deezer* users can connect with their *Facebook* or their *Google+* account). Music RS thus have access to a wider array of information regarding new users.

Research on music tastes can provide insights into how to take advantage of this information. More than a decade ago, similar reasoning led Uitdenbogerd and Schyndel [2] to review the literature on the subject to identify the factors affecting music tastes. In 2003, however, a paper published by Rentfrow and Gosling [3] on the relationship between music and personality generated a renewed interest for music tastes among researchers, which translated into a sharp increase in research on this topic.

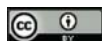
In this paper, we propose to review the recent literature on music preferences from social psychology and sociology of music to identify the correlates of music tastes and to understand how music tastes are formed and evolve through time. We first explain the process by which we identified and selected the articles and books reviewed. We then present the structure and the correlates of music preferences based on the literature review. We conclude with a brief discussion on the implications of these findings for music RS design.

## 2. METHODS

We used two databases to identify the literature on music preferences, one in psychology, *PsycINFO* (Ovid), and one in sociology, *Sociological Abstracts* (ProQuest). We used the thesaurus of each database to find the descriptors that were used to represent the two concepts of interest (i.e., music, preferences), which led to the queries presented in Table 1.

PsycINFO	music AND preferences
Sociological Abstracts:	(music OR "music/musical") AND ("preference/preferences" OR preferences)

**Table 1.** Queries used to retrieve articles in databases



© Audrey Laplante

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Audrey Laplante. "Improving Music Recommender Systems: What Can We Learn from Research on Music Tastes?", 15th International Society for Music Information Retrieval Conference, 2014.

Both searches were limited to the subject heading field. We also limited the search to peer-reviewed publications and to articles published in 1999 or later to focus on the articles published during the last 15 years. This yielded 155 articles in PsycINFO and 38 articles in Sociological Abstracts. Additional articles and books were identified through chaining (i.e., by following citations in retrieved articles), which allowed us to add a few important documents that had been published before 1999. Considering the limited space and the large number of documents on music tastes, further selection was needed. After ensuring that all aspects were covered, we rejected articles with a narrow focus (e.g., articles focusing on a specific music genre or personality trait). For topics on which there were several publications, we retained articles with the highest number to citations based on Google Scholar. We also decided to exclude articles on the relationship between music preferences and the functions of music to concentrate on individual characteristics.

### 3. REVIEW OF LITERATURE ON MUSIC TASTES

Research shows that people, especially adolescents, use their music tastes as a social badge through which they convey who they are, or rather how they would like to be perceived [4, 5]. This indicates that people consider that music preferences reflect personality, values, and beliefs. In the same line, people often make inferences about the personality of others based on their music preferences, as revealed by a study in which music was found to be the main topic of conversation between two young adults who are given the task of getting to know each other [6]. The same study showed that these inferences are often accurate: people can correctly infer several psychological characteristics based on one's music preferences, which suggests that they have an intuitive knowledge of the relationships that exist between music preferences and personality. Several researchers have studied these relationships systematically to identify the correlates of music preferences that pertain to personality and demographic characteristics, values and beliefs, and social influences and stratification.

#### 3.1 Dimensions of music tastes

There are numerous music genres and subgenres. However, as mentioned in [7], attitudes toward genres are not isolated from one another: there are genres that seem to go together while others seem to oppose. Therefore, to reduce the number of variables, prior to attempting to identify the correlates of music preferences, most researchers start by examining the nature of music preferences to identify the principal dimensions. The approach of Rentfrow and Gosling [3] is representative of the work of several researchers. To uncover the underlying structure of music preferences, they first asked 1,704 students from an American university to indicate their liking of 14 different music genres using a 7-point Likert scale. This questionnaire was called the Short Test of Music Preferences (STOMP). They then performed factor analysis by means of principal-components analysis

with varimax rotation on participants' ratings. This allowed them to uncover a factor structure of music preferences, composed of four dimensions, which they labeled *Reflective and Complex*, *Intense and Rebellious*, *Upbeat and Conventional*, and *Energetic and Rhythmic*. Table 2 shows the genres most strongly associated with each dimension. To verify the generalizability of this structure across samples, they replicated the study with 1,384 students of the same university, and examined the music libraries of individual users in a peer-to-peer music service. This allowed them to confirm the robustness of the model.

Music-preference dimension	Genres most strongly associated
Reflective and Complex	Blues, Jazz, Classical, Folk
Intense and Rebellious	Rock, Alternative, Heavy metal
Upbeat and Conventional	Country, Sound tracks, Religious, Pop
Energetic and Rhythmic	Rap/hip-hop, Soul/funk, Electronica/dance

**Table 2.** Music-preference dimensions of Rentfrow and Gosling (2003).

Several other researchers replicated Rentfrow and Gosling's study with other populations and slightly different methodologies. To name a few, [8] surveyed 2,334 Dutch adolescents aged 12–19; [9] surveyed 268 Japanese college students; [10, 11] surveyed 422 and 170 German students, respectively; and [12] surveyed 358 Canadian students. Although there is a considerable degree of similarity in the results across these studies, there also appears to be a few inconsistencies. Firstly, the number of factors varies: while 4 studies revealed a 4-factor structure [3, 8-10], one found 5 factors [11], and another, 9 factors<sup>1</sup> [12]. These differences could potentially be explained by the fact that researchers used different music preference tests: the selection of the genres to include in these tests depends on the listening habits of the target population and thus needs to be adapted. The grouping of genres also varies. In the 4 above-mentioned studies in which a 4-factor structure was found, rock and metal music were consistently grouped together. However, techno/electronic was not always grouped with the same genres: while it was grouped with rap, hip-hop, and soul music in 3 studies, it was grouped with popular music in the study with the Dutch sample [8]. Similarly, religious music was paired with popular music in Rentfrow and Gosling's study, but was paired with classical and jazz music in the 3 other studies. These discrepancies could come from the fact that some music genres might have different connotations in different cultures. It can also be added that music genres are problematic in themselves: they are broad, inconsistent, and ill-defined. To

<sup>1</sup> For this study, the researchers started with 30 genres, as opposed to others who used between 11 and 21 genres.

solve these problems, Rentfrow and colleagues [13, 14] replicated the study yet again, but used 52 music excerpts representing 26 different genres to measure music preferences instead of a list of music genres. The resulting structure was slightly different. It was composed of 5 factors labeled *Mellow*, *Unpretentious*, *Sophisticated*, *Intense*, and *Contemporary* (MUSIC). This approach also allowed them to examine the ties between the factors and the musical attributes. To do so, they asked non-experts to rate each music excerpt according to various attributes (i.e., auditory features, affect, energy level, perceived complexity) and used this information to identify the musical attributes that were more strongly associated with each factor.

### 3.2 Personality traits

Several researchers have examined the relationship between music preferences and personality traits [3, 8-10] using the 5-factor model of personality, commonly called the “Big Five” dimensions of personality (i.e., Extraversion, Emotional Stability, Agreeableness, Conscientiousness, and Openness to Experience). Rentfrow and Gosling [3] were the first to conduct a large-scale study focusing on this aspect, involving more than 3,000 participants. In addition to taking the STOMP test for measuring their music preferences, participants had to complete 6 personality tests, including the Big Five Inventory. The analysis of the results revealed associations between some personality traits and the 4 dimensions of music preferences. For instance, they found that liking *Reflective and Complex* music (e.g., classical, jazz) or *Intense and Rebellious* music (e.g., rock, metal) was positively related to Openness to Experience; and liking *Upbeat and Conventional* music (e.g., popular music) or *Energetic and Rhythmic* music (e.g., rap, hip-hop) was positively correlated with extraversion. Emotional Stability was the only personality dimension that had no significant correlation with any of the music-preference dimensions. Openness and Extraversion were the best predictors of music preferences.

As mentioned previously, since researchers use different genres and thus find different music-preference dimensions, comparing results from various studies is problematic. Nonetheless, subsequent studies seem to confirm most of Rentfrow and Gosling’s findings. Delsing et al. [8] studied Dutch adolescents and found a similar pattern of associations between personality and music preferences dimensions. Only two correlations did not match. However, it should be noted that the correlations were generally lower, a disparity the authors attribute to the age difference between the two samples (college student vs. adolescents): adolescents being more influenced than young adults by their peers, personality might have a lesser effect on their music preferences. Brown [9] found fewer significant correlations when studying Japanese university students. The strongest correlations concerned Openness, which was positively associated with liking *Reflective and Complex* music (e.g., classical, jazz) and negatively related to liking *Energetic and Rhythmic* music (e.g., hip-hop/rap). The positive correlation between

*Energetic and Rhythmic* music and Extraversion, which was found in most other studies [3, 8, 10], was not found with the Japanese sample.

### 3.3 Values and Beliefs

Fewer recent studies have focused on the relationship between music preferences and values or beliefs compared to personality. Nevertheless, several correlates of music preferences were found in this area, from political orientation to religion to vegetarianism [15].

#### 3.3.1 Political Orientation

In the 1980s, Peterson and Christenson [16] surveyed 259 American university students on their music preferences and political orientation. They found that liberalism was positively associated with liking jazz, reggae, soul, or hardcore punk, whereas linking 70s rock or 80s rock was negatively related to liberalism. They also uncovered a relationship between heavy metal and political alienation: heavy metal fans were significantly more likely than others to check off the “Don’t know/don’t care” box in response to the question about their political orientation. More recently, Rentfrow and Gosling [3] found that political conservatism was positively associated with liking *Upbeat & Conventional* music (e.g., popular music), whereas political liberalism was positively associated with liking *Energetic and Rhythmic* (e.g., rap, hip-hop) or *Reflective and Complex* (e.g., classical, jazz) music, although the last two correlations were weak. North and Hargreaves [15], who surveyed 2,532 British individuals, and Gardikiotis and Baltzis [17], who surveyed 606 Greek college students, also found that people who liked classical music, opera, and blues were more likely to have liberal, pro-social beliefs (e.g., public health care, protection of the environment, taking care of the most vulnerable). In contrast, fans of hip-hop, dance, and DJ-based music were found to be among the least likely groups to hold liberal beliefs (e.g., increased taxation to pay for public services, public health care) [15]. As we can see, liking jazz and classical music was consistently associated with liberalism, but no such clear patterns of associations emerged for other music genres, which suggests that further research is needed.

#### 3.3.2 Religious Beliefs

There are very few studies that examined the link between music preferences and religion. The only recent one we could find was the study by North and Hargreaves previously mentioned [15]. Their analysis revealed that fans of western, classical music, disco, and musicals were the most likely to be religious; whereas fans of dance, indie, or DJ-based music were least likely to be religious. They also found a significant relation between music preferences and the religion affiliation of people. Fans of rock, musicals, or adult pop were more likely to be Protestant; fans of opera or country/western were more likely to be Catholic; and fans of R&B and hip-hop/rap were more likely to adhere to other religions. Another older study used the 1993 General Social Survey to examine the attitude of American adults towards heavy metal and rap music and found that people who attended religious services were more likely to dislike heavy metal

(no such association was found with rap music) [18]. Considering that religious beliefs vary across cultures, further studies are needed to discern a clear pattern of associations between music preferences and religion.

### 3.4 Demographic Variables

#### 3.4.1 Gender

Several studies have revealed associations between gender and music tastes. It was found that women were more likely to be fans of chart pop or other types of easy listening music (e.g., country) [7, 12, 15, 19, 20], whereas men were more likely to prefer rock and heavy metal [12, 15, 19, 20]. This is not to say that women do not like rock: in Colley's study [19], which focused on gender differences in music tastes, rock was the second most highly rated music genre among women: the average rating for women was 4.1 (on an 8-point scale from 0 to 7) vs. 4.8 for men. There was, however, a much greater gap in the attitudes towards popular music between men and women, who attributed 3.17 and 4.62 on average, respectively. This was the genre for which gender difference was the most pronounced. Lastly, it is worth mentioning that most studies did not find any significant gender difference for rap [7, 12, 19], which indicates that music in this category appeals to both sexes. This is a surprising result considering the misogynistic message conveyed by many rap songs. Christenson and Roberts [7] speculated that this could be due to the fact that men appreciate rap for its subversive lyrics while women appreciate it for its danceability.

#### 3.4.2 Race and Ethnicity

Very few studies have examined the ties between music preferences and race and ethnicity. In the 1970s, a survey of 919 American college students revealed that, among the demographic characteristics, race was the strongest predictor of music preferences [21]. In a book published in 1998 [7], Christenson and Roberts affirmed that racial and ethnic origins of fans of a music genre mirror those of its musicians. To support their affirmation, they reported the results of a survey of adolescents conducted in the 1990s by Carol Dykers in which 7% of black adolescents reported rap as their favourite music genre, compared with 13% of white adolescents. On the other hand, 25% of white adolescents indicated either rock or heavy metal as their favourite genre, whereas these two genres had been only mentioned by a very small number of black adolescents (less than 5% for heavy metal). North & Hargreaves [15] also found a significant relationship between ethnic background and music preferences. This study was conducted more recently (in 2007), with British adults, and with a more diversified sample in terms of ethnic origins. Interestingly, they found that a high proportion of the respondents who were from an Asian background liked R&B, dance, and hip-hop/rap, which seems to challenge Christenson and Roberts' affirmation. [22] who studied 3,393 Canadian adolescents, performed a cluster analysis to group respondents according to their music preferences. They then examined the correlates of each music-taste cluster. The analysis revealed a different ethnic composition for different clusters. For instance, the

*Black Stylists* cluster was composed of fans of hip-hop and reggae who were largely black, with some South Asian representation. By contrast, the *Hard Rockers*, who like heavy metal and alternative music, were almost exclusively white.

#### 3.4.3 Age

Most researchers who study music preferences draw their participants from the student population of the university where they work. As a result, samples are mostly homogenous in terms of age, which explains the small number of studies that focused on the relationship between age and music preferences. Age was found to be significantly associated with music preferences. For instance, [23] compared the music preferences of different age groups and found that there were only two genres—rock and country—that appeared in the five most highly rated genres of both the 18-24 year olds and the 55-64 year olds. While the favourite genres of younger adults were rap, metal, rock, country, and blues; older adults preferred gospel, country, mood/easy listening, rock, and classical/chamber music. [15] also found a correlation between age and preferences for certain music genres. Unsurprisingly, their analysis revealed that people who liked what could be considered trendy music genres (e.g., hip-hop/rap, DJ-based music, dance, indie, chart pop) were more likely to be young, whereas people who liked more conventional music genres (e.g., classical music, sixties pop, musicals, country) were more likely to be older. [24] conducted a study involving more than 250,000 participants and found that the interest for music genres associated with the *Intense* (e.g., rock, heavy metal, punk) and the *Contemporary* (e.g., rap, funk, reggae) music-preference dimensions decreases with age, whereas the interest for music genres associated with the *Unpretentious* (e.g., pop, country) and the *Sophisticated* (e.g., classical, folk, jazz) dimensions increases.

Some researchers have also looked at the trajectory of music tastes. Studies on the music preferences of children and adolescents revealed that as they get older, adolescents tend to move away from mainstream rock and pop, although these genres remain popular throughout adolescence [7]. Research has also demonstrated that music tastes are already fairly stable in early adolescence and further crystallize in late adolescence or early adulthood [25, 26]. Using data from the American national *Survey of Public Participation in the Arts* (SPPA) of 1982, 1992, and 2002, [23] examined the relationship between age and music tastes, with a focus on older age. They looked at the number of genres liked per age group and found that in young adulthood, people had fairly narrow tastes. Their tastes expand into middle age (i.e., 55 year old), to then narrow again, suggesting that people disengage from music in older age. They also found that although music genres that are popular among younger adults change from generation to generation; they remain much more stable among older people.

#### 3.4.4 Education

Education was also found to be significantly correlated to music preferences. [15] found that individuals who held a master's degree or a Ph.D. were most likely to like opera,

jazz, classical music, or blues; whereas fans of country, musicals, or 1960s pop were most likely to have a lower level of education. [27] studied 325 adolescents and their parents and also found an association between higher education and a taste for classical and jazz music. Parents with lower education were more likely to like popular music and to dislike classical and jazz music.

### 3.5 Social influences

As mentioned before, research established that people use their music preferences as a social badge that conveys information about their personality, values, and beliefs. But music does not only play a role in the construction of personal identity. It is also important to social identity. Music preferences can also act as a social badge that indicates membership in a social group or a social class.

#### 3.5.1 Peers and Parents

Considering the importance adolescents ascribe to both friendship and music, it is not surprising to learn that social groups often identify with music subcultures during adolescence [4]. Therefore, it seems legitimate to posit that in the process of forming their social identity, adolescents may adopt music preferences similar to that of other members of the social group to which they belong or they aspire to belong. This hypothesis seems to be confirmed by recent studies. [28] examined the music preferences of 566 Dutch adolescents who formed 283 same-sex friendship dyads and found a high degree of similarity in the music preferences of mutual friends. Since they surveyed the same participants one year after the initial survey, they could also examine the role of music preferences in the formation of new friendships and found that adolescents who had similar music preferences were more likely to become friends, as long as their music preferences were not associated with the most mainstream dimensions. In the same line, Boer and colleagues [29] conducted three studies (two laboratory experiments involving German participants and one field study involving Hong Kong university students) to examine the relationship between similarity in music preferences and social attraction. They found that people were more likely to be attracted to others who shared their music tastes because it suggests that they might also share the same values.

Adolescents were also found to be influenced by the music tastes of their parents. ter Bogt and colleagues [27] studied the music tastes of 325 adolescents and their parents. Their analysis revealed some significant correlations. The adolescents whose parents liked classical or jazz music were also more likely to appreciate these music genres. Parents' preferences for popular music were associated with a preference for popular and dance music in their adolescent children. Parents were also found to pass on their liking of rock music to their adolescent daughters but not to their sons. One possible explanation for the influence of parents on their children's music tastes is that since family members live under the same roof, children are almost inevitably exposed to the favourite music of their parents.

#### 3.5.2 Social Class

In *La Distinction* [30], Bourdieu proposed a social stratification of tastes and cultural practices according to which a taste for highbrow music or other cultural products (and a disdain for lowbrow culture) is considered the expression of a high status. Recent research, however, suggests that a profound transformation in the tastes of the elite has occurred. In an article published in 1996, Peterson and Kern [31] reported the results of a study of the musical tastes of Americans based on data from the *Survey of Public Participation in the Arts* of 1982 and 1992. Their analysis revealed that far from being snobbish in their tastes, individuals with a high occupational status had eclectic tastes which spanned across the lowbrow/highbrow spectrum. In fact, people of high status were found to be more omnivorous than others, and their level of omnivorousness has increased over time. This highly cited study has motivated several other researchers to study the link between social class and music preferences. Similar studies were conducted in other countries, notably in France [32], Spain [33], and the Netherlands [34], and yielded similar results.

## 4. IMPLICATION FOR MUSIC RECOMMENDER SYSTEM DESIGN

A review of the literature on music tastes revealed many interesting findings that could be used to improve music RS. Firstly, we saw that researchers had been able to uncover the underlying structure of music preferences, which is composed of 4 or 5 factors. The main advantage for music RS is that these factors are fairly stable across populations and time, as opposed to genres, which are inconsistent and ill-defined. As suggested by Rentfrow, Goldberg, and Levitin themselves [13], music RS could characterize the music preferences of their users by calculating a score for each dimension.

Secondly, some personality dimensions were found to be correlated to music preferences. In most studies, Openness to experience was the strongest predictor of music tastes. It was positively related to liking *Reflective and Complex* music (e.g., jazz and classical) and, to a lesser extent, to *Intense and Rebellious* music (e.g., rock, heavy metal). This could indicate that users who like these music genres are more open to new music than other users. RS could take that into account and adapt the novelty level accordingly.

Finally, the demographic correlates of music preferences (e.g., age, gender, education, race), as well as religion and political orientation, could help ease the new user cold-start problem. As mentioned in the introduction, many music RS invite new users to create a profile and/or allow them to connect with a social networking site account, in which they have a profile. These profiles contain various types of information about users. Music RS could combine such information to make inferences about the music preferences of new users. In the same line, information about the education and the occupation of a user could be used to identify potential high-status, omnivore users.

## 5. CONCLUSION

The abundant research on music tastes in sociology and social psychology has been mostly overlooked by music RS developers. This review of selected literature on the topic allowed us to present the patterns of associations between music preferences and demographic characteristics, personality traits, values and beliefs. It also revealed the importance of social influences on music tastes and the role music plays in the construction of individual and social identities.

## 6. REFERENCES

- [1] Y. Koren: "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery Data*, Vol. 4, No. 1, pp. 1-24, 2010.
- [2] A. Uitendbogerd and R. V. Schyndel: "A review of factors affecting music recommender success," *ISMIR 2002: Proceedings of the Third International Conference on Music Information Retrieval*, M. Fingerhut, ed., pp. 204-208, Paris, France: IRCAM - Centre Pompidou, 2002.
- [3] P. J. Rentfrow and S. D. Gosling: "The do re mi's of everyday life: The structure and personality correlates of music preferences," *Journal of Personality and Social Psychology*, Vol. 84, No. 6, pp. 1236-1256, 2003.
- [4] S. Frith: *Sound effects : youth, leisure, and the politics of rock'n'roll*, New York: Pantheon Books, 1981.
- [5] A. C. North and D. J. Hargreaves: "Music and adolescent identity," *Music Education Research*, Vol. 1, No. 1, pp. 75-92, 1999.
- [6] P. J. Rentfrow and S. D. Gosling: "Message in a ballad: the role of music preferences in interpersonal perception," *Psychological Science*, Vol. 17, No. 3, pp. 236-242, 2006.
- [7] P. G. Christenson and D. F. Roberts: *It's not only rock & roll : popular music in the lives of adolescents*, Cresskill: Hampton Press, 1998.
- [8] M. J. M. H. Delsing, T. F. M. ter Bogt, R. C. M. E. Engels, and W. H. J. Meeus: "Adolescents' music preferences and personality characteristics," *European Journal of Personality*, Vol. 22, No. 2, pp. 109-130, 2008.
- [9] R. Brown: "Music preferences and personality among Japanese university students," *International Journal of Psychology*, Vol. 47, No. 4, pp. 259-268, 2012.
- [10] A. Langmeyer, A. Guglhor-Rudan, and C. Tarnai: "What do music preferences reveal about personality? A cross-cultural replication using self-ratings and ratings of music samples," *Journal of Individual Differences*, Vol. 33, No. 2, pp. 119-130, 2012.
- [11] T. Schäfer and P. Sedlmeier: "From the functions of music to music preference," *Psychology of Music*, Vol. 37, No. 3, pp. 279-300, 2009.
- [12] D. George, K. Stickle, R. Faith, and A. Wopnford: "The association between types of music enjoyed and cognitive, behavioral, and personality factors of those who listen," *Psychomusicology*, Vol. 19, No. 2, pp. 32-56, 2007.
- [13] P. J. Rentfrow, L. R. Goldberg, and D. J. Levitin: "The structure of musical preferences: A five-factor model," *Journal of Personality and Social Psychology*, Vol. 100, No. 6, pp. 1139-1157, 2011.
- [14] P. J. Rentfrow, L. R. Goldberg, D. J. Stillwell, M. Kosinski, S. D. Gosling, and D. J. Levitin: "The song remains the same: A replication and extension of the music model," *Music Perception*, Vol. 30, No. 2, pp. 161-185, 2012.
- [15] A. C. North and D. J. Hargreaves, Jr.: "Lifestyle correlates of musical preference: 1. Relationships, living arrangements, beliefs, and crime," *Psychology of Music*, Vol. 35m No. 1, pp. 58-87, 2007.
- [16] J. B. Peterson and P. G. Christenson: "Political orientation and music preference in the 1980s," *Popular Music and Society*, Vol. 11, No. 4, pp. 1-17, 1987.
- [17] A. Gardikiotis and A. Baltzis: "'Rock music for myself and justice to the world!': Musical identity, values, and music preferences," *Psychology of Music*, Vol. 40, No. 2, pp. 143-163, 2012.
- [18] J. Lynxwiler and D. Gay: "Moral boundaries and deviant music: public attitudes toward heavy metal and rap," *Deviant Behavior*, Vol. 21, No. 1, pp. 63-85, 2000.
- [19] A. Colley: "Young people's musical taste: relationship with gender and gender-related traits," *Journal of Applied Social Psychology*, Vol. 38, No. 8, pp. 2039-2055, 2008.
- [20] P. G. Christenson and J. B. Peterson: "Genre and gender in the structure of music preferences," *Communication Research*, Vol. 15, No. 3, pp. 282-301, 1988.
- [21] R. S. Denisoff and M. H. Levine: "Youth and popular music: A test of the taste culture hypothesis," *Youth & Society*, Vol. 4, No. 2, pp. 237-255, 1972.
- [22] J. Tanner, M. Asbridge, and S. Wortley: "Our favourite melodies: musical consumption and teenage lifestyles," *British Journal of Sociology*, Vol. 59, No. 1, pp. 117-144, 2008.
- [23] J. Harrison and J. Ryan: "Musical taste and ageing," *Ageing & Society*, Vol. 30, No. 4, pp. 649-669, 2010.
- [24] A. Bonneville-Roussy, P. J. Rentfrow, M. K. Xu, and J. Potter: "Music through the ages: Trends in musical engagement and preferences from adolescence through middle adulthood," *American Psychological Association*, pp. 703-717, 2013.
- [25] M. B. Holbrook and R. M. Schindler: "Some exploratory findings on the development of musical tastes," *Journal of Consumer Research*, Vol. 16, No. 1, pp. 119-124, 1989.
- [26] J. Hemming: "Is there a peak in popular music preference at a certain song-specific age? A replication of Holbrook & Schindler's 1989 study," *Musicae Scientiae*, Vol. 17, No. 3, pp. 293-304, 2013.
- [27] T. F. M. ter Bogt, M. J. M. H. Delsing, M. van Zalk, P. G. Christenson, and W. H. J. Meeus: "Intergenerational Continuity of Taste: Parental and Adolescent Music Preferences," *Social Forces*, Vol. 90, No. 1, pp. 297-319, 2011.
- [28] M. H. W. Selfhout, S. J. T. Branje, T. F. M. ter Bogt, and W. H. J. Meeus: "The role of music preferences in early adolescents' friendship formation and stability," *Journal of Adolescence*, Vol. 32, No. 1, pp. 95-107, 2009.
- [29] D. Boer, R. Fischer, M. Strack, M. H. Bond, E. Lo, and J. Lam: "How shared preferences in music create bonds between people: Values as the missing link," *Personality and Social Psychology Bulletin*, Vol. 37, No. 9, pp. 1159-1171, 2011.
- [30] P. Bourdieu: *La distinction: critique sociale du jugement*, Paris: Éditions de minuit, 1979.
- [31] R. A. Peterson and R. M. Kern: "Changing Highbrow Taste: From Snob to Omnivore," *American Sociological Review*, Vol. 61, No. 5, pp. 900-907, 1996.
- [32] P. Coulangeon and Y. Lemel: "Is 'distinction' really outdated? Questioning the meaning of the omnivorization of musical taste in contemporary France," *Poetics*, Vol. 35, No. 2-3, pp. 93-111, 2007.
- [33] J. López-Sintas, M. E. Garcia-Alvarez, and N. Filimon: "Scale and periodicities of recorded music consumption: reconciling Bourdieu's theory of taste with facts," *The Sociological Review*, Vol. 56, No. 1, pp. 78-101, 2008.
- [34] K. van Eijck: "Social Differentiation in Musical Taste Patterns," *Social Forces*, Vol. 79, No. 3, pp. 1163-1185, 2001.



## SOCIAL MUSIC IN CARS

Sally Jo Cunningham, David M. Nichols, David Bainbridge, Hasan Ali

Department of Computer Science, University of Waikato, New Zealand

{sallyjo, d.nichols, davidb}@waikato.ac.nz, hma4@students.waikato.ac.nz

Wayne: “I think we’ll go with a little *Bohemian Rhapsody*, gentlemen”

Garth: “Good call”

Wayne's World  
(1992)

### ABSTRACT

This paper builds an understanding of how music is currently experienced by a social group travelling together in a car—how songs are chosen for playing, how music both reflects and influences the group’s mood and social interaction, who supplies the music, the hardware/software that supports song selection and presentation. This fine-grained context emerges from a qualitative analysis of a rich set of ethnographic data (participant observations and interviews) focusing primarily on the experience of in-car music on moderate length and long trips. We suggest features and functionality for music software to enhance the social experience when travelling in cars, and prototype and test a user interface based on design suggestions drawn from the data.

### 1. INTRODUCTION

Automobile travel occupies a significant space in modern Western lives and culture. The car can become a ‘home-from-home’ for commuters in their largely solitary travels, and for groups of people (friends, families, work colleagues) in both long and short journeys [20]. Music is commonly seen as a natural feature of automotive travel, and as cars become increasingly computerized [17] the opportunities are increased for providing music tailored to the specific characteristics of a given journey. To achieve this goal, however, we must first come to a more fine-grained understanding of these car-based everyday music experiences. To that end, this paper explores the role of music in supporting the ‘peculiar sociality’ [20] of car travel.

### 2. BACKGROUND

Most work investigating the experience of music in cars focuses on single-users, (e.g. [4], [5]). Solo drivers are free to create their own audio environment: “the car is a space of performance and communication where drivers report being in dialogue with the radio or singing in their own audited/privatized space” [5]. Walsh [21] notes that “a

large majority of drivers in the United States declare they sing aloud when driving”.

Walsh provides the most detailed discussion of the social aspects of music in cars, noting the interaction with conversation (particularly through volume levels) and music’s role in filling “chasms of silence” [21]. Issues of impression management [9, 21] (music I like but wouldn’t want others to know I like) are more acute in the confined environment of a car and vary depending on the social relationships between the occupants [21]. Music selections are often the result of negotiations between the passengers and the driver [14, 21], where the driver typically has privileged access to the audio controls.

Bull [6] reports a particularly interesting example of the intersection between the private environment of personal portable devices and the social environment of a car with passengers:

*Jim points to the problematic nature of joint listening in the automobile due to differing musical tastes. The result is that he plays his iPod through the car radio whilst his children listen to theirs independently or playfully in ‘harmony’ resulting in multiple sound-worlds in the same space.*

Here, although the children have personal devices they try to synchronize the playback so that they can experience the same song at the same time; even though their activity will occur in the context of another piece of music on the car audio system. Alternative methods for sharing include explicit (and implicit) recommendation, as in *Push!Music* [15], and physical sharing of earbuds [3]. Bull [6] also highlights another aspect of music in cars: selection activities that occur prior to a journey. The classic ‘roadtrip’ activity of choosing music to accompany a long drive is also noted: “drivers would intentionally set up and prepare for their journey by explicitly selecting music to accompany the protracted journey “on the road”” [21].

*Sound Pryer* [18] is a joint-listening prototype that enables drivers to ‘pry’ into the music playing in other cars. This approach emphasizes driving as a social practice, though it focuses on inter-driver relationships rather than those involving passengers. *Sound Pryer* can also be thought of as a transfer of some of the mobile music sharing concepts in the *tunA* system [2] to the car setting.



© S.J. Cunningham, D.M. Nichols, D. Bainbridge, H. Ali.  
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** S.J. Cunningham, D.M. Nichols, D. Bainbridge, H. Ali.. “Social Music in Cars”, 15th International Society for Music Information Retrieval Conference, 2014.

Driver distraction is known to be a significant factor in vehicle accidents and has led to legislation around the world restricting the use of mobile phones whilst driving. In addition to distraction effects caused by operating audio devices there are the separate issues of how the music itself affects the driver. Driving style can be influenced by genre, volume and tempo of music [10]: “at high levels, fast and loud music has been shown to divert attention [from driving]” [11], although drivers frequently use music to relax [11]. Several reports indicate that drivers use music to relieve boredom on long or familiar routes [1, 21], e.g. “as repetitious scenery encourages increasing disinterest ... the personalized sounds of travel assume a greater role in allowing the driver-occupants respite via intermitting the sonic activity during protracted driving stints” [21].

Many accidents are caused by driver drowsiness; when linked with physiological sensors to assess the driver’s state, music can be used to assist in maintaining an appropriate level of driver vigilance [16]. Music can also counteract driver vigilance by masking external sounds and auditory warnings, particularly for older drivers where age-related hearing loss is more likely to occur [19].

In summary, music fulfils a variety of different roles in affecting the mental state of the driver. It competes and interacts with passenger conversation, the external environmental and with audio functions from the increasingly computerized driving interface of the car. When passengers are present, the selection and playing of music is a social activity that requires negotiation between the occupants of the vehicle.

### 3. DATA COLLECTION AND METHODOLOGY

Our research uses data collected in a third year university Human Computer Interaction (HCI) course in which students design and prototype a system for the set application, where their designs are informed by an ethnographic investigations into behavior associated with the application domain. This present paper focuses on the ethnographic data collected that relates to music and car travel, as gathered by 22 student investigators (Table 1). All data gathering for this study occurred within New Zealand.

To explore the problem of designing a system to support groups of people in selecting and playing music while traveling, The students performed participant observations, with the observations focusing on how the music is chosen for playing, how the music fits in with the other activities being conducted, who supplies the music, and how/who changes the songs or alters the volume. The students then explored subjective social music experiences through auto-ethnographies [8] and interviews of friends. The data comprises 19 participant observations, two self-interviews, and four interviews (approximately 45 printed pages). Of the 19 participant observations, four were of short drives (10 to 30 minutes), 14 were lengthier trips (50 minutes to 2 hours), and one was a classic ‘road trip’ (7 hours). The number of

people participating in a trip ranged from one to five (Table 2). Of the 69 total travelers across the nineteen journeys, 45 were male and 24 were female. One set of travelers were all female, 7 were all male, and the remainder (11) were mixed gender.

**Table 1.** Demographics of student investigators

Male	Female	National Origin	Count
17	5	NZ/Australia	5
		China	13
		Mid-East	3
		Other	1
<b>Age Range:</b>			
20 - 27			

Grounded Theory methods [13] were used to analyze the student summaries of their participant observations and interviews. This present paper teases out the social behaviors that influence, and are influenced by, music played during group car travel. Supporting evidence drawn from the ethnographic data is presented below in italics.

**Table 2.** Number of travelers in observed journeys

1	2	3	4	5
1	0	7	7	4

## 4. MUSIC BEHAVIOR IN CAR TRAVEL

This section explores: the physical car environment and the reported car audio devices; the different reported roles of the driver; observed behaviors surrounding the choice of songs and the setting of volume; music and driving safety; ordering of songs that are selected to be played; and the ‘activities’ that music supports and influences.

### 4.1 Pre-trip Activities

The owner of a car often keeps personal music on hand in the vehicle (CDs, an MP3 player loaded with ‘car music’) as well as carrying along a mobile or MP3 player loaded with his/her music collection). If only the owner’s music is played on the trip, then that person should, logically, also manage the selection of songs during the journey. Unfortunately the owner of the car is also often the driver as well—and so safety may be compromised when the driver is actively involved in choosing and ordering songs for play.

Passengers are also likely to have on hand a mobile or MP3 player, and for longer trips may select CDs to share. If two or more people contribute music to be played on the journey, the challenge then becomes to bring all the songs together onto a single device—otherwise they experience the hassle of juggling several players. A consequence of merging collections, however, is that no one person will be familiar with the full set of songs, making on-the-road construction of playlists more difficult (particularly given the impoverished display surface of most MP3 players).

A simple pooling of songs from the passengers’ and driver’s personal music devices is unlikely to provide an efficiently utilizable source for selection of songs for a specific journey. The music that an individual listens to during

a usual day's activities may not be suitable for a particular trip, or indeed for any car journey. People tend to tailor their listening to the activity at hand [7], and so songs that are perfect 'gym music' or 'study music' may not have the appropriate tempo, mood, or emotional tenor. Further, an individual's music collection may include 'guilty pleasures' that s/he may not want others to become aware of [9]:

*What mainly made [him] less comfortable in providing music that he likes is because he did [not] want to destroy the hyper atmosphere in the car as a result of the mostly energetic songs being played throughout the trip. His taste is mostly doom and death metal, with harsh emotion and so will create a bleak atmosphere in the car.*

#### 4.2 Physical Environment and Audio Equipment

The travel described in the participant observations primarily occurred in standard sized cars with two seating areas, comfortably seating at most two people in the front and three in the rear sections. In this environment physical movement is constrained. If the audio device controller is fixed in place then not everyone can easily reach it or view its display; if the controller is a handheld device, then it must be passed around (and even then it may be awkward to move the controller between the two sections).

As is typical of student vehicles in New Zealand, the cars tended to be older (10+ years) and so were less likely to include sophisticated audio options such as configurable speakers and built-in MP3 systems. The range of audio equipment reported included radio, built-in CD player, portable CD player, stand-alone MP3 player plus speakers, and MP3 player connected to the car audio system.

The overwhelming preference evinced in this study is for devices that give more fine-grained control over song selection (i.e., MP3 players over CD players, CD players over radio). The disadvantages of radio are that music choice is by station rather than by song, reception can be disrupted if the car travels out of range, and most channels include ads. On the other hand, radio can provide news and talkback, to break up a longer journey.

#### 4.3 Music in Support of Journey Social Activities

Music is seen as integral to the group experience on a trip; it would be unacceptable and anti-social for the car's occupants to simply each listen to their individual MP3 player, for example. We identify a wide variety of ways that travelers select songs so as to support group social activities during travel:

- Music can contribute to driving **safety**, by playing songs that will reduce driver drowsiness and keep the driver focused (*music... can liven up a drive and keep you entertained or awake much longer*). For passengers, it can reduce the tedium associated with trips through uninteresting or too-familiar scenery (*music can reduce the boredom for you and your friends with the journey*).

Conversely, loud, fast tempo music can adversely affect safety ([As the driver, I] *changed the volume very high... my body was shaking with the song. I stepped on the accelerator in my car; The driver [was] seen to increase the speed when the songs he liked is on*).

- Listening to music can be the main source of **entertainment** during a trip, as the driver and passengers focus on the songs played.
- Songs need not be listened to passively; travelers may engage in **group sing-alongs**, with the music providing support for their 'performances'. These sessions may be loud and include over-the-top emotive renditions for the amusement of the singer and the group, and be accompanied by clapping and 'dancing' in the seats (*The participants would sing along to the lyrics of the songs, and also sometimes dance along to the music, laughing and smiling throughout it*).
- A particular song may spark a **conversation about the music**—to identify a song (*they would know what song they wanted to hear but they would not know the artist or name of the song. When this happened, they would ... try to think of the artist name together*) or to discuss other aspects of the artist/song/genre/etc (*'In the air tonight, Phil Collins!' Ann asked Joan and I, 'did you know that it's top of the charts at the moment' ... There was conversation about Phil Collins re-releasing his music.*) A lively debate can surround the choice and ordering of the songs to play, if playlists are created during the trip itself.
- Music can provide a **background to conversation**; at this point the travelers pay little or no attention to the songs but they mask traffic noises (*when we were chatting... no one really cared what was on as long as there was some ambient sound*). By providing 'filler' for awkward silences, music is particularly useful in supporting conversations among groups who don't know each other particularly well (*it seemed more natural to talk when there was music to break the silence*).

For shorter trips, music might serve only one or two of these social purposes—playing as background to a debate over where to eat, for example. On longer journeys, the focus of group attention and activity is likely to shift over time, and with that shift the role of the music will vary as well: *At some times it would be the focus activity, with everyone having input on what song to choose and then singing along. While at other times the group just wanted to talk with each other and so the music was turned right down and became background music...*

#### 4.4 Selecting and Ordering Songs

The physical music device plays a significant role in determining who chooses the music on a car trip. If the device is fixed (typically in the center of the dashboard), then it is easily accessible only by the driver or front passenger—and so they are likely to have primary responsibility for choosing, or arbitrating the choice, of songs. The driver is often

the owner of the vehicle, and in that case is likely to be assertive at decision points (*Since I was the driver, I was basically the DJ. I would select the CD and the song to be played. I also changed the song if I didn't like it even if others in the car did.*). Given the small display surfaces of most music devices and the complexity of interactions with those devices, it is likely that safety is compromised when the driver acts as DJ. Consider, for example:

*I select some remixed trance music from the second CD at odd slots of the playlist, and then insert some pop songs from other CDs in the rest of the slots of the list. ... I manually change the play order to random. Also I disable the volume protect. And enable the max volume that from the subwoofer due to the noises from the outside of my car ...*

If the music system has a hand-held controller, then the responsibility for song selection can move through the car. At any one point, however, a single individual will assume responsibility for music management. Friends are often familiar with each other's tastes, and so decisions can be made amicably with little or no consultation (*I felt comfortable in choosing the music because they were mostly friends and I knew what kind of music they were all into and what music some friends were not into...*). Imposing one's will might go against the sense of a group experience and social expectations (*...having the last word means it could cause problems between friends*), or alternatively close ties might make unilateral decisions more acceptable (*I did occasionally get fed up from their music and put back my music again without even asking them for permission, you know we are all friends.*).

As noted in Section 4.1, song selection on the fly can be difficult because the chooser may not be familiar with the complete base collection, or because the base collection includes songs not suited to the current mood of the trip. A common strategy is to listen to the first few seconds of a song, and if it is unacceptable then to skip to the song that comes up 'next' in the CD / shuffle / predetermined playlist. This strategy provides a choppy listening experience, but does have the advantage of simplicity: a song is skipped if any one person in the car expresses an objection to it. It may, however, be embarrassing to ask for a change if one is not in current possession of the control device.

Song-by-song selection is appropriate for shorter trips, as the setup time for a playlist may be longer than the journey itself. Suggesting and ordering songs can also be a part of the fun of the event and engage travelers socially (*My friends would request any songs that they would like to hear, and the passenger in control of the iPod acted like a human playlist; trying to memorise the requests in order and playing them as each song finished.*)

For longer trips, a set of pre-created playlists or mixes (supporting the expected moods or phases of the journey) can create a smoother travel experience. A diverse set of playlists may be necessary to match the range of social mu-

sic behaviors reported in Section 4.2. Even with careful pre-planning, however, a song may be rejected at time of play for personal, idiosyncratic reasons (for example, one participant skips *particular songs ... associated with particular memories and events so I don't like to listen to them while driving for example*).

#### 4.5 Music Volume

Sound volume is likely to change during a trip, signaling a change in the mood of the gathering, an alteration in the group focus, or to intensify / downplay the effects of a given song. Participant observations included the following reasons for altering sound levels: to focus group attention on a particular song (louder); for the group to sing along with a song (louder); to switch the focus of group activity from the music to conversation (softer); to 'energize' the mood of the group (louder); to calm the group mood, and particularly to permit passengers to sleep (softer); and to move the group focus from conversation back to the music, particularly when conversation falters (louder).

Clearly the ability to modulate volume to fit to the current activity or mood is crucial. A finer control than is currently available would be desirable, as often speaker placement means perceived volume depends on one's seat in the car (*[he] asked the driver to turn the bass down ... because the bass effect was too strong, and the driver ... think[s] the bass is fine in the front*).

Further, the physical division of a car into separate rows of seats and its restriction of passenger movement can encourage separate activity 'zones' (for example, front seats / back seats)—and the appropriate volume for the music can differ between seating areas:

*One of our friends who sets beside the driver is paying more attentions on the music, the rest 3 of us set in the back were communicate a lot more, and didn't paying too much attention on the music... the front people can hear the music a lot more clear than the people sets in the back, and it's harder for the front people to join the communication with the back people because he need to turn his head around for the chat sometimes.*

### 5. IMPLICATIONS FOR A SOCIAL AUDIO SYSTEM FOR CAR TRAVEL

Leveraging upon music information retrieval capabilities, we now describe how our findings can inform the design of software specially targeted for song selection during car trips—personified, the software we seek in essence acts as a music host. In general a playlist generator [12] for song selection coupled with access to a distributed network of self-contained digital music libraries for storing, organizing, and retrieving items (the collections of songs the various people travelling have) are useful building blocks to developing such software; however, to achieve a digital music host, what is needed ultimately goes beyond this.

In broad terms, we envisage a software application with two phases: initial configuration and responsive adaptation. During configuration, the application gathers the pool of songs for the trip from the individuals' devices, taking into account preferences such as which songs they wish to keep private and which types of songs (genre, artist, tempo, etc.) that they wish to have considered for the trip playlist. The users are then prompted to enter the approximate length of the upcoming road trip, and an initial playlist is constructed based on the user preferences and pool of songs.

During the trip, the application can make use of a variety of inputs to dynamically adjust the sequence of songs played. Here significant gains can be made from inventive uses of MIR techniques coupled with temporal and spatial information—even data sensors from the car. For instance, if the application noticed the driver speeding for that section of road it could alter the selection of the next song to one that is quieter with a slower tempo (beat detection); alternatively, triggered by the detection of the conversation lapsing into silence (noise cancelling) the next song played could be altered to be one labeled with a higher “interest” value (tagged, for instance, using semantic web technologies, and captured in the playlist as metadata). News sourced from a radio signal (whichever is currently in range) can be interspersed with the songs being played.

As evidenced by our analysis, the role of the driver/owner of the car takes on special significance in terms of the interface and interaction design. As the host of the vehicle, there is a perception that they are more closely linked to the software (the digital music host) that is making the decision over what to play next. While it is not a strict requirement of the software, for the majority of situations it will be an instinctive decision that the key audio device used to play the songs on the trip will be the one owned by the driver. For the adaptive phase of the software then, there is a certain irony that the driver (for reasons of driving safely) has less opportunity to influence the song selection during the trip. To address this imbalance, an aspect the software could support is the prioritization of input from the “master” application at noted times that are deemed safe (such when the car is stationary).

More prosaically, the travellers will require support in tweaking the playlist as the trip progresses. We developed and tested a prototype of this aspect of the system, to evaluate the design's potential. The existing behaviors explored in Section 3 suggest that this system should be targeted at tablet devices rather than smaller mobiles: while the device should be lightweight enough to be easily passed between passengers in a vehicle, the users should be able to clearly see the screen details from an arm's length, and controls should be large and spaced to minimize input error.

Figure 1 presents screenshots for primary functionality of our prototype: the view of the trip playlist, which features the current song in context with the preceding and succeeding songs (Figure 1a); the lyrics display for the current song, sized to be viewable by all (Figure 1b); and a

screen allowing selected songs to be easily inserted into different points in the playlist (Figure 1c). While it was tempting on a technical level to include mobile-based wireless voting (using their smart phones) to move the currently playing item up or down as an expression of like/dislike (relevance feedback), we recognize that face-to-face discussion and argument over songs is often a source of enjoyment and bonding for fellow travelers—and so we deliberately support only manual playlist manipulation.

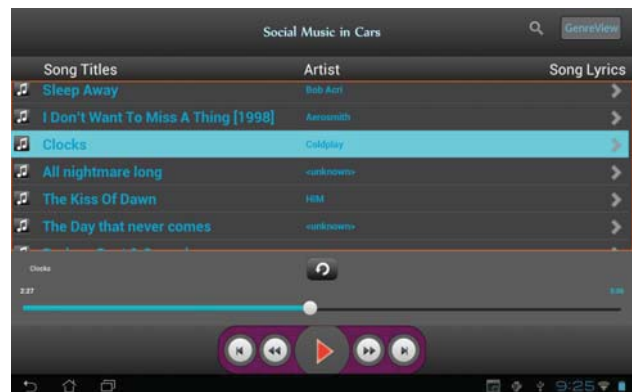


Figure 1a. Playlist view.

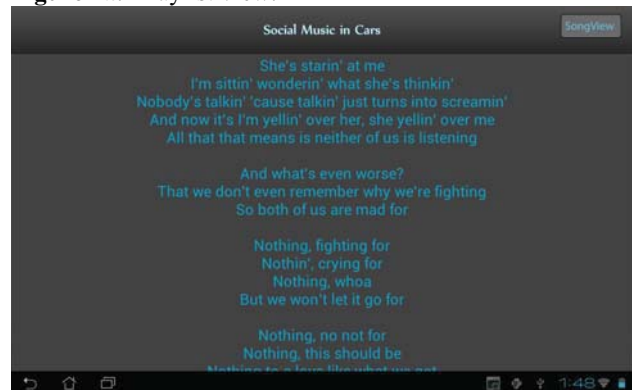


Figure 1b. Lyrics view for the active song.

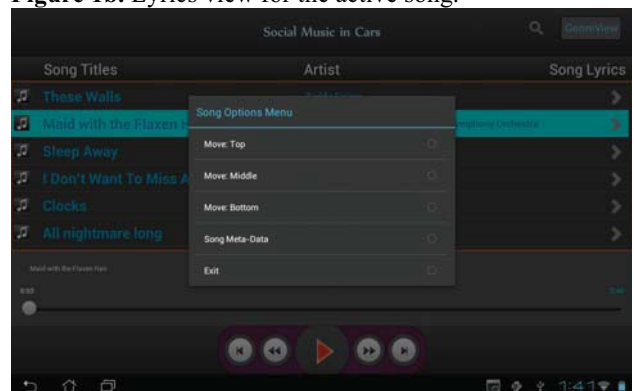


Figure 1c. After searching for a song, “smart options” for inserting the song into the current section of the playlist.

Given the practical and safety difficulties in evaluating our prototype system in a moving car, we instead used a stationary simulation. Two groups of four high school aged

males participated in the evaluation, with each trial consisting of approximately 30 minutes in which they listened to songs on a pre-prepared playlist, both collaboratively and individually selected additional songs, inserted them into the playlist, and viewed lyrics to sing along. The researchers took manual notes of the simulations, and participants engaged in focus group discussions post-simulation.

While the participants found the prototype to be generally usable (though usability tweaks were identified), we identified worrying episodes in which the drivers switched focus from the wheel to the tablet. While we recognize that behavior may be different in a simulation than in real driving conditions, we also saw strong evidence from the ethnographic data that drivers—particularly young, male drivers—can prioritize song selection over road safety. Further design iterations must recognize that drivers will inevitably seize control of a car’s music system, and so should prioritize design that supports fast, one-handed interactions.

## 6. CONCLUSIONS

The primary contribution of this paper is understanding of social music behavior of small groups of people while on ‘road trips’, developed through a qualitative analysis of ethnographic data (participant observations and interviews). We prototyped and evaluated the more prosaic aspects of a system to support social music listening on road trips, and suggest further extensions—including sensor-based input to modify the trip playlist—for future research.

## 7. REFERENCES

- [1] K.P. Åkesson, A. Nilsson: “Designing Leisure Applications for the Mundane Car-Commute,” *Personal and Ubiquitous Computing*, 6:3, 176–187, 2002.
- [2] A. Bassoli, J. Moore, S. Agamanolis: “tunA: Socialising Music Sharing on the Move,” In K. O’Hara and B. Brown (eds.), *Consuming Music Together: Social and Collaborative Aspects of Music Consumption Technologies*. Springer, 151-172, 2007.
- [3] T. Bickford: “Earbuds Are Good for Sharing: Children’s Sociable Uses of Headphones at a Vermont Primary School,” In J. Stanyek and S. Gopinath (eds.), *The Handbook of Mobile Music Studies*, Oxford University Press, 2011.
- [4] M. Bull: “Soundscapes of the car: a critical study of automobile habitation,” In M. Bull and L. Back, (eds.) *The Auditory Culture Reader*, Berg, 357–374, 2003.
- [5] M. Bull: “Automobility and the power of sound”, *Theory, Culture & Society*, 21:4/5, 243–259, 2004.
- [6] M. Bull: “Investigating the culture of mobile listening: from Walkman to iPod,” In K. O’Hara and B. Brown (eds.), *Consuming Music Together*. Springer, 131–149, 2006.
- [7] S.J. Cunningham, D. Bainbridge, A. Falconer. More of an art than a science: playlist and mix construction. *Proceedings of ISMIR ’06*, Vancouver, 2006.
- [8] S.J. Cunningham, M. Jones: “Autoethnography: a tool for practice and education,” *Proceedings of the 6<sup>th</sup> New Zealand International Conference on Computer-Human Interaction (CHINZ 2005)*, 1-8, 2005.
- [9] S.J. Cunningham, M. Jones, S. Jones: “Organizing digital music for use: an examination of personal music collections”. *Proceedings of ISMIR’04*, Barcelona, 447-454, 2004.
- [10] B.H. Dalton, D.G. Behm: “Effects of noise and music on human and task performance: A systematic review,” *Occupational Ergonomics*, 7:3, 143-152, 2007.
- [11] N. Dibben, V.J. Williamson: “An exploratory survey of in-vehicle music listening,” *Psychology of Music*, 35: 4, 571-589, 2007.
- [12] A. Flexer, D. Schnitzer, M. Gasser, G. Widmer. “Playlist generation using start and end songs”, *Proceedings of ISMIR’08*, 173-178, 2008.
- [13] B. Glaser, A. Strauss: *The Discovery of Grounded Theory: Strategies for Qualitative Research*, Chicago, 1967.
- [14] A. E. Greasley, A. Lamont: “Exploring engagement with music in everyday life using experience sampling methodology,” *Musicae Scientiae*, 15: 45, 45-71, 2011.
- [15] M. Håkansson, M. Rost, L.E. Holmquist: “Gifts from friends and strangers: a study of mobile music sharing,” *Proceedings of ECSCW’07*, 311-330, 2007.
- [16] C. Hasegawa, K. Oguri: “The effects of specific musical stimuli on driver’s drowsiness,” *Proceedings of the Intelligent Transportation Systems Conference (ITSC’06)*, 817-822, 2006.
- [17] O. Juhlin: *Social Media on the Road: The Future of Car Based Computing*, Springer, London, 2010.
- [18] M. Östergren, O. Juhlin: “Car Drivers Using Sound Pryer – Joint Music Listening in Traffic Encounters,” In K. O’Hara and B. Brown (eds.), *Consuming Music Together: Social and Collaborative Aspects of Music Consumption Technologies*. Springer, 173-190, 2006.
- [19] E.B. Slawinski, J.F. McNeil: (2002) “Age, Music, and Driving Performance: Detection of External Warning Sounds in Vehicles,” *Psychomusicology*, 18, 123-31, 2002.
- [20] Urry, J. “Inhabiting the car,” *The Sociological Review*, 54, 17–31, 2006.
- [21] M.J. Walsh: “Driving to the beat of one’s own hum: Automobility and musical listening,” In N. K. Denzin (ed.) *Studies in Symbolic Interaction*, 35, 201-221, 2010.



## Poster Session 3

This Page Intentionally Left Blank



# A COMBINED THEMATIC AND ACOUSTIC APPROACH FOR A MUSIC RECOMMENDATION SERVICE IN TV COMMERCIALS

Mohamed Morchid, Richard Dufour, Georges Linares

LIA - University of Avignon (France)

{mohamed.morchid, richard.dufour, georges.linares}@univ-avignon.fr

## ABSTRACT

Most of modern advertisements contain a song to illustrate the commercial message. The success of a product, and its economic impact, can be directly linked to this choice. Finding the most appropriate song is usually made manually. Nonetheless, a single person is not able to listen and choose the best music among millions. The need for an automatic system for this particular task becomes increasingly critical. This paper describes the LIA music recommendation system for advertisements using both textual and acoustic features. This system aims at providing a song to a given commercial video and was evaluated in the context of the MediaEval 2013 Soundtrack task [14]. The goal of this task is to predict the most suitable soundtrack from a list of candidate songs, given a TV commercial. The organizers provide a development dataset including multimedia features. The initial assumption of the proposed system is that commercials which sell the same type of product, should also share the same music rhythm. A two-fold system is proposed: find commercials with close subjects in order to determine the mean rhythm of this subset, and then extract, from the candidate songs, the music which better corresponds to this mean rhythm.

## 1. INTRODUCTION

The success of a product or a service essentially depends of the way to present it. Thus, companies pay much attention to choose the most appropriate advertisement that will make a difference in the customer choice. The advertisers have different media possibilities, such as journal paper, radio, TV or Internet. In this context, they can exploit the audio media (TV, radio...) to attract listeners using a song related to the commercial. The choice of an appropriate song is crucial and can have a significant economic impact [5, 18]. Usually, this choice is made by a human expert. Nonetheless, while millions of musics exist, a human agent could only choose a song among a limited subset. This choice could then be inappropriate, or simply not the best one, since the agent could not search into a large num-

ber of musics. For these reasons, the need for an automatic song recommendation system, to illustrate advertisements, becomes a critical subject for companies.

In this paper, an automatic system for songs recommendation is proposed. The proposed approach combines both textual (web pages) and audio (acoustic) features to select, among a large number of songs, the most appropriate and relevant music knowing the commercial content. The first step of the proposed system is to represent commercials into a thematic space built from a Latent Dirichlet Allocation (LDA) [4]. This pre-processing subtask uses the related textual content of the commercial. Then, acoustic features of each song are extracted to find a set of the most relevant songs for a given commercial.

An appropriate benchmark is needed to evaluate the effectiveness of the proposed recommendation system. For these reasons, the proposed system is evaluated in the context of the challenging MediaEval 2013 Soundtrack task for commercials [10]. Indeed, the MusiClef task seeks to make this process automated by taking into account both context- and content-based information about the video, the brand, and the music. The main difficulty of this task is to find the set of relevant features that best describes the most appropriate song for a video.

Next section describes related work in topic space modeling for information retrieval and music tasks. Section 3 presents the proposed music recommendation system using both textual content and acoustic features related to musics from commercials. Section 4 explains in details the unsupervised Latent Dirichlet Allocation (LDA) technique, while Section 4.2 describes how the acoustic features are used to evaluate the proximity of a music to a commercial. Finally, experiments are presented in Section 5, while Section 6 gives conclusions and perspectives.

## 2. RELATED WORKS

Latent Dirichlet Allocation (LDA) [4] is widely used in several tasks of information retrieval such as classification or keywords extraction. However, this unsupervised method is not much considered in the music processing tasks. Next sections describe related works using LDA techniques with text corpora (Section 2.1) and in the context of music tasks (Section 2.3).



## 2.1 Topic modeling

Several methods were proposed by Information Retrieval (IR) researchers to build topic spaces such as Latent Semantic Analysis or Indexing (LSA/LSI) [2, 6], that use a singular value decomposition (SVD) to reduce the space dimension.

This method was improved by [11] which proposed a probabilistic LSA/LSI (pLSA/pLSI). The pLSI approach models each word in a document as a sample from a mixture model, where the mixture components are multinomial random variables that can be viewed as representations of topics. This method demonstrated its performance on various tasks, such as sentence [3] or keyword [24] extraction. In spite of the effectiveness of the pLSI approach, this method has two main drawbacks. The distribution of topics in pLSI is indexed by training documents. Thus, the number of these parameters grows with the training document set size, and then, the model is prone to overfitting which is a main issue in an IR task such as document clustering. However, to address this shortcoming, a tempering heuristic is used to smooth the parameter of pLSI model for acceptable predictive performance. Nonetheless, authors showed in [20] that overfitting can occur even if tempering process is used.

As a result, IR researchers proposed the Latent Dirichlet allocation (LDA) [4] method to overcome these two drawbacks. Thus, the number of parameters of LDA does not grow with the size of the training corpus and LDA is not candidate for overfitting. LDA is a generative model which considers a document, seen as a *bag-of-words* [21], as a mixture of latent topics. In opposition to a multinomial mixture model, LDA considers that a theme is associated to each occurrence of a word composing the document, rather than associate a topic with the complete document. Thereby, a document can change of topics from a word to another. However, the word occurrences are connected by a latent variable which controls the global respect of the distribution of the topics in the document. These latent topics are characterized by a distribution of word probabilities which are associated with them. pLSI and LDA models have been shown to generally outperform LSI on IR tasks [12]. Moreover, LDA provides a direct estimate of the relevance of a topic knowing a word set or a document such as a web pages in the proposed system.

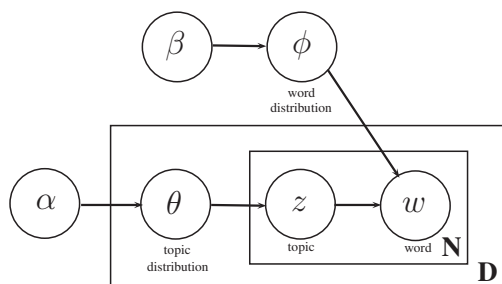


Figure 1. LDA Formalism.

Figure 1 presents the LDA formalism. For every docu-

ment  $d$  of a corpus  $\mathbf{D}$ , a first parameter  $\theta$  is drawn according to a Dirichlet law of parameter  $\alpha$ . A second parameter  $\phi$  is drawn according to the same Dirichlet law of parameter  $\beta$ . Then, to generate every word  $w$  of the document  $d$ , a latent topic  $z$  is drawn from a multinomial distribution on  $\theta$ . Knowing this topic  $z$ , the distribution of the words is a multinomial of parameters  $\phi$ . The parameter  $\theta$  is drawn for all the documents from the same *prior* parameter  $\alpha$ . This allows to obtain a parameter binding the documents all together [4].

## 2.2 Gibbs sampling

Several techniques have been proposed to estimate LDA parameters, such as Variational Methods [4], Expectation-Propagation [17] or Gibbs Sampling [8]. Gibbs Sampling is a special case of Markov-chain Monte Carlo (MCMC) [7] and gives a simple algorithm for approximate inference in high-dimensional models such as LDA [9]. This overcomes the difficulty to directly and exactly estimate parameters that maximize the likelihood of the whole data collection defined as:  $P(W|\vec{\alpha}, \vec{\beta}) = \prod_{m=1}^M P(\vec{w}_m|\vec{\alpha}, \vec{\beta})$  for the whole data collection  $W = \{\vec{w}_m\}_{m=1}^M$  knowing the Dirichlet parameters  $\vec{\alpha}$  and  $\vec{\beta}$ .

The first use of Gibbs Sampling for estimating LDA is reported in [8] and a more comprehensive description of this method can be found in [9]. One can refer to these papers for a better understanding of this sampling technique.

## 2.3 Topic modeling and Music

Topic modeling was already used in music processing, such as [13], where the authors presented a system which learns musical key as a key-profile. Thus, the proposed approach considered a song as a random mixture of key-profiles. In [25], authors described a classification method to assign a label to an unseen music. The authors use LDA to build a topic space from music-tags to get the probability of every music-tag belonging to each music genre. Then, each music is labeled to a genre knowing its tags. The purpose of the proposed approach is to find a set of relevant musics for a TV commercial.

## 3. PROPOSED APPROACH

The goal of the proposed automatic system is to recommend a set of musics given a TV commercial. The system uses external knowledge to find these songs. These external resources are composed with a set of TV commercials associated, for each one, with a song and a set of web pages (see [14] for more details about the MediaEval 2013 Soundtrack task). The idea behind the proposed approach is to assume that two commercials sharing same subjects or interests, also share the same kind of songs. The main issue in this approach is to find commercials, from the external dataset, that have sets of subjects close to those in commercials from the test set. As described in Section 2.1, a document can be represented as a set of latent topics. Thus, two documents sharing the same topics could be seen as *thematically* close.

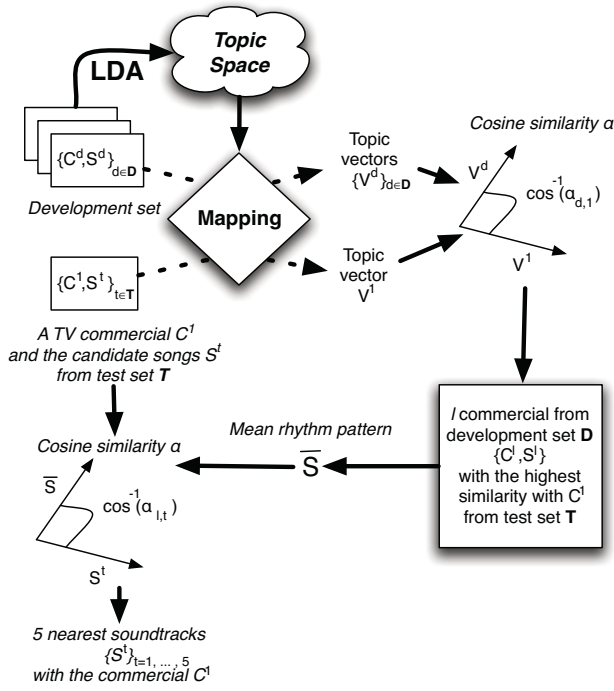


Figure 2. Global architecture of the proposed system.

Basically, the first process of the proposed three step system is to map each TV commercial from the test and development sets, into a topic space learnt with a LDA algorithm. A TV commercial from the test set is then linked to TV commercials from development set sharing a set of close topics. Moreover, each commercial of the development set is related to a music. Thus, as a result, a commercial from the test set is related to a subset of songs from the development set, considered as thematically close to the commercial textual content.

The second step has the responsibility to estimate a list of candidate songs (see Figure 2) using song audio features from the subset of songs thematically close associated during the first step. This subset of songs is used to evaluate a rhythm pattern of the *ideal* song for this commercial.

The last step retrieves, from all candidate songs from the test set, the closest song to the rhythm pattern estimated during the previous step.

In details, the development set  $\mathbf{D}$  is composed of TV commercials  $C^d$ , with for each, a soundtrack  $S^d$  and a vector representation  $V^d$  related to the  $d^{th}$  TV commercial. In the same manner, the test set  $\mathbf{T}$  is composed of TV commercials  $C^t$ , with, for the  $t^{th}$  one, a vector representation  $V^t$  and a soundtrack  $S^t$  to predict. Then a similarity score  $\{\alpha_{d,t}\}_{d=1,\dots,\mathbf{D}}^{t=1,\dots,\mathbf{T}}$  is computed for each commercial  $C_i^d$  of the development set given one from the test set  $C^t$ :

$$\begin{aligned} \mathbf{D} &= \{C^d, V^D, S^d\}_{d=1,\dots,\mathbf{D}} \\ \mathbf{T} &= \{C^t, V^T, S^t\}_{t=1,\dots,5000} \end{aligned} \quad (1)$$

In the next sections, the topic space representation and

the mapping of a commercial in this topic representation to evaluate both  $V^d$  and  $V^t$  are described. Then, the computed similarity score is detailed. Finally, the soundtrack prediction process from a TV commercial is explained.

#### 4. TOPIC REPRESENTATION OF A TV COMMERCIAL

Let's consider a corpus  $\mathbf{D}$  from the development set of TV commercials with a word vocabulary  $\mathbf{V} = \{w_1, \dots, w_N\}$  of size  $N$ . A topic representation from corpus  $\mathbf{D}$  is then performed using a Latent Dirichlet Allocation (LDA) [4] approach. At the final LDA analysis, a topic space  $m$  of  $n$  topics is obtained with, for each theme  $z$ , the probability of each word  $w$  of  $\mathbf{V}$  knowing  $z$ , and for the entire model  $m$ , the probability of each theme  $z$  knowing the model  $m$ . Each TV commercial from both development and test sets is mapped into the topic space (see Figure 3) to obtain a vector representation ( $V^d$  and  $V^t$ ) of web pages related to a commercial into the thematic space computed as follow:

$$V^d[i](C_j^d) = P(z_i | C_j^d) \quad (2)$$

where  $P(z_i | C_j^d)$  is the probability of a topic  $z_i$  to be generated by the web pages from the commercial  $C_j^d$ , estimated using Gibbs sampling as described in Section 2.2. In the same way,  $V^t$  is estimated with the same topic space, and with the use of web pages of commercials of test set  $C_j^t$  (see Figure 3).

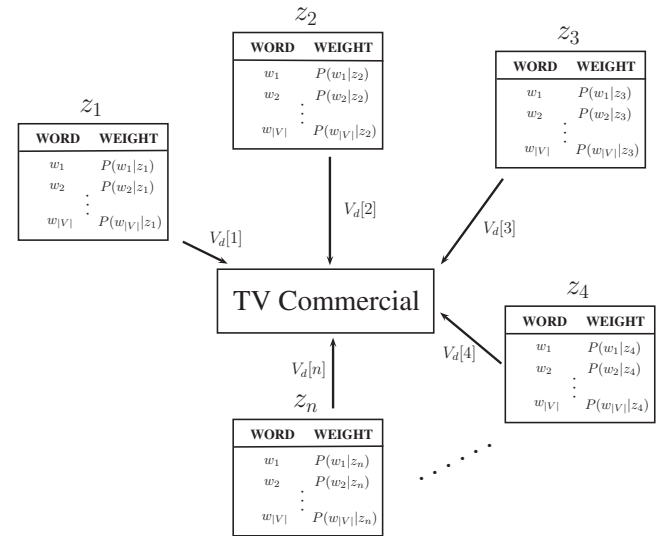


Figure 3. Mapping of a TV commercial in the topic space.

##### 4.1 Similarity measure

Each commercial from both development and test set, is mapped into the topic space to produce a vector representation for each one, respectively  $V^d$  and  $V^t$  as outcomes. Then, given a TV commercial  $C^1$  from the test set  $\mathbf{T}$ , a subset of other TV commercials from the development set  $\mathbf{D}$  is selected knowing their thematic proximity with  $C^1$ .

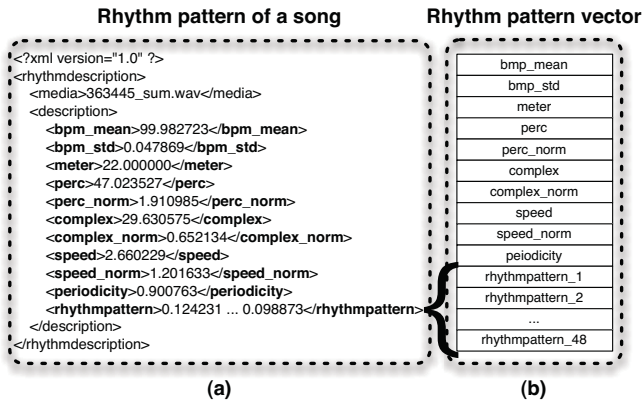
To estimate the similarity between  $C^1$  and commercials from development set, the cosine metric  $\alpha$  is used. This similarity metric is expressed thereafter:

$$\begin{aligned} \text{cosine}(V^d, V^t) &= \alpha_{d,t} \\ &= \frac{\sum_{i=1}^n V^d[i] \times V^t[i]}{\sqrt{\sum_{i=1}^n V^d[i]^2} \sqrt{\sum_{i=1}^n V^t[i]^2}} \end{aligned} \quad (3)$$

This metric allows to extract a subset of commercials from  $\mathbf{D}$  thematically close to  $C^1$ .

## 4.2 Rhythm pattern

The cosine measure, presented in previous section, is also used to evaluate the similarity between a mean rhythm pattern vector  $S^d$  of a song, and all the candidate songs  $S_k^t$  of the test set.



**Figure 4.** Rhythm pattern of a song from the development set in xml (a) and vector (b) representations.

In details, each commercial from  $\mathbf{D}$  is related with a soundtrack that is represented with a rhythm pattern vector. The organizers provide for each song contained into the MusicClef 2013 dataset:

- *video features* (MPEG-7 Motion Activity and Scalable Color Descriptor [15]),
- web pages about the respective brands and music artists,
- *music features*:
  - MFCC or BLF [22],
  - PS209 [19],
  - beat, key, harmonic pattern extracted with the Ircam software [1].

In our experiments, 10 rhythm features of songs are used (*speed*, *percussion*, ..., *periodicity*) as shown in Figure 4. These features of beat, key or harmonic pattern are

extracted using the Ircam software available at [1]. More information about features extraction from songs are detailed in [14].

As an outcome, each commercial is represented by a rhythm pattern vector of size 58 (10 from song features and 48 from rhythm pattern). From the subset of soundtracks of the  $l$  nearest commercials from  $\mathbf{D}$ , a mean rhythm vector  $\bar{S}$  is performed as:

$$\bar{S} = \frac{1}{l} \sum_{d \in l} S^d.$$

Finally, the cosine measure between this mean rhythm  $\bar{S}$  of the  $l$  nearest commercials from  $\mathbf{D}$ , and each commercial ( $\text{cosine}(\bar{S}, S^t)_{t \in T}$ ), is used to find, from the soundtrack  $S^t$  of the test set  $\mathbf{T}$ , the 5 songs from all the candidates having the closest rhythm pattern.

## 5. EXPERIMENTS AND RESULTS

Previous sections described the proposed automatic music recommendation system for TV commercials. This system is decomposed into three sub-processes. The first one maps the commercials into a topic space to evaluate the proximity of a commercial from the test set and all commercials from the development set. Then, the mean rhythm pattern of the thematically close commercials is computed. Finally, this rhythm pattern is computed with all ones from the test set of candidate songs to find a set of relevant musics.

### 5.1 Experimental protocol

The first step of the proposed approach, detailed in previous section, maps TV commercial textual content into a topic space of size  $n$  ( $n = 500$ ). This one is learnt from a LDA in a large corpus of documents. Section 4 describes the corpus  $\mathbf{D}$  of web pages. This corpus contains 10,724 Web pages related to brands of the commercials contained in  $\mathbf{D}$ . This corpus is composed of 44,229,747 words for a vocabulary of 4,476,153 unique words. More details about this text corpus, and the way to collect it, is explained into [14].

The first step of the proposed approach is to map each commercial textual content into a topic space learnt from a latent Dirichlet allocation (LDA). During the experiments, the MALLET tool is used [16] to perform a topic model. The proposed system is evaluated in the MediaEval 2013 MusicClef benchmark [14]. The aim of this task is to predict, for each video of the test set, the most suitable soundtrack from 5,000 candidate songs. The dataset is split into 3 sets. The development set contains multimodal information on 392 commercials (various metadata including Youtube uploader comments, audio features, video features, web pages and text features). The test set is a set of 55 videos to which a song should be associated using the recommendation set of 5,000 soundtracks (30 seconds long excerpts).

## 5.2 Experimental metrics

For each video in the test set, a ranked list of 5 candidate songs should be proposed. The song prediction evaluation is manually performed using the Amazon Mechanical Turk platform. This novel task is non-trivial in terms of “ground truth”, that is why human ratings for evaluation are used. Three scores have been computed from our system output. Let  $V$  be the full collection of test set videos, and let  $s_r(v)$  be the average suitability score for the audio file suggested at rank  $r$  for the video  $v$ . Then, the evaluation measures are computed as follows:

- Average suitability score of the first-ranked song:

$$\frac{1}{|V|} \sum_{i=1}^{|V|} s_1(v_i)$$

- Average suitability score for the full top-5:

$$\frac{1}{|V|} \sum_{i=1}^{|V|} \frac{1}{5} \overline{s_r(v_i)}$$

- Weighted average suitability score of the full top-5. Here, we apply a weighted harmonic mean score instead of an arithmetic mean:

$$\frac{1}{|V|} \sum_{i=1}^{|V|} \frac{\sum_{r=1}^5 s_r(v_i)}{\sum_{r=1}^5 \frac{s_r(v_i)}{r}}$$

The previously presented measures are used to study both rating and ranking aspects of the results.

## 5.3 Results

The measures defined in the previous section are used to evaluate the effectiveness of songs selected to be associated to TV commercials from the test set. The proposed topic space-based approach is evaluated in the same way, and obtained the results detailed thereafter:

- First rank average score: **2.16**
- Top 5 average score (arithmetic mean): **2.24**
- Top 5 average score (harmonic mean, taking rank into account): **2.22**

Considering that human judges rate the predicted songs from 1 (*very poor*) to 4 (*very well*), we can consider that our system is slightly better than the mean evaluation score (2) no matter the metric considered. While the system proposed in [23] is clearly different from ours, results are very similar. This shows the difficulty to build an automatic song recommendation system for TV commercials, the evaluation being also a critical point to discuss.

## 6. CONCLUSIONS AND PERSPECTIVES

In this paper, an automatic system to assign a soundtrack to a TV commercial has been proposed. This system combines two media: textual commercial content and audio rhythm pattern. The proposed approach obtains good results in spite of the fact that the system is automatic and unsupervised. Indeed, both subtasks are unsupervised (LDA learning and commercials mapping into the topic space)

and songs extraction (rhythm pattern estimation of the *ideal* songs for a commercial from the test set). Moreover, this promising approach, combining thematic representation of the textual content of a set of web pages describing a TV commercial and acoustic features, shows the relevance of topic-based representation in automatic recommendation using external resources (development set).

The choice of a relevant song to describe the idea behind a commercial, is a challenging task when the framework does not take into account relevant features related to:

- mood, such as harmonic content, harmonic progressions and timbre,
- music rhythm, such as musical style, texture, spectral centroid, or tempo.

The proposed automatic music recommendation system is limited by this small number (58) of features which not describe all music aspects. For these reasons, in future works, we plan to use others features, such as the song lyrics or the audio transcription of the TV commercials, and evaluate the effectiveness of the proposed hybrid framework into other information retrieval tasks such as classification of music genre or music clustering.

## 7. REFERENCES

- [1] Ircam. analyse-synthse: Software. In <http://anasynth.ircam.fr/home/software.>, Accessed: Sept. 2013.
- [2] J.R. Bellegarda. A latent semantic analysis framework for large-span language modeling. In *Fifth European Conference on Speech Communication and Technology*, 1997.
- [3] J.R. Bellegarda. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, 88(8):1279–1296, 2000.
- [4] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.
- [5] Claudia Bullerjahn. The effectiveness of music in television commercials. *Food Preferences and Taste: Continuity and Change*, 2:207, 1997.
- [6] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.
- [7] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):721–741, 1984.
- [8] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

- [9] Gregor Heinrich. Parameter estimation for text analysis. Web: <http://www.arbylon.net/publications/text-est.pdf>, 2005.
- [10] Nina Hoerberichts. Music and advertising: The effect of music in television commercials on consumer attitudes. *Bachelor Thesis*, 2012.
- [11] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI '99*, page 21. Citeseer, 1999.
- [12] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, 2001.
- [13] Diane Hu and Lawrence K Saul. A probabilistic topic model for unsupervised learning of musical key-profiles. In *ISMIR*, pages 441–446, 2009.
- [14] Cynthia C. S. Liem, Nicola Orio, Geoffroy Peeters, and Markus Schedl. MusiClef 2013: Soundtrack Selection for Commercials. In *MediaEval*, 2013.
- [15] Bangalore S Manjunath, Philippe Salembier, and Thomas Sikora. *Introduction to MPEG-7: multimedia content description interface*, volume 1. John Wiley & Sons, 2002.
- [16] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>, 2002.
- [17] Thomas Minka and John Lafferty. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 352–359. Morgan Kaufmann Publishers Inc., 2002.
- [18] C Whan Park and S Mark Young. Consumer response to television commercials: The impact of involvement and background music on brand attitude formation. *Journal of Marketing Research*, pages 11–24, 1986.
- [19] Tim Pohle, Dominik Schnitzer, Markus Schedl, Peter Knees, and Gerhard Widmer. On rhythm and general music similarity. In *ISMIR*, pages 525–530, 2009.
- [20] Alexandrin Popescul, David M Pennock, and Steve Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 437–444. Morgan Kaufmann Publishers Inc., 2001.
- [21] G. Salton. Automatic text processing: the transformation. *Analysis and Retrieval of Information by Computer*, 1989.
- [22] Klaus Seyerlehner, Gerhard Widmer, and Tim Pohle. Fusing block-level features for music similarity estimation. In *Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10)*, pages 225–232, 2010.
- [23] Han Su, Fang-Fei Kuo, Chu-Hsiang Chiu, Yen-Ju Chou, and Man-Kwan Shan. Mediaeval 2013: Soundtrack selection for commercials based on content correlation modeling. In *MediaEval 2013*, volume 1043 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.
- [24] Y. Suzuki, F. Fukumoto, and Y. Sekiguchi. Keyword extraction using term-domain interdependence for dictation of radio news. In *17th international conference on Computational linguistics*, volume 2, pages 1272–1276. ACL, 1998.
- [25] Chao Zhen and Jieping Xu. Multi-modal music genre classification approach. In *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, volume 8, pages 398–402. IEEE, 2010.

# ARE POETRY AND LYRICS ALL THAT DIFFERENT?

Abhishek Singhi

Daniel G. Brown

University of Waterloo  
Cheriton School of Computer Science  
{asinghi,dan.brown}@uwaterloo.ca

## ABSTRACT

We hypothesize that different genres of writing use different adjectives for the same concept. We test our hypothesis on lyrics, articles and poetry. We use the English Wikipedia and over 13,000 news articles from four leading newspapers for the article data set. Our lyrics data set consists of lyrics of more than 10,000 songs by 56 popular English singers, and our poetry dataset is made up of more than 20,000 poems from 60 famous poets. We find the probability distribution of synonymous adjectives in all the three different categories and use it to predict if a document is an article, lyrics or poetry given its set of adjectives. We achieve an accuracy level of 67% for lyrics, 80% for articles and 57% for poetry. Using these probability distribution we show that adjectives more likely to be used in lyrics are more rhymable than those more likely to be used in poetry, but they do not differ significantly in their semantic orientations. Furthermore we show that our algorithm is successfully able to detect poetic lyricists like Bob Dylan from non-poetic ones like Bryan Adams, as their lyrics are more often misclassified as poetry.

## 1. INTRODUCTION

The choice of a particular word, from a set of words that can instead be used, depends on the context we use it in, and on the artistic decision of the authors. We believe that for a given concept, the words that are more likely to be used in lyrics will be different from the ones which are more likely to be used in articles or poems, because lyricists have different objectives typically. We test our hypothesis on adjective usage in these categories of documents. We use adjectives, as a majority have synonyms that can be used depending on context. To our surprise, just the adjective usage is sufficient to separate documents quite effectively.

Finding the synonyms of a word is still an open problem. We used three different sources to obtain synonyms for a word – the WordNet, Wikipedia and an online thesaurus. We prune synonyms, obtained from the three sources, which fall below an experimentally determined threshold for the semantic distance between the synonyms

and the word. The list of relevant synonyms obtained after pruning was used to obtain the probability distribution over words.

A key requirement of our study is that there exists a difference, albeit a hazy one, between poetry and lyrics. Poetry attracts a more educated and sensitive audience while lyrics are written for the masses. Poetry, unlike lyrics, is often structurally more constrained, adhering to a particular meter and style. Lyrics are often written keeping the music in mind while poetry is written against a silent background. Lyrics, unlike poetry, often repeat lines and segments, causing us to believe that lyricists tend to pick more rhymable adjectives; of course, some poetic forms also repeat lines, such as the villanelle. For twenty different concepts we compare adjectives which are more likely to be used in lyrics rather than poetry and vice versa.

**Even** in my heart I see  
You're not **bein'** **true** to me  
**Deep** within my soul I feel  
Nothing's like it used to be  
Sometimes I wish I could turn back time  
**Impossible** as it may seem  
But I wish I could so **bad** baby  
Quit playin' games with my heart

**Figure 1.** The bold-faced words are the adjectives our algorithm takes into account while classifying a document, which in this case in a snippet of lyrics by the Backstreet Boys.

We use a bag of words model for the adjectives, where we do not care about their relative positions in the text, but only their frequencies. Finding synonyms of a given word is a vital step in our approach and since it is still considered a difficult task improvement in synonyms finding approaches will lead to an improvement in our classification accuracy. Our algorithm has a linear run time as it scans through the document once to come up with the prediction, giving us an accuracy of 68% overall. Lyricists with a relatively high percentage of lyrics misclassified as poetry tend to be recognized for their poetic style, such as Bob Dylan and Annie Lennox.

## 2. RELATED WORK

We do not know of any work on the classification of documents based on the adjective usage into lyrics, poetry or articles nor are we aware of any computational



© Abhishek Singhi, Daniel G. Brown.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Abhishek Singhi, Daniel G. Brown. "Are Poetry And Lyrics All That Different?", 15th International Society for Music Information Retrieval Conference, 2014.

work which discerns poetic from non-poetic lyricists. Previous works have used adjectives for various purposes like sentiment analysis [1]. Furthermore in Music Information Retrieval, work on poetry has focused on poetry translator, automatic poetry generation.

Chesley et al. [1] classifies blog posts according to sentiment using verb classes and adjective polarity, achieving accuracy levels of 72.4% on objective posts, 84.2% for positive posts, and 80.3% for negative posts. Entwisle et al. [2] analyzes the free verbal productions of ninth-grade males and females and conclude that girls use more adjectives than boys but fail to reveal differential use of qualifiers by social class.

Smith et al. [13] use of tf-idf weighting to find typical phrases and rhyme pairs in song lyrics and conclude that the typical number one hits, on average, are more clichéd. Nichols et al. [14] studies the relationship between lyrics and melody on a large symbolic database of popular music and conclude that songwriters tend to align salient notes with salient lyrics.

There is some existing work on automatic generation of synonyms. Zhou et al. [3] extracts synonyms using three sources - a monolingual dictionary, a bilingual corpus and a monolingual corpus, and use a weighted ensemble to combine the synonyms produced from the three sources. They get improved results when compared to the manually built thesauri, WordNet and Roget.

Christian et al. [4] describe an approach for using Wikipedia to automatically build a dictionary of named entities and their synonyms. They were able to extract a large amount of entities with a high precision, and the synonyms found were mostly relevant, but in some cases the number of synonyms was very high. Niemi et al. [5] add new synonyms to the existing synsets of the Finnish WordNet using Wikipedia's links between the articles of the same topic in Finnish and English.

As to computational poetry, Jiang et al. [6] use statistical machine translation to generate Chinese couplets while Genzel et al. [7] use statistical machine translation to translate poetry keeping the rhyme and meter constraints.

### 3. DATA SET

The training set consists of articles, lyrics and poetry and is used to calculate the probability distribution of adjectives in the three different types of documents. We use these probability distributions in our document classification algorithms, to identify poetic from non-poetic lyricists and to determine adjectives more likely to be used in lyrics rather than poetry and vice versa.

#### 3.1 Articles

We take the English Wikipedia and over 13,000 news articles from four major newspapers as our article data set. Wikipedia, an enormous and freely available data set is

edited by experts. Both of these are extremely rich sources of data on many topics. To remove the influence of the presence of articles about poems and lyrics in Wikipedia we set the pruning threshold frequency of adjectives to a high value, and we ensured that the articles were not about poetry or music.

#### 3.2 Lyrics

We took more than 10,000 lyrics from 56 very popular English singers. Both the authors listen to English music and hence it was easy to come up with a list which included singers from many popular genres with diverse backgrounds. We focus on English-language popular music in our study, because it is the closest to "universally" popular music, due to the strength of the music industry in English-speaking countries. We do not know if our work would generalize to non-English Language songs. Our data set includes lyrics from the US, Canada, UK and Ireland.

#### 3.3 Poetry

We took more than 20,000 poems from more than 60 famous poets, like Robert Frost, William Blake and John Keats, over the last three hundred years. We selected the top poets from Poem Hunter [19]. We selected a wide time range for the poets, as many of the most famous English poets are from that time period. None of the poetry selected were translations from another language. Most of the poets in our dataset are poets from North America and Europe. We believe that our training data, is representative of the mean, as a majority of poetry and poetic style are inspired by the work of these few extremely famous poets.

#### 3.4 Test Data

For the purpose of document classification we took 100 from each category, ensuring that they were not present in the training set. While collecting the test data we ensured the diversity, the lyrics and poets came from different genres and artists and the articles covered different topics and were selected from different newspapers.

To determine poetic lyricists from non-poetic ones we took eight of each of the two types of lyricists, none of whom were present in our lyrics data sets. We ensured that the poetic lyricists we selected were indeed poetic by looking up popular news articles or ensuring that they were poet along with being lyricists. Our list for poetic lyricists included Bob Dylan and Annie Lennox etc. while the non-poetic ones included Bryan Adams and Michael Jackson.

## 4. METHOD

These are the main steps in our method:



- 1) Finding the synonyms of all the words in the training data set.
- 2) Finding the probability distribution of word for all the three types of documents.
- 3) The document classification algorithm.

#### 4.1 Extracting Synonyms

We extract the synonyms for a term from three sources: WordNet, Wikipedia and an online thesaurus.

**WordNet** is a large lexical database of English where words are grouped into sets of cognitive synonyms (synsets) together based on their meanings. WordNet interlinks not just word forms but specific senses of words. As a result, words that are found in close proximity to one another in the network are semantically disambiguated. The synonyms returned by WordNet need some pruning.

We use **Wikipedia** redirects to discover terms that are mostly synonymous. It returns a large number of words, which might not be synonyms, so we need to prune the results. This method has been widely used for obtaining the synonyms of named entities *e.g.* [4], but we get decent results for adjectives too.

We also used an **online Thesaurus** that lists words grouped together according to similarity of meaning. Though it gives very accurate synonyms, pruning is necessary to get better results.

We prune synonyms obtained from the three sources, which fall below an experimentally determined threshold for the semantic distance between the synonyms and the word. To calculate the semantic similarity distance between words we use the method described by Pirro et al. [8]. Extracting synonyms for a given word is an open problem and with improvement in this area our algorithm will achieve better classification accuracy levels.

#### 4.2 Probability Distribution

We believe that the choice of an adjective to express a given concept depends on the genre of writing: adjectives used in lyrics will be different from ones used in poems or in articles. We calculate the probability of a specific adjective for each of the three document types.

First, WordNet is used to identify the adjectives in our training sets. For each adjective we compute the frequency of that were in the training set and the frequency of it and its synonyms; the ratio of these is the frequency with which that adjective represents its synonym group in that class of writing.

We exclude adjectives that occur infrequently (fewer than 5 times in our lyrics/poetry set or 50 in articles). The enormous size of the Wikipedia justifies the high threshold value.

#### 4.3 Document classification algorithm

We use a simple linear time algorithm which takes as input the probability distributions for adjectives, calculated

above, and the document(s) to be classified, calculates the score of the document being an article, lyrics or poetry, and labels it with the class with the highest score. The algorithm takes a single pass along the whole document and identifies adjectives using WordNet.

For each word in the document we check its presence in our word list. If found, we add the probability to the score, with a special penalty of -1 for adjectives never found in the training set and a special bonus of +1 for words with probability 1. The penalty and boosting values used in the algorithm were determined experimentally. Surprisingly, this simple approach gives us much better accuracy rates than Naïve Bayes, which we thought would be a good option since it is widely used in classification tasks like spam filtering. We have decent accuracy rates with this simple, naïve algorithm; one future task could be to come up with a better classifier.

## 5. RESULTS

First, we look at the classification accuracies between lyrics, articles and poems obtained by our classifier. We show that the adjectives used in lyrics are much more rhymable than the ones used in poems but they do not differ significantly in their semantic orientations. Furthermore, our algorithm is able to identify poetic lyricists from non-poetic ones using the word distributions, calculated in earlier section. We also compare adjectives for a given concepts which are more likely to be used in lyrics rather than poetry and vice versa.

### 5.1 Document Classification

Our test set consists of the text of 100 each of our three categories. Using our algorithm with the adjective distributions we get an accuracy of 67% for lyrics, 80% for articles and 57% for poems.

The confusion matrix, Table 1 we find the best accuracy for articles. This might be because of the enormous size of the article training set which consisted of all English Wikipedia articles. A slightly more number of articles get misclassified as lyrics than poetry.

Surprisingly, a large number of misclassified poems get classified as articles rather than poetry, but most misclassified lyrics get classified as poems.

### 5.2 Adjective Usage in Lyrics versus Poems

Poetry is written against a silent background while lyrics are often written keeping the melody, rhythm, instrumentation, the quality of the singer's voice and other qualities of the recording in mind. Furthermore, unlike most poetry, lyrics include repeated lines. This led us to believe the adjectives which were more likely to be used in lyrics rather than poetry would be more rhymable.

We counted the number of words an adjective in our lyrics and poetry list rhymes with from the website rhymezone.com. The values are tabulated in Table 2.

From the values in Table 2, we can clearly see that the adjectives which are more likely to be used in lyrics to be much more rhymable than the adjectives which are more likely to be used in poetry.

	Predicted		
Actual	Lyrics	Articles	Poems
Lyrics	67	11	22
Articles	11	80	6
Poems	10	33	57

**Table 1.** The confusion matrix for document classification. Many lyrics are categorized as poems, and many poems as articles.

	Lyrics	Poetry
Mean	33.2	22.9
Median	11	5
25 <sup>th</sup> percentile	2	0
75 <sup>th</sup> percentile	38	24

**Table 2.** Statistical values for the number of words an adjective rhymes with.

	Lyrics	Poetry
Mean	-.05	-.053
Median	0.0	0.0
25 <sup>th</sup> percentile	-0.27	-0.27
75 <sup>th</sup> percentile	0.13	0.13

**Table 3.** Statistical values for the semantic orientation of adjectives used in lyrics and poetry.

We were also interested in finding if the adjectives used in lyrics and poetry differed significantly in their semantic orientations. SentiWordNet assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity. We calculated the semantic orientations, which take a value between -1 and +1, using SentiWordNet, of all the adjectives in the lyrics and poetry list, the values are in Table 3. They show no difference between adjectives in poetry and those in lyrics.

### 5.3 Poetic vs non-Poetic Lyricists

There are lyricists like Bob Dylan [15], Ani DiFranco [16], and Stephen Sondheim [17,18], whose lyrics are considered to be poetic, or indeed, who are published poets in some cases. The lyrics of such poetic lyricists possibly could be structurally more constrained than a majority of the lyrics or might adhere to a particular meter and style. While selecting the poetic lyricists we ensured that popular articles supported our claim or by going to their Wikipedia page and ensuring that they were poets along with being lyricists and hence the influence of their poetry on lyrics.

Our algorithm consistently misclassifies a large fraction of the lyrics of such poetic lyricists as poetry while the percentage of misclassified lyrics as poetry for the non-poetic lyricists is significantly much less. These values for poetic and non-poetic lyricists are tabulated in table 4 and table 5 respectively.

Poetic Lyricists	% of lyrics misclassified as poetry
Bob Dylan	42%
Ed Sheeran	50%
Ani Di Franco	29%
Annie Lennox	32%
Bill Callahan	34%
Bruce Springsteen	29%
Stephen Sondheim	40%
Morrissey	29%
<b>Average misclassification rate</b>	<b>36%</b>

**Table 4.** Percentage of misclassified lyrics as poetry for poetic lyricists.

Non-Poetic Lyricists	% of lyrics misclassified as poetry
Bryan Adams	14%
Michael Jackson	22%
Drake	7%
Backstreet Boys	23%
Radiohead	26%
Stevie Wonder	17%
Led Zeppelin	8%
Kesha	18%
<b>Average misclassification rate</b>	<b>17%</b>

**Table 5.** Percentage of misclassified lyrics as poetry for non-poetic lyricists.

From the values in table 4 and 5 we see that there is a clear separation between the misclassification rate between poetic and non-poetic lyricists. The maximum misclassification rate for the non-poetic lyricists i.e. 26% is less than the minimum mis-classification rate for poetic lyricists i.e. 29%. Furthermore the difference in average misclassification rate between the two groups of lyricists is 19%. Hence our simple algorithm can accurately identify poetic lyricists from non-poetic ones, based only on adjective usage.

### 5.4 Concept representation in Lyrics vs Poetry

We compare adjective uses for common concepts. To represent physical beauty we are more likely to use words like “sexy” and “hot” in lyrics but “gorgeous” and “handsome” in poetry. For 20 of these, results are tabulated in Table 6. The difference could possibly be because unlike lyrics, which are written for the masses, poetry is generally written for people who are interested in literature. It

has been shown that the typical number one hits, on average, are more clichéd [13].

Lyrics	Poetry
proud, arrogant, cocky	haughty, imperious
sexy, hot, beautiful, cute	gorgeous, handsome
merry, ecstatic, elated	happy, blissful, joyous
heartbroken, brokenhearted	sad, sorrowful, dismal
real	genuine
smart	wise, intelligent
bad, shady	lousy, immoral, dishonest
mad, outrageous	wrathful, furious
royal	noble, aristocratic, regal
pissed	angry, bitter
greedy	selfish
cheesy	poor, worthless
lethal, dangerous, fatal	mortal, harmful, destructive
afraid, nervous	frightened, cowardly, timid
jealous	envious, covetous
lax, sloppy	lenient, indifferent
weak, fragile	feeble, powerless
black	ebon
naïve, ignorant	innocent, guileless, callow
corny	dull, stale

**Table 6.** For twenty different concepts, we compare adjectives which are more likely to be used in lyrics rather than poetry and vice versa.

## 6. APPLICATIONS

The algorithm developed has many practical applications in Music Information Retrieval (MIR). They could be used for automatic poetry/lyrics generation to identify adjectives more likely to be used in a particular type of document. As we have shown we can analyze documents, analyze how lyrical, poetic or article-like a document is. For lyricists or poets we can come up with alternate better adjectives to make a document fit its genre better. Using the word distributions we can come up with a better measure of distance between documents where the weights are assigned to a word depending on its probability of usage in a particular type of document. And, of course, our work here can be extended to different genres of writings like prose or fiction.

## 7. CONCLUSION

Our key finding is that the choice of synonym for even a small number of adjectives are sufficient to reliably identify genre of documents. In accordance with our hypothesis, we show that there exist differences in the kind of adjectives used in different genres of writing. We calculate the probability distribution of adjectives over the three kinds of documents and using this distribution and a simple algorithm we are able to distinguish among lyrics, poetry and article with an accuracy of 67%, 57% and 80% respectively.

Adjectives likely to be used in lyrics are more rhymable than the ones used in poetry. This might be because lyrics are written keeping in mind the melody, rhythm, instrumentation, quality of the singer's voice and other qualities of the recording while poetry is without such concerns. There is no significant difference in the semantic orientation of adjectives which are more likely to be used in lyrics and those which are more likely to be used in poetry. Using the probability distributions, obtained from training data, we present adjectives more likely to be used in lyrics rather than poetry and vice versa for twenty common concepts.

Using the probability distributions and our algorithm we show that we can discern poetic lyricists from non-poetic ones. Our algorithm consistently misclassifies a majority of the lyrics of such poetic lyricists as poetry while the percentage of misclassified lyrics as poetry for the non-poetic lyricists is significantly much less.

Calculating the probability distribution of adjectives over the various document types is a vital step in our method which in turn depends on the synonyms extracted for an adjective. Synonym extraction is still an open problem and with improvements in it our algorithm will give better accuracy levels. We extract synonyms from three different sources – Wikipedia, WordNet and an online Thesaurus, and prune the results based on the semantic similarity between the adjectives and the obtained synonyms.

We use a simple naïve algorithm, which gives us better result than Naïve Bayes. An extension to the work can be coming up with an improved version of the algorithm with better accuracy levels. Future works can use a larger dataset for lyrics and poetry (we have an enormous dataset for articles) to come up with better probability distribution for the two document types or to identify parts of speech that effectively separates genres of writing. Our work here can be extended to different genres of writings like prose, fiction etc. to analyze the adjective usage in those writings. It would be interesting to do similar work for verbs and discern if different words, representing the same action, are used in different genres of writings.

## 8. ACKNOWLEDGEMENTS

Our research is supported by a grant from the Natural Sciences and Engineering Research Council of Canada to DGB.

## 9. REFERENCES

- [1] P. Chesley, B. Vincent., L. Xu, and R. Srihari, “Using Verbs and Adjectives to Automatically Classify Blog Sentiment”, *Training*, volume 580, number 263, pages 233, 2006.
- [2] D.R. Entwisle and C. Garvey, “Verbal productivity and adjective usage”, *Language and Speech*, volume 15, number 3, pages 288-298, 1972.
- [3] H. Wu and M. Zhou, “Optimizing Synonym Extraction Using Monolingual and Bilingual Resources”, in *Proceedings of the 2nd International Workshop on Paraphrasing*, volume 16, pages 72-79, 2003.
- [4] C. Bohn and K. Norvag, “Extracting Named Entities and Synonyms from Wikipedia”, in *Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications*, (AINA '10), pages 1300–1307.
- [5] J. Niemi, K. Linden and M. Hyvarinen, “Using a bilingual resource to add synonyms to a wordnet: FinnWordNet and Wikipedia as an example”, in *Proceedings of the Global WordNet Association*, pages 227–231, 2012.
- [6] L. Jiang and M. Zhou, “Generating Chinese couplets using a statistical MT approach”, in *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 377–384, 2008.
- [7] D. Genzel, J. Uszkoreit and F. Och, “Poetic statistical machine translation: rhyme and meter”, in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 158–166, 2010.
- [8] G. Pirro and J. Euzenat, “A Feature and Information Theoretic Framework for Semantic Similarity and Relatedness”, in *Proceedings of the 9th International Semantic Web Conference (ISWC '10)*, pages 615-630, 2010.
- [9] G. Miller, “WordNet: A Lexical Database for English”, *Communications of the ACM*, volume 38, number 11, pages 39-41, 1995.
- [10] G. Miller and F. Christiane, *WordNet: An Electronic Lexical Database*, 1998.
- [11] S. Baccianella, A. Esuli, and F. Sebastiani, “SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining”, in *Proceedings of the 7<sup>th</sup> Conference on International Language Resources and Evaluation (LREC '10)*, pages 2200–2204, 2010.
- [12] A. Esuli and F. Sebastiani, “SentiWordNet: A Publicly Available Lexical Resource for Opinion Mining”, in *Proceedings of the 5<sup>th</sup> Conference on Language Resources and Evaluation (LREC '06)*, pages 417–422, 2006.
- [13] A.G. Smith, C. X. S. Zee and A. L. Uitdenbogerd, “In your eyes: Identifying cliché in song lyrics”, in *Proceedings of the Australasian Language Technology Association Workshop*, pages 88–96, 2012.
- [14] E. Nichols, D. Morris, S. Basu, S. Christopher, “Relationships between lyrics and melody in popular music”, in *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR '09)*, 2009.
- [15] K. Negus, *Bob Dylan*, Equinox London, 2008.
- [16] A. DiFranco, *Verses*, Seven Stories, 2007.
- [17] S. Sondheim, *Look, I Made a Hat!* New York: Knopf, 2011.
- [18] S. Sondheim, *Finishing the Hat*, New York: Knopf, 2010.
- [19] <http://www.poemhunter.com>.

# SINGING-VOICE SEPARATION FROM MONAURAL RECORDINGS USING DEEP RECURRENT NEURAL NETWORKS

Po-Sen Huang<sup>†</sup>, Minje Kim<sup>‡</sup>, Mark Hasegawa-Johnson<sup>†</sup>, Paris Smaragdis<sup>†§</sup>

<sup>†</sup>Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, USA

<sup>‡</sup>Department of Computer Science, University of Illinois at Urbana-Champaign, USA

<sup>§</sup>Adobe Research, USA

{huang146, minje, jhasegaw, paris}@illinois.edu

## ABSTRACT

Monaural source separation is important for many real world applications. It is challenging since only single channel information is available. In this paper, we explore using deep recurrent neural networks for singing voice separation from monaural recordings in a supervised setting. Deep recurrent neural networks with different temporal connections are explored. We propose jointly optimizing the networks for multiple source signals by including the separation step as a nonlinear operation in the last layer. Different discriminative training objectives are further explored to enhance the source to interference ratio. Our proposed system achieves the state-of-the-art performance, 2.30~2.48 dB GNSDR gain and 4.32~5.42 dB GSIR gain compared to previous models, on the MIR-1K dataset.

## 1. INTRODUCTION

Monaural source separation is important for several real-world applications. For example, the accuracy of automatic speech recognition (ASR) can be improved by separating noise from speech signals [10]. The accuracy of chord recognition and pitch estimation can be improved by separating singing voice from music [7]. However, current state-of-the-art results are still far behind human capability. The problem of monaural source separation is even more challenging since only single channel information is available.

In this paper, we focus on singing voice separation from monaural recordings. Recently, several approaches have been proposed to utilize the assumption of the low rank and sparsity of the music and speech signals, respectively [7, 13, 16, 17]. However, this strong assumption may not always be true. For example, the drum sounds may lie in the sparse subspace instead of being low rank. In addition, all these models can be viewed as linear transformations in the spectral domain.

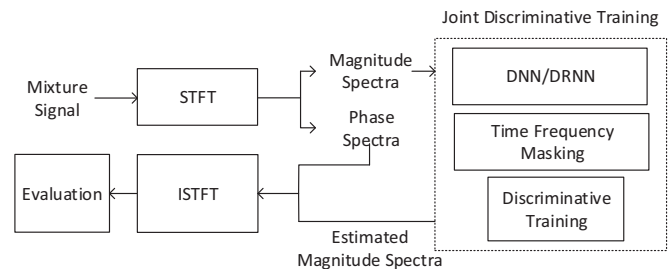


Figure 1. Proposed framework.

With the recent development of deep learning, without imposing additional constraints, we can further extend the model expressibility by using multiple nonlinear layers and learn the optimal hidden representations from data. In this paper, we explore the use of deep recurrent neural networks for singing voice separation from monaural recordings in a supervised setting. We explore different deep recurrent neural network architectures along with the joint optimization of the network and a soft masking function. Moreover, different training objectives are explored to optimize the networks. The proposed framework is shown in Figure 1.

The organization of this paper is as follows: Section 2 discusses the relation to previous work. Section 3 introduces the proposed methods, including the deep recurrent neural networks, joint optimization of deep learning models and a soft time-frequency masking function, and different training objectives. Section 4 presents the experimental setting and results using the MIR-1K dataset. We conclude the paper in Section 5.

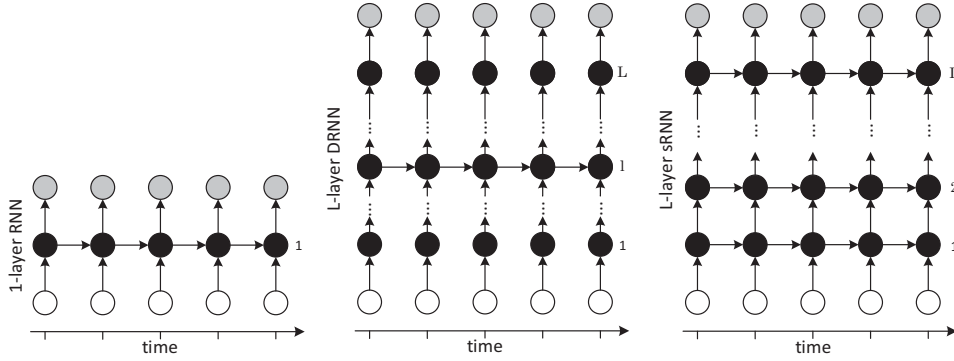
## 2. RELATION TO PREVIOUS WORK

Several previous approaches utilize the constraints of low rank and sparsity of the music and speech signals, respectively, for singing voice separation tasks [7, 13, 16, 17]. Such strong assumption for the signals might not always be true. Furthermore, in the separation stage, these models can be viewed as a single-layer linear network, predicting the clean spectra via a linear transform. To further improve the expressibility of these linear models, in this paper, we use deep learning models to learn the representations from



© Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, Paris Smaragdis.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, Paris Smaragdis. "Singing-Voice Separation From Monaural Recordings Using Deep Recurrent Neural Networks", 15th International Society for Music Information Retrieval Conference, 2014.



**Figure 2.** Deep Recurrent Neural Networks (DRNNs) architectures: Arrows represent connection matrices. Black, white, and grey circles represent input frames, hidden states, and output frames, respectively. (Left): standard recurrent neural networks; (Middle):  $L$  intermediate layer DRNN with recurrent connection at the  $l$ -th layer. (Right):  $L$  intermediate layer DRNN with recurrent connections at all levels (called stacked RNN).

data, without enforcing low rank and sparsity constraints.

By exploring deep architectures, deep learning approaches are able to discover the hidden structures and features at different levels of abstraction from data [5]. Deep learning methods have been applied to a variety of applications and yielded many state of the art results [2, 4, 8]. Recently, deep learning techniques have been applied to related tasks such as speech enhancement and ideal binary mask estimation [1, 9–11, 15].

In the ideal binary mask estimation task, Narayanan and Wang [11] and Wang and Wang [15] proposed a two-stage framework using deep neural networks. In the first stage, the authors use  $d$  neural networks to predict each output dimension separately, where  $d$  is the target feature dimension; in the second stage, a classifier (one layer perceptron or an SVM) is used for refining the prediction given the output from the first stage. However, the proposed framework is not scalable when the output dimension is high. For example, if we want to use spectra as targets, we would have 513 dimensions for a 1024-point FFT. It is less desirable to train such large number of neural networks. In addition, there are many redundancies between the neural networks in neighboring frequencies. In our approach, we propose a general framework that can jointly predict all feature dimensions at the same time using one neural network. Furthermore, since the outputs of the prediction are often smoothed out by time-frequency masking functions, we explore jointly training the masking function with the networks.

Maas et al. proposed using a deep RNN for robust automatic speech recognition tasks [10]. Given a noisy signal  $\mathbf{x}$ , the authors apply a DRNN to learn the clean speech  $\mathbf{y}$ . In the source separation scenario, we found that modeling one target source in the denoising framework is suboptimal compared to the framework that models all sources. In addition, we can use the information and constraints from different prediction outputs to further perform masking and discriminative training.

### 3. PROPOSED METHODS

#### 3.1 Deep Recurrent Neural Networks

To capture the contextual information among audio signals, one way is to concatenate neighboring features together as input features to the deep neural network. However, the number of parameters increases rapidly according to the input dimension. Hence, the size of the concatenating window is limited. A recurrent neural network (RNN) can be considered as a DNN with indefinitely many layers, which introduce the memory from previous time steps. The potential weakness for RNNs is that RNNs lack hierarchical processing of the input at the current time step. To further provide the hierarchical information through multiple time scales, deep recurrent neural networks (DRNNs) are explored [3, 12]. DRNNs can be explored in different schemes as shown in Figure 2. The left of Figure 2 is a standard RNN, folded out in time. The middle of Figure 2 is an  $L$  intermediate layer DRNN with temporal connection at the  $l$ -th layer. The right of Figure 2 is an  $L$  intermediate layer DRNN with full temporal connections (called stacked RNN (sRNN) in [12]).

Formally, we can define different schemes of DRNNs as follows. Suppose there is an  $L$  intermediate layer DRNN with the recurrent connection at the  $l$ -th layer, the  $l$ -th hidden activation at time  $t$  is defined as:

$$\begin{aligned} \mathbf{h}_t^l &= f_h(\mathbf{x}_t, \mathbf{h}_{t-1}^l) \\ &= \phi_l(\mathbf{U}^l \mathbf{h}_{t-1}^l + \mathbf{W}^l \phi_{l-1}(\mathbf{W}^{l-1}(\dots \phi_1(\mathbf{W}^1 \mathbf{x}_t))))), \end{aligned} \quad (1)$$

and the output,  $\mathbf{y}_t$ , can be defined as:

$$\begin{aligned} \mathbf{y}_t &= f_o(\mathbf{h}_t^l) \\ &= \mathbf{W}^L \phi_{L-1}(\mathbf{W}^{L-1}(\dots \phi_l(\mathbf{W}^l \mathbf{h}_t^l))), \end{aligned} \quad (2)$$

where  $\mathbf{x}_t$  is the input to the network at time  $t$ ,  $\phi_l$  is an element-wise nonlinear function,  $\mathbf{W}^l$  is the weight matrix

for the  $l$ -th layer, and  $\mathbf{U}^l$  is the weight matrix for the recurrent connection at the  $l$ -th layer. The output layer is a linear layer.

The stacked RNNs have multiple levels of transition functions, defined as:

$$\begin{aligned} \mathbf{h}_t^l &= f_h(\mathbf{h}_t^{l-1}, \mathbf{h}_{t-1}^l) \\ &= \phi_l(\mathbf{U}^l \mathbf{h}_{t-1}^l + \mathbf{W}^l \mathbf{h}_t^{l-1}), \end{aligned} \quad (3)$$

where  $\mathbf{h}_t^l$  is the hidden state of the  $l$ -th layer at time  $t$ .  $\mathbf{U}^l$  and  $\mathbf{W}^l$  are the weight matrices for the hidden activation at time  $t-1$  and the lower level activation  $\mathbf{h}_t^{l-1}$ , respectively. When  $l=1$ , the hidden activation is computed using  $\mathbf{h}_t^0 = \mathbf{x}_t$ .

Function  $\phi_l(\cdot)$  is a nonlinear function, and we empirically found that using the rectified linear unit  $f(\mathbf{x}) = \max(0, \mathbf{x})$  [2] performs better compared to using a sigmoid or tanh function. For a DNN, the temporal weight matrix  $\mathbf{U}^l$  is a zero matrix.

### 3.2 Model Architecture

At time  $t$ , the training input,  $\mathbf{x}_t$ , of the network is the concatenation of features from a mixture within a window. We use magnitude spectra as features in this paper. The output targets,  $\mathbf{y}_{1t}$  and  $\mathbf{y}_{2t}$ , and output predictions,  $\hat{\mathbf{y}}_{1t}$  and  $\hat{\mathbf{y}}_{2t}$ , of the network are the magnitude spectra of different sources.

Since our goal is to separate one of the sources from a mixture, instead of learning one of the sources as the target, we adapt the framework from [9] to model all different sources simultaneously. Figure 3 shows an example of the architecture.

Moreover, we find it useful to further smooth the source separation results with a time-frequency masking technique, for example, binary time-frequency masking or soft time-frequency masking [7, 9]. The time-frequency masking function enforces the constraint that the sum of the prediction results is equal to the original mixture.

Given the input features,  $\mathbf{x}_t$ , from the mixture, we obtain the output predictions  $\hat{\mathbf{y}}_{1t}$  and  $\hat{\mathbf{y}}_{2t}$  through the network. The soft time-frequency mask  $\mathbf{m}_t$  is defined as follows:

$$\mathbf{m}_t(f) = \frac{|\hat{\mathbf{y}}_{1t}(f)|}{|\hat{\mathbf{y}}_{1t}(f)| + |\hat{\mathbf{y}}_{2t}(f)|}, \quad (4)$$

where  $f \in \{1, \dots, F\}$  represents different frequencies.

Once a time-frequency mask  $\mathbf{m}_t$  is computed, it is applied to the magnitude spectra  $\mathbf{z}_t$  of the mixture signals to obtain the estimated separation spectra  $\hat{\mathbf{s}}_{1t}$  and  $\hat{\mathbf{s}}_{2t}$ , which correspond to sources 1 and 2, as follows:

$$\begin{aligned} \hat{\mathbf{s}}_{1t}(f) &= \mathbf{m}_t(f) \mathbf{z}_t(f) \\ \hat{\mathbf{s}}_{2t}(f) &= (1 - \mathbf{m}_t(f)) \mathbf{z}_t(f), \end{aligned} \quad (5)$$

where  $f \in \{1, \dots, F\}$  represents different frequencies.

The time-frequency masking function can be viewed as a layer in the neural network as well. Instead of training the network and applying the time-frequency masking to the results separately, we can jointly train the deep learning models with the time-frequency masking functions. We

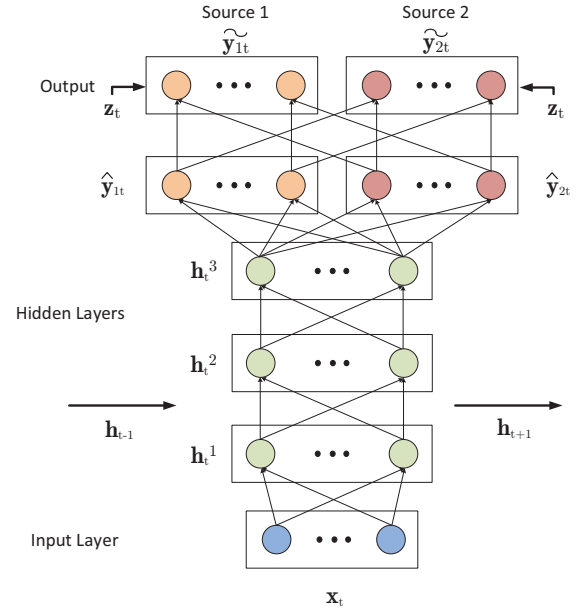


Figure 3. Proposed neural network architecture.

add an extra layer to the original output of the neural network as follows:

$$\begin{aligned} \tilde{\mathbf{y}}_{1t} &= \frac{|\hat{\mathbf{y}}_{1t}|}{|\hat{\mathbf{y}}_{1t}| + |\hat{\mathbf{y}}_{2t}|} \odot \mathbf{z}_t \\ \tilde{\mathbf{y}}_{2t} &= \frac{|\hat{\mathbf{y}}_{2t}|}{|\hat{\mathbf{y}}_{1t}| + |\hat{\mathbf{y}}_{2t}|} \odot \mathbf{z}_t, \end{aligned} \quad (6)$$

where the operator  $\odot$  is the element-wise multiplication (Hadamard product). In this way, we can integrate the constraints to the network and optimize the network with the masking function jointly. Note that although this extra layer is a deterministic layer, the network weights are optimized for the error metric between and among  $\tilde{\mathbf{y}}_{1t}$ ,  $\tilde{\mathbf{y}}_{2t}$  and  $\mathbf{y}_{1t}$ ,  $\mathbf{y}_{2t}$ , using back-propagation. To further smooth the predictions, we can apply masking functions to  $\tilde{\mathbf{y}}_{1t}$  and  $\tilde{\mathbf{y}}_{2t}$ , as in Eqs. (4) and (5), to get the estimated separation spectra  $\tilde{\mathbf{s}}_{1t}$  and  $\tilde{\mathbf{s}}_{2t}$ . The time domain signals are reconstructed based on the inverse short time Fourier transform (ISTFT) of the estimated magnitude spectra along with the original mixture phase spectra.

### 3.3 Training Objectives

Given the output predictions  $\hat{\mathbf{y}}_{1t}$  and  $\hat{\mathbf{y}}_{2t}$  (or  $\tilde{\mathbf{y}}_{1t}$  and  $\tilde{\mathbf{y}}_{2t}$ ) of the original sources  $\mathbf{y}_{1t}$  and  $\mathbf{y}_{2t}$ , we explore optimizing neural network parameters by minimizing the squared error and the generalized Kullback-Leibler (KL) divergence criteria, as follows:

$$J_{MSE} = \|\hat{\mathbf{y}}_{1t} - \mathbf{y}_{1t}\|_2^2 + \|\hat{\mathbf{y}}_{2t} - \mathbf{y}_{2t}\|_2^2 \quad (7)$$

and

$$J_{KL} = D(\mathbf{y}_{1t} \|\hat{\mathbf{y}}_{1t}) + D(\mathbf{y}_{2t} \|\hat{\mathbf{y}}_{2t}), \quad (8)$$

where the measure  $D(A\|B)$  is defined as:

$$D(A\|B) = \sum_i \left( A_i \log \frac{A_i}{B_i} - A_i + B_i \right). \quad (9)$$

$D(\cdot||\cdot)$  reduces to the KL divergence when  $\sum_i A_i = \sum_i B_i = 1$ , so that  $A$  and  $B$  can be regarded as probability distributions.

Furthermore, minimizing Eqs. (7) and (8) is for increasing the similarity between the predictions and the targets. Since one of the goals in source separation problems is to have high signal to interference ratio (SIR), we explore discriminative objective functions that not only increase the similarity between the prediction and its target, but also decrease the similarity between the prediction and the targets of other sources, as follows:

$$\|\hat{\mathbf{y}}_{1_t} - \mathbf{y}_{1_t}\|_2^2 - \gamma \|\hat{\mathbf{y}}_{1_t} - \mathbf{y}_{2_t}\|_2^2 + \|\hat{\mathbf{y}}_{2_t} - \mathbf{y}_{2_t}\|_2^2 - \gamma \|\hat{\mathbf{y}}_{2_t} - \mathbf{y}_{1_t}\|_2^2 \quad (10)$$

and

$$D(\mathbf{y}_{1_t} || \hat{\mathbf{y}}_{1_t}) - \gamma D(\mathbf{y}_{1_t} || \hat{\mathbf{y}}_{2_t}) + D(\mathbf{y}_{2_t} || \hat{\mathbf{y}}_{2_t}) - \gamma D(\mathbf{y}_{2_t} || \hat{\mathbf{y}}_{1_t}), \quad (11)$$

where  $\gamma$  is a constant chosen by the performance on the development set.

## 4. EXPERIMENTS

### 4.1 Setting

Our system is evaluated using the MIR-1K dataset [6].<sup>1</sup> A thousand song clips are encoded with a sample rate of 16 KHz, with durations from 4 to 13 seconds. The clips were extracted from 110 Chinese karaoke songs performed by both male and female amateurs. There are manual annotations of the pitch contours, lyrics, indices and types for unvoiced frames, and the indices of the vocal and non-vocal frames. Note that each clip contains the singing voice and the background music in different channels. Only the singing voice and background music are used in our experiments.

Following the evaluation framework in [13, 17], we use 175 clips sung by one male and one female singer ('ab-jones' and 'amy') as the training and development set.<sup>2</sup> The remaining 825 clips of 17 singers are used for testing. For each clip, we mixed the singing voice and the background music with equal energy (i.e. 0 dB SNR). The goal is to separate the singing voice from the background music.

To quantitatively evaluate source separation results, we use Source to Interference Ratio (SIR), Source to Artifacts Ratio (SAR), and Source to Distortion Ratio (SDR) by BSS-EVAL 3.0 metrics [14]. The Normalized SDR (NSDR) is defined as:

$$\text{NSDR}(\hat{\mathbf{v}}, \mathbf{v}, \mathbf{x}) = \text{SDR}(\hat{\mathbf{v}}, \mathbf{v}) - \text{SDR}(\mathbf{x}, \mathbf{v}), \quad (12)$$

where  $\hat{\mathbf{v}}$  is the resynthesized singing voice,  $\mathbf{v}$  is the original clean singing voice, and  $\mathbf{x}$  is the mixture. NSDR is for estimating the improvement of the SDR between the preprocessed mixture  $\mathbf{x}$  and the separated singing voice  $\hat{\mathbf{v}}$ . We report the overall performance via Global NSDR

(GNSDR), Global SIR (GSIR), and Global SAR (GSAR), which are the weighted means of the NSDRs, SIRs, SARs, respectively, over all test clips weighted by their length. Higher values of SDR, SAR, and SIR represent better separation quality. The suppression of the interfering source is reflected in SIR. The artifacts introduced by the separation process are reflected in SAR. The overall performance is reflected in SDR.

For training the network, in order to increase the variety of training samples, we circularly shift (in the time domain) the singing voice signals and mix them with the background music.

In the experiments, we use magnitude spectra as input features to the neural network. The spectral representation is extracted using a 1024-point short time Fourier transform (STFT) with 50% overlap. Empirically, we found that using log-mel filterbank features or log power spectrum provide worse performance.

For our proposed neural networks, we optimize our models by back-propagating the gradients with respect to the training objectives. The limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm is used to train the models from random initialization. We set the maximum epoch to 400 and select the best model according to the development set. The sound examples and more details of this work are available online.<sup>3</sup>

### 4.2 Experimental Results

In this section, we compare different deep learning models from several aspects, including the effect of different input context sizes, the effect of different circular shift steps, the effect of different output formats, the effect of different deep recurrent neural network structures, and the effect of the discriminative training objectives.

For simplicity, unless mentioned explicitly, we report the results using 3 hidden layers of 1000 hidden units neural networks with the mean squared error criterion, joint masking training, and 10K samples as the circular shift step size using features with a context window size of 3 frames. We denote the DRNN- $k$  as the DRNN with the recurrent connection at the  $k$ -th hidden layer. We select the models based on the GNSDR results on the development set.

First, we explore the case of using single frame features, and the cases of concatenating neighboring 1 and 2 frames as features (context window sizes 1, 3, and 5, respectively). Table 1 reports the results using DNNs with context window sizes 1, 3, and 5. We can observe that concatenating neighboring 1 frame provides better results compared with the other cases. Hence, we fix the context window size to be 3 in the following experiments.

Table 2 shows the difference between different circular shift step sizes for deep neural networks. We explore the cases without circular shift and the circular shift with a step size of {50K, 25K, 10K} samples. We can observe that the separation performance improves when the number of training samples increases (i.e. the step size of circular

<sup>1</sup> <https://sites.google.com/site/unvoicedsoundseparation/mir-1k>

<sup>2</sup> Four clips, abjones\_5\_08, abjones\_5\_09, amy\_9\_08, amy\_9\_09, are used as the development set for adjusting hyper-parameters.

<sup>3</sup> <https://sites.google.com/site/deeplearningsourceseparation/>



Model (context window size)	GNSDR	GSIR	GSAR
DNN (1)	6.63	10.81	9.77
DNN (3)	6.93	10.99	10.15
DNN (5)	6.84	10.80	10.18

**Table 1.** Results with input features concatenated from different context window sizes.

Model (circular shift step size)	GNSDR	GSIR	GSAR
DNN (no shift)	6.30	9.97	9.99
DNN (50,000)	6.62	10.46	10.07
DNN (25,000)	6.86	11.01	10.00
DNN (10,000)	6.93	10.99	10.15

**Table 2.** Results with different circular shift step sizes.

Model (num. of output sources, joint mask)	GNSDR	GSIR	GSAR
DNN (1, no)	5.64	8.87	9.73
DNN (2, no)	6.44	9.08	11.26
DNN (2, yes)	6.93	10.99	10.15

**Table 3.** Deep neural network output layer comparison using single source as a target and using two sources as targets (with and without joint mask training). In the “joint mask” training, the network training objective is computed after time-frequency masking.

shift decreases). Since the improvement is relatively small when we further increase the number of training samples, we fix the circular shift size to be 10K samples.

Table 3 presents the results with different output layer formats. We compare using single source as a target (row 1) and using two sources as targets in the output layer (row 2 and row 3). We observe that modeling two sources simultaneously provides better performance. Comparing row 2 and row 3 in Table 3, we observe that using the joint mask training further improves the results.

Table 4 presents the results of different deep recurrent neural network architectures (DNN, DRNN with different recurrent connections, and sRNN) and the results of different objective functions. We can observe that the models with the generalized KL divergence provide higher GSARs, but lower GSIRs, compared to the models with the mean squared error objective. Both objective functions provide similar GNSDRs. For different network architectures, we can observe that DRNN with recurrent connection at the second hidden layer provides the best results. In addition, all the DRNN models achieve better results compared to DNN models by utilizing temporal information.

Table 5 presents the results of different deep recurrent neural network architectures (DNN, DRNN with different recurrent connections, and sRNN) with and without discriminative training. We can observe that discriminative training improves GSIR, but decreases GSAR. Overall, GNSDR is slightly improved.

Model (objective)	GNSDR	GSIR	GSAR
DNN (MSE)	6.93	10.99	10.15
DRNN-1 (MSE)	7.11	11.74	9.93
DRNN-2 (MSE)	7.27	11.98	9.99
DRNN-3 (MSE)	7.14	11.48	10.15
sRNN (MSE)	7.09	11.72	9.88
DNN (KL)	7.06	11.34	10.07
DRNN-1 (KL)	7.09	11.48	10.05
DRNN-2 (KL)	7.27	11.35	10.47
DRNN-3 (KL)	7.10	11.14	10.34
sRNN (KL)	7.16	11.50	10.11

**Table 4.** The results of different architectures and different objective functions. The “MSE” denotes the mean squared error and the “KL” denotes the generalized KL divergence criterion.

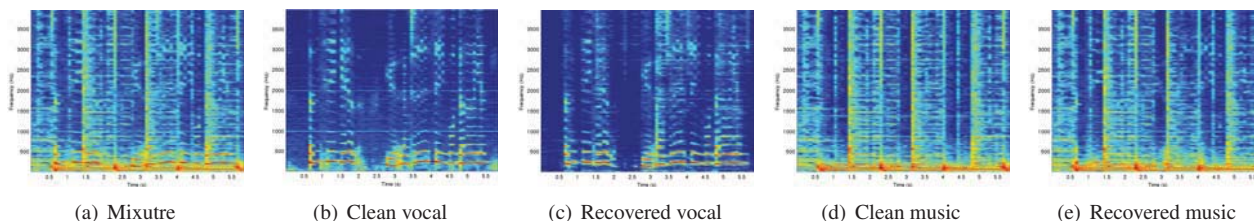
Model	GNSDR	GSIR	GSAR
DNN	6.93	10.99	10.15
DRNN-1	7.11	11.74	9.93
DRNN-2	7.27	11.98	9.99
DRNN-3	7.14	11.48	10.15
sRNN	7.09	11.72	9.88
DNN + discrim	7.09	12.11	9.67
DRNN-1 + discrim	7.21	12.76	9.56
DRNN-2 + discrim	7.45	13.08	9.68
DRNN-3 + discrim	7.09	11.69	10.00
sRNN + discrim	7.15	12.79	9.39

**Table 5.** The comparison for the effect of discriminative training using different architectures. The “discrim” denotes the models with discriminative training.

Finally, we compare our best results with other previous work under the same setting. Table 6 shows the results with unsupervised and supervised settings. Our proposed models achieve 2.30~2.48 dB GNSDR gain, 4.32~5.42 dB GSIR gain with similar GSAR performance, compared with the RNMF model [13]. An example of the separation results is shown in Figure 4.

## 5. CONCLUSION AND FUTURE WORK

In this paper, we explore using deep learning models for singing voice separation from monaural recordings. Specifically, we explore different deep learning architectures, including deep neural networks and deep recurrent neural networks. We further enhance the results by jointly optimizing a soft mask function with the networks and exploring the discriminative training criteria. Overall, our proposed models achieve 2.30~2.48 dB GNSDR gain and 4.32~5.42 dB GSIR gain, compared to the previous proposed methods, while maintaining similar GSARs. Our proposed models can also be applied to many other applications such as main melody extraction.



**Figure 4.** (a) The mixture (singing voice and music accompaniment) magnitude spectrogram (in log scale) for the clip Ani\_1\_01 in MIR-1K; (b) (d) The groundtruth spectrograms for the two sources; (c) (e) The separation results from our proposed model (DRNN-2 + discrim).

Unsupervised			
Model	GNSDR	GSIR	GSAR
RPCA [7]	3.15	4.43	11.09
RPCAh [16]	3.25	4.52	11.10
RPCAh + FASST [16]	3.84	6.22	9.19
Supervised			
Model	GNSDR	GSIR	GSAR
MLRR [17]	3.85	5.63	10.70
RNMF [13]	4.97	7.66	10.03
<b>DRNN-2</b>	<b>7.27</b>	<b>11.98</b>	<b>9.99</b>
<b>DRNN-2 + discrim</b>	<b>7.45</b>	<b>13.08</b>	<b>9.68</b>

**Table 6.** Comparison between our models and previous proposed approaches. The “discrim” denotes the models with discriminative training.

## 6. ACKNOWLEDGEMENT

We thank the authors in [13] for providing their trained model for comparison. This research was supported by U.S. ARL and ARO under grant number W911NF-09-1-0383. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1053575.

## 7. REFERENCES

- [1] N. Boulanger-Lewandowski, G. Mysore, and M. Hoffman. Exploiting long-term temporal dependencies in NMF using recurrent neural networks with application to source separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.
- [2] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *JMLR W&CP: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, 2011.
- [3] M. Hermans and B. Schrauwen. Training and analysing deep recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 190–198, 2013.
- [4] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, 29:82–97, Nov. 2012.
- [5] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.
- [6] C.-L. Hsu and J.-S.R. Jang. On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(2):310–319, Feb. 2010.
- [7] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 57–60, 2012.
- [8] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck. Learning deep structured semantic models for web search using clickthrough data. In *ACM International Conference on Information and Knowledge Management (CIKM)*, 2013.
- [9] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Deep learning for monaural speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.
- [10] A. L. Maas, Q. V Le, T. M O’Neil, O. Vinyals, P. Nguyen, and A. Y. Ng. Recurrent neural networks for noise reduction in robust ASR. In *INTERSPEECH*, 2012.
- [11] A. Narayanan and D. Wang. Ideal ratio mask estimation using deep neural networks for robust speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2013.
- [12] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio. How to construct deep recurrent neural networks. In *International Conference on Learning Representations*, 2014.
- [13] P. Sprechmann, A. Bronstein, and G. Sapiro. Real-time online singing voice separation from monaural recordings using robust low-rank modeling. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, 2012.
- [14] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(4):1462–1469, July 2006.
- [15] Y. Wang and D. Wang. Towards scaling up classification-based speech separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(7):1381–1390, 2013.
- [16] Y.-H. Yang. On sparse and low-rank matrix decomposition for singing voice separation. In *ACM Multimedia*, 2012.
- [17] Y.-H. Yang. Low-rank representation of both singing voice and music accompaniment via learned dictionaries. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, November 4-8 2013.

# IMPACT OF LISTENING BEHAVIOR ON MUSIC RECOMMENDATION

**Katayoun Farrahi**

Goldsmiths, University of London  
London, UK  
k.farrahi@gold.ac.uk

**Markus Schedl, Andreu Vall, David Hauger, Marko Tkalčič**

Johannes Kepler University  
Linz, Austria  
firstname.lastname@jku.at

## ABSTRACT

The next generation of music recommendation systems will be increasingly intelligent and likely take into account user behavior for more personalized recommendations. In this work we consider user behavior when making recommendations with features extracted from a user's history of listening events. We investigate the impact of listener's behavior by considering features such as play counts, "mainstreamness", and diversity in music taste on the performance of various music recommendation approaches. The underlying dataset has been collected by crawling social media (specifically Twitter) for listening events. Each user's listening behavior is characterized into a three dimensional feature space consisting of play count, "mainstreamness" (i.e. the degree to which the observed user listens to currently popular artists), and diversity (i.e. the diversity of genres the observed user listens to). Drawing subsets of the 28,000 users in our dataset, according to these three dimensions, we evaluate whether these dimensions influence figures of merit of various music recommendation approaches, in particular, collaborative filtering (CF) and CF enhanced by cultural information such as users located in the same city or country.

## 1. INTRODUCTION

Early attempts in collaborative filtering (CF) recommender systems for music content have generally treated all users as equivalent in the algorithm [1]. The predicted score (i.e. the likelihood that the observed user would like the observed music piece) was a weighted average of the  $K$  nearest neighbors in a given similarity space [8]. The only way the users were treated differently was the weight, which reflected the similarity between users. However, users' behavior in the consumption of music (and other multimedia material in general) has more dimensions than just ratings. Recently, there has been an increase of research in music consumption behavior and recommender systems that draw inspiration from psychology research on personality. Personality accounts for the individual difference in

users in their behavioral styles [9]. Studies showed that personality affects rating behavior [6], music genre preferences [11] and taste diversity both in music [11] and other domains (e.g. movies in [2]).

The aforementioned work inspired us to investigate how user features intuitively derived from personality traits affect the performance of a CF recommender system in the music domain. We chose three user features that are arguably proxies of various personality traits for user clustering and fine-tuning of the CF recommender system. The chosen features are *play counts*, *mainstreamness* and *diversity*. Play count is a measure of how often the observed user engages in music listening (intuitively related to extraversion). Mainstreamness is a measure that describes to what degree the observed user prefers currently popular songs or artists over non-popular (and is intuitively related to openness and agreeableness). The diversity feature is a measure of how diverse the observed user's spectrum of listened music is (intuitively related to openness).

In this paper, we consider the music listening behavior of a set of 28,000 users, obtained by crawling and analyzing microblogs. By characterizing users across a three dimensional space of play count, mainstreamness, and diversity, we group users and evaluate various recommendation algorithms across these behavioral features. The goal is to determine whether or not the evaluated behavioral features influence the recommendation algorithms, and if so which directions are most promising. Overall, we find that recommending with collaborative filtering enhanced by continent and country information generally performs best. We also find that recommendations for users with large play counts, higher diversity and mainstreamness values are better.

## 2. RELATED WORK

The presented work stands at the crossroads of personality-inspired user features and recommender systems based on collaborative filtering.

Among various models of personality, the Five-factor model (FFM) is the most widely used and is composed of the following traits: *openness*, *conscientiousness*, *extraversion*, *agreeableness* and *neuroticism* [9]. The personality theory inspired several works in the field of recommender systems. For example, Pu et al. [6] showed that user rating behavior is correlated with personality factors. Tkalčič et al. [13] used FFM factors to calculate similarities in a CF recommender system for images. A study by

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

© 2014 International Society for Music Information Retrieval.

Rentfrow et al. [11] showed that scoring high on certain personality traits is correlated with genre preferences and other listening preferences like diversity. Chen et al. [2] argue that people who score high in openness to new experiences prefer more diverse recommendations than people who score low. The last two studies explore the relations between personality and diversity. In fact, the study of diversity in recommending items has become popular after the publishing of two popular books, *The Long Tail* [4] and *The Filter Bubble* [10]. However, most of the work was focused on the trade-off between recommending diverse and similar items (e.g. in [7]). In our work, we treat diversity not as a way of presenting music items but as a user feature, which is a novel way of addressing the usage of diversity in recommender systems.

The presented work builds on collaborative filtering (CF) techniques that are well established in the recommender systems domain [1]. CF methods have been improved using context information when available [3]. Recently, [12] incorporated geospatial context to improve music recommendations on a dataset gathered through microblog crawling [5]. In the presented work, we advance this work by including personality-inspired user features.

### 3. USER BEHAVIOR MODELING

#### 3.1 Dataset

We use the “Million Musical Tweets Dataset”<sup>1</sup> (MMTD) dataset of music listening activities inferred from microblogs. This dataset is freely available [5], and contains approximately 1,100,000 listening events of 215,000 users listening to a total of 134,000 unique songs by 25,000 artists, collected from Twitter. The data was acquired crawling Twitter and identifying music listening events in tweets, using several databases and rule-based filters. Among others, the dataset contains information on location for each post, which enables location-aware analyses and recommendations. Location is provided both as GPS coordinates and semantic identifiers, including continent, country, state, county, and city.

The MMTD contains a large number of users with only a few listening events. These users are not suitable for reliable recommendation and evaluation. Therefore, we consider a subset of users who had at least five listening events over different artists. This subset consists of 28,000 users.

Basic statistics of the data used in all experiments are given in Table 1. The second column shows the total amount of the entities in the corresponding first row, whereas the right-most six columns show principal statistics based on the number of tweets.

#### 3.2 Behavioral Features

Each user is defined by a set of three behavioral features: play count, diversity, and mainstreamness, defined next. These features are used to group users and to determine how they influence the recommendation process.

**Play count** The play count of a user,  $P(u)$ , is a measure of the quantity of listening events for a user  $u$ . It is computed as the total number of listening events recorded over all time for a given user.

**Diversity** The diversity of a user,  $D(u)$ , can be thought of as a measure which captures the range of listening tastes by the user. It is computed as the total number of unique genres associated with all of the artists listened to by a given user. Genre information was obtained by gathering the top tags from Last.fm for each artist in the collection. We then identified genres within these tags by matching the tags to a selection of 20 genres indicated by Allmusic.com.

**Mainstreamness** The mainstreamness  $M(u)$  is a measure of how mainstream a user  $u$  is in terms of her/his listening behavior. It reflects the share of most popular artists within all the artists user  $u$  has listened to. Users that listen mostly to artists that are popular in a given time window tend to have high  $M(u)$ , while users who listen more to artists that are rarely among the most popular ones tend to score low.

For each time window  $i \in \{1 \dots I\}$  within the dataset (where  $I$  is the number of all time windows in the dataset) we calculated the set of the most popular artists  $A_i$ . We calculated the most popular artists in an observed time period as follows. For the given period we sorted the artists by the aggregate of the listening events they received in a decreasing order. Then, the top  $k$  artists, that cover at least 50% of all the listening events of the observed period are regarded as popular artists. For each user  $u$  in a given time window  $i$  we counted the number of play counts of popular artists  $P_i^p(u)$  and normalized it with all the play counts of that user in the observed time window  $P_i^a(u)$ . The final value  $M(u)$  was aggregated by averaging the partial values for each time window:

$$M(u) = \frac{1}{I} \sum_{i=1}^I \frac{P_i^p(u)}{P_i^a(u)} \quad (1)$$

In our experiments, we investigated time windows of six months and twelve months.

Table 3 shows the correlation between individual user features. No significant correlation was found, except for the mainstreamness using an interval of six months and an interval of twelve months, which is expected.

#### 3.3 User Groups

Each user is characterized by a three dimensional feature vector consisting of  $M(u)$ ,  $D(u)$ ,  $P(u)$ . The distribution of users across these features are illustrated in Figures 1 and 2. In Figure 3, mainstreamness is considered with a 6 month interval. The results illustrate the even distribution of users across these features. Therefore, for grouping users, we consider each feature individually and divide users between groups considering a threshold.

For mainstreamness, we consider the histogram of  $M(u)$  (Figure 2 for a 6 month (top) and 12 month (bottom)) in making the groups. We consider 2 different cases for grouping users. First, we divide the users into 2 groups according to the median value (referred to as M6(12)-median-G1(2)).

<sup>1</sup> <http://www.cp.jku.at/datasets/MMTD>

Level	Amount	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Users	27,778	5	7	10	27.69	17	89,320
Artists	21,397	1	1	2	35.95	9	11,850
Tracks	108,676	1	1	1	7.08	4	2,753
Continents	7	9	4,506	101,400	109,900.00	142,200	374,300
Countries	166	1	12	71	4,633.00	555	151,600
States	872	1	7	40	882.00	195	148,900
Counties	3557	1	2	10	216.20	41	191,900
Cities	15123	1	1	5	50.86	16	148,900

**Table 1.** Basic dataset characteristics, where “Amount” is the number of items, and the statistics correspond to the values of the data.

	<i>RB</i>	<i>C<sub>cnt</sub></i>	<i>C<sub>cry</sub></i>	<i>C<sub>sta</sub></i>	<i>C<sub>cty</sub></i>	<i>C<sub>cit</sub></i>	<i>CF</i>	<i>CC<sub>cnt</sub></i>	<i>CC<sub>cry</sub></i>	<i>CC<sub>sta</sub></i>	<i>CC<sub>cty</sub></i>	<i>CC<sub>cit</sub></i>
P-top10	10.28	<b>11.75</b>	11.1	5.70	5.70	5.70	11.22	10.74	10.47	5.89	5.89	5.89
P-mid5k	1.33	1.75	2.25	2.43	1.46	1.96	4.47	<b>4.59</b>	4.51	3.56	1.96	2.56
P-bottom22k	0.64	0.92	1.10	1.03	0.77	1.07	1.85	<b>1.95</b>	<b>1.95</b>	1.56	0.96	1.16
P-G1	0.45	0.67	0.72	0.68	0.44	0.56	1.13	<b>1.26</b>	1.17	0.78	0.26	0.35
P-G2	0.65	1.32	1.34	1.01	0.69	0.92	1.71	<b>1.78</b>	1.77	1.32	0.80	0.89
P-G3	1.08	2.04	2.02	1.88	1.30	1.73	3.51	<b>3.60</b>	3.59	2.90	1.68	2.16
D-G1	0.64	0.85	1.16	1.04	0.87	0.88	2.22	<b>2.24</b>	2.16	1.59	0.97	0.93
D-G2	0.73	0.93	1.05	1.23	0.84	1.02	2.04	<b>2.21</b>	2.20	1.68	0.98	1.08
D-G3	0.93	1.63	1.49	1.56	0.93	1.41	2.49	2.56	<b>2.59</b>	2.03	1.08	1.54
M6-03-G1	0.50	0.88	0.95	0.96	0.64	0.88	1.76	<b>1.84</b>	<b>1.84</b>	1.43	0.81	1.00
M6-03-G2	1.34	2.73	2.43	2.22	1.49	2.00	3.36	<b>3.50</b>	<b>3.50</b>	2.81	1.67	2.08
M6-median-G1	0.35	0.58	0.62	0.65	0.48	0.61	1.35	<b>1.46</b>	1.45	1.04	0.56	0.66
M6-median-G2	1.25	2.49	2.89	2.25	1.47	1.97	3.14	3.27	<b>3.29</b>	2.67	1.66	2.07
M12-05-G1	1.35	2.02	2.27	2.25	1.50	1.93	2.90	3.02	<b>3.04</b>	2.47	1.54	1.94
M12-05-G2	0.36	0.59	0.69	0.61	0.41	0.57	1.30	<b>1.38</b>	<b>1.38</b>	1.01	0.52	0.66
M12-median-G1	0.36	0.62	0.71	0.64	0.43	0.59	1.41	<b>1.50</b>	<b>1.50</b>	1.10	0.56	0.71
M12-median-G2	1.34	2.09	2.33	2.34	1.57	2.01	3.10	3.24	<b>3.26</b>	2.66	1.67	2.10

**Table 2.** Maximum F-score for all combinations of methods and user sets. *C* refers to the CULT approaches, *CC* to CF\_CULT; *cnt* indicates continent, *cry* country, *sta* state, *cty* county, and *cit* city. The best performing recommenders for a given group are in bold.

Second, we divide users into 2 groups for which borders are defined by a mainstreamness of 0.3 and 0.5, respectively, for the 6 month case and the 12 month case (referred to as M6(12)-03(05)-G1(2)). These values were chosen by considering the histograms in Figure 2 and choosing values which naturally grouped users. For the diversity, we create 3 groups according to the 0.33 and 0.67 percentiles (referred to as D-G1(2,3)). For play counts, we consider 2 different groupings. The first is the same as for diversity, i.e. dividing groups according to the 0.33 and 0.67 percentiles (referred to as P-G1(2,3)). The second splits the users according to the accumulative play counts into the following groups, each of which accounts for approximately a third of all play counts: top 10 users, mid 5,000 users, bottom 22,000 users (referred to as P-top10(mid5k,bottom22k)).

#### 4. RECOMMENDATION MODELS

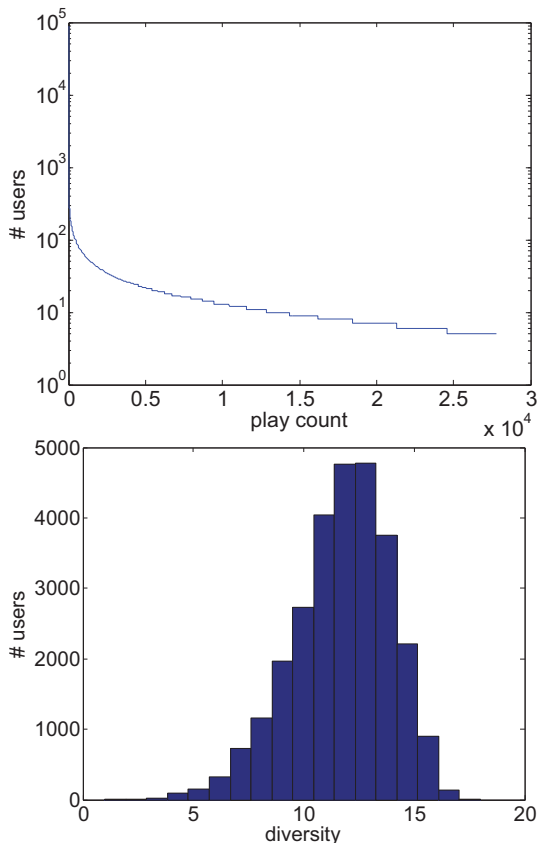
In the considered music recommendation models, each user  $u \in U$  is represented by a list of artists listened to  $A(u)$ . All approaches determine for a given seed user  $u$  a number  $K$  of most similar neighbors  $V_K(u)$ , and recommend the artists listened to by these  $V_K(u)$ , excluding the artists

	<b>D(u)</b>	<b>M(u) (6 mo.)</b>	<b>P(u)</b>
<b>D(u)</b>	-	0.119	0.292
<b>M(u) (12 mo.)</b>	0.069	0.837	0.013
<b>P(u)</b>	0.292	0.021	-

**Table 3.** Feature correlations. Note due to the symmetry of these features, mainstreamness is presented for 6 months on one dimension and 12 months on another. Overall, none of the features are highly correlated other than the mainstreamness 6 and 12 month features, which is expected.

$A(u)$  already known by  $u$ . The recommended artists  $R(u)$  for user  $u$  are computed as  $R(u) = \bigcup_{v \in V_K(u)} A(v) \setminus A(u)$  and  $V_K(u) = \text{argmax}_{v \in U \setminus \{u\}}^K \text{sim}(u, v)$ , where  $\text{argmax}_v^K$  denotes the  $K$  users  $v$  with highest similarities to  $u$ . In considering geographical information for user-context models, we investigate the following approaches, which differ in the way this similarity term  $\text{sim}(u, v)$  is computed. The following approaches were investigated:

**CULT:** In the cultural approach, we select the neighbors for the seed user only according to a geographical similarity computed by means of the Jaccard index on listening distributions over semantic locations. We consider as such



**Figure 1.** Histogram of (top) play counts (note the log scale on the y-axis) and (bottom) diversity over users.

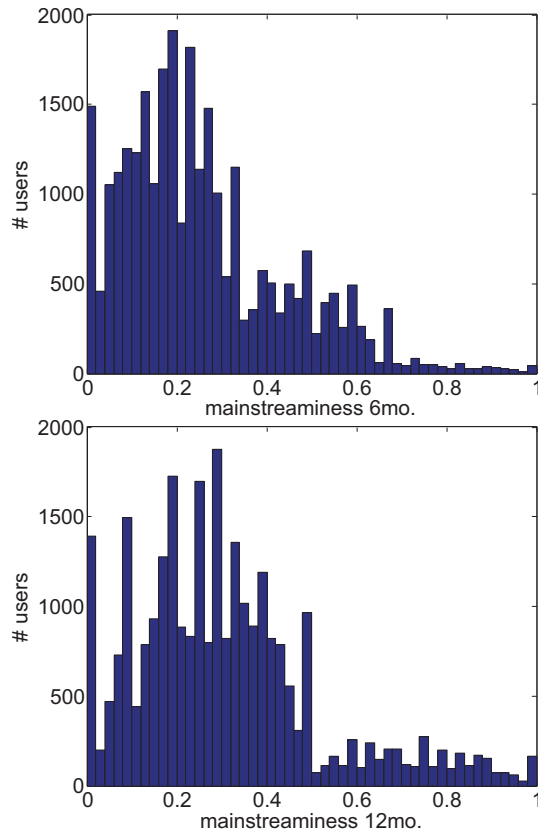
semantic categories continent, country, state, county, and city. For each user, we obtain the relevant locations by computing the relative frequencies of his listening events over all locations. To exclude the aforementioned geonities that are unlikely to contribute to the user’s cultural circle, we retain only locations at which the user has listened to music with a frequency above his own average<sup>2</sup>. On the corresponding listening vectors over locations of two users  $u$  and  $v$ , we compute the Jaccard index to obtain  $sim(u, v)$ . Depending on the location category user similarities are computed on, we distinguish CULT\_continent, CULT\_country, CULT\_state, CULT\_county, and CULT\_city.

**CF:** We also consider a user-based collaborative filtering approach. Given the artist play counts of seed user  $u$  as a vector  $\vec{P}(u)$  over all artists in the corpus, we first omit the artists that occur in the test set (i.e. we set to 0 the play count values for artists we want our algorithm to predict). We then normalize  $\vec{P}(u)$  so that its Euclidean norm equals 1 and compute similarities  $sim(u, v)$  as the inner product between  $\vec{P}(u)$  and  $\vec{P}(v)$ .

**CF\_CULT:** This approach works by combining the CF similarity matrix with the CULT similarity matrix via point-wise multiplication, in order to incorporate both music preference and cultural information.

**RB:** For comparison, we implemented a random baseline model that randomly picks  $K$  users and recommends

<sup>2</sup>This way we exclude, for instance, locations where the user might have spent only a few days during vacation.



**Figure 2.** Histogram of mainstreamness considering a time interval of (top) 6 months and (bottom) 12 months.

the artists they listened to. The similarity function can thus be considered  $sim(u, v) = rand[0,1]$ .

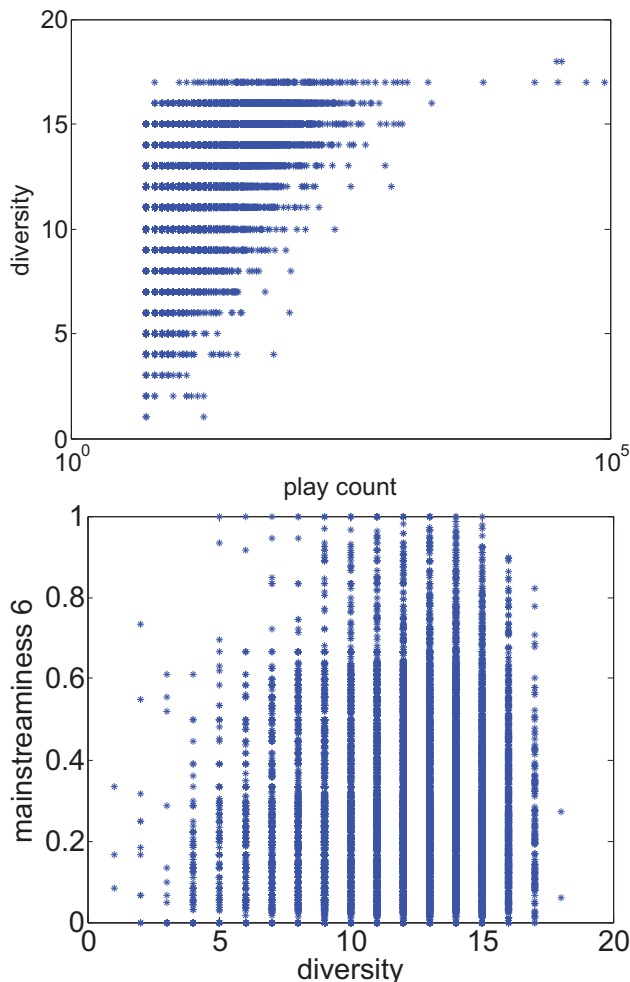
## 5. EVALUATION

### 5.1 Experimental Setup

For experiments, we perform 10-fold cross validation on the user level. For each user, we predict 10% of the artists based on the remaining 90% used for training. We compute precision, recall, and F-measure by averaging the results over all folds per user and all users in the dataset. To compare the performance between approaches, we use a parameter  $N$  for the number of recommended artists, and adapt dynamically the number of neighbors  $K$  to be considered for the seed user  $u$ . This is necessary since we do not know how many artists should be predicted for a given user (this number varies over users and approaches). To determine a suited value of  $K$  for a given recommendation approach and a given  $N$ , we start the approach with  $K = 1$  and iteratively increase  $K$  until the number of recommended artists equals or exceeds  $N$ . In the latter case, we sort the returned artists according to their overall popularity among the  $K$  neighbors and recommend the top  $N$ .

### 5.2 Results

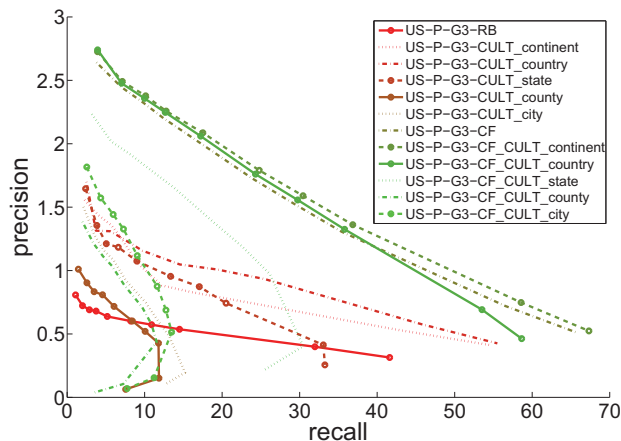
Table 2 depicts the maximum F-score (over all values of  $N$ ) for each combination of user set and method. We decided to report the maximum F-scores, because recall and



**Figure 3.** Users plot as a function of (top)  $D(u)$  vs  $P(u)$  and (bottom)  $M(u)$  (6 months) vs  $D(u)$ . Note the log scale for  $P(u)$  only. These figures illustrate the widespread, even distribution of users across the feature space.

precision show an inverse characteristics over  $N$ . Since the F-score equals the harmonic mean of precision and recall, it is less influenced by variations of  $N$ , nevertheless aggregate performance in a meaningful way. We further plot precision/recall-curves for several cases reported in Table 2. In Figure 4, we present the results of all of the recommendation algorithms for one group on the play counts. For this case, the CF approach with integrated continent and country information performed best, followed by the CF approach. Predominantly, these three methods outperformed all of the other approaches for the various groups, which is also apparent in Table 2. The only exception was the P-top10 case, where the CULT\_continent approach outperformed CF approaches. However, considering the small number of users in this subset (10), the difference of one percentage point between CULT\_continent and CF\_CULT\_continent is not significant. We observe the CF approach with the addition of the continent and country information are very good recommenders in general for the data we are using.

Now we are interested to know how the recommendations performed across user groups and respective features.



**Figure 4.** Recommendation performance of investigated methods on user group P-G3.

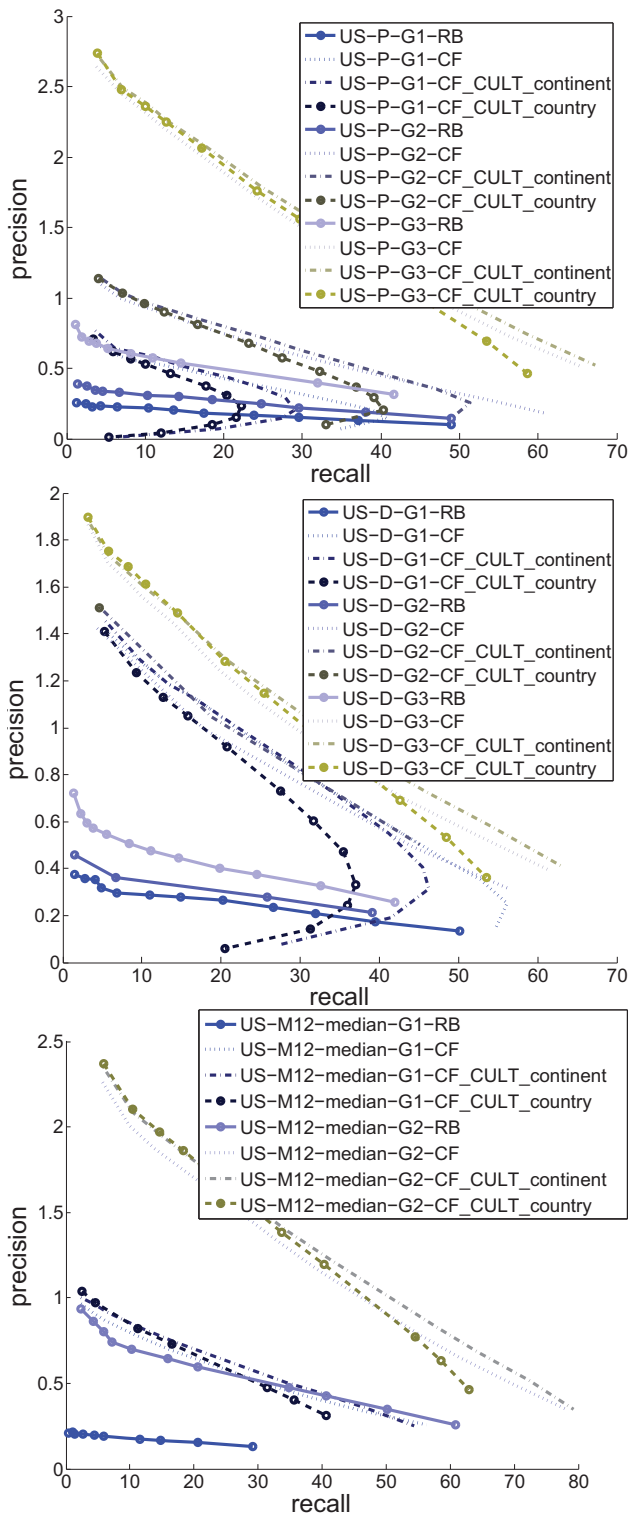
In terms of play counts, we observe as the user has a larger number of events in the dataset, the performance increases significantly (P-G3 and P-top10). This can be explained by the fact that more comprehensive user models can be created for users about whom we know more, which in turn yields better recommendations.

Also in terms of diversity, there are performance differences across groups given a particular recommender algorithm. Especially between the high diversity listeners D-G3 and low diversity listeners D-G1, results differ substantially. This can be explained by the fact that it is easier to find a considerable amount of like-minded users for seeds who have a diverse music taste, in technical terms, less sparse  $A(u)$  vector.

When considering mainstreamness, taking either a 6 month or 12 month interval does not appear to have a significant impact on recommendation performance. There are minor differences depending on the recommendation algorithm. However, in general, the groups with larger mainstreamness (M6-03-G2, M6-med-G2, M12-med-G2) always performed much better for all approaches than the groups with smaller mainstreamness. It hence seems easier to satisfy users with a mainstream music taste than users with diverging taste.

## 6. CONCLUSIONS AND FUTURE WORK

In this paper, we consider the role of user listening behavior related to the history of listening events in order to evaluate how this may effect music recommendation, particularly considering the direction of personalization. We investigate three user characteristics, play count, mainstreamness, and diversity, and form groups of users along these dimensions. We evaluate several different recommendation algorithms, particularly collaborative filtering (CF), and CF augmented by location information. We find the CF and CF approaches augmented by continent and country information about the listener to outperform the other methods. We also find recommendation algorithms for users with large play counts, higher diversity, and higher



**Figure 5.** Precision vs. recall for play count (top), diversity (middle), and mainstreamness with a 12 month interval (bottom) experiments over groups and various recommendation approaches.

mainstreamness have better performance.

As part of future work, we will investigate content-based music recommendation models as well as combinations of content-based, CF-based, and location-based models. Additional characteristics of the user, such as age, gender, or musical education, will be addressed, too.

## 7. ACKNOWLEDGMENTS

This research is supported by the Austrian Science Funds (FWF): P22856 and P25655, and by the EU FP7 project no. 601166 (“PHENICX”).

## 8. REFERENCES

- [1] G. Adomavicius and A. Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.
- [2] L. Chen, W. Wu, and L. He. How personality influences users’ needs for recommendation diversity? *CHI ’13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA ’13*, 2013.
- [3] N. Hariri, B. Mobasher, and R. Burke. Context-aware music recommendation based on latent topic sequential patterns. In *Proc. ACM RecSys ’12*, New York, NY, USA, 2012.
- [4] M. Hart. The long tail: Why the future of business is selling less of more by chris anderson. *Journal of Product Innovation Management*, 24(3):274–276, 2007.
- [5] D. Hauger, M. Schedl, A. Košir, and M. Tkalčič. The Million Musical Tweets Dataset: What Can We Learn From Microblogs. In *Proc. ISMIR*, Curitiba, Brazil, November 2013.
- [6] R. Hu and P. Pu. Exploring Relations between Personality and User Rating Behaviors. *1st Workshop on Emotions and Personality in Personalized Services (EMPIRE)*, June 2013.
- [7] N. Hurley and M. Zhang. Novelty and diversity in top-n recommendation – analysis and evaluation. *ACM Trans. Internet Technol.*, 10(4):14:1–14:30, March 2011.
- [8] J. Konstan and J. Riedl. Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22(1-2):101–123, March 2012.
- [9] R. McCrae and O. John. An Introduction to the Five-Factor Model and its Applications. *Journal of Personality*, 60(2):175–215, 1992.
- [10] E. Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011.
- [11] P. Rentfrow and S. Gosling. The do re mi’s of everyday life: The structure and personality correlates of music preferences. *Journal of Personality and Social Psychology*, 84(6):1236–1256, 2003.
- [12] M. Schedl and D. Schnitzer. Hybrid Retrieval Approaches to Geospatial Music Recommendation. In *Proc. ACM SIGIR*, Dublin, Ireland, July–August 2013.
- [13] M. Tkalčič, M. Kunaver, A. Košir, and J. Tasič. Addressing the new user problem with a personality based user similarity measure. *Joint Proc. DEMRA and UMMS*, 2011.



# TOWARDS SEAMLESS NETWORK MUSIC PERFORMANCE: PREDICTING AN ENSEMBLE’S EXPRESSIVE DECISIONS FOR DISTRIBUTED PERFORMANCE

**Bogdan Vera**

Queen Mary University of London  
Centre for Digital Music  
*b.vera@qmul.ac.uk*

**Elaine Chew**

Queen Mary University of London  
Centre for Digital Music  
*elaine.chew@qmul.ac.uk*

## ABSTRACT

Internet performance faces the challenge of network latency. One proposed solution is music prediction, wherein musical events are predicted in advance and transmitted to distributed musicians ahead of the network delay. We present a context-aware music prediction system focusing on expressive timing: a Bayesian network that incorporates stylistic model selection and linear conditional gaussian distributions on variables representing proportional tempo change. The system can be trained using rehearsals of distributed or co-located ensembles.

We evaluate the model by comparing its prediction accuracy to two others: one employing only linear conditional dependencies between expressive timing nodes but no stylistic clustering, and one using only independent distributions for timing changes. The three models are tested on performances of a custom-composed piece that is played ten times, each in one of two styles. The results are promising, with the proposed system outperforming the other two. In predictable parts of the performance, the system with conditional dependencies and stylistic clustering achieves errors of 15ms; in more difficult sections, the errors rise to 100ms; and, in unpredictable sections, the error is too great for seamless timing emulation. Finally, we discuss avenues for further research and propose the use of predictive timing cues using our system.

## 1. INTRODUCTION

Ensemble performance between remote musicians playing over the Internet is generally made difficult or impossible by high latencies in data transmission [3] [5]. While many composers and musicians have chosen to treat latency as a feature of network music, performance of conventional music, such as that of classical repertoire, remains extremely difficult in network scenarios. Audio latency frequently results in progressively decreasing tempo

and difficulty in synchronizing.

One aspect that has received less attention than the latency is the lack of visual contact when performing over the internet. Visual cues can be transmitted via video, but such data is at least as slow as audio, and was previously found to not be of significant use for transmitting synchronization cues even when the audio had an acceptable latency [6].

Since the start of network music research, several researchers have posited theoretically that music prediction could be the solution to network latency (see, for example, Chafe [2]). Ideally, if the music can be predicted ahead of time with sufficient accuracy, then it can be replicated at all connected end-points with no apparent latency. Recent efforts have made limited progress towards this goal. One example is a system for predicting tabla drumming patterns [12], and recent proposals by Alexandraki [1]. Both assume that the tempo of the piece will be at least locally smooth and, in the case Alexandraki’s system, timing alterations are always based on one reference recording.

In many styles of music, such as romantic classical music, the tempo can vary widely, with musicians interacting on fine-scale note-to-note timing changes and using visual cues to synchronize. The tempo cannot be expected to always evolve in the exact same way as one previous performance, rather the musicians significantly improvise timing deviations to some constraints.

In this paper we propose a system for predicting timing in network performance in real time, loosely inspired by Raphael’s approach based on Bayesian networks [11]. We propose and test a way to incorporate abstract notions of expressive context within a probabilistic framework, making use of time series clustering. Flossman et al. [8] employed similar ideas when they extended the YQX model for expressive offline rendering of music by using conditional gaussian distributions to link expressive predictions over time. Our model contains an extra layer of stylistic abstraction and is applied to modeling and real-time tracking of one performer or ensemble’s expressive choice at the inter-onset interval level. We also describe how the method could be used for predicting musical timing in network performance, and discuss ideas for further work.



© Bogdan Vera, Elaine Chew.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Bogdan Vera, Elaine Chew. “Towards Seamless Network Music Performance: Predicting an Ensemble’s Expressive Decisions for Distributed Performance”, 15th International Society for Music Information Retrieval Conference, 2014.

## 2. MOTIVATION

Our goal is to use observable sources of information during a live performance to predict the timing of future notes so as to counter the effects of network latency. The sources of information we can use include the timing of previous notes and the intensity with which the notes are played.

The core idea is reminiscent of Raphael’s approach to automatic accompaniment [11], which uses a Bayesian network relating note onset times, tempo and its change over time. In Raphael’s model, changes in tempo and local note timing are represented as independent gaussian variables, with distributions estimated from rehearsals. During a performance, the system generates an accompaniment that emulates the rehearsals by applying similar alterations of timing and tempo at each note event in the performance. The model has been demonstrated in live performances and proven to be successful, however as long as the system generates musically plausible expression in the accompaniment, it is difficult to determine an error value, as it is simply meant to follow a musician and replicate a performance style established in rehearsals. An underlying assumption of this statistical model is that the solo musician leading the performance tends to perform the piece with the same expressive style each time.

In an ensemble performance scenario, two-way communication exists between musicians. The requirement for the system to simply ‘follow’ is no longer enough. As a step towards tighter ensemble, we set as a goal a stringent accuracy requirement for our prediction system: to have errors small enough—no higher than 20-40ms—as to be indistinguishable from the normal fluctuations in ensemble playing. Note that actual playing may have higher errors, even in ideal conditions, due to occasional mistakes and fluctuations in motor control.

The same ensemble might also explore a variety of ways to perform a piece expressively. When expressive possibilities are explored during rehearsals, the practices establish a common ‘vocabulary’ for possible variations in timing that the musicians can then anticipate. Another goal of our system is to account for several distinct ways of applying expression to the same piece. This is accomplished in two ways. Like Flossman et al. [8], we deliberately encode the context of the local expression by introducing dependencies between the expressive tempo changes at each time step. We additionally propose and test a form of model selection using discrete variables that represent the chosen stylistic *mode* of the expression. For example, given two samples exhibiting the same tempo change, one may be part of a longer term tempo increase, while another may be part of an elastic time-stretching gesture. Knowing the stylistic context for a tempo change will allow us to better predict its trajectory.

## 3. CONTEXTUALIZING TIMING PREDICTION

We combine two techniques to implement ensemble performance prediction. First, we condition the expressive ‘update’ distributions characterizing temporal expression

on those from preceding events, making the timing changes dependent on both musicians’ previous timing choices, while also allowing the system to respond to the interplay between the two musicians. Secondly, we abstract different ways of performing the piece by summarizing these larger scale differences in an unsupervised manner in a new discrete node in the network: a stylistic cluster node.

### 3.1 Linear Gaussian Conditional Timing Prediction

Our goal is to predict the timing of events such as notes, chords, articulations, and rests. In particular, we wish to determine the time until the next event given the score information and a timing model. We collapse all chords into single events. Assume that the performance evolves according to the following equations,

$$\begin{aligned} t_{n+1} &= s_n l_n + t_n, \text{ and} \\ s_{n+1} &= s_n \cdot \delta_n, \end{aligned} \quad (1)$$

where  $t_n$  is the onset time of the  $n$ -th event,  $s_n$  is the corresponding inter-beat period,  $l_n$  is the length of the event in beats, and  $\delta_n$  is a proportional change in beat duration that is drawn from the gaussian distributions  $\Delta_n$ . For simplicity, there is no distinction between tempo and local timing in our model, though it could be extended to include this separation.

Because  $\delta_n$ ’s reflect proportional change in beat duration, prediction of future beat durations are done on a logarithmic scale:

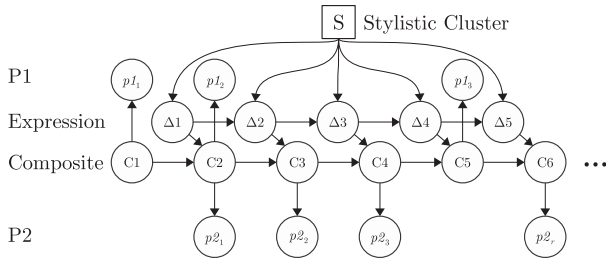
$$\log_2 s_{n+1} = \log_2 s_n + \log_2 \delta_n.$$

$\log(\text{tempo}) = \log(1/s_n)$ , thus  $\log s_n$  as well, has been shown in recent research to be a more consistent measure of tempo variation in expressive performance [4].

The parameters of the  $\Delta_n$  distributions are predicted during the performance from previous observations, such as  $\delta_{n-1}$ . Thus, each inter-beat interval,  $s_n$ , is shaped from event to event by the random changes,  $\delta_n$ . The conditional dependencies between the random variables are illustrated in Figure 1. The first and last layers in the network, labeled P1 and P2 in the diagram, are the observed onset times. The 3rd layer, labeled ‘Composite’ following Raphael’s terminology, embodies the time and tempo information at each event, regardless of which ensemble musician is playing, and it is on this layer that our model focuses. The 2nd layer, Expression, consists of the variables  $\Delta_n$ .

The  $\Delta_n$  variables are conditioned upon their predecessors, using any number of previous timing changes as input; formally, they are represented by linear conditional gaussian distributions [9]. Let there be a Bayesian network node with a normal distribution  $Y$ . We can condition  $Y$  on its  $k$  continuous parents  $C = \{C_1, \dots, C_k\}$  and discrete parents  $D = \{D_1, \dots, D_k\}$  by using a linear regression model to predict the mean and variance of  $Y$  given the values of  $C$  and  $D$ . The following equation describes the conditional probability of  $Y$  given only continuous parent nodes:

$$P(Y|C = \mathbf{c}) = \mathcal{N}(\beta_0 + \sum_{i=1}^k \beta_i c_i, \sigma^2).$$



**Figure 1:** A section of the graphical model. Round nodes are continuous gaussian variables, and the square node ( $S$ ) is a discrete stylistic cluster node.

This is the equation for both continuous and discrete parents:

$$P(Y|D = \mathbf{d}, C = \mathbf{c}) = \mathcal{N}(\beta_{\mathbf{d},0} + \sum_{j=1}^k \beta_{\mathbf{d},j} c_j, \sigma_{\mathbf{d}}^2).$$

Simply speaking, the mean and variance of each linear conditional gaussian node is calculated from the values of its continuous and discrete parent nodes. The mean is derived through linear regression from its continuous parents' values with one weight matrix per configuration of its discrete parents.

The use of conditional gaussian distributions means that rather than having fixed statistics for how the timing should occur at each point, the parameters for the timing distributions are predicted in real time from previous observations using linear regression. This simple linear relationship provides a means of predicting the extent of temporal expression as an ongoing gesture. For example, if the performance is slowing down, the model can capture the rate of slowdown, or a sharp tempo turnaround if this occurred during rehearsals.

Our network music approach involves interaction between two actual musicians rather than a musician and a computer. Thus, each event observed is a 'real' event, and we update the  $\Delta_n$  probability distributions at each step during run-time with the present actions of the musicians themselves. Unlike a system playing in automatic accompaniment or an expressive rendering system, our system is never left to play on its own, and its task is simply to continue from the musicians' choices, leaving less opportunity for errors to accumulate. Additionally, we can correct the musicians' intended timing by compensating for latency post-hoc - this implies that we can make predictions that emulate what the musicians would have done without the interference of the latency.

We may also choose the number of previous changes to consider. Experience shows that adding up to 3 previous inputs improves the performance moderately, but the performance decreases thereafter with more inputs. For simplicity, we currently use only one previous input, which provides the most significant step improvement.

In contrast to a similar approach by Flossman et al. [8], we do not attempt to link score features to the performance; we only consider the local context of their temporal expression. Our goal is to capture the essence of one particular ensemble's interpretation of a particular piece rather

than attempting to construct a universal model for mapping score to performance. As a result, the amount of training data will generally be much smaller as we may only use the most recent recorded and annotated rehearsals of the ensemble. The next section describes a clustering method we use to account for large-scale differences in timing.

### 3.2 Unsupervised Stylistic Characterization

Although we could add a large number of previous inputs to each of the  $\Delta_n$  nodes, we cannot tractably condition these variables' distributions on potentially hundreds of previous observations. This would require a large amount of training data to estimate the parameters in a meaningful way. Instead, we propose to summarize larger-scale expression using a small number of discrete nodes representing the stylistic mode. For example, a musician may play the same section of music in few distinct ways, and a listener may describe it as 'static', 'swinging' or 'loose'. If these playing styles could be classified in real time, prediction could be improved by considering this stylistic context. Our ultimate goal is to perform this segmentally on a piece of music, discovering distinct stylistic choices that occurred in the ensemble's rehearsals. In this paper, we present the first steps towards this goal: we characterize the style of the entire performance using a single discrete stylistic node.

The stylistic node is shown at the top of Figure 1. In our model this node links to all of the  $\Delta_n$  nodes in the piece, so that each of the  $\Delta_n$ 's is now linearly dependent on the previous timing changes with weights that are dependent on the stylistic node. Assuming that each  $\Delta_n$  node is linked to one previous one, the parameters of the  $\Delta_n$  distributions are then predicted at run-time using

$$P(\Delta_t | S = s, \Delta_{t-1} = \delta) = \mathcal{N}(\beta_{s,0} + \beta_{s,1} \delta, \sigma_s^2),$$

where  $S$  is the style node.

To predict note events, we can simply take the means of the  $\Delta_n$  distributions, and use Equation 1 to find the onset time of the next event given the current one.

To use this model, we must first discover the distinct ways (if any) in which the rehearsing musicians perform the piece. We apply k-means clustering to the  $\log(\delta_n)$  time series obtained from each rehearsal. We find the optimal number of clusters by using the Bayes Information Criterion (BIC) as described by Pelleg and Moore [10]. Note that other methods exist for estimating an optimal number of clusters. To train the Bayesian network, a training set is generated containing all of the  $\delta_n$  values for each rehearsal as well as the cluster to which each time series is allocated. We then use the algorithm by Murphy [9] to find all the parameters of the linear conditional nodes. Note that all of the nodes are observable and we have training data for the  $\Delta_n$ .

During the performance, the system can update its belief about the stylistic node's value from the note timings that have been observed at any point; we do not need to re-cluster the performance, as the network has learned the relationships between the  $\Delta_n$ 's and the stylistic node. We

use the message passing algorithm of Bayesian networks to infer the most likely state of the node. As the performance progresses, the belief about the state of the node is gradually established. Intuitively, the system arrives at a stable answer after some observations, otherwise the overall style is ambiguous. The state of the node is then used to place future predictions into some higher level context. The next section shows that the prediction performance is improved by using the stylistic node to select the best regression parameters to predict the subsequent timing changes, which can be thought of as a form of model selection.

## 4. EVALUATION

### 4.1 Methodology

In this section we present an evaluation of the basic form of our model. Evaluation of such predictive models remains a challenge because testing in live performance requires further work on performance tracking and optimization, while offline testing necessitates a large number of annotated performances from the same ensemble. We present initial results on a small dataset; in our future work we will study real time performances of more complex pieces.

We evaluate the performance of three models: one uses linear conditional nodes and a stylistic cluster node; the second uses only linear conditional nodes; and, the third has independent gaussian distributions for the  $\Delta$  variables.

Our dataset consists of 20 performances by one pianist of the short custom-composed piece shown in Figure 2. Notice that we have not added any dynamics or tempo-related markings - the interpretation is left entirely to the musicians. While this is not an ensemble piece, the performances are sufficient to test the prediction accuracy of our model in various conditions. In this simple example, we consider only the composite layer in the model, without P1 and P2.



Figure 2: Custom-composed piano test piece.

The piece was played on an M-Audio AXIOM MIDI keyboard in one of two expressive styles decided beforehand, ten times for each style. We used IRCAM's Antescofo score follower [7] for live tracking of the performance in our system, and annotation of the note and chord events. The log-period plots for every performance in the dataset are shown in Figure 4a. The changes in log-period per event are shown in Figure 4b, and we also show the same changes but for the data in each cluster found, to demonstrate the difference between the two playing styles.

We evaluated the system using a ‘leave-one-out’ approach, where out of the 20 performances we always trained on 19 of them and tested on the remaining one. We always used one previous input to the  $\Delta_n$  nodes, using the actual observations in the performances rather than our predictions (like the extended YQX), simulating the process of live performance. We evaluated the prediction accuracy by measuring timing errors, which we define as the absolute difference between the true event times and those predicted by the model (in seconds).

The training performances were clustered correctly in all cases, dividing the dataset into the two styles, with the first 10 performances being grouped with cluster 1 and the second 10 becoming part of cluster 2. Figure 3 shows the stylistic inference process. In the matrix, performances are arranged as rows, with events on the  $x$ -axis. Recall that we predict the time between events rather than just notes. So, we also consider the timing of rests, and chords are combined into single events rather than individual notes. The colors indicate the inferred value of the style node: grey for Style 1 and white for Style 2. We see that the system correctly infers the stylistic cluster of each performance within the first 19 events. In many cases the classification assigns the performance to the correct cluster after only two events.

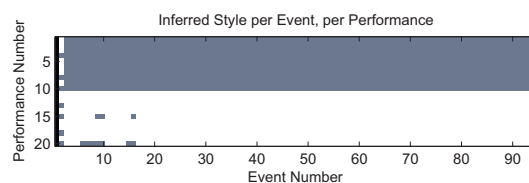


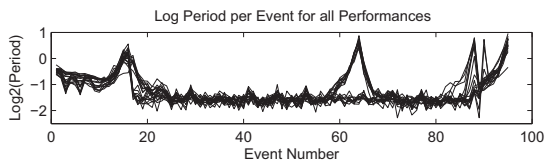
Figure 3: Matrix showing most likely style state after each event's observed  $\delta$ . Performances 1-10 are in Style 1, and 11-20 are in Style 2. Classification result: grey = Style 1, white = Style 2.

Figure 4 shows the tempo information for the dataset. Figure 4(a) shows the inter-beat period contours of all of the performances, while Figure 4(b) shows boxplots (indicating the mean and variability) of the period at each musical event, for the entire dataset and for the two clusters.

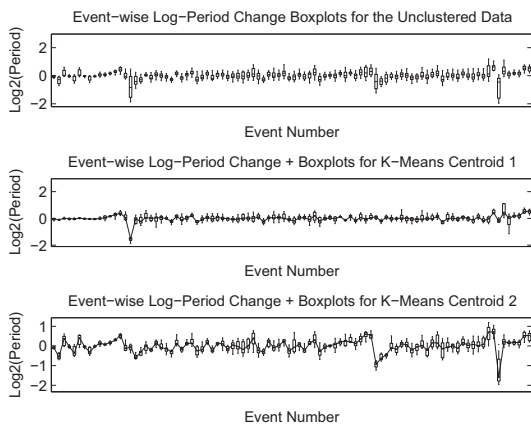
### 4.2 Results

Figure 5a and Figure 5b show the performance of the models, measured using mean absolute error averaged over events in each performance, and over performances for each event, respectively. We also show a detailed ‘zoomed in’ plot of the errors between events 20-84 to make the different models’ mean errors clearer in Figure 5c. For network music performance, we would want to predict at least as far forward as needed to counter the network (and other system) latency. As some inter-event time differences may be shorter than the latency, we may occasionally need to predict more than one event ahead.

The model with stylistic clustering and linear conditional nodes performed best, followed by the one with only linear conditional nodes, then the model with independent



(a) Log-period per event for every performance in the dataset.



(b) Boxplots showing median and variability for the log-period change at each event. Top: unclustered data, Middle: first centroid, Bottom: second centroid.

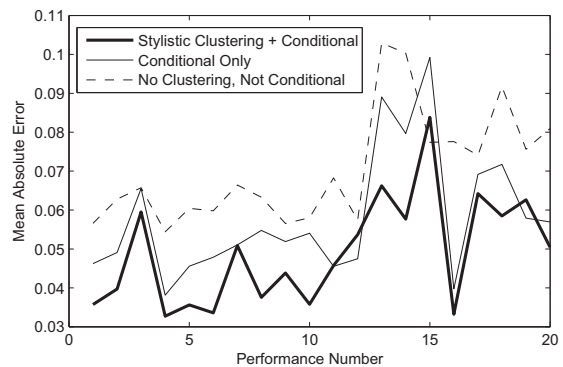
**Figure 4: Tempo Data**

$\Delta_n$  nodes. In all cases the errors were higher for the second style (the latter 10 performances), which was much looser than the first. The mean absolute errors for each model, considering all of the events in all of the performances are summarized in Table 1.

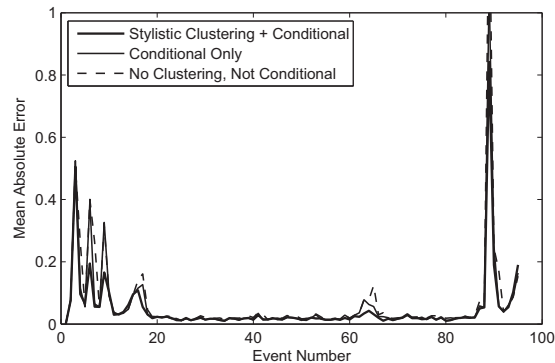
Observe in Figure 5b that some parts of the performance were very difficult to predict. For example, we note high prediction errors in the first 12 events of the piece and one large spike in the error at the end of the piece. These are 1-bar and 2-bar long chords, for which musicians in an ensemble would have to use visual gestures or other information to synchronize. We would not expect any prediction system to do better than a musician anticipating the same timing without any form of extra-musical information. We discuss potential applications of music prediction for virtual cueing in the next section. The use of clustering and conditional timing distributions reduced the error rate for the events which were poorly predicted with independent timing distributions. For much of the piece the mean error was as low as 15ms, but even for these predictable parts of the performance, the models with conditional distributions and clustering lowered the error, as can be seen from Figure 5c.

## 5. CONCLUSIONS AND FUTURE WORK

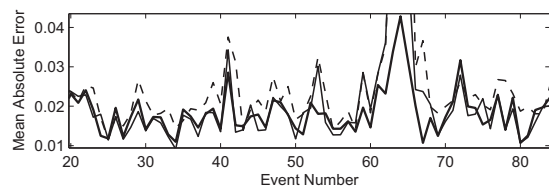
We have outlined a novel approach to network music prediction using a Bayesian network incorporating contextual inference and linear gaussian conditional distributions. In an evaluation comparing the model with stylistic clustering and linear conditional nodes, one with only linear conditional nodes without clustering, and one with indepen-



(a) Mean absolute error for each performance.



(b) Mean absolute error per event, over the whole performance.



(c) A 'zoomed-in' view of the error rates between events 20-84.

**Figure 5: Mean absolute error per event.**

dent nodes, we have shown that the proposed approach produces promising results. Specifically, we have shown evidence that considering a notion of large scale expressive context, drawn from performance styles of a particular ensemble, can intuitively increase the accuracy of timing prediction. The model remains to be tested on more data. As creative musicians are infinitely diverse in their expressive interpretations, the true test of the model would ultimately be in live performances.

The end goal of this research is to implement and evaluate network music performance systems based on the prediction model. Whether music prediction can ever be precise enough to allow seamless network performance remains an open question. Important questions arise in pur-

Model	Mean Abs. Error
Independent	69.8ms
Conditional	57.4ms
<b>Clustering and Conditional</b>	<b>48.5ms</b>

**Table 1: Overall Timing Errors for Each Model**

suit of this goal: how much should the system lead the musicians to help them stay in time without making the performance artificial? Predicting musical timing with sufficient accuracy will open up interesting avenues for network music research, especially when we consider parallel research into predicting other information such as intensity and even pitch information, but whether any musician would truly want to let a machine impersonate them expressively remains to be seen, which is why we propose that a ‘minimally-invasive’ conductor-like approach to regulating tempo would be more appropriate than complete audio prediction.

### 5.1 The Bayesian Network

It would be straightforward to extend our model by implementing prediction of timing from other forms of expression that tend to correlate with tempo. For example, using event loudness in the prediction would simply require the addition of another layer of variables in the Bayesian network and conditioning the timing variables on these nodes as well.

### 5.2 Capturing Style

Much work remains to expand on the characterization of stylistic mode. As previously mentioned, we plan to explore segmental stylistic characterization, considering different contextual information for each part of the performance. In our current model we use only one stylistic node. This may be a plausible for a small segment of music, but in a longer performance the choice of performance style may vary over time. If the predicted performance starts within one style but changes to another, the model is ill-informed to predict the parameters. In our future work we would like to extend the model to capture such stylistic tendencies over time. One approach would require pre-segmentation of the piece based on the choice of expressive choices during the rehearsal stage, and introduction of one stylistic node per segment. The prediction context would then be local to each part of the performance. We may then, for example, have causal conditional dependencies between the stylistic nodes in each segment of the piece, which would allow the system to both infer the style within a part of the performance from what is being played and from the previous stylistic choices.

In practice, a musician or ensemble’s rehearsals may not comprise of completely distinct interpretations; however, capturing expression contextually will likely offer a larger degree of freedom to the musicians in an internet performance, who may then explore a greater variety of temporal and other articulations.

### 5.3 Virtual Cueing

Virtual cueing forms an additional application of interest. As mentioned at the start of the paper, visual communication is generally absent or otherwise delayed in network music performance. If we could predict with reasonable

accuracy the timing in sections of a piece requiring temporal coordination, then we could help musicians synchronize by providing them with perfectly simultaneous predicted cues. We regard the use of predictive virtual cues as less invasive to networked ensembles than complete predictive sonification. In situations where the audio latency is low enough for performance to be feasible but video latency is still too high for effective transmission of gestural cues, predictive sonification may be omitted completely, and virtual cues could be implemented as a regulating factor.

## 6. ACKNOWLEDGEMENTS

This research was funded in part by the Engineering and Physical Sciences Research Council.

## 7. REFERENCES

- [1] C. Alexandraki and R. Bader. Using computer accompaniment to assist networked music performance. In *Proc. of the AES 53rd Conference on Semantic Audio, London, UK, 2013*.
- [2] C. Chafe. Tapping into the internet as an acoustical/musical medium. *Contemporary Music Review*, 28, Issue 4:413–420, 2010.
- [3] C. Chafe and M. Gurevich. Network time delay and ensemble accuracy: Effects of latency, asymmetry. In *Proc. of the 117th Audio Engineering Society Convention, 2004*.
- [4] E. Chew and C. Callender. Conceptual and experiential representations of tempo: Effects on expressive performance comparisons. In *Proc. of the 4th International Conference on Mathematics and Computation in Music*, pages 76–87, 2013.
- [5] E. Chew, A. Sawchuk, C. Tanoue, and R. Zimmermann. Segmental tempo analysis of performances in user-centered experiments in the distributed immersive performance project. In *Proc. of the Sound and Music Computing Conference, 2005*.
- [6] E. Chew, R. Zimmermann, A. Sawchuk, C. Kyriakakis, and C. Papadopolous. Musical interaction at a distance: Distributed immersive performance. In *Proc. of the 4th Open Workshop of MUSICNETWORK, Barcelona, 2004*.
- [7] A. Cont. Antescofo: Anticipatory synchronization and control of interactive parameters in computer music. In *Proc. of the International Computer Music Conference, 2008*.
- [8] S. Flossmann, M. Grachten, and G. Widmer. *Guide to Computing for Expressive Music Performance*, chapter Expressive Performance Rendering with Probabilistic Models, pages 75–98. Springer Verlag, 2013.
- [9] K. P. Murphy. Fitting a conditional linear gaussian distribution. Technical report, University of British Columbia, 1998.
- [10] D. Pelleg and A. W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proc. of the Seventeenth International Conference on Machine Learning*, pages 727–734, 2000.
- [11] C. Raphael. Music plus one and machine learning. In *Proc. of the 27th International Conference on Machine Learning*, pages 21–28, 2010.
- [12] M. Sarkar. Tablanet: a real-time online musical collaboration system for indian percussion. Master’s thesis, MIT, 2007.

# DETECTION OF MOTOR CHANGES IN VIOLIN PLAYING BY EMG SIGNALS

Ling-Chi Hsu, Yu-Lin Wang, Yi-Ju Lin, Alvin  
W.Y. Su

Department of CSIE, National Cheng-Kung  
University, Taiwan  
t19897843@gmail.com;  
daphne.yl.wang@gmail.com;  
lyjca.cs96@g2.nctu.edu.tw;  
alvinsu@mail.ncku.edu.tw

Cheryl D. Metcalf

Faculty of Health Sciences, University of  
Southampton, United Kingdom  
c.d.metcalf@soton.ac.uk

## ABSTRACT

Playing a music instrument relies on the harmonious body movements. Motor sequences are trained to achieve the perfect performances in musicians. Thus, the information from audio signal is not enough to understand the sensorimotor programming in players. Recently, the investigation of muscular activities of players during performance has attracted our interests. In this work, we propose a multi-channel system that records the audio sounds and electromyography (EMG) signal simultaneously and also develop algorithms to analyze the music performance and discover its relation to player's motor sequences. The movement segment was first identified by the information of audio sounds, and the direction of violin bowing was detected by the EMG signal. Six features were introduced to reveal the variations of muscular activities during violin playing. With the additional information of the audio signal, the proposed work could efficiently extract the period and detect the direction of motor changes in violin bowing. Therefore, the proposed work could provide a better understanding of how players activate the muscles to organize the multi-joint movement during violin performance.

## 1. INTRODUCTION

For musicians, their motor skills must be honed by many hours of daily practice to maintain the performing quality. Motor sequences are trained to achieve the perfect performances. Playing a musical instrument relies on the harmonious coordination of body movements, arm and fingers. This is fundamental to understanding the neurophysiological mechanisms that underpin learning. It therefore becomes important to understand the sensorimotor programming in players. In the late 20th century, Harding et al. [1] directly measured the force between player's fingers and piano keys with different skill levels. Engel et al. [2] found there is an anticipatory change of sequential hand movements in pianists. Parlitz et al. [3]

explored the dynamic pressures to analyze how pianists depressed the piano keys and hold them down during playing. The pressure measurement advances the evaluation of the keystroke in piano playing [4-5]. The use of muscle activity via electromyography (EMG) signals allows further investigation into the motor control sequences that produce the music. EMG is a technique which evaluates the electrical activity of the muscle by recording the electrical potentials when muscles generate an electrical voltage during activation, which results in a movement or coordinated action.

EMG is generally recorded in two protocols; invasive electromyography (IEMG) and surface electromyography (SEMG). IEMG is used to measure deep muscles and discrete positions using a fine-wire needle; however, it is not a preferable model for subjects due to the invasiveness and being less repetitive. Compared to IEMG, SEMG has the following characteristics: (1) it is non-invasive; (2) it provides global information; (3) it is comparatively simple and inexpensive; (4) it is applicable by non-medical personnel; and (5) it can be used over a longer time during work and sport activities [6]. Therefore, the SEMG is suitable for use within biomechanics and movement analysis, and was used in this paper.

For the analysis of musical performance, EMG has been used to evaluate behavioral changes of the fingers [7-8], upper limbs [9-10] shoulder [11-12] and wrist [13] in piano, violin, cello and drum players. The EMG method allows for differentiating the variations and reproducibility of muscular activities in individual players. Comparing the EMG activity between expert pianists and novice players [7-14] has also been studied.

There have been many approaches developed for segmentation of EMG signals [15]. Prior EMG segmentation techniques were mainly used to detect the time period for a certain muscle contraction, but we found that the potential variations from various muscles maybe different during a movement. It causes the conventional EMG segmentation to fail to extract the accurate timing of movement in instrument playing.

In this paper, the timing activation of the muscle group is assessed, and the changes in motor control of players during performance are investigated. We propose a system with the function of concurrently recording the audio signal and behavioral changes (EMG) while playing an instrument. This work is particularly focused on violin playing, which is considered difficult to segment with the

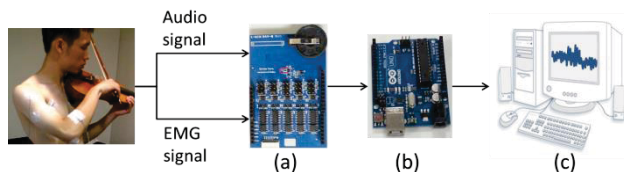


soft onsets of the notes. The segment with body movements was first identified by the information of audio sounds. It is believed that if there is an audio signal, then there is a corresponding movement. Six features were then introduced to EMG signals to discover the variation of movements. This work identifies the individual movement segments, i.e. up-bowing and down-bowing, during violin playing. Thus, how motor systems operated in musicians and affected during performance could be explored using this methodology.

This paper is organized as follows. The multi-channel signal recording system and its experimental protocol are shown in section 2. In section 3, we introduce the proposed algorithms for segmenting the EMG signal with additional audio information. The experimental results are shown in section 4 and the conclusion and future work are given in section 5.

## 2. AUDIO SOUNDS AND BIOSIGNAL RECORDING SYSTEM

This work proposed a multi-channel signal recording system capable of recording audio and EMG signals concurrently. The system is illustrated in Figure 1 and comprises: (a) a signal pre-amplifier acquisition board, (b) an analog to digital signal processing unit, and (c) a host-system.



**Figure 1.** The proposed multi-channel recording system for recording audio signal and EMG concurrently.

The violin signal was recorded in a chamber and the microphone was placed 30cm from the player with a sampling rate of 44100Hz. With this real violin recording, the sound is supposedly embedded with the noise and the artifacts.

Furthermore, there is three subjects in the experiment database. The violinist play music and be recorded. Each participant was requested to press one string during playing. This experiment included two tasks for performance evaluation, and each task contained 10 movements. The movements for task#1 and task#2 are defined as follows.

Movements for task#1:

- (1) Player presses the 2<sup>nd</sup> string then is idle for 2s (begin the bow at the frog).
- (2) Pulls the bow from the frog to the tip for 4s (whole bow down).
- (3) Pulls the whole bow up for 4s.

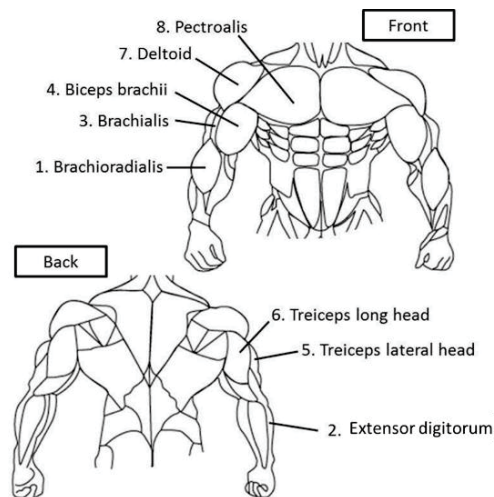
Movements for task#2:

- (1) Player presses the 3<sup>rd</sup> string then is idle for 2s (begin the bow at the tip).
- (2) Pulls the whole bow up for 4s.
- (3) Pulls the whole bow down for 4s.

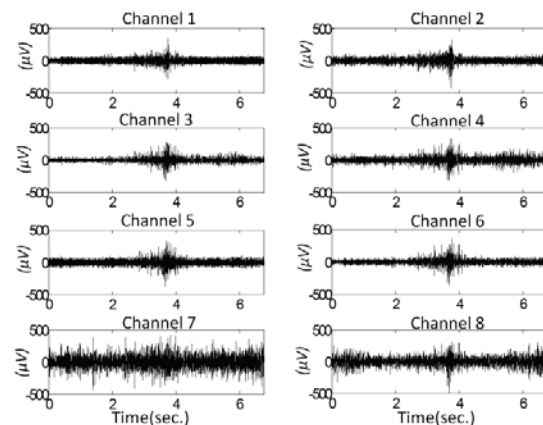
Two seconds resting time was given between the two consecutive movements.

The EMG sampling rate was 1000Hz. The electrodes attached on the surface of the player's skin as shown Figure 2. In this study, the direction of violin bowing, i.e. up-bowing and down-bowing, is detected by the corresponding muscle activity (EMG signal). The total of 8 muscles in the upper limb and body is measured in our system. Figure 3 shows the 8-channel EMG signals of up-bowing movement, and potential variations were shown in all channels when bowing. Three types of variations were observed and grouped:

- (1) Channel#1 to Channel#6: it is seen that the trend of six channels is similar; additionally, the average noise floor between channel#3 and channel#6 are lower than others; finally, we choose channel#6 because the position is convenient to place the electrode.
- (2) Channel#7: the channel involving the most noise.
- (3) Channel#8: although it has more noise than Channel#1 to Channel#6, it is the important part when we have a whole-bowing movement.



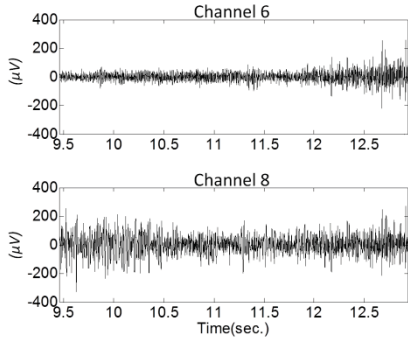
**Figure 2.** The placement of the electrodes attached on the player's skin [16, 17].



**Figure 3.** The 8-channel EMG signals of up-down bowing movements.



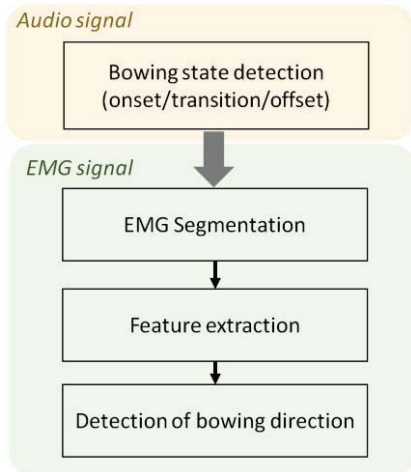
To reduce the computation and retain the variety of features, only channel#6 and channel#8 were thereafter used for further analysis. Figure 4 shows the EMG signals of channel#6 and channel#8 while during down-bowing.



**Figure 4.** The EMG signals of triceps (channel#6) and pectoralis (channel#8) during down-bowing movements.

### 3. METHOD

The following section will introduce the proposed algorithm for detecting the bowing states during violin playing. The proposed system is capable of recording audio and EMG signals concurrently, and in this study a bowing state detection algorithm was developed, which was implemented the embedded system. The flowchart of the proposed method is shown in Figure 5.



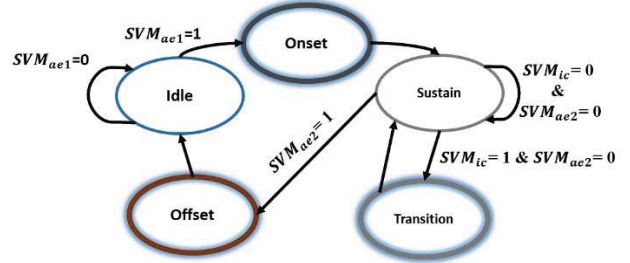
**Figure 5.** Flowchart of the proposed system.

The EMG signals were segmented according to the violin sounds. Then, six features were identified to detect the direction of bowing movements. For analyzing the audio signal, the window size of a frame is 2048 samples and the hop size 256 samples.

#### 3.1 Onset/Transition/Offset detection

This section elaborates on the state detection of audio sounds. The states of audio sounds are defined as *Onset*, *Transition* and *Offset* in this study. The *Onset* is the beginning of bowing; the *Transition* is the timing when the next bowing movement occurred; the *Offset* is the end of

the bowing; the *Sustain* is the duration of the note segment. Both frequency and spatial features were calculated and used as the inputs to our developed finite state machine (FSM). The diagram of our proposed FSM is illustrated in Figure 6. The output of FSM identifies the result of note detection and further used for EMG segmentation.



**Figure 6.** The state diagram of audio sounds.

The violin signal was analyzed both in frequency and time domains. For frequency analysis, the violin signal was first transformed by short time Fourier transform. The inverse correlation (IC) was then applied to calculate the possible note onset period. The inverse correlation (IC) coefficients are computed from the correlation coefficients of two consecutive discrete Fourier transform spectra [18]. A support vector machine (SVM), denoted as  $SVM_{ic}$  (1), was applied for detecting the accurate timing of onset. SVM is a popular methodology, with high speed and simple implementation, for classification and regression analysis [19].

$$SVM_{ic} = \begin{cases} 0 & , \text{ non - transition} \\ 1 & , \text{ transition} \end{cases} \quad (1)$$

For spatial analysis, the amplitude envelop (AE) was used to detect the segment of the sound data. AE is evaluated as the maximum value of a frame. There are two similar classifiers, called  $SVM_{ae1}$  (2) and  $SVM_{ae2}$  (3).  $SVM_{ae1}$  is used to identify the possible onsets and  $SVM_{ae2}$  is used to identify the possible offsets.

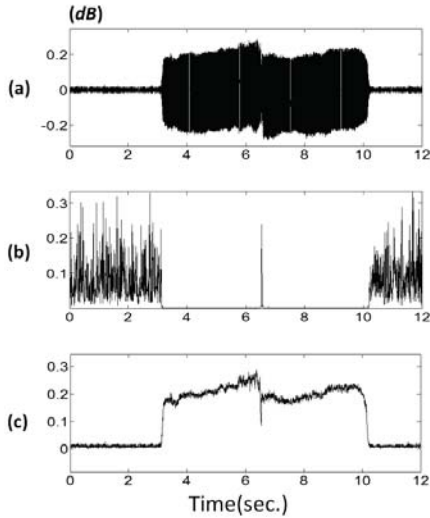
$$SVM_{ae1} = \begin{cases} 0 & , \text{ non - onset} \\ 1 & , \text{ onset} \end{cases} \quad (2)$$

$$SVM_{ae2} = \begin{cases} 0 & , \text{ non - offset} \\ 1 & , \text{ offset} \end{cases} \quad (3)$$

Figure 7 shows (a) a segment of audio sounds with one sequence of down-bowing and up-bowing, while Figure 7(b) and (c) display the results of IC and AE, respectively.

During the bowing state, the IC value is extremely small when compared to the results of the non-bowing state. IC seems to be a good index to identify the state of whether the violin is being played, or not. However, it can be seen that a time deviation is introduced if the system simply applies a hard threshold, e.g. 0.3. Alternatively, the AE value becomes larger at the playing state. But the issue of time deviation is also present in this feature, if a hard threshold is applied.

After calculating the IC and AE values, their variation is considered as one set of input data for SVM. The time period of each data is 100ms. Therefore,  $SVM_{ic}$ ,  $SVM_{ae1}$  and  $SVM_{ae2}$  are designed to detect the most plausible timing of onset, transition and offset.



**Figure 7.** (a) The audio sounds of down-bowing and up-bowing; (b) the results of IC; (c) the results of AE

### 3.2 Detection of bowing direction

In each movement, there are one onset, one offset, and several transitions. However, the total number of transitions will differ from the number of notes. After detection of the bowing state is completed, the duration between onset and offset is applied for segmenting the EMG signal of triceps (channel#6) and pectoralis (channel#8). For each note duration, there are three cases:

- (1) The duration from the onset to the first transition.
- (2) The duration from the current transition to the next transition.
- (3) The duration from the last transition of the offset.

This note duration extracted from the audio sound is called an active frame and the active frames are variant lengths from each other. The segment extracted by the audio sounds is called an *active frame* and the active frames are variant lengths from each other.

For each active frame, six features in [20] were applied to calculate the variations of EMG signal while bowing. The features are:

- Mean absolute value (MAV)
- Mean absolute value slope (MAVS)
- Zero crossings (ZC)
- Slope sign changes (SSC)
- Waveform length (WL)
- Correlation variation (CV)

Here, the active frame is experimentally divided into 20 segments for calculating MAV and WL, thus each active frame has 20 values of MAV and WL. For CV, we calculate the auto-correlation and cross-correlation of channel#6 and channel#8, and therefore there are 3 values

of CV for each active frame. Table 1 lists the number of each feature for each channel.

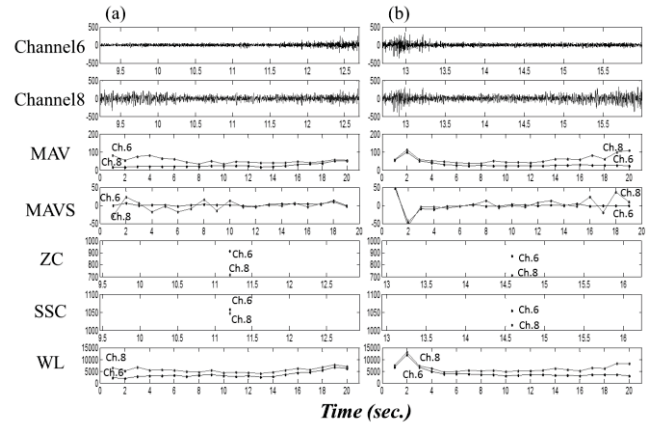
**Table 1.** The number of each feature per channel

Feature	MAV	MAVS	ZC	SSC	WL
Number	20	19	1	1	20

A more detailed description of those applied features could be found in [20]. Figure 8 displays the triceps EMG signal of one active frame (8s ~ 16s) and the results calculated by MAV, MAVS, ZC, SSC and WL. It can be seen that variations are exhibited for 6 features in violin playing with a down-up bowing movement.

The detection of bowing direction is also determined by a SVM classifier which is denoted as  $SVM_{dir}$  (3). For  $SVM_{dir}$ , a total of 125 inputs are used (61 inputs for channel#6 and channel#8 each, plus 3 values of CV) and it identifies whether the active EMG frame is in the up-bowing or down-bowing state.

$$SVM_{dir} = \begin{cases} 0 & , \quad Up - bowing \\ 1 & , \quad Down - bowing \end{cases} \quad (3)$$



**Figure 8.** One down-up bowing movement and its six features: (a) the down-bowing movement, (b) the up-bowing movement.

### 3.3 Performance evaluation

In our experiment, 10-fold cross-validation is used for  $SVM_{ic}$ ,  $SVM_{ae}$  and  $SVM_{dir}$ , and the performance evaluation calculates the accuracy (4), precision (5), recall (6) and F-score (7) of each detecting function.

$$Accuracy = \frac{True\ Positive + True\ Negative}{Positive + Negative}; \quad (4)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}; \quad (5)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}; \quad (6)$$

$$F\text{-score} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}; \quad (7)$$

The true positive means it correctly detected the movement; the false positive is a falsely detected movement; and the false negative is a missed detection.

4. EXPERIMENTAL RESULTS

In this section, the efficiency of the proposed SVMs is observed. An example of the proposed EMG segmentation is then compared to the prior work [15]. Finally, the averaged and overall simulation results are given.

4.1 The performance of SVM classifications

To illustrate both the proposed IC and AE effectively identify the sound states of onset and offset, respectively, Figure 9 shows the trend of IC and AE values in one down-up bowing movement by using the classification results for  $SVM_{ic}$  and  $SVM_{ae1}$  and  $SVM_{ae2}$ . Table 2 shows that, with the given FSM, the detection rate of onsets, transitions and offsets are 90%, 100%, 100%, respectively.

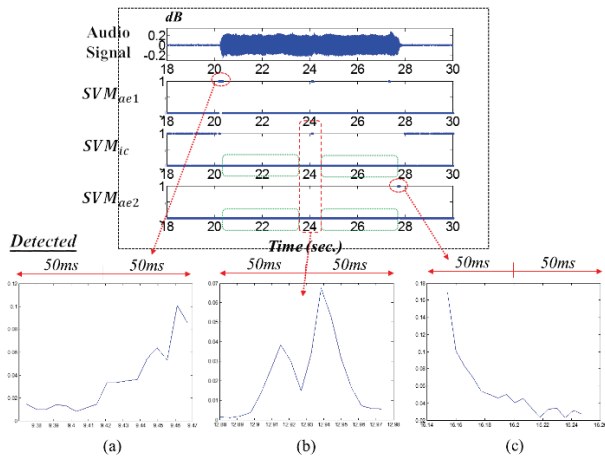


Figure 9. The results of 3 classifiers: (a) onsets, (b) transitions, (c) offsets.

Table 2. The detection results of the bowing states with the given FSM.

	Onset	Transition	Offset
Accuracy	90.00%	100%	100%
Precision	90.00%	100%	100%
Recall	90.00%	100%	100%
F-score	90.00%	100%	100%

Figure 10 shows the distribution of active EMG frames during up-bowing and down-bowing states, and it displays the distribution of MAV, MAVS and WL. The  $SVM_{dir}$  classifies the data with 85% accuracy.

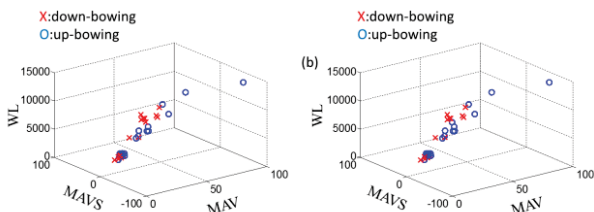


Figure 10. (a) The original distribution of up-bowing and down-bowing EMG frames; (b) the results of  $SVM_{dir}$  classification.

4.2 EMG segmentation

The results of EMG segmentation and its comparison to [15] are both illustrated in Figure 11. Figure 11 shows the

violin signal of task#1 with three movements. Figure 11 (b) and (c) are the EMG segmentations of our proposed method and [15], respectively. Channel#6 is used in this example to illustrate a sample output. It is believed that if there is an audio signal, then there is a corresponding movement. It can be seen that the results segmented by [15], without the additional information of the audio signal, could not precisely identify the segment of movements during bowing. However, the proposed method is based on the information from audio signals and clearly identifies the segment of behavioral changes during violin playing.

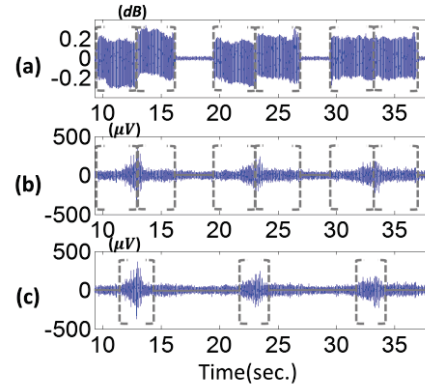


Figure 11. (a) The violin signal; (b) the proposed EMG segmentations; (c) the EMG segmentations of [15].

4.3 The simulation results

The detection result of violin bowing direction was given in Table 3 where accuracy, precision, recall and F-score are presented.

Table 3. The detection results of the bowing direction: (1) the detection results of ground truths of active frames; (2) the detection results of extracted active frames.

	(1)	(2)
Accuracy	85%	87.5%
Precision	76.92%	82.61%
Recall	100%	95%
F-score	86.96%	88.37%

The average detection results were shown to have excellent performance with an accuracy of 85%~87.5%. The results show that the proposed method efficiently identifies the bowing direction in violin playing.

5. CONCLUSION AND FUTURE WORK

The proposed biomechanical system for recording the audio sounds and EMG signals during playing an instrument was developed. The proposed method not only extracts the segment during movement and detects the moving direction of bowing, but with the additional information of violin sounds, changes in muscle activity as an element of motor control, could be efficiently detected when compared to the prior EMG segmentation (without any sound information). To the authors' knowledge, this is the first study which proposes such concept.

Future work will improve the detection rate of onset, transition and offset to extract the period of an active frame more precisely. The detection of the bowing direction will be also improved. Furthermore, the relationship between the musical sounds and the muscular activities of players in musical performance will be observed and analyzed. By measuring the music and the player's muscular activity, better insights can be made into the neurophysiological control during musical performances and may even prevent players from the injuries as greater insights into these mechanisms are made.

## 6. REFERENCES

- [1] DC. Harding, KD. Brandt, and BM. Hillberry: "Minimization of finger joint forces and tendon tensions in pianists," *Med. Probl. Perform Art* pp.103-104, 1989.
- [2] KC. Engel, M. Flanders, and JF. Soechting: "Anticipatory and sequential motor control in piano playing," *Exp Brain Res.* pp. 189-199, 1997.
- [3] D. Parlitz, T. Peschel, and E. Altenmüller: "Assessment of dynamic finger forces in pianists: Effects of training and expertise," *J. Biomech.* pp.1063-1067, 1998.
- [4] H. Kinoshita, S. Furuya, T. Aoki, and E. Altenmüller: "Loudness control in pianists as exemplified in keystroke force measurements on different touches," *J Acoust Soc Am.* pp. 2959-69, 2007.
- [5] AE. Minetti, LP. Ardigò, and T. McKee: "Keystroke dynamics and timing: Accuracy, precision and difference between hands in pianist's performance," *J Biomech.* pp. 3738-43, 2007.
- [6] R. Merletti and P. Parker: *Electromyography: physiology, engineering, and noninvasive applications*, Wiley-IEEE Press, 2004.
- [7] C.-J. Lai, R.-C. Chan, and T.-F. Yang *et al.*: "EMG changes during graded isometric exercise in pianists: comparison with non-musicians," *Journal of the Chinese Medical Association*, vol. 71, no. 11, pp. 571-575, 2008.
- [8] M. Candidi, L. M. Sacheli, and I. Mega *et al.*: "Somatotopic mapping of piano fingering errors in sensorimotor experts: TMS studies in pianists and visually trained musically naïves," *Cerebral Cortex*, vol. 24, no. 2, pp. 435-443, 2014.
- [9] S. Furuya, T. Aoki, and H. Nakahara *et al.*: "Individual differences in the biomechanical effect of loudness and tempo on upper-limb movements during repetitive piano keystrokes," *Human movement science*, vol. 31, no. 1, pp. 26-39, 2012.
- [10] D. L. Rickert and M. Halaki, K. A. Ginn *et al.*: "The use of fine-wire EMG to investigate shoulder muscle recruitment patterns during cello bowing: The results of a pilot study," *Journal of Electromyography and Kinesiology*, vol. 23, no. 6, pp. 1261-1268, 2013.
- [11] A. Fjellman-Wiklund and H. Grip, J. S. Karlsson *et al.*: "EMG trapezius muscle activity pattern in string players: Part I—is there variability in the playing technique?," *International journal of industrial ergonomics*, vol. 33, no. 4, pp. 347-356, 2004.
- [12] J. G. Bloemsaat, R. G. Meulenbroek, and G. P. Van Galen: "Differential effects of mental load on proximal and distal arm muscle activity," *Experimental brain research*, vol. 167, no. 4, pp. 622-634, 2005.
- [13] S. Fujii and T. Moritani: "Spike shape analysis of surface electromyographic activity in wrist flexor and extensor muscles of the world's fastest drummer," *Neuroscience letters*, vol. 514, no. 2, pp. 185-188, 2012.
- [14] S. Furuya and H. Kinoshita: "Organization of the upper limb movement for piano key-depression differs between expert pianists and novice players," *Experimental brain research*, vol. 185, no. 4, pp. 581-593, 2008.
- [15] P. Mazurkiewicz, "Automatic Segmentation of EMG Signals Based on Wavelet Representation," *Advances in Soft Computing Volume 45*, 2007, pp 589-595
- [16] Bodybuilding is lifestyle! "Chest - Bodybuilding is lifestyle!"<http://www.bodybuildingislifestyle.com/chest/>.
- [17] Bodybuilding is lifestyle! "Chest - Bodybuilding is lifestyle!"  
<http://www.bodybuildingislifestyle.com/hamstrings/>.
- [18] WJJ. Boo, Y. Wang, and A. Loscos, "A violin music transcriber for personalized learning," pp 2081-2084, *IEEE International Conference on Multimedia and Expo*, 2006.
- [19] BE. Boser, IM. Guyon and VN. Vapnik: "A training algorithm for optimal margin classifiers," In Fifth Annual Workshop on Computational Learning Theory, ACM 1992.
- [20] AJ. Andrews: "Finger movement classification using forearm EMG signals," *M. Sc. dissertation, Queen's University*, Kingston, ON, Canada, 2008.

# AUTOMATIC KEY PARTITION BASED ON TONAL ORGANIZATION INFORMATION OF CLASSICAL MUSIC

Lam Wang Kong, Tan Lee

Department of Electronic Engineering

The Chinese University of Hong Kong Hong Kong SAR, China

{wklam, tanlee}@ee.cuhk.edu.hk

## ABSTRACT

Key information is a useful information for tonal music analysis. It is related to chord progressions, which follows some specific structures and rules. In this paper, we describe a generative account of chord progression consisting of phrase-structure grammar rules proposed by Martin Rohrmeier. With some modifications, these rules can be used to partition a chord symbol sequence into different key areas, if modulation occurs. Exploiting tonal grammar rules, the most musically sensible key partition of chord sequence is derived. Some examples of classical music excerpts are evaluated. This rule-based system is compared against another system which is based on dynamic programming of harmonic-hierarchy information. Using Kostka-Payne corpus as testing data, the experimental result shows that our system is better in terms of key detection accuracy.

## 1. INTRODUCTION

*Chord* progression is the foundation of harmony in tonal music and it can determine the key. The *key* involves certain melodic tendencies and harmonic relations that maintain the tonic as the centre of attention [4]. Key is an indicator of the musical style or character. For example, the key C major is related to innocence and pureness, whereas F minor is related to depression or funereal lament [16]. Key detection is useful for music analysis. A classical music piece may have several modulations (key changes). A change of key means a change of tonal center, the adoption of a different tone to which all the other tones are to be related [10]. Key change allows tonal music to convey a sense of long-range motion and drama [17].

Keys and chord labels are interdependent. Even if the chord labels are free from errors, obtaining the key path is often a non-trivial task. For example, if a music excerpt has been analyzed with the chord sequence  $[B\flat, F, G_{min}, A_{min}, G, C]$ , how would you analyze its key? Is it a phrase entirely in  $B\flat$  major or C major, as

they are the beginning or ending chords? Seems it is not, as  $B\flat$  major chord is normally not a member chord of C major and vice versa. It seems that there must be a key change in the middle. But how would you find out the point of key change, and how does the key change? With the help of the tonal grammar tree analysis in §2.1, a good estimate of the key path can be obtained. To start with, we assume that the excerpt consists of harmonically complete phrase(s) and the chord labels are free from errors.

There are some existing algorithms to estimate the key based on chord progression. These algorithms can be classified into two categories: *statistical-based* and *rule-based* approach. Hidden Markov model is very often used in the statistical approach. Lee & Stanley [7] extracted key information by performing harmonic analysis on symbolic training data and estimated the model parameters from them. They built 24 key-specific HMMs (all major and minor keys) for recognizing a single global key which has the highest likelihood. Raphael & Stoddard [11] performed harmonic analysis on pitch and rhythm. They divided the music into a fixed musical period, usually a measure, and associate a key and chord to each of period. They performed functional analysis of chord progression to determine the key. Unlabeled MIDI files were used to train the transition and output distributions of HMM. Instead of recognizing the global key, it can track the local key. Cateau *et al.* [2] described a probabilistic framework for simultaneous chord and key recognition. Instead of using training data, Lerdahl's representation of tonal space [8] were used as a distance metric to model the key and chord transition probabilities. Shenoy *et al.* [15] proposed a rule-based approach for determining the key from chord sequence. They created a reference vector for each of the 12 major and minor keys, including the possible chords within the key. Higher weights were assigned to primary chords (tonic, subdominant and dominant chords). The chord vector obtained from audio data were compared against the reference vector using weighted cosine similarity. The pattern with the highest rank is chosen as the selected global key.

This paper uses a rule-based approach to model tonal harmony. A context-free dependency structure is used to exhaust all the possible combinations of key paths, and the best one is selected according to music knowledge. The main objective of this research is to exploit this tonal context-free dependency structure in order to partition an excerpt of classical music into several key sections.



© Lam Wang Kong, Tan Lee.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Lam Wang Kong, Tan Lee. "Automatic key partition based on Tonal Organization Information of Classical Music", 15th International Society for Music Information Retrieval Conference, 2014.

Functional level	Scale degree level
$TR \rightarrow DR T$	$T \rightarrow I$
$DR \rightarrow SR D$	$T \rightarrow I IV I$
$TR \rightarrow TR DR$	$S \rightarrow IV$
$XR \rightarrow XR XR$	$D \rightarrow V   vii$
$phrase \rightarrow TR$	$T \rightarrow vi   III$
	$D \rightarrow VII (minor)$
Added rules for scale degree level:	$S \rightarrow ii (major)$
$S \rightarrow ii (minor)$	$S \rightarrow VI   bII (minor)$
$T \rightarrow I IV VI   VI IV I   I bII I$	$X \rightarrow D(X) X$
$D \rightarrow I V, \text{ after } S \text{ or } D(V)$	$D(X) \rightarrow V/X   vii/X$

TR	tonic region	S	predominant function
DR	dominant region	X	any specific function
SR	predominant region	$D(\cdot)$	secondary dominant
XR	any specific region	X / Y	X of Y chord
T	tonic function	$I, III...$	major chords
D	dominant function	$ii, vi...$	minor chords

**Table 1.** Rules (top) and labels (bottom) used in our system

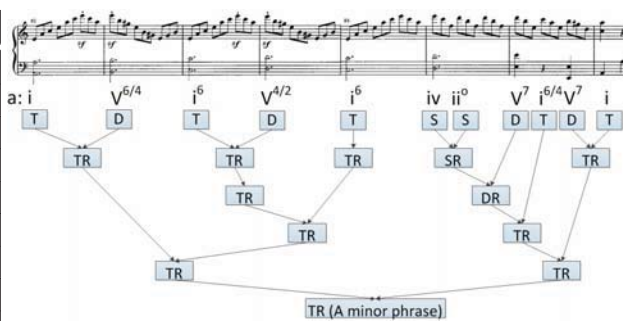
## 2. TONAL THEORY OF CLASSICAL MUSIC

### 2.1 Schenkerian analysis and formalization

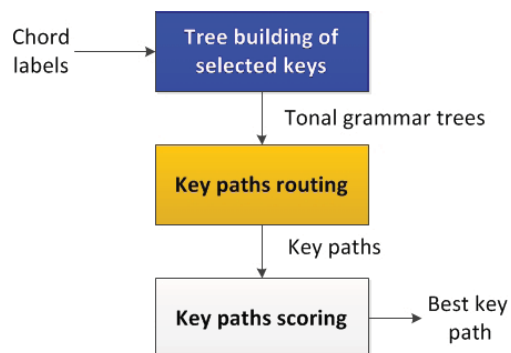
To interpret the structure of the tonal music, Schenkerian analysis [14] is used. The input is assumed to be classical music with one or more tonal centre (tonal region). Each tonal centre can be elaborated into tonic – dominant – tonic regions [1]. The dominant region can be further elaborated into predominant-dominant regions. Each region can be recursively elaborated to form a *tonal grammar tree*. We can derive the key information by referring to the top of the tree, which groups the chord sequence into a tonal region.

Context-free grammar can be used to formalize this tree structure. A list of generative syntax is proposed by Rohrmeier [13] in the form of  $V \rightarrow w$ .  $V$  is a single non-terminal symbol, while  $w$  is a string of terminals and/or non-terminals. Chord symbols (eg.  $IV$ ) are represented by terminals. They are the leaves of the grammar tree. Tonal functions (eg.  $T$  for tonic) or regions (eg.  $TR$  for tonic region) are represented by non-terminals. They can be the internal nodes or the root of the grammar tree. For instance, the rule  $D \rightarrow V | vii$  indicates that the  $V$  or  $vii$  chord can be represented by the dominant function. The rule  $S \rightarrow ii (major)$  indicates that  $ii$  chord can be represented by the predominant function only when the current key is major. Originally Rohrmeier has proposed 28 rules. Some of them were modified to suit classical music and were listed in Table 1.

Based on this set of rules, Cocke–Younger–Kasami parsing algorithm [18] is used to construct a tonal grammar tree. If a music input is harmonically valid, a single tonal grammar tree can be built like in Figure 1. Else some scattered tree branches are resulted and cannot be connected to one single root.



**Figure 1.** Example of a tonal grammar tree (single key)



**Figure 3.** Flow diagram of our key partitioning system

### 2.2 Modulation

In Rohrmeier’s generative syntax of tonal harmony, modulation is formalized as a new local tonic [13]. Each functional region (new key section) is grouped as a single non-tonic chord in the original passage, and they may relate this (elaborated) chord to the neighbouring chords.

In this research we have a more general view of modulation. As a music theorist, Reger had published a book *Modulation*, showing how to modulate from C major / minor to every other key [12]. Modulation to every other key is possible, but modulation to harmonically closer keys is more common [10]. For instance, if the music is originally in C major, it is more probable to modulate to G major instead of B major. Lerdahl’s chordal distance [8] is used to measure the distance between different keys. Here Rohrmeier’s modulation rules in [13] are not used. Instead, a tonal grammar tree is built for each new key section, and the key path with the best score is chosen. Any key changes explainable by tonicization (temporary borrowing of chords from other keys), such as the chords  $[I V/V V I]$ , is not considered as a modulation. Figure 2 shows an example of tonal grammar tree with modulation, from E minor to  $D\sharp$  minor. It is presented by two disjunct trees.

## 3. SYSTEM BUILDING BLOCKS

### 3.1 Overview

The proposed key partitioning system is shown as in Figure 3. This system takes a sequence of chord labels (e.g. A minor, E major) and outputs the best key path. The path may consist of only one key, or several keys. For example,  $[F F F F F F]$  or  $[Am Am Am C C C]$  (m indicates minor

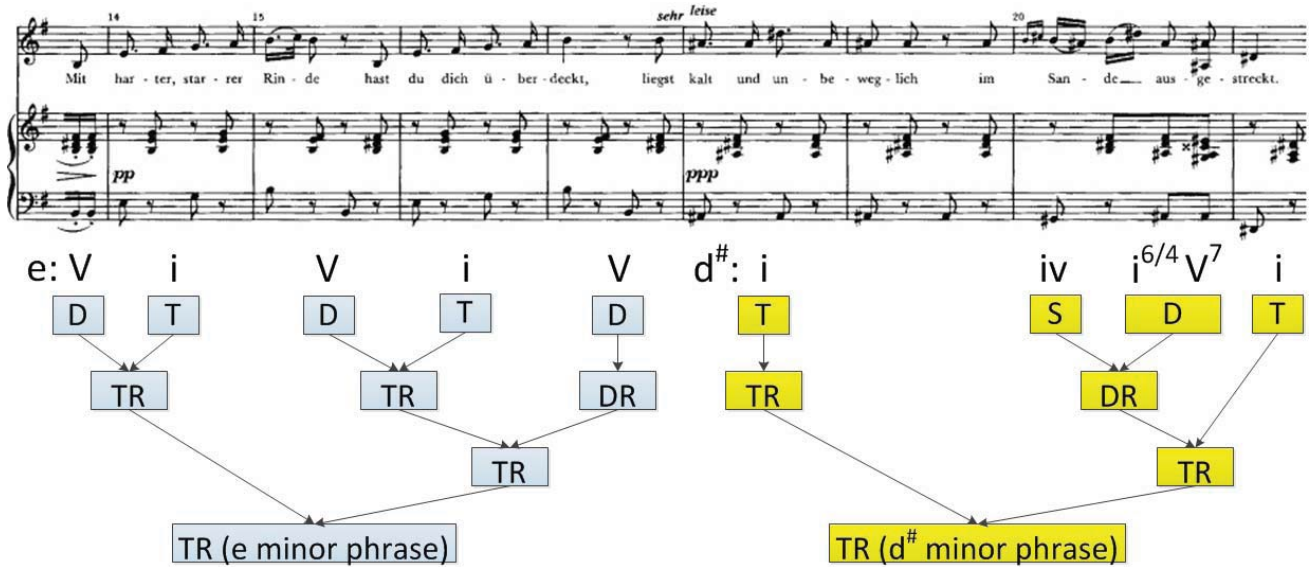


Figure 2. Example of a tonal grammar tree with modulation

chords, other chords are major) are both valid key paths. The *Tonal Grammar Tree* mentioned in §2.1 is the main tool used in this system.

### 3.2 Algorithm for key partitioning

Each key section is assumed to have at least one tonic chord. The top of each grammar tree must be TR (tonic region), so the key section is a complete tonal grammar tree by itself. Furthermore, the minimum length of each key section is assumed to be 3 chords. However, if no valid paths can be found, key sections with only 2 chords are also considered.

The algorithm is as follows:

1. In a chord sequence, hypothesize any of the chord label as the tonic of a key. Derive the tonal grammar tree of each key.
2. Find if there is any key that can build a single complete tree for the entire sequence. If yes, limit the valid paths to these single-key paths and go to step 7. This phrase is assumed to have a single key only. Else go to next step.
3. For each chord label in the sequence, find the maximum possible accumulated chord sequence length of each key section (up to that label). Determine if this sequence is breakable at that label (The secondary dominant chord is dependent on the subsequent chord. For example, the tonicization segment V/V V cannot be broken in the middle, as V/V is dependent on V chord).
4. Find out all possible key sections with at least 3 chords including at least one tonic chord.
5. Find out all valid paths traversing all the possible key sections, from beginning to end, in a brute-force manner.

Path no.	Key paths					
1	Gm	Gm	Gm	Am	Am	Am
2	Gm	Gm	Gm	C	C	C
3	B $\flat$	B $\flat$	B $\flat$	Am	Am	Am
4	B $\flat$	B $\flat$	B $\flat$	C	C	C

Table 2. All valid key paths in the example

6. If no valid paths can be found, go back to step 4 and change the requirement to “at least 2 chords”. Else proceed to step 7.
7. Evaluate the path score of all valid paths and select the one with the highest score to be the best key path.

A simple example is used to illustrate this process. The input chord sequence is [B $\flat$  F Gm Am G C]. Incomplete trees with the keys (B $\flat$ , F, Gm, Am, G, C) are built. As all the trees are incomplete, proceed to step 3 and the accumulated length is calculated. The B $\flat$  major tree is shown in Figure 4 as an example. Other five trees (F, Gm, Am, G, C) were built in the same fashion. Either key sections 1-3 or 1-4 of B $\flat$ major are valid key sections as they can all be grouped into a single TR and they have at least 3 chords. Then all the valid key paths were found and they are listed in Table 2. All the path scores were evaluated by the equation (1) of the next section.

### 3.3 Formulation

We have several criteria for choosing the best key path. A good choice of a key section should be rich in tonic and dominant chords, as they are the most important chords to define and establish a key [10]. It is more preferable if the key section starts and ends with the tonic chord, and with less tonicizations as a simpler explanation is better than a complicated one. In a music excerpt, less modulations and modulations to closer keys are preferred. We formulate

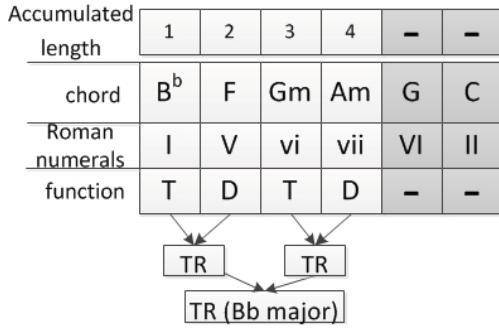


Figure 4. The incomplete B<sup>b</sup> major Tree

these criteria with equation (1):

$$S_{total} = aS_{td} - bS_{ton} - cS_{cost} + dS_{stend} - eS_{sect} \quad (1)$$

where  $S_{td}$  is the no. of tonic and dominant chords,  $S_{ton}$  is the total number of tonicization steps. For example, in chord progression V/V/ii V/ii ii, the first chord has two steps, while the second chord has one step.  $S_{ton} = 2 + 1 + 0 = 3$ .  $S_{cost}$  is the total modulation cost: the total tonal distance of each modulation measured by Lerdahl's distance defined in [8].  $S_{stend}$  indicates whether the excerpt starts and ends with tonic or not.  $S_{sect}$  is the total number of key sections. If a key section has only 2 chords, it is counted as 3 in  $S_{sect}$  as a penalty. These parameters control how well chords fit in a key section against how often the modulation occurs.  $S_{td}$ ,  $S_{ton}$  and  $S_{stend}$  maximizes fitness of the chord sequence to a key section.  $S_{cost}$  and  $S_{sect}$  induce penalty whenever modulation occurs. The parameters  $S_{td}$ ,  $S_{ton}$ ,  $S_{cost}$ ,  $S_{stend}$  and  $S_{sect}$  are normalized so that their mean and standard deviation are 0 and 1 respectively. All the coefficients, namely  $a, b, c, d, e$ , are determined experimentally, although a slightly different set of values does not have a large effect on the key partitioning results. They are set at  $[a, b, c, d, e] = [1, 0.4, 2, 2, 0.4]$ . Key structure is generally thought to be hierarchical. An excerpt may have one level of large-scale key changes and another level of tonicizations [17], and the boundary is not well-defined. So it seemed fair to adjust these parameters in order to match the level of key changes labeled by the ground truth. The key path with the highest  $S_{total}$  is chosen as the best path.

## 4. EXPERIMENTS

### 4.1 Settings

To test the system, we have chosen the Kostka-Payne corpus, which contains classical music excerpts in a theory book [5]. This selection has 46 excerpts, covering compositions of many famous composers. They serve as representative examples of classical music in common practice period (around 1650-1900). All of the excerpts were examined. This corpus has ground truth key information labeled by David Temperley<sup>1</sup>. The mode (major or minor) of the key was labeled by an experienced musician. The chord labels are also available from the website, with the mode

<sup>1</sup> <http://www.theory.esm.rochester.edu/temperley/kp-stats/>

added by the experienced musician<sup>2</sup>. All the chord types have been mapped to their roots: major or minor. There are 25 excerpts with a single key and 21 excerpts with key changes (one to four key changes). The longest excerpt has 47 chords whereas the shortest excerpt has 8 chords. The instrumentation ranges from solo piano to orchestral. As we assume the input chord sequence to be harmonically complete, the last chord of excerpts 9, 14 and 15 were truncated as they are the starting chord of another phrase. There are 866 chords in total. For every excerpt, the partitioning algorithm in §3.2 is used to obtain the best path.

### 4.2 Baseline system

To the best of author's knowledge, there is currently no key partitioning algorithm directly use chord labels as input. To compare the performance of our key partitioning system, another system based on Krumhansl's harmonic-hierarchy information and dynamic programming were set up. Krumhansl's key profile has been used in many note-based key tracking systems such as [3, 9]. Here Krumhansl's harmonic-hierarchy ratings (listed in Chapter 7 of [6]) are used to obtain the perceptual closeness of a chord in a particular key. A higher rating corresponds to a higher tendency to be part of the key. As a fair comparison, the number of chords in a key section is restricted to be at least three, which is the same in our system. To prevent fluctuations of the key, a penalty term  $D(x, y)$  is imposed on key changes. The multiplicative constant of penalty term  $\alpha$  is determined experimentally to give the best result. The best key path is found iteratively by the dynamic programming technique presented by equations (2) and (3):

$$A_x[1] = H_x[1] \quad \forall x \in K \quad (2)$$

$$A_x[n] = \max \left\{ \begin{array}{l} A_x[n-1] + H_x[n], \\ A_y[n-1] + H_x[n] - \alpha D(x, y) \end{array} \right\} \quad \forall x, y \in K, \text{ where } y \neq x \quad (3)$$

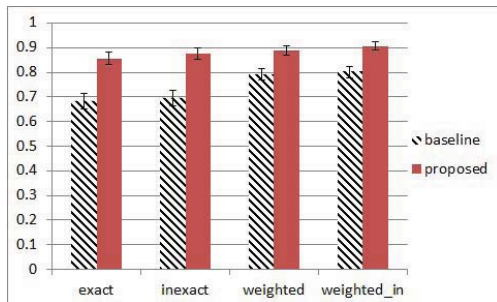
$H_x[n]$  is the harmonic-hierarchy rating of the  $n^{\text{th}}$  chord with the key  $x$ .  $A_x[n]$  is the accumulated key strength of the  $n^{\text{th}}$  chord when the current key is  $x$ .  $K$  is the set of all possible keys.  $D(x, y)$  is the distance between keys  $x, y$  based on the key distance in [6] derived from multidimensional scaling. The best path can be found by obtaining the largest  $A_x$  of the last chord and tracking all the way back to  $A_x[1]$ . The same Kostka-Payne corpus chord labels were used to test this baseline system. The best result was obtained by setting  $\alpha = 4.5$ .

### 4.3 Results

The key partitioning result of our proposed system and the baseline system were compared against the ground truth provided by Temperley. Four kinds of result metrics were used. The average matching score is shown in Figure 5.

<sup>2</sup> All the chord and key labels can be found here: <https://drive.google.com/file/d/0B0Td6LwTULvMVJ6MFcyYWsxVzQ/edit?usp=sharing>





**Figure 5.** Key partitioning result, with 95% confidence interval

*Exact* indicates the exact matches between the obtained key path and the ground truth. As modulation is a gradual process, the exact location of key changes may not be definitive. It is more meaningful to consider *Inexact*. For *inexact*, the obtained key is also considered as correct if it matches the key of the previous or next chord. *MIREX* refers to the MIREX 2014 Audio key detection evaluation standard<sup>3</sup>. Harmonically close keys will be given a partial point. Perfect fifth is awarded with 0.5 points, relative minor/ major 0.3 points, whereas parallel major/ minor 0.2 points. This is useful as sometimes a chord progression may be explainable by two different related keys. *MIREX\_in* refers to the MIREX standard, but with the addition that the points of previous or next chord will also be considered and the maximum point will be chosen as the matching score of that chord.

The proposed system outperforms the baseline system by about 18% for exact or inexact matching and 0.1 points for MIREX-related scores. It shows that our knowledge-based tonal grammar tree system is better than the baseline system which is based on perceptual closeness. Tonal structural information is exploited, so we have a better understanding of the chord progression and modulations.

#### 4.4 Error analysis

The ground truth key information are compared against the key labels generated by the proposed algorithm. 17 boundary errors were detected, i.e. the key label of the previous or next chord was recognized instead. In classical music, modulation is usually not a sudden event. It occurs gradually through several *pivot chords* (chords common to both keys) [10]. Therefore it is sometimes subjective to determine the boundary between two key sections. It may not be a wrong labeling if the boundary is different from the ground truth. Other types of error are listed in Table 3.

The most common error is the misclassification as dominant key, which is the closest related key [10]. It shares many common chords with the tonic key. From Table 4, the same chord sequence can be analyzed by two keys that are dominantly-related. Although the B $\flat$  major analysis contains more tonicizations, the resultant score disadvantage may be outweighed by the cost of key changes, if it is followed by a B $\flat$  major section.

Key relation	Semitone difference	total no.	%
Dominant	7	35	32.7
Supertonic	2	32	29.9
Relative	3	11	10.3
Parallel	0	11	10.3
Minor 3 <sup>rd</sup>	3	9	8.4
Major 3 <sup>rd</sup>	4	8	7.5
Leading tone	1	3	2.8
Tritone	6	2	1.9

**Table 3.** Eight categories of the 107 error labels

chord symbols	Gm	C	F	B $\flat$	Gm	C	F
F major	ii	V	I	IV	ii	V	I
B $\flat$ major	vi	V/V	V	I	vi	V/V	V

**Table 4.** Analysis with two different keys

Modulations between keys that are supertonic-related (differs by 2 semitones) or relative major / minor have a similar problem as the dominant key modulation. Many common chords are shared among both keys, so it is easy to confuse these two keys. It is worth to mention that nine of the supertonic-related errors came from excerpt 45. In Temperley's key labels, the whole excerpt is labeled as C major with measures 10-12 considered as a passage of tonicization. However, in [5], it was written that "*Measures 10-12 can be analyzed in terms of secondary functions or as a modulation*". If the measures 10-12 are considered as a modulation to D minor, then the analysis of these nine chords is correct.

The parallel key modulation, for example from C major to C minor, has a different problem. Sometimes composers tend to start the phrase with a new mode (major or minor) without much preparation, as the tonic is the same. Fluctuation between major and minor of the same key has always been common [10]. When the phrase information is absent, the exact position of modulation cannot be found by the proposed system.

In another way, there may exist some ornament notes that obscure the real identity of a chord, so that the chord symbol analyzed acoustically is different from the chord symbol analyzed structurally or grammatically. For example, in Figure 6, the first two bars should be analyzed as  $IV^6-vii^{\phi7}-I$  progression in A major. However, the C $\sharp$  of the *I* chord is delayed to the next chord. The appoggiatura B $\sharp$  made the *I* chord sound as a *i* chord, the tonic minor chord instead. Similarly, the last two bars should be analyzed as  $IV^{6/5}-vii^{\phi7}-i$  in F $\sharp$  minor. However, the passing note A $\sharp$  made the *i* chord sound as a *I* chord, the original A is delayed to the next chord. In these two cases, the key derived by the last chord in the progression is in conflict with the other chords. Hence the key will be recognized wrongly if the acoustic chord symbol is provided instead of the structural chord symbol.

## 5. DIFFICULTIES

The biggest problem of this research is lack of labeled data. To the best of our knowledge, large chord label database

<sup>3</sup> [http://www.music-ir.org/mirex/wiki/2014:Audio\\_Key\\_Detection](http://www.music-ir.org/mirex/wiki/2014:Audio_Key_Detection)



**Figure 6.** Excerpt from Mozart’s Piano Concerto no. 23, 2<sup>nd</sup> movement

for classical music is absent. The largest database we could find is the Kostka-Payne corpus used in this paper. In the future, we may consider manually label more music pieces to check if the system works generally well in classical music.

Moreover, key partitioning is sometimes subjective to listener’s perception. In some cases, there are several pivot chords to establish the new key center. “Ground truth” boundaries of key sections are sometimes set arbitrarily. Or there are several sets of acceptable and sensible partitions of key sections. This problem is yet to be studied. Inconsistency between acoustic and structural chord symbols mentioned in §4.4 is also yet to be solved. For any rule-based systems, exceptions may occur. Composers may deliberately break some traditions in the creative process. It is not possible to handle all these exceptional cases.

## 6. FUTURE WORK AND CONCLUSION

We have only considered major and minor chords in this paper. As dominant 7<sup>th</sup> and diminished chords are common in classical music, we may consider expanding the chord type selection to make chord labels more accurate. The current system assumes chord labels to be free of errors. We plan to study the method of key tracking in the presence of chord label errors. Then we may incorporate this system to the chord classification system for audio key detection, as the key and chord progression is interdependent. Currently the input phrases must be complete in order to make this tree building process work. We plan to find the key partition method for incomplete input phrases. A more efficient algorithm for tree building process, instead of brute-force, is yet to be discovered. Then less trees are required to be built.

In this paper, we have discussed the uses of tonal grammar to partition key sections of classical music. The proposed system outperforms the baseline system which uses dynamic programming on Krumhansl’s harmonic-hierarchy ratings. This tonal grammar is useful for tonal classical music information retrieval and hopefully more uses can be found.

## 7. REFERENCES

- [1] A. Cadwallader and D. Gagné. *Analysis of Tonal Music: A Schenkerian Approach*. Oxford University Press, Oxford, 1998.
- [2] B. Cateau, J. Martens, and M. Leman. A probabilistic framework for audio-based tonal key and chord recognition. *Advances in Data Analysis*, (2005):1–8, 2007.
- [3] E. Gómez and P. Herrera. Estimating The Tonality Of Polyphonic Audio Files: Cognitive Versus Machine Learning Modelling Strategies. In *ISMIR*, pages 1–4, 2004.
- [4] B. Hyer. Key (i). In S. Sadie, editor, *The New Grove Dictionary of Music and Musicians*. Macmillan Publishers, London, 1980.
- [5] S. M. Kostka and D. Payne. *Workbook for tonal harmony, with an introduction to twentieth-century music*. McGraw-Hill, New York, 3rd ed. edition, 1995.
- [6] C. L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford University Press, New York, 1990.
- [7] K. Lee and M. Slaney. Acoustic Chord Transcription and Key Extraction From Audio Using Key-Dependent HMMs Trained on Synthesized Audio. In Array, editor, *Ieee Transactions On Audio Speech And Language Processing*, volume 16, pages 291–301. Ieee, 2008.
- [8] F. Lerdahl. *Tonal pitch space*. Oxford University Press, Oxford, 2001.
- [9] H. Papadopoulos and G. Peeters. Local Key Estimation From an Audio Signal Relying on Harmonic and Metrical Structures. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(4):1297–1312, May 2012.
- [10] W. Piston. *Harmony*. W. W. Norton, New York, rev. ed. edition, 1948.
- [11] C. Raphael and J. Stoddard. Functional harmonic analysis using probabilistic models. *Computer Music Journal*, pages 45–52, 2004.
- [12] M. Reger. *Modulation*. Dover Publications, Mineola, N.Y., dover ed. edition, 2007.
- [13] M. Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1):35–53, Mar. 2011.
- [14] H. Schenker. *Free Composition*. Longman, New York, London, 1979.
- [15] A. Shenoy and R. Mohapatra. Key determination of acoustic musical signals. *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, pages 1771–1774, 2004.
- [16] R. Steblin. *A history of key characteristics in the eighteenth and early nineteenth centuries*. University of Rochester Press, Rochester, NY, 2nd edition, 2002.
- [17] D. Temperley. *The cognition of basic musical structures*. MIT Press, Cambridge, Mass., 2001.
- [18] D. H. Younger. Recognition and parsing of context-free languages in time  $n^3$ . *Information and Control*, 10(2):189–208, 1967.

# BAYESIAN SINGING-VOICE SEPARATION

Po-Kai Yang, Chung-Chien Hsu and Jen-Tzung Chien

Department of Electrical and Computer Engineering, National Chiao Tung University, Taiwan  
 {niceallen.cm01g, chien.cm97g, jtchien}@nctu.edu.tw

## ABSTRACT

This paper presents a Bayesian nonnegative matrix factorization (NMF) approach to extract singing voice from background music accompaniment. Using this approach, the likelihood function based on NMF is represented by a Poisson distribution and the NMF parameters, consisting of basis and weight matrices, are characterized by the exponential priors. A variational Bayesian expectation-maximization algorithm is developed to learn variational parameters and model parameters for monaural source separation. A clustering algorithm is performed to establish two groups of bases: one is for singing voice and the other is for background music. Model complexity is controlled by adaptively selecting the number of bases for different mixed signals according to the variational lower bound. Model regularization is tackled through the uncertainty modeling via variational inference based on marginal likelihood. The experimental results on MIR-1K database show that the proposed method performs better than various unsupervised separation algorithms in terms of the global normalized source to distortion ratio.

## 1. INTRODUCTION

Singing voice conveys important information of a song. This information is practical for many music-related applications including singer identification [11], music emotion annotation [21], melody extraction, lyric recognition and lyric synchronization [6]. However, singing voice is usually mixed with background accompaniment in a music signal. How to extract the singing voice from a single-channel mixed signal is known as a crucial issue for music information retrieval. Some approaches have been proposed to deal with single-channel singing-voice separation.

There are two categories of approaches to source separation: supervised learning [2] and unsupervised learning [8, 9, 13, 22]. Supervised approach conducts the single-channel source separation given by the labeled training data from different sources. In the application of singing-voice separation, the separate training data of singing voice

and background music should be collected. But, it is more practical to conduct the unsupervised learning for blind source separation by using only the mixed test data. In [13], the repeating structure of the spectrogram of the mixed music signal was extracted and applied for separation of music and voice. The repeating components from accompaniment signal were separated from the non-repeating components from vocal signal. A binary time-frequency masking was applied to identify the repeating background accompaniment. In [9], a robust principal component analysis was proposed to decompose the spectrogram of mixed signal into a low-rank matrix for accompaniment signal and a sparse matrix for vocal signal. System performance was improved by imposing the harmonicity constraints [22]. A pitch extraction algorithm was inspired by the computational auditory scene analysis [3] and was applied to extract the harmonic components of singing voice.

In general, the issue of singing-voice separation is seen as a single-channel source separation problem which could be solved by using the learning approach based on the nonnegative matrix factorization (NMF) [10, 19]. Using NMF, a nonnegative matrix is factorized into a product of a basis matrix and a weight matrix which are nonnegative [10]. NMF can be directly applied in Fourier spectrogram domain for audio signal processing. In [7], the nonnegative sparse coding was proposed to conduct sparse learning for overcomplete representation based on NMF. Such sparse coding provides efficient and robust solution to NMF. However, how to determine the regularization parameter for sparse representation is a key issue for NMF. In addition, the time-varying envelopes of spectrogram convey important information. In [16], one dimensional convolutive NMF was proposed to extract the bases, which considered the dependencies across successive columns of input spectrogram, and was applied for supervised single-channel speech separation. In [14], two dimensional NMF was proposed to discover fundamental bases for blind musical instrument separation in presence of harmonic variations from piano and trumpet. Number of bases was empirically determined. Nevertheless, the selection of the number of bases is known as a model selection problem in signal processing and machine learning. How to tackle this regularization issue plays an important role to assure generalization for future data in ill-posed condition [1].

Basically, uncertainty modeling via probabilistic framework is helpful to improve model regularization for NMF.



The uncertainties in singing-voice separation may come from improper model assumption, incorrect model order and possible noise interference, nonstationary environment, reverberant distortion. Under probabilistic framework, nonnegative spectral signals are drawn from probability distributions. The nonnegative parameters are also represented by prior distributions. Bayesian learning is introduced to deal with uncertainty decoding and build a robust source separation by maximizing the marginal likelihood over the randomness of model parameters. In [15], Bayesian NMF (BNMF) was proposed for image feature extraction based on the assumption of Gaussian likelihood and exponential prior. In the BNMF [4], an approximate Bayesian inference based on variational Bayesian (VB) algorithm using Poisson likelihood for observation data and Gamma prior for model parameters was proposed for image reconstruction. Implementation cost was demanding due to the numerical calculation of shape parameter. Although NMF was presented for singing-voice separation in [19, 23], the regularization issue was ignored and the sensitivity of system performance due to uncertain model and ill-posed condition was serious.

This paper presents a new model-based singing-voice separation. The novelties of this paper are twofold. The first one is to develop Bayesian approach to unsupervised singing-voice separation. Model uncertainty is compensated to improve the performance of source separation of vocal signal and background accompaniment signal. Number of bases is adaptively determined from the mixed signal according to the variational lower bound of the logarithm of a marginal likelihood over NMF basis and weight matrices. The second one is the theoretical contribution in Bayesian NMF. We construct a new Bayesian NMF where the likelihood function in NMF is drawn from Poisson distribution and the model parameters are characterized by exponential distributions. A closed-form solution to hyperparameters using the VB expectation-maximization (EM) [5] algorithm is derived for ease of implementation and computation. This BNMF is connected to standard NMF with sparseness constraint. But, using the BNMF, the regularization parameters or hyperparameters are optimally estimated from training data without empirical selection from validation data. Beyond the approaches in [4, 15], the proposed BNMF completely considers the dependencies of the variational objective on hyperparameters and derives the analytical solution to singing-voice separation.

## 2. NONNEGATIVE MATRIX FACTORIZATION

Lee and Seung [10] proposed the standard NMF where no probabilistic distribution was assumed. Given a nonnegative data matrix  $\mathbf{X} \in \mathcal{R}_+^{M \times N}$ , NMF aims to decompose data matrix  $\mathbf{X}$  into a product of two nonnegative matrices  $\mathbf{B} \in \mathcal{R}_+^{M \times K}$  and  $\mathbf{W} \in \mathcal{R}_+^{K \times N}$ . The  $(m, n)$ -th entry of  $\mathbf{X}$  is approximated by  $X_{mn} \approx [\mathbf{B}\mathbf{W}]_{mn} = \sum_k B_{mk}W_{kn}$ . NMF parameters  $\Theta = \{\mathbf{B}, \mathbf{W}\}$  consist of basis matrix  $\mathbf{B}$  and weight matrix  $\mathbf{W}$ . The approximation based on NMF is optimized by minimizing the Kullback-Leibler (KL) divergence  $D_{\text{KL}}(\mathbf{X} \parallel \mathbf{B}\mathbf{W})$  between the observed data  $\mathbf{X}$  and

the approximated data  $\mathbf{B}\mathbf{W}$

$$\sum_{m,n} (X_{mn} \log \frac{X_{mn}}{[\mathbf{B}\mathbf{W}]_{mn}} + [\mathbf{B}\mathbf{W}]_{mn} - X_{mn}) \quad (1)$$

### 2.1 Maximum Likelihood Factorization

NMF approximation is revisited by introducing the probabilistic framework based on maximum likelihood (ML) theory. The nonnegative latent variable  $Z_{mkn}$  is embedded in data entry  $X_{mn}$  by  $X_{mn} = \sum_k Z_{mkn}$  and is represented by a Poisson distribution with mean  $B_{mk}W_{kn}$ , i.e.  $Z_{mkn} \sim \text{Pois}(Z_{mkn}; B_{mk}W_{kn})$  [4]. Log likelihood function of data matrix  $\mathbf{X}$  given parameters  $\Theta$  is expressed by

$$\begin{aligned} \log p(\mathbf{X}|\mathbf{B}, \mathbf{W}) &= \log \prod_{m,n} \text{Pois}(X_{mn}; \sum_k B_{mk}W_{kn}) \\ &= \sum_{m,n} (X_{mn} \log [\mathbf{B}\mathbf{W}]_{mn} - [\mathbf{B}\mathbf{W}]_{mn} - \log \Gamma(X_{mn} + 1)) \end{aligned} \quad (2)$$

where  $\Gamma(\cdot)$  is the gamma function. Maximizing the log likelihood function in Eq. (2) based on Poisson distribution is equivalent to minimizing the KL divergence between  $\mathbf{X}$  and  $\mathbf{B}\mathbf{W}$  in Eq. (1). This ML problem with missing variables  $\mathbf{Z} = \{Z_{mkn}\}$  can be solved according to EM algorithm. In E step, the expectation function of the log likelihood of data  $\mathbf{X}$  and latent variable  $\mathbf{Z}$  given new parameters  $\mathbf{B}^{(\tau+1)}$  and  $\mathbf{W}^{(\tau+1)}$  is calculated with respect to  $\mathbf{Z}$  under current parameters  $\mathbf{B}^{(\tau)}$  and  $\mathbf{W}^{(\tau)}$ . In M step, we maximize the resulting auxiliary function to obtain the updating of NMF parameters which is equivalent to that of standard NMF in [10].

### 2.2 Bayesian Factorization

ML estimation is prone to find an over-trained model [1]. To improve model regularization, Bayesian approach is introduced to establish NMF for single-source separation. ML NMF was improved by considering the priors of basis matrix  $\mathbf{B}$  and weight matrix  $\mathbf{W}$  for Bayesian NMF (BNMF). Different specifications of likelihood function and prior distribution result in different solutions with different inference procedures. In [15], the approximation error of  $X_{mn}$  using  $\sum_k B_{mk}W_{kn}$  is modeled by a zero-mean Gaussian distribution

$$X_{mn} \sim \mathcal{N}(X_{mn}; \sum_k B_{mk}W_{kn}, \sigma^2) \quad (3)$$

with the variance parameter  $\sigma^2$  which is distributed by an inverse gamma prior. The priors of nonnegative  $B_{mk}$  and  $W_{kn}$  are modeled by the exponential distributions

$$B_{mk} \sim \text{Exp}(B_{mk}; \lambda_{mk}^b), \quad W_{kn} \sim \text{Exp}(W_{kn}; \lambda_{kn}^w) \quad (4)$$

where  $\text{Exp}(x; \theta) = \theta \exp(-\theta x)$ , with means  $(\lambda_{mk}^b)^{-1}$  and  $(\lambda_{kn}^w)^{-1}$ , respectively. Typically, the larger the exponential hyperparameter  $\theta$  is involved, the sparser the exponential distribution is shaped. The sparsity of basis parameter  $B_{mk}$  and weight parameter  $W_{kn}$  is controlled by hyperparameters  $\lambda_{mk}^b$  and  $\lambda_{kn}^w$ , respectively. In [15], the hyperparameters  $\{\lambda_{mk}^b, \lambda_{kn}^w\}$  were fixed and empirically determined. The Gaussian likelihood does not adhere to

the assumption of nonnegative data matrix  $\mathbf{X}$ . The other weakness in the BNMF [15] is that the exponential distribution is not conjugate prior to the Gaussian likelihood function for NMF. There was no closed-form solution. The parameters  $\Theta = \{\mathbf{B}, \mathbf{W}, \sigma^2\}$  were accordingly estimated by Gibbs sampling procedure where a sequence of posterior samples of  $\Theta$  was drawn by the corresponding conditional posterior probabilities.

Cemgil [4] proposed the BNMF for image reconstruction based on the Poisson likelihood function as given in Eq. (2) and the gamma priors for basis and weight matrices. The gamma distribution, represented by a shape parameter and a scale parameter, is known as the conjugate prior to Poisson likelihood function. Variational Bayesian (VB) inference procedure was developed for NMF implementation. However, the shape parameter was implemented by the numerical solution. The computation cost was relatively high. Some dependencies of variational lower bound on model parameters were ignored in [4]. The resulting parameters did not reach true optimum of variational objective.

### 3. NEW BAYESIAN FACTORIZATION

This study aims to find an analytical solution to full Bayesian NMF by considering all dependencies of variational lower bound on regularization parameters. Regularization parameters are optimally estimated.

#### 3.1 Bayesian Objectives

In accordance with the Bayesian perspective and the spirit of standard NMF, we adopt the Poisson distribution as likelihood function and the exponential distribution as *conjugate prior* for NMF parameters  $B_{mk}$  and  $W_{kn}$  with hyperparameters  $\lambda_{mk}^b$  and  $\lambda_{kn}^w$ , respectively. Maximum *a posteriori* (MAP) estimates of parameters  $\Theta = \{\mathbf{B}, \mathbf{W}\}$  are obtained by maximizing the posterior distribution or minimizing  $-\log p(\mathbf{B}, \mathbf{W}|\mathbf{X})$  which is arranged as a regularized KL divergence between  $\mathbf{X}$  and  $\mathbf{B}\mathbf{W}$

$$D_{\text{KL}}(\mathbf{X}||\mathbf{B}\mathbf{W}) + \sum_{m,k} \lambda_{mk}^b B_{mk} + \sum_{k,n} \lambda_{kn}^w W_{kn} \quad (5)$$

where the terms independent of  $B_{mk}$  and  $W_{kn}$  are treated as constants. Notably, the regularization terms (2nd and 3rd terms) in this objective are nonnegative and seen as the  $\ell_1$  regularizers [18] which are controlled by hyperparameters  $\{\lambda_{mk}^b, \lambda_{kn}^w\}$ . These regularizers impose sparseness in the estimated MAP parameters.

However, MAP estimates are seen as point estimates. The randomness of parameters is not considered in model construction. To conduct full Bayesian treatment, BNMF is developed by maximizing the marginal likelihood  $p(\mathbf{X}|\Theta)$  over latent variables  $\mathbf{Z}$  as well as NMF parameters  $\{\mathbf{B}, \mathbf{W}\}$

$$\int \sum_{\mathbf{Z}} p(\mathbf{X}|\mathbf{Z}, \mathbf{B}, \mathbf{W}) p(\mathbf{Z}|\mathbf{B}, \mathbf{W}) p(\mathbf{B}, \mathbf{W}|\Theta) d\mathbf{B} d\mathbf{W} \quad (6)$$

and estimating the sparsity-controlled hyperparameters or regularization parameters  $\Theta = \{\lambda_{mk}^b, \lambda_{kn}^w\}$ . The resulting

evidence function is meaningful to act as an objective for model selection which balances the tradeoff between data fitness and model complexity [1]. In the singing-voice separation based on NMF, this objective is used to judge which number of bases  $K$  should be selected. The selected number is adaptive to fit different experimental conditions with varying lengths and the variations from different singers, genders, songs, genres, instruments and music accompaniments. Model regularization is tackled accordingly. But, using NMF without Bayesian treatment, the number of bases was fixed and empirically determined.

#### 3.2 Variational Bayesian Inference

The exact Bayesian solution to optimization problem in Eq. (6) does not exist because the posterior probability of three latent variables  $\{\mathbf{Z}, \mathbf{B}, \mathbf{W}\}$  given the observed mixtures  $\mathbf{X}$  could not be factorized. To deal with this issue, the variational Bayesian expectation-maximization (VB-EM) algorithm is developed to implement Poisson-Exponential BNMF. VB-EM algorithm applies the Jensen's inequality and maximizes the lower bound of the logarithm of marginal likelihood

$$\log p(\mathbf{X}|\Theta) \geq \int \sum_{\mathbf{Z}} q(\mathbf{Z}, \mathbf{B}, \mathbf{W}) \log \frac{p(\mathbf{X}, \mathbf{Z}, \mathbf{B}, \mathbf{W}|\Theta)}{q(\mathbf{Z}, \mathbf{B}, \mathbf{W})} \quad (7)$$

$$\times d\mathbf{B} d\mathbf{W} = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}, \mathbf{B}, \mathbf{W}|\Theta)] + H[q(\mathbf{Z}, \mathbf{B}, \mathbf{W})]$$

where  $H[\cdot]$  is an entropy function. The factorized variational distribution  $q(\mathbf{Z}, \mathbf{B}, \mathbf{W}) = q(\mathbf{Z})q(\mathbf{B})q(\mathbf{W})$  is assumed to approximate the true posterior distribution  $p(\mathbf{Z}, \mathbf{B}, \mathbf{W}|\mathbf{X}, \Theta)$ .

##### 3.2.1 VB-E Step

In VB-E step, a general solution to variational distribution  $q_j$  of an individual latent variable  $j \in \{\mathbf{Z}, \mathbf{B}, \mathbf{W}\}$  is obtained by [1]

$$\log \hat{q}_j \propto \mathbb{E}_{q(i \neq j)}[\log p(\mathbf{X}, \mathbf{Z}, \mathbf{B}, \mathbf{W}|\Theta)]. \quad (8)$$

Given the variational distributions defined by

$$\begin{aligned} q(B_{mk}) &= \text{Gam}(B_{mk}; \alpha_{mk}^b, \beta_{mk}^b) \\ q(W_{kn}) &= \text{Gam}(W_{kn}; \alpha_{kn}^w, \beta_{kn}^w) \\ q(Z_{mkn}) &= \text{Mult}(Z_{mkn}; P_{mkn}) \end{aligned} \quad (9)$$

the variational parameters  $\{\alpha_{mk}^b, \beta_{mk}^b, \alpha_{kn}^w, \beta_{kn}^w, P_{mkn}\}$  in three distributions are estimated by

$$\begin{aligned} \hat{\alpha}_{mk}^b &= 1 + \sum_n \langle Z_{mkn} \rangle, \quad \hat{\beta}_{mk}^b = \left( \sum_n \langle W_{kn} \rangle + \lambda_{mk}^b \right)^{-1} \\ \hat{\alpha}_{kn}^w &= 1 + \sum_m \langle Z_{mkn} \rangle, \quad \hat{\beta}_{kn}^w = \left( \sum_k \langle B_{mk} \rangle + \lambda_{kn}^w \right)^{-1} \\ \hat{P}_{mkn} &= \frac{\exp(\langle \log B_{mk} \rangle + \langle \log W_{kn} \rangle)}{\sum_j \exp(\langle \log B_{mj} \rangle + \langle \log W_{jn} \rangle)} \end{aligned} \quad (10)$$

where the expectation function  $\mathbb{E}_q[\cdot]$  is replaced by  $\langle \cdot \rangle$  for simplicity. By substituting the variational distribution into

Eq. (7), the variational lower bound is obtained by

$$\begin{aligned}
\mathcal{B}_L = & - \sum_{m,n,k} \langle B_{mk} \rangle \langle W_{kn} \rangle \\
& + \sum_{m,n} (-\log \Gamma(X_{mn} + 1) - \sum_k \langle Z_{mkn} \rangle \log \hat{P}_{mkn}) \\
& + \sum_{m,k} \langle \log B_{mk} \rangle \sum_n \langle Z_{mkn} \rangle + \sum_{k,n} \langle \log W_{kn} \rangle \sum_m \langle Z_{mkn} \rangle \\
& + \sum_{m,k} (\log \lambda_{mk}^b - \lambda_{mk}^b \langle B_{mk} \rangle) + \sum_{k,n} (\log \lambda_{kn}^w - \lambda_{kn}^w \langle W_{kn} \rangle) \\
& + \sum_{m,k} (-\langle \hat{\alpha}_{mk}^b - 1 \rangle \Psi(\hat{\alpha}_{mk}^b) + \log \hat{\beta}_{mk}^b + \hat{\alpha}_{mk}^b + \log \Gamma(\hat{\alpha}_{mk}^b)) \\
& + \sum_{k,n} (-\langle \hat{\alpha}_{kn}^w - 1 \rangle \Psi(\hat{\alpha}_{kn}^w) + \log \hat{\beta}_{kn}^w + \hat{\alpha}_{kn}^w + \log \Gamma(\hat{\alpha}_{kn}^w))
\end{aligned} \tag{11}$$

where  $\Psi(\cdot)$  is the derivative of the log gamma function, and is known as a digamma function.

### 3.2.2 VB-M Step

In VB-M step, the optimal regularization parameters  $\Theta = \{\lambda_{mk}^b, \lambda_{kn}^w\}$  are derived by maximizing Eq. (11) with respect to  $\Theta$  and yielding

$$\begin{aligned}
\frac{\partial \mathcal{B}_L}{\partial \lambda_{mk}^b} &= \frac{1}{\lambda_{mk}^b} - \langle B_{mk} \rangle + \frac{\partial \log \beta_{mk}^b}{\partial \lambda_{mk}^b} = 0 \\
\frac{\partial \mathcal{B}_L}{\partial \lambda_{kn}^w} &= \frac{1}{\lambda_{kn}^w} - \langle W_{kn} \rangle + \frac{\partial \log \beta_{kn}^w}{\partial \lambda_{kn}^w} = 0.
\end{aligned} \tag{12}$$

Accordingly, the solution to BNMF hyperparameters is derived by solving a quadratic equation where nonnegative constraint is considered to find positive values of hyperparameters by

$$\begin{aligned}
\hat{\lambda}_{mk}^b &= \frac{1}{2} \left( -\sum_n \langle W_{kn} \rangle + \sqrt{\left(\sum_n \langle W_{kn} \rangle\right)^2 + 4 \frac{\sum_n \langle W_{kn} \rangle}{\langle B_{mk} \rangle}} \right) \\
\hat{\lambda}_{kn}^w &= \frac{1}{2} \left( -\sum_m \langle B_{mk} \rangle + \sqrt{\left(\sum_m \langle B_{mk} \rangle\right)^2 + 4 \frac{\sum_m \langle B_{mk} \rangle}{\langle W_{kn} \rangle}} \right)
\end{aligned} \tag{13}$$

where  $\langle B_{mk} \rangle = \alpha_{mk}^b \beta_{mk}^b$  and  $\langle W_{kn} \rangle = \alpha_{kn}^w \beta_{kn}^w$  are obtained as the means of gamma distributions. VB-E step and VB-M step are alternatively and iteratively performed to estimate BNMF parameters  $\Theta$  with convergence. It is meaningful to select the best number of bases ( $K$ ) with the largest lower bound of the log marginal likelihood which integrates out the parameters of weight and basis matrices.

### 3.3 Poisson-Exponential Bayesian NMF

To the best of our knowledge, this is the first study where a Bayesian approach is developed for singing-voice separation. The uncertainties in singing-voice separation due to a variety of singers, songs and instruments could be compensated. Model selection problem is tackled as well. In this study, total number of basis vectors  $K$  is adaptively selected for individual mixed signal according to the variational lower bound in Eq. (11) with the converged variational parameters  $\{\hat{\alpha}_{mk}^b, \hat{\beta}_{mk}^b, \hat{\alpha}_{kn}^w, \hat{\beta}_{kn}^w, \hat{P}_{mkn}\}$  and model parameters  $\{\hat{\lambda}_{mk}^b, \hat{\lambda}_{kn}^w\}$ .

Considering the pairs of likelihood function and prior distribution in NMF, the proposed method is also called the Poisson-Exponential BNMF which is different from

the Gaussian-Exponential BNMF in [15] and the Poisson-Gamma BNMF in [4]. The superiorities of the proposed method to the BNMFs in [15, 4] are twofold. First, assuming the exponential priors provides a BNMF approach with tractable solution as given in Eq. (13). Gibbs sampling in [15] and Newton's solution in [4] are computationally expensive. Second, the dependencies of three terms of the variational lower bound in Eq. (11) on hyperparameters  $\lambda_{mk}^b$  or  $\lambda_{kn}^w$  are all considered in finding the true optimum while some dependencies were ignored in the solution to Poisson-Gamma BNMF [4]. Also, the observations in Gaussian-Exponential BNMF [15] were not constrained to be nonnegative.

## 4. EXPERIMENTS

### 4.1 Experimental Setup

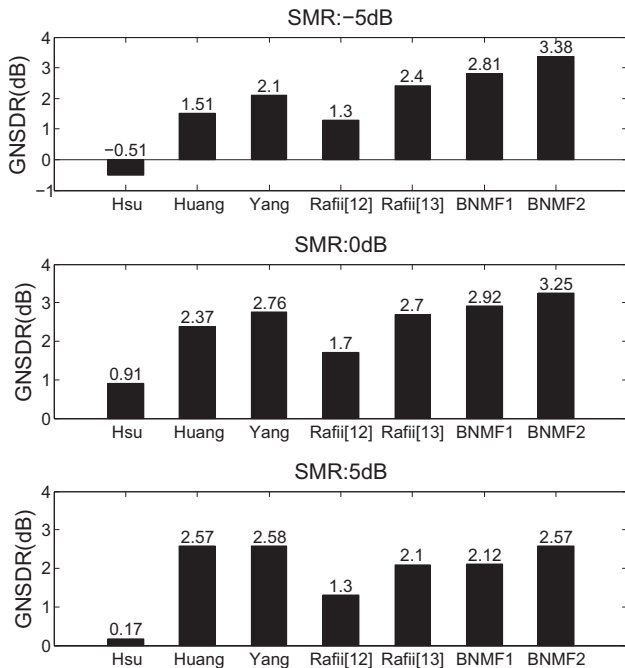
We used the MIR-1Kdataset [8] to evaluate the proposed method for unsupervised singing-voice separation from background music accompaniment. The dataset consisted of 1000 song clips extracted from 110 Chinese karaoke pop songs performed by 8 female and 11 male amateurs. Each clip recorded at 16 KHz sampling frequency with the duration ranging from 4 to 13 seconds. Since the music accompaniment and the singing voice were recorded at left and right channels, we followed [8, 9, 13] and simulated three different sets of monaural mixtures at signal-to-music-ratios (SMRs) of 5, 0, and -5 dB where the singing-voice was treated as signal and the accompaniment was treated as music. The separation problem was tackled in the short-time Fourier transform (STFT) domain. The 1024-point STFT was calculated to obtain the Fourier magnitude spectrograms with frame duration of 40 ms and frame shift of 10 ms. In the implementation of BNMF, ML-NMF was adopted as the initialization and 50 iterations were run to find the posterior means of basis and weight parameters. To evaluate the performance of singing-voice separation, we measure the signal-to-distortion ratio (SDR) [20] and then calculate the normalized SDR (NSDR) and the global NSDR (GNSDR) as

$$\begin{aligned}
\text{NSDR}(\hat{\mathbf{V}}, \mathbf{V}, \mathbf{X}) &= \text{SDR}(\hat{\mathbf{V}}, \mathbf{V}) - \text{SDR}(\mathbf{X}, \mathbf{V}) \\
\text{GNSDR}(\hat{\mathbf{V}}, \mathbf{V}, \mathbf{X}) &= \frac{\sum_{n=1}^{\tilde{N}} l_n \text{NSDR}(\hat{\mathbf{V}}_n, \mathbf{V}_n, \mathbf{X}_n)}{\sum_{n=1}^{\tilde{N}} l_n}
\end{aligned} \tag{14}$$

where  $\hat{\mathbf{V}}, \mathbf{V}, \mathbf{X}$  denote the estimated singing voice, the original clean singing voice, and the mixture signal, respectively,  $\tilde{N}$  is the total number of the clips and  $l_n$  is the length of the  $n$ th clip. NSDR is used to measure the improvement of SDR between the estimated singing voice  $\hat{\mathbf{V}}$  and the mixture signal  $\mathbf{X}$ . GNSDR is used to calculate the overall separation performance by taking the weighted mean of the NSDRs.

### 4.2 Unsupervised Singing-Voice Separation

We implemented the unsupervised singing-voice separation where total number of bases ( $K$ ) and the grouping of these bases into vocal source and music source were both

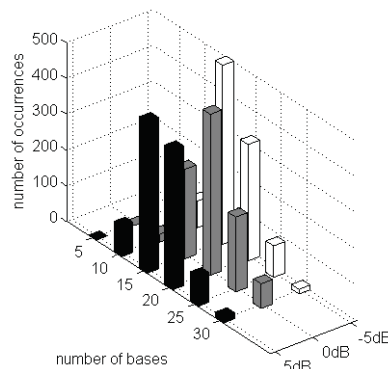


**Figure 1.** Performance comparison using BNMF1 (K-means clustering) and BNMF2 (NMF-clustering) and five competitive methods (Hsu [8], Huang [9], Yang [22], Rafii [12], Rafii [13]) in terms of GNSDR under various SMRs.

learned from test data in an unsupervised way. No training data were required. Model complexity based on  $K$  was determined in accordance with the variational lower bound of log marginal likelihood in Eq. (11) while the grouping of bases for two sources was simply performed via the clustering algorithms using the estimated basis vectors in  $\mathbf{B}$  or equivalently from the estimated variational parameters  $\{\alpha_{mk}^b, \beta_{mk}^b\}$ . Following [17], we conducted the K-means clustering algorithm based on the basis vectors  $\mathbf{B}$  in Mel-frequency cepstral coefficient (MFCC) domain. Each basis vector was first transformed to the Mel-scaled spectrum by applying 20 overlapping triangle filters spaced on the Mel scale. Then, we took the logarithm and applied the discrete cosine transform to obtain nine MFCCs. Finally, we normalized each coefficient to zero mean and unit variance. The K-means clustering algorithm was applied to partition the feature set into two clusters through an iterative procedure until convergence. However, it is more meaningful to conduct NMF-based clustering for the proposed BNMF method. To do so, we transformed the basis vectors  $\mathbf{B}$  into Mel-scaled spectrum to form the Mel-scaled basis matrix. ML-NMF was applied to factorize this Mel-scaled basis matrix into two matrices  $\tilde{\mathbf{B}}$  of size  $N$ -by-2 and  $\tilde{\mathbf{W}}$  of size 2-by- $K$ . The soft mask scheme based on Wiener gain was applied to smooth the separation of  $\mathbf{B}$  into basis vectors for vocal signal and music signal. This same soft mask was performed for the separation of mixed signal  $X$  into vocal signal and music signal based on the K-means clustering and NMF clustering. Finally, the separated singing voice and music accompaniment signals were obtained by the overlap-and-add method using the original phase.

	NMF (30)	NMF (40)	NMF (50)	BNMF (adaptive)
K-means clustering	2.69	2.58	2.47	2.92
NMF clustering	3.15	3.13	2.97	3.25

**Table 1.** Comparison of GNSDR at SMR = 0 dB using NMF with fixed number of bases  $\{30, 40, 50\}$  and BNMF with adaptive number of bases.



**Figure 2.** Histogram of the selected number of bases using BNMF under various SMRs.

### 4.3 Experimental Results

The unsupervised single-channel separation using BNMFs (BNMF1 using K-means clustering and BNMF2 using NMF clustering) and the other five competitive systems (Hsu [8], Huang [9], Yang [22], Rafii [12], Rafii [13]) is compared in terms of GNSDR as depicted in Figure 1. Using K-means clustering in MFCC domain, the resulting BNMF1 outperforms the other five methods under SMRs of 0 dB and -5 dB while the results using Huang [9] and Yang [22] perform better than BNMF1 under 5 dB condition. This is because the methods in [9, 22] used additional pre- and/or post-processing techniques as provided in [13, 22] which were not applied in BNMF1 and BNMF2. Nevertheless, using BNMF factorization with NMF clustering (BNMF2), the overall evaluation consistently achieves around 0.33~0.57 dB relative improvement in GNSDR compared with BNMF1 including the SMR condition at 5dB. In addition, we evaluate the effect on the adaptive basis selection using BNMF. Table 1 reports the comparison of BNMF1 and BNMF2 with adaptive basis selection and ML-NMF with fixed number of bases under SMR of 0 dB. Two clustering methods were also carried out for NMF with different  $K$ . BNMF factorization combined with NMF clustering achieves the best performance in this comparison. Figure 2 shows the histogram of the selected number of bases  $K$  using BNMF. It is obvious that this adaptive basis selection plays an important role to find suitable amount of bases to fit different experimental conditions.

## 5. CONCLUSIONS

We proposed a new unsupervised Bayesian nonnegative matrix factorization approach to extract the singing voice from background music accompaniment and illustrated the novelty on an analytical and true optimum solution to the Poisson-Exponential BNMF. Through the VB-EM inference procedure, the proposed method automatically selected different number of bases to fit various experimental conditions. We conducted two clustering algorithms to find the grouping of bases into vocal and music sources. Experimental results showed the consistent improvement of using BNMF factorization with NMF clustering over the other singing-voice separation methods in terms of GNSDR. In future works, the proposed BNMF shall be extended to multi-layer source separation and applied to detect unknown number of sources.

## 6. REFERENCES

- [1] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Science, 2006.
- [2] N. Boulanger-Lewandowski, G. J. Mysore, and M. Hoffman. Exploiting long-term temporal dependencies in NMF using recurrent neural networks with application to source separation. In *Proc. of ICASSP*, pages 337–344, 2014.
- [3] A. S. Bregman. *Auditory Scene Analysis: the Perceptual Organization of Sound*. MIT Press, 1990.
- [4] A. T. Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, (Article ID 785152), 2009.
- [5] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society (B)*, 39(1):1–38, 1977.
- [6] H. Fujihara, M. Goto, J. Ogata, and H. G. Okuno. Lyricsynchronizer: automatic synchronization system between musical audio signals and lyrics. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1252–1261, 2011.
- [7] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.
- [8] C.-L. Hsu and J.-S. R. Jang. On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset. *IEEE Transactions on Audio, Speech, Language Processing*, 18(2):310–319, 2010.
- [9] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *Proc. of ICASSP*, pages 57–60, 2012.
- [10] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, pages 556–562, 2000.
- [11] A. Mesaros, T. Virtanen, and A. Klapuri. Singer identification in polyphonic music using vocal separation and pattern recognition methods. In *Proc. of Annual Conference of International Society for Music Information Retrieval*, pages 375–378, 2007.
- [12] Z. Rafii and B. Pardo. A simple music/voice separation method based on the extraction of the repeating musical structure. In *Proc. of ICASSP*, pages 221–224, 2011.
- [13] Z. Rafii and B. Pardo. Repeating pattern extraction technique (REPET): A simple method for music/voice separation. *IEEE Transactions on Audio, Speech, Language Processing*, 21(1):73–84, Jan. 2013.
- [14] M. N. Schmidt and M. Morup. Non-negative matrix factor 2-D deconvolution for blind single channel source separation. In *Proc. of ICA*, pages 700–707, 2006.
- [15] M. N. Schmidt, O. Winther, and L. K. Hansen. Bayesian non-negative matrix factorization. In *Proc. of ICA*, pages 540–547, 2009.
- [16] P. Smaragdis. Convolutional speech bases and their application to speech separation. *IEEE Transactions on Audio, Speech, Language Processing*, 15(1):1–12, 2007.
- [17] M. Spiertz and V. Gnann. Source-Filter based clustering for monaural blind source separation. In *Proc. of International Conference on Digital Audio Effects*, pages 1–4, 2009.
- [18] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.
- [19] S. Vembu and S. Baumann. Separation of vocals from polyphonic audio recordings. In *Proc. of ISMIR*, pages 375–378, 2005.
- [20] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *IEEE Transaction on Audio, Speech and Language Processing*, 14(4):1462–1469, 2006.
- [21] D. Yang and W. Lee. Disambiguating music emotion using software agents. In *Proc. of ISMIR*, pages 52–57, 2004.
- [22] Y.-H. Yang. On sparse and low-rank matrix decomposition for singing voice separation. In *Proc. of ACM International Conference on Multimedia*, pages 757–760, 2012.
- [23] B. Zhu, W. Li, R. Li, and X. Xue. Multi-stage non-negative matrix factorization for monaural singing voice separation. *IEEE Transactions on Audio, Speech, Language Processing*, 21(10):2096–2107, 2013.



# PROBABILISTIC EXTRACTION OF BEAT POSITIONS FROM A BEAT ACTIVATION FUNCTION

Filip Korzeniowski, Sebastian Böck, and Gerhard Widmer

Department of Computational Perception  
Johannes Kepler University, Linz, Austria

filip.korzeniowski@jku.at

## ABSTRACT

We present a probabilistic way to extract beat positions from the output (activations) of the neural network that is at the heart of an existing beat tracker. The method can serve as a replacement for the greedy search the beat tracker currently uses for this purpose. Our experiments show improvement upon the current method for a variety of data sets and quality measures, as well as better results compared to other state-of-the-art algorithms.

## 1. INTRODUCTION

Rhythm and pulse lay the foundation of the vast majority of musical works. Percussive instruments like rattles, stampers and slit drums have been used for thousands of years to accompany and enhance rhythmic movements or dances. Maybe this deep connection between movement and sound enables humans to easily tap to the pulse of a musical piece, accenting its beats. The computer, however, has difficulties determining the position of the beats in an audio stream, lacking the intuition humans developed over thousands of years.

Beat tracking is the task of locating beats within an audio stream of music. Literature on beat tracking suggests many possible applications: practical ones such as automatic time-stretching or correction of recorded audio, but also as a support for further music analysis like segmentation or pattern discovery [4]. Several musical aspects hinder tracking beats reliably: syncopation, triplets and off-beat rhythms create rhythmical ambiguousness that is difficult to resolve; varying tempo increases musical expressivity, but impedes finding the correct beat times. The multitude of existing beat tracking algorithms work reasonably well for a subset of musical works, but often fail for pieces that are difficult to handle, as [11] showed.

In this paper, we further improve upon the beat tracker presented in [2]. The existing algorithm uses a neural network to detect beats in the audio. The output of this neural network, called activations, indicates the likelihood of a

beat at each audio position. A post-processing step selects from these activations positions to be reported as beats. However, this method struggles to find the correct beats when confronted with ambiguous activations.

We contribute a new, probabilistic method for this purpose. Although we designed the method for audio with a steady pulse, we show that using the proposed method the beat tracker achieves better results even for datasets containing music with varying tempo.

The remainder of the paper is organised as follows: Section 2 reviews the beat tracker our method is based on. In Section 3 we present our approach, describe the structure of our model and show how we infer beat positions. Section 4 describes the setup of our experiments, while we show their results in Section 5. Finally, we conclude our work in Section 6.

## 2. BASE METHOD

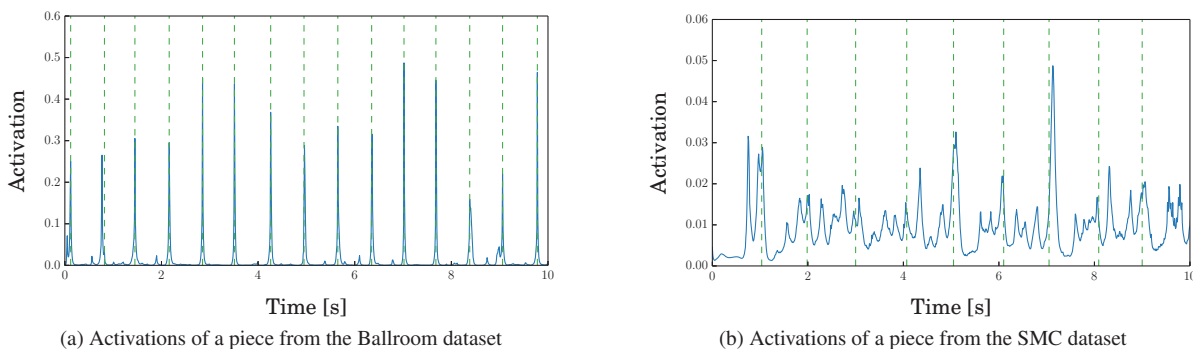
In this section, we will briefly review the approach presented in [2]. For a detailed discourse we refer the reader to the respective publication. First, we will outline how the algorithm processes the signal to emphasise onsets. We will then focus on the neural network used in the beat tracker and its output in Section 2.2. After this, Section 3 will introduce the probabilistic method we propose to find beats in the output activations of the neural network.

### 2.1 Signal Processing

The algorithm derives from the signal three logarithmically filtered power spectrograms with window sizes  $W$  of 1024, 2048 and 4096 samples each. The windows are placed 441 samples apart, which results in a frame rate of  $f_r = 100$  frames per second for audio sampled at 44.1kHz. We transform the spectra using a logarithmic function to better match the human perception of loudness, and filter them using 3 overlapping triangular filters per octave.

Additionally, we compute the first order difference for each of the spectra in order to emphasise onsets. Since longer frame windows tend to smear spectral magnitude values in time, we compute the difference to the last, second to last, and third to last frame, depending on the window size  $W$ . Finally, we discard all negative values.





**Figure 1.** Activations of pieces from two different datasets. The activations are shown in blue, with green, dotted lines showing the ground truth beat annotations. On the left, distinct peaks indicate the presence of beats. The prominent rhythmical structure of ballroom music enables the neural network to easily discern frames that contain beats from those that do not. On the right, many peaks in the activations do not correspond to beats, while some beats lack distinguished peaks in the activations. In this piece, a single woodwind instrument is playing a solo melody. Its soft onsets and lack of percussive instruments make detecting beats difficult.

## 2.2 Neural Network

Our classifier consists of a bidirectional recurrent neural network of Long Short-Term Memory (LSTM) units, called *bidirectional Long Short-Term Memory* (BLSTM) recurrent neural network [10]. The input units are fed with the log-filtered power spectra and their corresponding positive first order differences. We use three fully connected hidden layers of 25 LSTM units each. The output layer consists of a single sigmoid neuron. Its value remains within  $[0, 1]$ , with higher values indicating the presence of a beat at the given frame.

After we initialise the network weights randomly, the training process adapts them using standard gradient descent with back propagation and early stopping. We obtain training data using 8-fold cross validation, and randomly choose 15% of the training data to create a validation set. If the learning process does not improve classification on this validation set for 20 training epochs, we stop it and choose the best performing neural network as final model. For more details on the network and the learning process, we refer the reader to [2].

The neural network’s output layer yields activations for every feature frame of an audio signal. We will formally represent this computation as mathematical function. Let  $N$  be the number of feature frames for a piece, and  $\mathbb{N}_{\leq N} = \{1, 2, \dots, N\}$  the set of all frame indices. Furthermore, let  $v_n$  be the feature vector (the log-filtered power spectra and corresponding differences) of the  $n^{\text{th}}$  audio frame, and  $\Upsilon = (v_1, v_2, \dots, v_N)$  denote all feature vectors computed for a piece. We represent the neural network as a function

$$\Psi : \mathbb{N}_{\leq N} \rightarrow [0, 1], \quad (1)$$

such that  $\Psi(n; \Upsilon)$  is the activation value for the  $n^{\text{th}}$  frame when the network processes the feature vectors  $\Upsilon$ . We will call this function “*activations*” in the following.

Depending on the type of music the audio contains, the activations show clear (or, less clear) peaks at beat positions. Figure 1 depicts the first 10 seconds of activations

for two different songs, together with ground truth beat annotations. In Fig. 1a, the peaks in the activations clearly correspond to beats. For such simple cases, thresholding should suffice to extract beat positions. However, we often have to deal with activations as those in Fig. 1b, with many spurious and/or missing peaks. In the following section, we will propose a new method for extracting beat positions from such activations.

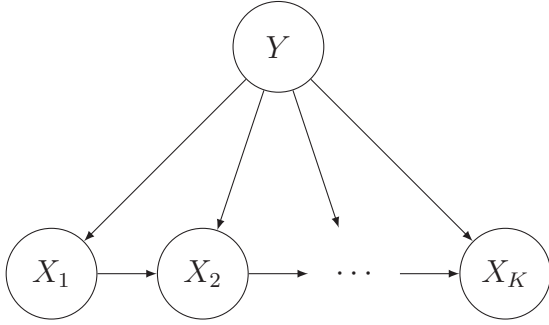
## 3. PROBABILISTIC EXTRACTION OF BEAT POSITIONS

Figure 1b shows the difficulty in deriving the position of beats from the output of the neural network. A greedy local search, as used in the original system, runs into problems when facing ambiguous activations. It struggles to correct previous beat position estimates even if the ambiguity resolves later in the piece. We therefore tackle this problem using a probabilistic model that allows us to globally optimise the beat sequence.

Probabilistic models are a frequently used to process time-series data, and are therefore popular in beat tracking (e.g. [3, 9, 12, 13, 14]). Most systems favour generative time-series models like hidden Markov models (HMMs), Kalman filters, or particle filters as natural choices for this problem. For a more complete overview of available beat trackers using various methodologies and their results on a challenging dataset we refer the reader to [11].

In this paper, we use a different approach: our model represents each beat with its own random variable. We model time as dimension in the sample space of our random variables as opposed to a concept of time driving a random process in discrete steps. Therefore, all activations are available at any time, instead of one at a time when thinking of time-series data.

For each musical piece we create a model that differs from those of other pieces. Different pieces have different lengths, so the random variables are defined over different sample spaces. Each piece contains a different number of beats, which is why each model consists of a different



**Figure 2.** The model depicted as Bayesian network. Each  $X_k$  corresponds to a beat and models its position.  $Y$  represents the feature vectors of a signal.

number of random variables.

The idea to model beat positions directly as random variables is similar to the HMM-based method presented in [14]. However, we formulate our model as a Bayesian network with the observations as topmost node. This allows us to directly utilise the whole observation sequence for each beat variable, without potentially violating assumptions that need to hold for HMMs (especially those regarding the observation sequence). Also, our model uses only a single factor to determine potential beat positions in the audio – the output of a neural network – whereas [14] utilises multiple features on different levels to detect beats and downbeats.

### 3.1 Model Structure

As mentioned earlier, we create individual models for each piece, following the common structure described in this section. Figure 2 gives an overview of our system, depicted as Bayesian network.

Each  $X_k$  is a random variable modelling the position of the  $k^{\text{th}}$  beat. Its domain are all positions within the length of a piece. By position we mean the frame index of the activation function – since we extract features with a frame rate of  $f_r = 100$  frames per second, we discretise the continuous time space to 100 positions per second.

Formally, the number of possible positions per piece is determined by  $N$ , the number of frames. Each  $X_k$  is then defined as random variable with domain  $\mathbb{N}_{\leq N}$ , the natural numbers smaller or equal to  $N$ :

$$X_k \in \mathbb{N}_{\leq N} \quad \text{with } 1 \leq k \leq K, \quad (2)$$

where  $K$  is the number of beats in the piece. We estimate this quantity by detecting the dominant interval  $\tau$  of a piece using an autocorrelation-based method on the smoothed activation function of the neural network (see [2] for details). Here, we restrict the possible intervals to a range  $[\tau_l, \tau_u]$ , with both bounds learned from data. Assuming a steady tempo and a continuous beat throughout the piece, we simply compute  $K = N/\tau$ .

$Y$  models the features extracted from the input audio. If we divide the signal into  $N$  frames,  $Y$  is a sequence of vectors:

$$Y \in \{(y_1, \dots, y_N)\}, \quad (3)$$

where each  $y_n$  is in the domain defined by the input features. Although  $Y$  is formally a random variable with a distribution  $P(Y)$ , its value is always given by the concrete features extracted from the audio.

The model’s structure requires us to define dependencies between the variables as conditional probabilities. Assuming these dependencies are the same for each beat but the first, we need to define

$$P(X_1 | Y) \quad \text{and} \\ P(X_k | X_{k-1}, Y).$$

If we wanted to compute the joint probability of the model, we would also need to define  $P(Y)$  – an impossible task. Since, as we will elaborate later, we are only interested in  $P(X_{1:K} | Y)$ <sup>1</sup>, and  $Y$  is always given, we can leave this aside.

### 3.2 Probability Functions

Except for  $X_1$ , two random variables influence each  $X_k$ : the previous beat  $X_{k-1}$  and the features  $Y$ . Intuitively, the former specifies the spacing between beats and thus the rough position of the beat compared to the previous one. The latter indicates to what extent the features confirm the presence of a beat at this position. We will define both as individual factors that together determine the conditional probabilities.

#### 3.2.1 Beat Spacing

The pulse of a musical piece spaces its beats evenly in time. Here, we assume a steady pulse throughout the piece and model the relationship between beats as factor favouring their regular placement according to this pulse. Future work will relax this assumption and allow for varying pulses.

Even when governed by a steady pulse, the position of beats is far from rigid: slight modulations in tempo add musical expressivity and are mostly artistic elements intended by performers. We therefore allow a certain deviation from the pulse. As [3] suggests, tempo changes are perceived relatively rather than absolutely, i.e. halving the tempo should be equally probable as doubling it. Hence, we use the logarithm to base 2 to define the intermediate factor  $\tilde{\Phi}$  and factor  $\Phi$ , our beat spacing model. Let  $x$  and  $x'$  be consecutive beat positions and  $x > x'$ , we define

$$\tilde{\Phi}(x, x') = \phi(\log_2(x - x'); \log_2(\tau), \sigma_\tau^2), \quad (4)$$

$$\Phi(x, x') = \begin{cases} \tilde{\Phi}(x, x') & \text{if } 0 < x - x' < 2\tau \\ 0 & \text{else} \end{cases}, \quad (5)$$

where  $\phi(x; \mu, \sigma^2)$  is the probability density function of a Gaussian distribution with mean  $\mu$  and variance  $\sigma^2$ ,  $\tau$  is the dominant inter-beat interval of the piece, and  $\sigma_\tau^2$  represents the allowed tempo variance. Note how we restrict the non-zero range of  $\Phi$ : on one hand, to prevent computing the logarithm of negative values, and on the other hand, to reduce the number of computations.

<sup>1</sup> We use  $X_{m:n}$  to denote all  $X_k$  with indices  $m$  to  $n$

The factor yields high values when  $x$  and  $x'$  are spaced approximately  $\tau$  apart. It thus favours beat positions that correspond to the detected dominant interval, allowing for minor variations.

Having defined the beat spacing factor, we will now elaborate on the activation vector that connects the model to the audio signal.

### 3.2.2 Beat Activations

The neural network's activations  $\Psi$  indicate how likely<sup>2</sup> each frame  $n \in N_{\leq N}$  is a beat position. We directly use this factor in the definition of the conditional probability distributions.

With both factors in place we can continue to define the conditional probability distributions that complete our probabilistic model.

### 3.2.3 Conditional Probabilities

The conditional probability distribution  $P(X_k | X_{k-1}, Y)$  combines both factors presented in the previous sections. It follows the intuition we outlined at the beginning of Section 3.2 and molds it into the formal framework as

$$P(X_k | X_{k-1}, Y) = \frac{\Psi(X_k; Y) \cdot \Phi(X_k, X_{k-1})}{\sum_{X_k} \Psi(X_k; Y) \cdot \Phi(X_k, X_{k-1})}. \quad (6)$$

The case of  $X_1$ , the first beat, is slightly different. There is no previous beat to determine its rough position using the beat spacing factor. But, since we assume that there is a steady and continuous pulse throughout the audio, we can conclude that its position lies within the first interval from the beginning of the audio. This corresponds to a uniform distribution in the range  $[0, \tau]$ , which we define as beat position factor for the first beat as

$$\Phi_1(x) = \begin{cases} 1/\tau & \text{if } 0 \leq x < \tau, \\ 0 & \text{else} \end{cases}. \quad (7)$$

The conditional probability for  $X_1$  is then

$$P(X_1 | Y) = \frac{\Psi(X_1; Y) \cdot \Phi_1(X_1)}{\sum_{X_1} \Psi(X_1; Y) \cdot \Phi_1(X_1)}. \quad (8)$$

The conditional probability functions fully define our probabilistic model. In the following section, we show how we can use this model to infer the position of beats present in a piece of music.

## 3.3 Inference

We want to infer values  $x_{1:K}^*$  for  $X_{1:K}$  that maximise the probability of the beat sequence given  $Y = \Upsilon$ , that is

$$x_{1:K}^* = \operatorname{argmax}_{x_{1:K}} P(X_{1:K} | \Upsilon). \quad (9)$$

Each  $x_k^*$  corresponds to the position of the  $k^{\text{th}}$  beat.  $\Upsilon$  are the feature vectors computed for a specific piece. We use

<sup>2</sup> technically, it is not a likelihood in the probabilistic sense – it just yields higher values if the network thinks that the frame contains a beat than if not

a dynamic programming method similar to the well known Viterbi algorithm [15] to obtain the values of interest.

We adapt the standard Viterbi algorithm to fit the structure of model by changing the definition of the ‘‘Viterbi variables’’  $\delta$  to

$$\begin{aligned} \delta_1(x) &= P(X_1 = x | \Upsilon) \quad \text{and} \\ \delta_k(x) &= \max_{x'} P(X_k = x | X_{k-1} = x', \Upsilon) \cdot \delta_{k-1}(x'), \end{aligned}$$

where  $x, x' \in \mathbb{N}_{\leq N}$ . The backtracking pointers are set accordingly.

$P(x_{1:K}^* | \Upsilon)$  gives us the probability of the beat sequence given the data. We use this to determine how well the deduced beat structure fits the features and in consequence the activations. However, we cannot directly compare the probabilities of beat sequences with different numbers of beats: the more random variables a model has, the smaller the probability of a *particular* value configuration, since there are more *possible* configurations. We thus normalise the probability by dividing by  $K$ , the number of beats.

With this in mind, we try different values for the dominant interval  $\tau$  to obtain multiple beat sequences, and choose the one with the highest normalised probability. Specifically, we run our method with multiples of  $\tau$  ( $1/2$ ,  $2/3$ ,  $1$ ,  $3/2$ ,  $2$ ) to compensate for errors when detecting the dominant interval.

## 4. EXPERIMENTS

In this section we will describe the setup of our experiments: which data we trained and tested the system on, and which evaluation metrics we chose to quantify how well our beat tracker performs.

### 4.1 Data

We ensure the comparability of our method by using three freely available data sets for beat tracking: the *Ballroom* dataset [8, 13]; the *Hainsworth* dataset [9]; the *SMC* dataset [11]. The order of this listing indicates the difficulty associated with each of the datasets. The Ballroom dataset consists of dance music with strong and steady rhythmic patterns. The Hainsworth dataset includes of a variety of musical genres, some considered easier to track (like pop/rock, dance), others more difficult (classical, jazz). The pieces in the SMC dataset were specifically selected to challenge existing beat tracking algorithms.

We evaluate our beat tracker using 8-fold cross validation, and balance the splits according to dataset. This means that each split consists of roughly the same relative number of pieces from each dataset. This way we ensure that all training and test splits represent the same distribution of data.

All training and testing phases use the same splits. The same training sets are used to learn the neural network and to set parameters of the probabilistic model (lower and upper bounds  $\tau_l$  and  $\tau_u$  for dominant interval estimation and  $\sigma_\tau$ ). The test phase feeds the resulting tracker with data from the corresponding test split. After detecting the beats

for all pieces, we group the results according to the original datasets in order to present comparable results.

## 4.2 Evaluation Metrics

A multitude of evaluation metrics exist for beat tracking algorithms. Some accent different aspects of a beat tracker's performance, some capture similar properties. For a comprehensive review and a detailed elaboration on each of the metrics, we refer the reader to [5]. Here, we restrict ourselves to the following four quantities, but will publish further results on our website<sup>3</sup>.

**F-measure** The standard measure often used in information retrieval tasks. Beats count as correct if detected within  $\pm 70$ ms of the annotation.

**Cemgil** Measure that uses a Gaussian error window with  $\sigma = 40$ ms instead of a binary decision based on a tolerance window. It also incorporates false positives and false negatives.

**CMLt** The percentage of correctly detected beats at the correct metrical level. The tolerance window is set to 17.5% of the current inter-beat interval.

**AMLt** Similar to CMLt, but allows for different metrical levels like double tempo, half tempo, and off-beat.

In contrast to common practice<sup>4</sup>, we do not skip the first 5 seconds of each audio signal for evaluation. Although skipping might make sense for on-line algorithms, it does not for off-line beat trackers.

## 5. RESULTS

Table 1 shows the results of our experiments. We obtained the raw beat detections on the Ballroom dataset for [6, 12, 13] from the authors of [13] and evaluated them using our framework. The results are thus directly comparable to those of our method. For the Hainsworth dataset, we collected results for [6, 7, 12] from [7], who does skip the first 5 seconds of each piece in the evaluation. In our experience, this increases the numbers obtained for each metric by about 0.01.

The approaches of [6, 7] do not require any training. In [12], some parameters are set up based on a separate dataset consisting of pieces from a variety of genres. [13] is a system that is specialised for and thus only trained on the Ballroom dataset.

We did not include results of other algorithms for the SMC dataset, although available in [11]. This dataset did not exist at the time most beat trackers were crafted, so the authors could not train or adapt their algorithms in order to cope with such difficult data.

Our method improves upon the original algorithm [1, 2] for each of the datasets and for all evaluation metrics. While F-Measure and Cemgil metric rises only marginally (except for the SMC dataset), CMLt and AMLt improves

SMC	F	Cg	CMLt	AMLt
Proposed	0.545	0.436	0.442	0.580
Böck [1, 2]	0.497	0.402	0.360	0.431
Hainsworth	F	Cg	CMLt	AMLt
Proposed	0.840	0.718	0.784	0.875
Böck [1, 2]	0.837	0.717	0.763	0.811
Degara* [7]	-	-	0.629	0.815
Klapuri* [12]	-	-	0.620	0.793
Davies* [6]	-	-	0.609	0.763
Ballroom	F	Cg	CMLt	AMLt
Proposed	0.903	0.864	0.833	0.910
Böck [1, 2]	0.889	0.857	0.796	0.831
Krebs [13]	0.855	0.772	0.786	0.865
Klapuri [12]	0.728	0.651	0.539	0.817
Davies [6]	0.764	0.696	0.574	0.864

**Table 1.** Beat tracking results for the three datasets. **F** stands for F-measure and **Cg** for the Cemgil metric. Results marked with a star skip the first five seconds of each piece and are thus better by about 0.01 for each metric, in our experience.

considerably. Our beat tracker also performs better than the other algorithms, where metrics were available.

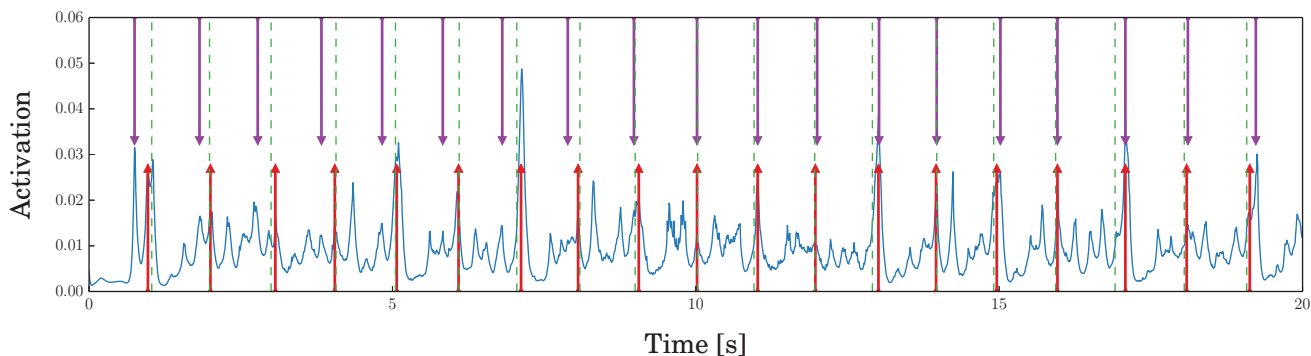
The proposed model assumes a stable tempo throughout a piece. This assumption holds for certain kinds of music (like most of pop, rock and dance), but does not for others (like jazz or classical). We estimated the variability of the tempo of a piece using the standard deviation of the local beat tempo. We computed the local beat tempo based on the inter-beat interval derived from the ground truth annotations. The results indicate that most pieces have a steady pulse: 90% show a standard deviation lower than 8.61 bpm. This, of course, depends on the dataset, with 97% of the ballroom pieces having a deviation below 8.61 bpm, 89% of the Hainsworth dataset but only 67.7% of the SMC data.

We expect our approach to yield inferior results for pieces with higher tempo variability than for those with a more constant pulse. To test this, we computed Pearson's correlation coefficient between tempo variability and AMLt value. The obtained value of  $\rho = -0.46$  indicates that our expectation holds, although the relationship is not linear, as a detailed examination showed. Obviously, multiple other factors also influence the results. Note, however, that although the tempo of pieces from the SMC dataset varies most, it is this dataset where we observed the strongest improvement compared to the original approach.

Figure 3 compares the beat detections obtained with the proposed method to those computed by the original approach. It exemplifies the advantage of a globally optimised beat sequence compared to a greedy local search.

<sup>3</sup> <http://www.cp.jku.at/people/korzeniowski/ismir2014>

<sup>4</sup> As implemented in the MatLab toolbox for the evaluation of beat trackers presented in [5]



**Figure 3.** Beat detections for the same piece as shown in Fig. 1b obtained using the proposed method (red, up arrows) compared to those computed by the original approach (purple, down arrows). The activation function is plotted solid blue, ground truth annotations are represented by vertical dashed green lines. Note how the original method is not able to correctly align the first 10 seconds, although it does so for the remaining piece. Globally optimising the beat sequence via back-tracking allows us to infer the correct beat times, even if the peaks in the activation function are ambiguous at the beginning.

## 6. CONCLUSION AND FUTURE WORK

We proposed a probabilistic method to extract beat positions from the activations of a neural network trained for beat tracking. Our method improves upon the simple approach used in the original algorithm for this purpose, as our experiments showed.

In this work we assumed close to constant tempo throughout a piece of music. This assumption holds for most of the available data. Our method also performs reasonably well on difficult datasets containing tempo changes, such as the SMC dataset. Nevertheless we believe that extending the presented method in a way that enables tracking pieces with varying tempo will further improve the system's performance.

## ACKNOWLEDGEMENTS

This work is supported by the European Union Seventh Framework Programme FP7 / 2007-2013 through the GiantSteps project (grant agreement no. 610591).

## 7. REFERENCES

- [1] MIREX 2013 beat tracking results. [http://nema.lis.illinois.edu/nema\\_out/mirex2013/results/abt/](http://nema.lis.illinois.edu/nema_out/mirex2013/results/abt/), 2013.
- [2] S. Böck and M. Schedl. Enhanced Beat Tracking With Context-Aware Neural Networks. In *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, Paris, France, Sept. 2011.
- [3] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing. On tempo tracking: Tempogram Representation and Kalman filtering. *Journal of New Music Research*, 28:4:259–273, 2001.
- [4] T. Collins, S. Böck, F. Krebs, and G. Widmer. Bridging the Audio-Symbolic Gap: The Discovery of Repeated Note Content Directly from Polyphonic Music Audio. In *Proceedings of the Audio Engineering Society's 53rd Conference on Semantic Audio*, London, 2014.
- [5] M. E. P. Davies, N. Degara, and M. D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. *Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06*, 2009.
- [6] M. E. P. Davies and M. D. Plumbley. Context-Dependent Beat Tracking of Musical Audio. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1009–1020, Mar. 2007.
- [7] N. Degara, E. A. Rua, A. Pena, S. Torres-Guijarro, M. E. P. Davies, and M. D. Plumbley. Reliability-Informed Beat Tracking of Musical Signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):290–301, Jan. 2012.
- [8] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, 2006.
- [9] S. W. Hainsworth and M. D. Macleod. Particle Filtering Applied to Musical Tempo Tracking. *EURASIP Journal on Advances in Signal Processing*, 2004(15):2385–2395, Nov. 2004.
- [10] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computing*, 9(8):1735–1780, Nov. 1997.
- [11] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. a. L. Oliveira, and F. Gouyon. Selective Sampling for Beat Tracking Evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2539–2548, Nov. 2012.
- [12] A. P. Klapuri, A. J. Eronen, and J. T. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, 2006.
- [13] F. Krebs, S. Böck, and G. Widmer. Rhythmic Pattern Modeling for Beat and Downbeat Tracking in Musical Audio. In *Proc. of the 14th International Conference on Music Information Retrieval (ISMIR)*, 2013.
- [14] G. Peeters and H. Papadopoulos. Simultaneous beat and downbeat-tracking using a probabilistic framework: theory and large-scale evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1754–1769, 2011.
- [15] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

# GEOGRAPHICAL REGION MAPPING SCHEME BASED ON MUSICAL PREFERENCES

**Sanghoon Jun**  
Korea University  
ysbhjun@korea.ac.kr

**Seungmin Rho**  
Sungkyul University  
smrho@sungkyul.edu

**Eenjun Hwang**  
Korea University  
ehwang04@korea.ac.kr

## ABSTRACT

Many countries and cities in the world tend to have different types of preferred or popular music, such as pop, K-pop, and reggae. Music-related applications utilize geographical proximity for evaluating the similarity of music preferences between two regions. Sometimes, this can lead to incorrect results due to other factors such as culture and religion. To solve this problem, in this paper, we propose a scheme for constructing a music map in which regions are positioned close to one another depending on the similarity of the musical preferences of their populations. That is, countries or cities in a traditional map are rearranged in the music map such that regions with similar musical preferences are close to one another. To do this, we collect users' music play history and extract popular artists and tag information from the collected data. Similarities among regions are calculated using the tags and their frequencies. And then, an iterative algorithm for rearranging the regions into a music map is applied. We present a method for constructing the music map along with some experimental results.

## 1. INTRODUCTION

To recommend suitable music pieces to users, various methods have been proposed and one of them is the joint consideration of music and location information. In general, users in the same place tend to listen to similar kinds of music and this is shown by the statistics of music listening history. Context-aware computing utilizes this human tendency to recommend songs to a user.

However, the current approach of exploring geographical proximity for obtaining a user's music preferences might have several limitations due to various factors such as region scale, culture, religion, and language. That is, neighboring regions can show significant differences in music listening statistics and vice versa.

In fact, the geographical distance between two regions is not always proportional to the degree of difference in music preferences. For instance, assume that there are two neighboring countries having different music prefer-

ences. In the case of two regions near the border of the two countries, the people might show very different music preferences from those living in a region far from the border but in the same country. The degree of preference differences can be varied because of the difference in the sizes of the countries. Furthermore, the water bodies that cover 71% of the Earth's surface can lead to a disjunction of the differences.

Music from countries that have a high cultural influence might gain global popularity. For instance, pop music from the United States is very popular all over the world. Countries that have a common cultural background might have similar musical preferences irrespective of the geographical distance between them. Language is another important factor that can lead to different countries, such as the US and the UK, having similar popular music charts.

For these reasons, predicting musical preferences on the basis of geographical proximity can lead to incorrect results. In this paper, we present a scheme for constructing a music map where regions are positioned close to one another depending on the musical preferences of their populations. That is, regions such as cities in a traditional map are rearranged in the music map such that regions with similar musical preferences are close to one another. As a result, regions with similar musical preferences are concentrated in the music map and regions with distinct musical preferences are far away from the group.

The rest of this paper is organized as follows: In Section 2, we present a brief overview of the related works. Section 3 presents the scheme for mapping a geographical region to a new music space. Section 4 describes the experiments that we performed and some of the results. In the last section, we conclude the paper with directions for future work.

## 2. RELATED WORK

Many studies have tried to utilize location information for various music-related applications such as music search and recommendation. Kaminskis et al. presented a context-aware music recommender system that suggests music items on the basis of the users' contextual conditions, such as the users' mood or location [1]. They defined the term "place of interest (POI)" and considered the selection of suitable music tracks on the basis of the POI. In [2], Schedl et al. presented a music recommenda-



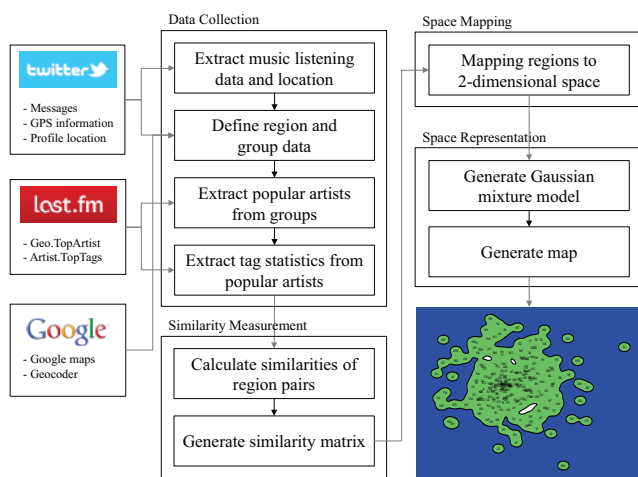


Figure 1. Overall scheme



Figure 2. Collected data from twitter

tion algorithm that combines information on the music content, music context, and user context by using a data set of geo-located music listening activities. In [3], Schedl et al derived and analyzed culture-specific music listening patterns by collecting music listening patterns of different countries (cities). They utilized social microblog such as Twitter and its tags in order to collect music-related information and measure the similarities between artists. Jun et al. presented a music recommender that considers personal and general musical predilections on the basis of time and location [4]. They analyzed massive social network streams from twitter and extracted the music listening histories. On the basis of a statistical analysis of the time and location, a collection of songs is selected and blended using automatic mixing techniques. These location-aware methods show a reasonable music search and recommendation performance when the range of the place of interest is small. However, the aforementioned problems might occur when the location range increases. Furthermore, these methods do not consider the case where remote regions have similar music preferences, which is often the case.

On the basis of these observations, in this paper, we propose a new data structure called a “music map”, where regions with similar musical preferences are located close to one another. Some pioneering studies to represent music by using visualization techniques have been reported. Lamere et al. presented an application for exploring and discovering new music by using a three-

dimensional (3D) visualization model [5]. Using the music similarity model, they provided new tools for exploring and interacting with a music collection. In [6], Knees et al. presented a user interface that creates a virtual landscape for music collection. By extracting features from audio signals and clustering the music pieces, they created a 3D island landscape. In [7], Pampalk et al. presented a system that facilitates the exploration of music libraries. By estimating the perceived sound similarities, music pieces are organized on a two-dimensional (2D) map so that similar pieces are located close to one another. In [8], Rauber et al. proposed an approach to automatically create an organization of music collection based on sound similarities. A 3D visualization of music collection offers an interface for an interactive exploration of large music repositories.

### 3. GEOGRAPHICAL REGION MAPPING

In this paper, we propose a scheme for geographical region mapping on the basis of the musical preferences of the people residing in these regions. The proposed scheme consists of three parts as shown in Figure 1. Firstly, the music listening history and the related location data are collected from Twitter. After defining regions, the collected data are refined to tag the statistics per region by querying popular artists and their popularities from last.fm. Similarities between the defined regions are calculated and stored in the similarity matrix. The similarity matrix is represented into a 2D space by using an iterative algorithm. Then, a Gaussian mixture model (GMM) is generated for constructing the music map on the basis of the relative location of the regions.

#### 3.1 Music Listen History and Location Collection

By analyzing the music listening history and location data, we can find out the music type that is popular in a certain city or country. In order to construct a music map, we need to collect the music listening history and location information on a global scale. To do this, we utilize last.fm, which is a popular music database. However, last.fm has several limitations related to the coverage of the global music listening history. The most critical one is that the database provides the listening data of a particular country only. In other words, we cannot obtain the data for a detailed region. Users in some countries (not all countries) use last.fm, and it does not contain sufficient data to cover the preferences of all the regions of these countries. Because of this, we observed that popular music in the real world does not always match with the last.fm data.

On the other hand, an explosive number of messages are generated all over the world through Twitter. Twitter is one of the most popular social network services. In this study, we use Twitter for collecting a massive amount of music listening history data. By filtering music-related messages from Twitter, we can collect various types of



#nowplaying	#np	#music
#soundcloud	#musicfans	#listenlive
#hiphop	#musicmondays	#pandora
#mp3	#itunes	#newmusic

Table 1. Music-related hashtags.

<Phrase A> by <Phrase B>
<Phrase A> - <Phrase B>
<Phrase A> / <Phrase B>
“<Phrase A>” - <Phrase B>

Table 2. Typical syntax for parsing song title and artist

music-related information, such as artist name, song title, and the published location. Figure 2 shows the distribution of the collected music-related tweets from around the world.

We used the Tweet Stream provided through a Twitter application processing interface (API) for collecting tweets. In order to select only the music-related tweets, we used music-related hashtags. Hashtags are very useful for searching the relevant tweets or for grouping tweets on the basis of topics. As shown in Table 1, we used the music-related hashtag lists that have been defined in [4]. Music-related tweet messages contain musical information such as song title and artist name. These textual data are represented in various forms. In particular, we considered the patterns shown in Table 2 for finding the artist names and the song titles. We employed a local MusicBrainz [9] server to validate the artist names.

For collecting location information, we gathered global positioning system (GPS) data that are included in tweet messages. However, we observed that the number of tweets that contain GPS data is quite small considering the total number of tweets. To solve this, we collected the profile location of the user who published a tweet message. Profile location contains the text address of the country or the city of the user. We employed the Google Geocoding API [10] for validating the location name and converting the address to GPS coordinates.

### 3.2 Region Definition and Tag Representation

Using the collected GPS information, we created a set of regions on the basis of the city or country. For grouping data by city name or country name, the collected GPS information is converted into its corresponding city or country name. In this study, we got 1327 cities or 198 countries from the music listening history collected through Twitter.

For each region, we collect two sets  $A_r$  and  $AC_r$  of referred artist names and their play counts, respectively:

$$A_r = \{a_1, \dots, a_n\} \quad (1)$$

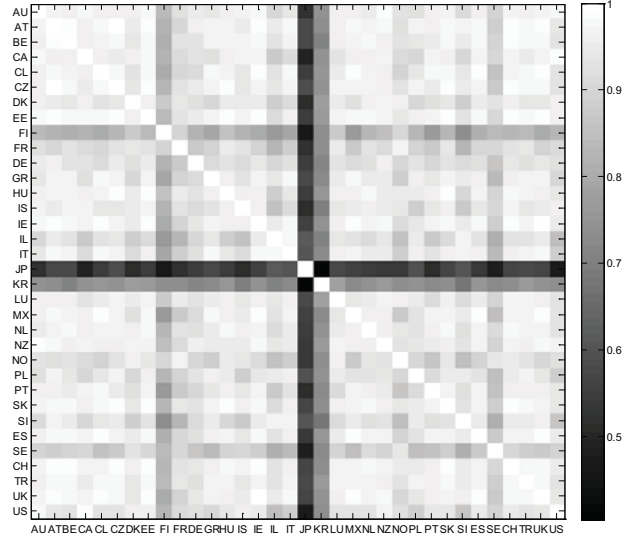


Figure 3. Tag similarity matrix of 34 countries

$$AC_r = \{ac_1, \dots, ac_n\} \quad (2)$$

where  $n$  is the number of referred artists. Also, using an artist name, we can collect his/her tag list. For a region  $r$ , we construct a set  $T_r$  of top tags by querying top tags to last.fm using the artist names of the region  $r$  as follows:

$$T_r = \{getTopTags(a_1) \cup \dots \cup getTopTags(a_n) \mid a_i \in A_r\} \\ = \{t_1, \dots, t_m\} \quad (3)$$

where  $getTopTags(a)$  returns a list of top tags of artist  $a$  and  $m$  is the number of collected tags for the region  $r$ . We define a function  $RTC(r, t)$  that calculates the total count of tag  $t$  in region  $r$  using the following equation:

$$RTC(r, t) = \sum_{a_i \in A_r} ac_i \times getTagCount(a_i, t) \quad (4)$$

Here,  $getTagCount(a, t)$  returns the count of tag  $t$  for the artist  $a$  in last.fm. In the same vein,  $RTC$  can return a set of tag counts when the second argument is a tag set  $T$ .

$$RTC(r, T) = \{RTC(r, t_1), \dots, RTC(r, t_m) \mid t_i \in T\} \quad (5)$$

### 3.3 Similarity Measurement

To construct a music map of regions, we need a measurement for estimating musical similarity. In this paper, we assume that music proximity between regions is closely related to the artists and their tags because the musical characteristics of a region can be explained by the artists' tags of the region. In particular, in order to measure the similarity among the regions represented by the tag groups, we employed a cosine similarity measurement as shown in the following equation:

$$TSM(r_1, r_2) = \frac{RTC(r_1, T_u) \times RTC(r_2, T_u)}{|RTC(r_1, T_{r_1})| \times |RTC(r_2, T_{r_2})|} \quad (6)$$

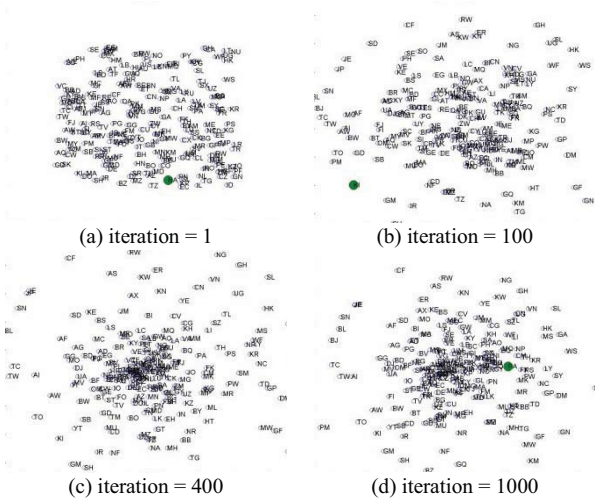


Figure 4. Example of mapped space in iterations

$$T_u = T_{r_1} \cap T_{r_2} \quad (7)$$

The cosine similarities of all possible pairs of regions were calculated and stored in the tag similarity matrix TSM. Hence, if there were  $m$  regions in the collection, we obtained a TSM of  $m \times m$ . A sample TSM for 34 countries is shown in Figure 3.

### 3.4 2D Space Mapping

On the basis of the TSM, we generated a 2D space for a music map by converting tag similarities between regions into proper metric for 2D space mapping. In this paper, this conversion is done approximately using an iterative algorithm. The proposed algorithm is based on the computational model such as a self-organizing map and an artificial neural network algorithm. By using an iterative phase, the algorithm gradually separates the regions in inverse proportion to the tag similarity.

#### 3.4.1 Initialization

In the initialization phase, 2D space is generated where X-axis and Y-axis of the space have ranges from 0 to 1. Each region is randomly placed on the 2D space. We observed that our random initialization does not provide deterministic result of the 2D space mapping.

#### 3.4.2 Iterations

In each iteration, a region in the 2D space is randomly selected and the tag distance  $TD$  between the selected region  $r_s$  and any other region  $r_i$  is computed using the similarity matrix.

$$TD(r_s, r_i) = 1 - TSM(r_s, r_i) \quad (8)$$

Subsequently, Euclidean distances  $ED$  between the selected region  $r_s$  and other region  $r_i$  is computed using the following equation

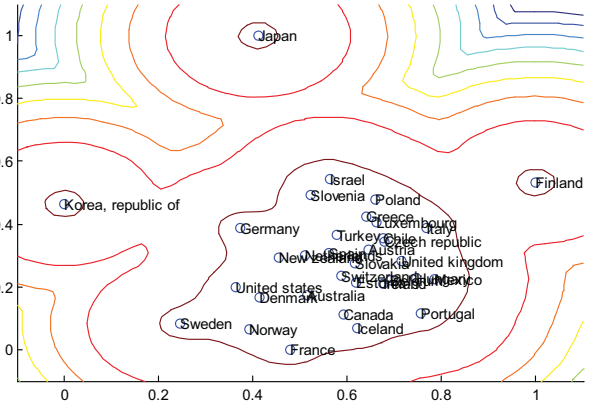


Figure 5. Gaussian mixture model of 34 countries

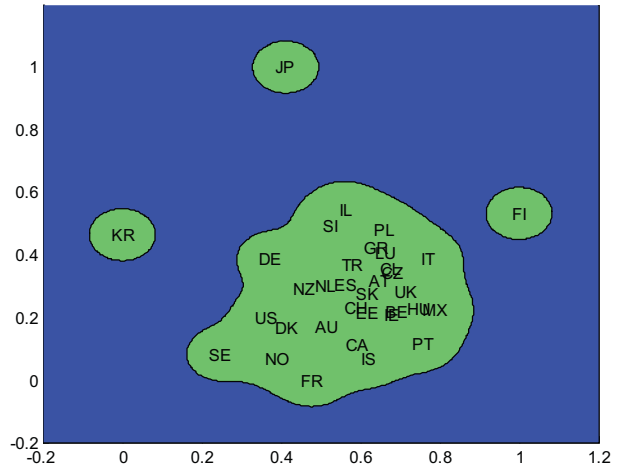


Figure 6. Music map of 34 countries

$$ED(r_s, r_i) = \sqrt{(x(r_s) - x(r_i))^2 + (y(r_s) - y(r_i))^2} \quad (9)$$

where  $x(r_i)$  and  $y(r_i)$  returns  $x$  and  $y$  positions of the region  $r_i$  in 2D space, respectively. In order for  $TD$  and  $ED$  to have same value as much as possible, the following equation is applied

$$x(r_i) = x(r_i) + \lambda(t)(ED(r_s, r_i) - TD(r_s, r_i)) \frac{(x(r_s) - x(r_i))}{ED(r_s, r_i)} \quad (10)$$

$$y(r_i) = y(r_i) + \lambda(t)(ED(r_s, r_i) - TD(r_s, r_i)) \frac{(y(r_s) - y(r_i))}{ED(r_s, r_i)} \quad (11)$$

Here,  $\lambda(t)$  is a learning rate in  $t$ -th iteration. The learning rate is monotonically decreased during iteration according to the following equation

$$\lambda(t) = \lambda_0 \exp(-t/T) \quad (12)$$

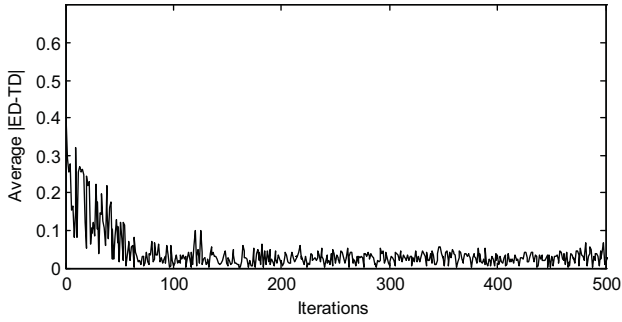


Figure 7. Average difference of distances in iterations

$\lambda_0$  denotes the initial learning rate, and T represents the total number of iterations. After each iteration, regions having higher TD are located far away from the selected region and regions having lower TD are located closer. Figure 4 shows examples of the mapped space after iterations.

### 3.5 Space Representation

After 2D space mapping, the regions are mapped such that regions having similar music preferences are placed close to one another. As a result, they form distinct crowds in the 2D space. In contrast, regions having unique preferences are placed apart from the crowds. To represent them as a map, a 2D distribution on the space is not sufficient. In this paper, in order to represent the information like a real world map, we employed the GMM. The Gaussian with diagonal matrix is constructed using the following equations:

$$\mu(i) = \{x(r_i), y(r_i)\} \tag{13}$$

$$\sigma(i) = \begin{bmatrix} 1/8n & 0 \\ 0 & 1/8n \end{bmatrix} \tag{14}$$

$$p(i) = \frac{1}{nm(r_i)} \tag{15}$$

Here,  $n$  is total number of regions and  $nm(r_i)$  returns the number of neighboring regions of region  $r_i$  in the 2D space. To model the GMM in the crowded area of 2D space, mixing proportion  $p(i)$  is adjusted based on the number of neighbors  $nm(r_i)$ . In other words,  $nm(r_i)$  has a higher value when  $p(i)$  is crowded and it reduces the proportion of  $i$ -th Gaussian. It helps to prevent Gaussian from over-height. An example of generated GMM is shown in Figure 5.

To generate a music map using the GMM, the probabilistic density function (pdf) of the GMM is simplified by applying a threshold. By projecting the GMM on the 2D plane after applying the threshold to the pdf, the boundaries of the GMM are created. We empirically found that the threshold value 0 gives an appropriate boundary. A boundary represents regions as a continent

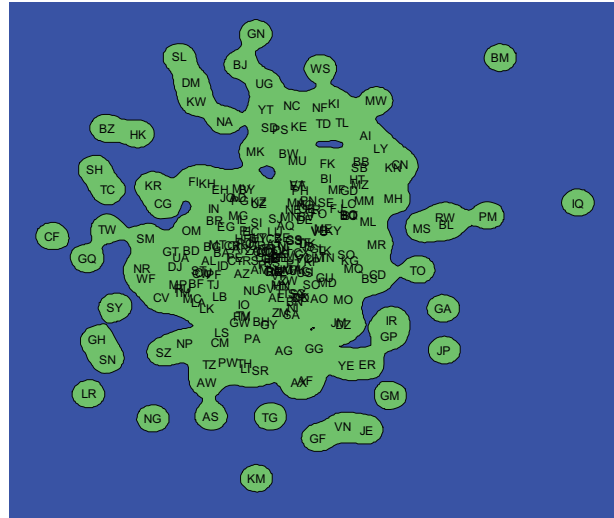


Figure 8. Music map of 239 countries

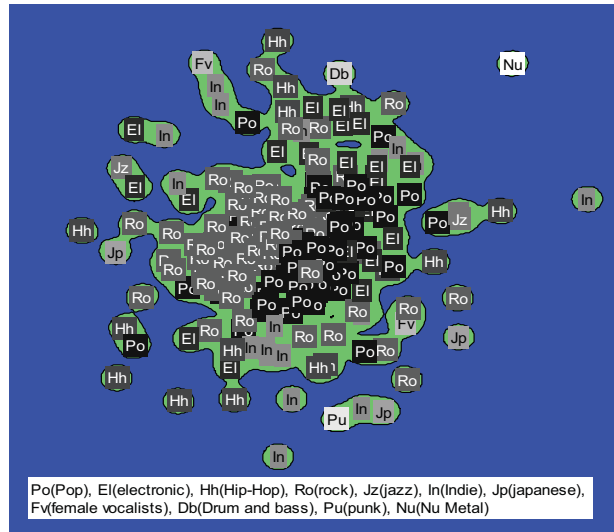


Figure 9. Top tags of music map.

or a small island on the basis of their distribution. As a result, the mapped result is visualized as a music map having an appearance similar to that of a real world map. An example of a music map for 34 countries is shown in Figure 6. Although the generated music map contains less information than the contour graph of GMM, it could be more intuitive to the casual users to understand the relations between regions in terms of music preferences.

## 4. EXPERIMENT

### 4.1 Experiment Setup

To collect the music-related tweets, we gathered the tweet streams from the Twitter server in real time in order to collect the music information of Twitter users. During one week, we collected 4.57 million tweets that had the hashtags listed in Table 1. After filtering the tweets through regular expressions, 1.56 million music listening history records were collected. We got 1327 cities or 198

countries from the music listening history collected through Twitter. We collected the lists of the top artists for 249 countries from last.fm. For these countries, 2735 artists and their top tags were collected from last.fm.

#### 4.2 Differences of ED and TD

In the proposed scheme, the iterative algorithm gradually reduces the difference between ED and TD, as mentioned above. In order to show that the algorithm reduces the difference and moves the regions appropriately, the average difference between ED and TD is measured in each iteration. Figure 7 shows the average distances during 500 iterations. The early phases in the computation show high average distance differences due to the random initialization. As the iteration proceeds, the average distance differences are gradually reduced and converged.

#### 4.3 Map Generation for 249 Countries

In order to evaluate the effectiveness of the proposed scheme, we defined a region group that contained 249 countries. After collecting the music listening history from Twitter and last.fm, we generated a music map by using the proposed scheme. Figure 8 shows the resulting music map. We observed that the map consisted of a big island (continent) and a few small islands. In the center of the big island, countries that had a high musical influence, such as the US and the UK, were located. On the other hand, countries having unique music preferences such as Japan and Hong Kong were formed as small islands and located far away from the big island.

#### 4.4 Top Tag Representation

A music map is based on the musical preferences between regions, and these preferences were calculated on the basis of the similarities of the musical tags. In the last experiment, we first find out the top tag of each country and show the distribution of the top tags in the music map. Figure 9 shows the top tags of the map in Figure 8. In the map, “Rock” and “Pop”, which are the most popular tags in the collected data, are located in the center and occupies a significant portion of the big island. On the north side of the big island, “Electronic” tag is located and in the south, “Indie” tag is placed. The “Pop” tag, which is popular in almost every country, is located throughout the map.

### 5. CONCLUSION

In this paper, we proposed a scheme for constructing a music map in which regions such as cities and countries are located close to one another depending on the musical preferences of the people residing in them. To do this, we collected the music play history and extracted the popular artists and tag information from Twitter and last.fm. A similarity matrix for each region pair was calculated by using the tags and their frequencies. By applying an iterative algorithm and GMM, we reorganized the regions into

a music map according to the tag similarities. The possible application domains of the proposed scheme span a broad range—from music collection, browsing services, and music marketing tools, to a worldwide music trend analysis.

### 6. ACKNOWLEDGEMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2012627) and the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2014-H0301-14-1001) supervised by the NIPA (National IT Industry Promotion Agency)

### 7. REFERENCES

- [1] M. Kaminskas and F. Ricci, “Location-Adapted Music Recommendation Using Tags,” in *User Modeling, Adaption and Personalization*, Springer Berlin Heidelberg, 2011, pp. 183–194.
- [2] M. Schedl and D. Schnitzer, “Location-Aware Music Artist Recommendation,” in *MultiMedia Modeling*, Springer International Publishing, 2014, pp. 205–213.
- [3] M. Schedl and D. Hauger, “Mining Microblogs to Infer Music Artist Similarity and Cultural Listening Patterns,” in *Proceedings of the 21st International Conference Companion on World Wide Web*, New York, USA, 2012, pp. 877–886.
- [4] S. Jun, D. Kim, M. Jeon, S. Rho, and E. Hwang, “Social mix: automatic music recommendation and mixing scheme based on social network analysis,” *Journal of Supercomputing*, pp. 1–22, Apr. 2014.
- [5] P. Lamere and D. Eck, “Using 3d visualizations to explore and discover music,” in *Int. Conference on Music Information Retrieval*, 2007.
- [6] P. Knees, M. Schedl, T. Pohle, and G. Widmer, “Exploring Music Collections in Virtual Landscapes,” *IEEE MultiMedia*, vol. 14, no. 3, pp. 46–54, Jul. 2007.
- [7] E. Pampalk, A. Rauber, and D. Merkl, “Content-based Organization and Visualization of Music Archives,” in *Proceedings of the Tenth ACM International Conference on Multimedia*, New York, NY, USA, 2002, pp. 570–579.
- [8] A. Rauber, E. Pampalk, and D. Merkl, “The SOM-enhanced JukeBox: Organization and Visualization of Music Collections Based on Perceptual Models,” *Journal of New Music Research*, vol. 32, no. 2, pp. 193–210, 2003.
- [9] “MusicBrainz - The Open Music Encyclopedia.” [Online]. Available: <http://musicbrainz.org/>. [Accessed: 03-May-2014].
- [10] “The Google Geocoding API” [Online]. Available: <https://developers.google.com/maps/documentation/geocoding/>. [Accessed: 03-May-2014]

## ON COMPARATIVE STATISTICS FOR LABELLING TASKS: WHAT CAN WE LEARN FROM MIREX ACE 2013?

**John Ashley Burgoyne**

Universiteit van Amsterdam  
j.a.burgoyne@uva.nl

**W. Bas de Haas**

Universiteit Utrecht  
w.b.dehaas@uu.nl

**Johan Pauwels**

STMS IRCAM-CNRS-UPMC  
johan.pauwels@gmail.com

### ABSTRACT

FOR MIREX 2013, the evaluation of audio chord estimation (ACE) followed a new scheme. Using chord vocabularies of differing complexity as well as segmentation measures, the new scheme provides more information than the ACE evaluations from previous years. With this new information, however, comes new interpretive challenges. What are the correlations among different songs and, more importantly, different submissions across the new measures? Performance falls off for all submissions as the vocabularies increase in complexity, but does it do so directly in proportion to the number of more complex chords, or are certain algorithms indeed more robust? What are the outliers, song-algorithm pairs where the performance was substantially higher or lower than would be predicted, and how can they be explained? Answering these questions requires moving beyond the Friedman tests that have most often been used to compare algorithms to a richer underlying model. We propose a logistic-regression approach for generating comparative statistics for MIREX ACE, supported with generalised estimating equations (GEES) to correct for repeated measures. We use the MIREX 2013 ACE results as a case study to illustrate our proposed method, including some of interesting aspects of the evaluation that might not appear from the headline results alone.

### 1. INTRODUCTION

Automatic chord estimation (ACE) has a long tradition within the music information retrieval (MIR) community, and chord transcriptions are generally recognised as a useful mid-level representation in academia as well as in industry. For instance, in an academic context it has been shown that chords are interesting for addressing musicological hypotheses [3,13], and that they can be used as a mid-level feature to aid in retrieval tasks like cover-song detection [7,10]. In

Johan Pauwels is no longer affiliated with STMS. Data and source code to reproduce this paper, including all statistics and figures, are available from <http://bitbucket.org/jaburgoyne/ismir-2014>.



© John Ashley Burgoyne, W. Bas de Haas, Johan Pauwels. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** John Ashley Burgoyne, W. Bas de Haas, Johan Pauwels. "On comparative statistics for labelling tasks: What can we learn from MIREX ACE 2013?", 15th International Society for Music Information Retrieval Conference, 2014.

an industrial setting, music start-ups like Riffstation<sup>1</sup> and Chordify<sup>2</sup> use ACE in their music teaching tools, and at the time of writing, Chordify attracts more than 2million unique visitors every month [6].

In order to compare different algorithmic approaches in an impartial setting, the Music Information Retrieval Evaluation eXchange (MIREX) introduced an annual ACE task in 2008. Since then, between 11 and 18 algorithms have been submitted each year by between 6 and 13 teams. Despite the fact that ACE algorithms are used outside of academic environments, and even though the number of MIREX participants has decreased slightly over the last three years, the problem of automatic chord estimation is nowhere near solved. Automatically extracted chord sequences have classically been evaluated by calculating the *chord symbol recall* (CSR), which reflects the proportion of correctly labelled chords in a single song, and a *weighted chord symbol recall* (WCSR), which weights the average CSR of a set of songs by their length. On fresh validation data, the best-performing algorithms in 2013 achieved WCSR of only 75 percent, and that only when the range of possible chords was restricted exclusively to the 25 major, minor and "no-chord" labels; the figure drops to 60 percent when the evaluation is extended to include seventh chords (see Table 1).

MIREX is a terrific platform for evaluating the performance of ACE algorithms, but by 2010 it was already being recognised that the metrics could be improved. At that time, they included only CSR and WCSR using a vocabulary of 12 major chords, 12 minor chords and a "no-chord" label. At ISMIR 2010, a group of ten researchers met to discuss their dissatisfaction. In the resulting 'Utrecht Agreement',<sup>3</sup> it was proposed that future evaluations should include more diverse chord vocabularies, such as seventh chords and inversions, as the 25-chord vocabulary was considered a rather coarse representation of tonal harmony. Furthermore, the group agreed that it was important to include a measure of segmentation quality in addition to CSR and WCSR.

At approximately the same time, Christopher Harte proposed a formalisation of measures that implemented the aspirations indicated in the Utrecht agreement [8]. Recently, Pauwels and Peeters reformulated and extended Harte's work with the precise aim of handling differences in chord vocabulary between annotated ground truth and algorithmic

<sup>1</sup> <http://www.riffstation.com/>

<sup>2</sup> <http://chordify.net>

<sup>3</sup> [http://www.music-ir.org/mirex/wiki/The\\_Utrecht\\_Agreement\\_on\\_Chord\\_Evaluation](http://www.music-ir.org/mirex/wiki/The_Utrecht_Agreement_on_Chord_Evaluation)

Algorithm	# Types	Inversions?	Training?	I	II	III	IV	V	VI	VII	VIII
KO2	7		•	76	74	72	60	58	84	79	89
NMSD2	10			75	71	69	59	57	82	79	86
CB4	13		•	76	72	70	59	57	85	80	90
NMSDI	10			74	71	69	58	56	83	79	86
CB3	13			76	72	70	58	56	85	81	89
KO1	7			75	71	69	54	52	83	80	88
PP4	5			69	66	64	51	49	83	78	87
PP3	2			70	68	65	50	48	83	82	84
CF2	10	•		71	67	65	49	47	83	83	83
NG1	2			71	67	65	49	46	82	79	86
NG2	5			67	63	61	44	43	82	81	83
SB8	2			9	7	6	5	5	51	92	35

**Table 1.** Number of supported chord types, inversion support, training support, and MIREX results on the *Billboard 2013* test set for all 2013 ACE submissions. I: root only; II: major-minor vocabulary; III: major-minor vocabulary with inversions; IV: major-minor vocabulary with sevenths; V: major-minor vocabulary with sevenths and inversions; VI: mean segmentation score; VII: under-segmentation; VIII: over-segmentation. Adapted from the MIREX Wiki.

output on one hand, and among the output of different algorithms on the other hand [15]. They also performed a rigorous re-evaluation of all MIREX ACE submissions from 2010 to 2012. As of MIREX 2013, these revised evaluation procedures, including the chord-sequence segmentation evaluation suggested by Harte [8] and Mauch [12], have been adopted in the context of the MIREX ACE task.

MIREX ACE evaluation has also typically included comparative statistics to help determine whether the differences in performance between pairs of algorithms are statistically significant. Traditionally, Friedman’s ANOVA has been used for this purpose, accompanied by Tukey’s Honest Significant Difference tests for each pair of algorithms. Friedman’s ANOVA is equivalent to a standard two-way ANOVA with the actual measurements (in our case WCSR or directional Hamming distance [DHD], the new segmentation measure) replaced by the rank of each treatment (in our case, each algorithm) on that measure within each block (in our case, for each song) [11]. The rank transformation makes Friedman’s ANOVA an excellent ‘one size fits all’ approach that can be applied with minimal regard to the underlying distribution of the data, but these benefits come with costs. Like any non-parametric test, Friedman’s ANOVA can be less powerful than parametric alternatives where the distribution is known, and the rank transformation can obscure information inherent to the underlying measurement, magnifying trivial differences and neutralising significant inter-correlations.

But there is no need to pay the costs of Friedman’s ANOVA for evaluating chord estimation. Fundamentally, WCSR is a proportion, specifically the expected proportion of audio frames that an estimation algorithm will label correctly, and as such, it fits naturally into *logistic regression* (i.e., a *logit model*). Likewise, DHD is constrained to fall between 0 and 100 percent, and thus it is also suitable for the same type of analysis. The remainder of this paper describes how logistic regression can be used to compare chord estimation algorithms, using MIREX results from 2013 to illustrate four key benefits: easier interpretation, greater statistical power, built-in correlation estimates for identifying relationships among algorithms, and better detection of outliers.

## 2. LOGISTIC REGRESSION WITH GEES

Proportions cannot be distributed normally because they are supported exclusively on  $[0, 1]$ , and thus they present challenges for traditional techniques of statistical analysis. Logit models are designed to handle these challenges without sacrificing the simplicity of the usual linear function relating parameters and covariates [1, ch.4]:

$$\pi(\mathbf{x}; \boldsymbol{\beta}) = \frac{e^{\mathbf{x}'\boldsymbol{\beta}}}{1 + e^{\mathbf{x}'\boldsymbol{\beta}}}, \quad (1)$$

or equivalently

$$\log \frac{\pi(\mathbf{x}; \boldsymbol{\beta})}{1 - \pi(\mathbf{x}; \boldsymbol{\beta})} = \mathbf{x}'\boldsymbol{\beta}, \quad (2)$$

where  $\pi$  represents the relative frequency of ‘success’ given the values of covariates in  $\mathbf{x}$  and parameters  $\boldsymbol{\beta}$ . In the case of a basic model for MIREX ACE,  $\mathbf{x}$  would identify the algorithm and  $\pi$  would be the relative frequency of correct chord labels for that algorithm (i.e., WCSR). In the case of data like ACE results, where there are proportions  $p_i$  of correct labels over  $n_i$  analysis frames rather than binary successes or failures,  $i$  indexing all combinations of individual songs and algorithms, logistic regression assumes that each  $p_i$  represents the observed proportion of successes among  $n_i$  conditionally-independent binary observations, or more formally, that the  $p_i$  are distributed binomially:

$$f_{P|N, \mathbf{X}}(p | n, \mathbf{x}; \boldsymbol{\beta}) = \binom{n}{pn} \pi^{pn} (1 - \pi)^{(1-p)n}. \quad (3)$$

The expected value for each  $p_i$  is naturally  $\pi_i = \pi(\mathbf{x}_i; \boldsymbol{\beta})$ , the overall relative frequency of success given  $\mathbf{x}_i$ :

$$\mathbf{E}[P | N, \mathbf{X}] = \pi(\mathbf{x}; \boldsymbol{\beta}). \quad (4)$$

Logistic regression models are most often fit by the maximum-likelihood technique, i.e., one is seeking a vector  $\hat{\boldsymbol{\beta}}$  to maximise the log-likelihood given the data:

$$\ell_{P|N, \mathbf{X}}(\boldsymbol{\beta}; \mathbf{p}, \mathbf{n}, \mathbf{X}) = \sum_i \left[ \log \binom{n_i}{p_i n_i} + p_i n_i \log \pi_i + (1 - p_i) n_i \log (1 - \pi_i) \right]. \quad (5)$$

One thus solves the system of likelihood equations for  $\hat{\boldsymbol{\beta}}$ , whereby the gradient of Equation 5 is set to zero:

$$\nabla_{\boldsymbol{\beta}} \ell_{P|N, \mathbf{X}}(\boldsymbol{\beta}; \mathbf{p}, \mathbf{N}, \mathbf{X}) = \sum_i (p_i - \pi_i) n_i \mathbf{x}_i = \mathbf{0} \quad (6)$$

and so

$$\sum_i p_i n_i \mathbf{x}_i = \sum_i \pi_i n_i \mathbf{x}_i . \quad (7)$$

In the case of MIREX ACE evaluation, each  $\mathbf{x}_i$  is simply an indicator vector to partition the data by algorithm, and thus  $\hat{\boldsymbol{\beta}}$  is the parameter vector for which  $\pi_i$  equals the song-length-weighted mean over all  $p_i$  for that algorithm.

## 2.1 Quasi-Binomial Models

Under a strict logit model, the variance of each  $p_i$  is inversely proportional to  $n_i$ :

$$\text{var}[P | N, \mathbf{X}] = \left(\frac{1}{n}\right) \pi(1 - \pi) . \quad (8)$$

Equation 8 only holds, however, if the estimates of chord labels for each audio frame are independent. For ACE, this is unrealistic: only the most naïve algorithms treat every frame independently. Some kind of time-dependence structure is standard, most frequently a hidden Markov model or some close derivative thereof. Hence one would expect that the variance of WCSR estimates should be rather larger than the basic logit model would suggest.

This type of problem is extremely common across disciplines, so much so that it has been given a name, *over-dispersion*, and some authors go so far as to state that ‘unless there are good external reasons for relying on the binomial assumption [of independence], it seems wise to be cautious and to assume that over-dispersion is present to some extent unless and until it is shown to be absent’ [14, p.125]. One standard approach to handling over-dispersion is to use a so-called *quasi-likelihood* [1, §4.7]. In case of logistic regression, this typically entails a modification to the assumption on the distribution of the  $p_i$  that includes an additional *dispersion parameter*  $\phi$ . The expected values are the same as a standard binomial model, but

$$\text{var}[P | N, \mathbf{X}] = \left(\frac{\phi}{n}\right) \pi(1 - \pi) . \quad (9)$$

These models are known as quasi-likelihood models because one loses a closed-form solution for the actual probability distribution  $f_{P|N, \mathbf{X}}$ ; one knows only that the  $p_i$  behave something like binomially-distributed variables, with identical means but proportionally more variance. The parameter estimates  $\hat{\boldsymbol{\beta}}$  and predictions  $\pi(\cdot; \hat{\boldsymbol{\beta}})$  for a quasi-binomial model are the same as ordinary logistic regression, but the estimated variance-covariance matrices are scaled by the estimated dispersion parameter  $\hat{\phi}$  (and likewise the standard errors are scaled by its square root). The dispersion parameter is estimated so that the theoretical variance matches the empirical variance in the data, and because of the form of Equation 9, it renders any scaling considerations for the  $n_i$  moot.

Other approaches to handling over-dispersion include *beta-binomial models* [1, §13.3] and *beta regression* [5], but we prefer the simplicity of the quasi-likelihood model.

## 2.2 Generalised Estimating Equations (GEES)

The quasi-binomial model achieves most of what one would be looking for when evaluating ACE for MIREX: it handles proportions naturally, is consistent with the weighted averaging used to compute WCSR, and adjusts for over-dispersion in a way that also eliminates any worries about scaling. Nonetheless, it is slightly over-conservative for evaluating ACE. As discussed earlier, quasi-binomial models are necessary to account for over-dispersion, and one important source of over-dispersion in these data is the lack of independence of chord estimates from most algorithms within the same song. MIREX exhibits another important violation of the independence assumption, however: all algorithms are tested on the same sets of songs, and some songs are clearly more difficult than others. Put differently, one does not expect the algorithms to perform completely independently of one another on the same song but rather expects a certain correlation in performance across the set of songs. By taking that correlation into account, one can improve the precision of estimates, particularly the precision of pair-wise comparisons [1, §10.1].

A relatively straightforward variant of quasi-likelihood known as *generalised estimating equations* (GEES) incorporates this type of correlation [1, ch.11]. With the GEE approach, rather than predicting each  $p_i$  individually, one predicts complete vectors of proportions  $\mathbf{p}_i$  for each relevant group, much as Friedman’s test seeks to estimate ranks within each group. For ACE, the groups are songs, and thus one considers the observations to be vectors  $\mathbf{p}_i$ , one for each song, where  $p_{ij}$  represents the CSR or segmentation score for algorithm  $j$  on song  $i$ . Analogous to the case of ordinary quasi-binomial or logistic regression,

$$\mathbf{E}[P_j | N, \mathbf{X}_j] = \pi(\mathbf{x}_j; \boldsymbol{\beta}) . \quad (10)$$

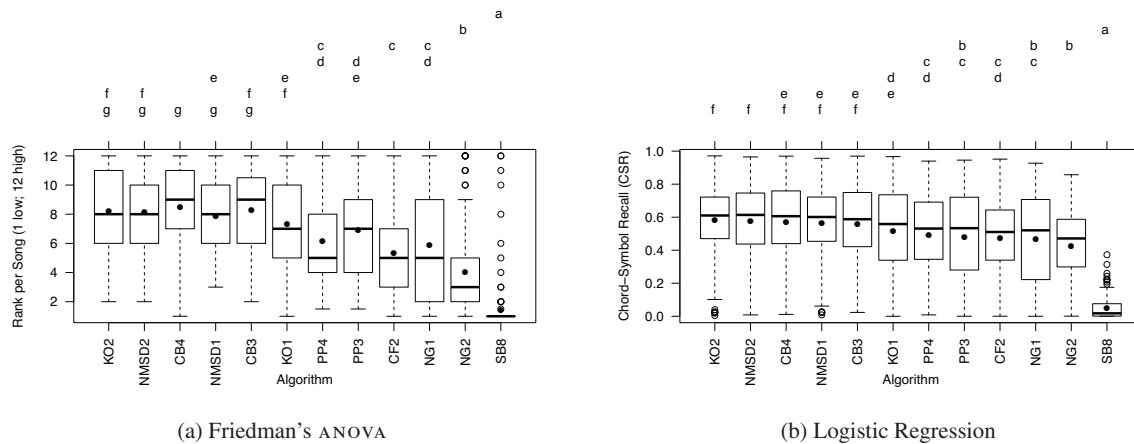
Likewise, analogous to the quasi-binomial variance,

$$\text{var}[P_j | N, \mathbf{X}_j] = \left(\frac{\phi}{n}\right) \pi_j(1 - \pi_j) . \quad (11)$$

Because the GEE approach is concerned with vector-valued estimates rather than point estimates, it also involves estimating a full variance-covariance matrix. In addition to  $\boldsymbol{\beta}$  and  $\phi$ , the approach requires a further vector of parameters  $\boldsymbol{\alpha}$  and an *a priori* assumption on the correlation structure of the  $P_j$  in the form of a function  $R(\boldsymbol{\alpha})$  that yields a correlation matrix. (One might, for example, assume that the  $P_j$  are *exchangeable*, i.e., that every pair shares a common correlation coefficient.) Then if  $B$  is a diagonal matrix such that  $B_{jj} = \text{var}[P_j | N, \mathbf{X}_j]$ ,

$$\text{cov}[\mathbf{P} | N, \mathbf{X}] = B^{1/2} R(\boldsymbol{\alpha}) B^{1/2} . \quad (12)$$

If all of the  $P_j$  are uncorrelated with each other, then this formula reduces to the basic quasi-binomial model, which assumes a diagonal covariance matrix. The final step of GEE estimation adjusts Equation 12 according to the actual correlations observed in the data, and as such, GEES are quite robust in practice even when the *a priori* assumptions about the correlation structure are incorrect [1, §11.4.2].



**Figure 1**. Boxplots and compact letter displays for the MIREX ACE 2013 results on the *Billboard* 2013 test set with vocabulary V (seventh chords and inversions), weighted by song length. Bold lines represent medians and filled dots means.  $N = 161$  songs per algorithm. Given the respective models, there are insufficient data to distinguish among algorithms sharing a letter, correcting to hold the FDR at  $\alpha = .005$ . Although Friedman's ANOVA detects 2 more significant pairwise differences than logistic regression (45 vs. 43), it operates on a different scale than CSR and misorders algorithms relative to WCSR.

### 3. ILLUSTRATIVE RESULTS

MIREX ACE 2013 evaluated 12 algorithms according to a battery of eight rubrics (WCSR on five harmonic vocabularies and three segmentation measures) on each of three different data sets (the Isophonics set, including music from the Beatles, Queen, and Zweieck [12] and two versions of the McGill *Billboard* set, including music from the American pop charts [4]). There is insufficient space to present the results of logistic regression on all combinations, and so we will focus on a single one of the data sets, the *Billboard* 2013 test set. In some cases, logistic regression allows us to speak to all measures (11 592 observations), but in general, we will also restrict ourselves to discussing the newest and most challenging of the harmonic vocabularies for WCSR: Vocabulary V (1932 observations), which includes major chords, minor chords, major sevenths, minor sevenths, dominant sevenths, and the complete set of inversions of all of the above. We are interested in four key questions.

1. How do pairwise comparisons under logistic regression compare to pairwise comparisons with Friedman's ANOVA? Is logistic regression more powerful?
2. Are there differences among algorithms as the harmonic vocabularies get more difficult, or is the drop performance uniform? In other words, is there a benefit to continuing with so many vocabularies?
3. Are all ACE algorithms making similar mistakes, or do they vary in their strengths and weaknesses?
4. Which algorithm-song pairs exhibited unexpectedly good or bad performance, and is there anything to be learned from these observations?

#### 3.1 Pairwise Comparisons

The boxplots in Figure 1 give a more detailed view of the performance of each algorithm than Table 1. The figure

is restricted to Vocabulary V, with the algorithms in descending order by WCSR. Figure 1 a comes from Friedman's ANOVA weighted by song length, and thus its y-axis reflects not CSR directly but the per-song ranks with respect to CSR. Figure 1 b comes from quasi-binomial regression estimated with GEES, as described in Section 2. Its y-axis does reflect per-song CSR. Above the boxplots, all significant pairwise differences are recorded as a *compact letter display*. In the interest of reproducible research, we used a stricter  $\alpha = .005$  threshold for reporting pairwise comparisons with the more contemporary false-discovery-rate (FDR) approach of Benjamini and Hochberg, as opposed to more traditional Tukey tests at  $\alpha = .05$  [2, 9]. Within either of the subfigures, the difference in performance between two algorithms that share any letter in the compact letter display is *not* statistically significant. Overall, Friedman's ANOVA found 2 more significant pairwise differences than logistic regression.

#### 3.2 Effect of Vocabulary

To test the utility of the new evaluation vocabularies, we ran both Friedman ANOVAs (ranked separately for each vocabulary) and logistic regressions and looked for significant interactions among the algorithm, inversions (present or absent from the vocabulary) and the complexity of the vocabulary (root only, major-minor, or major-minor with 7ths). Under Friedman's ANOVA, there was a significant Algorithm  $\times$  Complexity interaction,  $F(22, 9440) = 3.21$ ,  $p < .001$ . The logistic regression model identified a significant three-way Algorithm  $\times$  Complexity  $\times$  Inversions interaction,  $\chi^2(12) = 37.35$ ,  $p < .001$ , but the additional interaction with inversions should be interpreted with care: only one algorithm (CF2) attempts to recognise inversions.

#### 3.3 Correlation Matrices

Table 2 presents the inter-correlations of WCSR between algorithms, rank-transformed (Spearman's correlations, ana-



Algorithm	KO2	NMSD2	CB4	NMSDI	CB3	KOI	PP4	PP3	CF2	NG1	NG2	SB8
KO2	–	.07	.11	–.05	.10	.03	–.41*	–.44*	–.03	–.35*	.05	–.01
NMSD2	.25*	–	–.01	.49*	–.25*	–.20	–.19	–.36*	.00	–.33*	.02	–.06
CB4	.41*	.39*	–	.12	.47*	–.46*	–.30*	–.48*	.09	–.38*	.08	–.09
NMSDI	.30*	.60*	.53*	–	–.17	–.45*	–.08	–.45*	.27*	–.44*	.17	–.10
CB3	.34*	.10	.76*	.42*	–	–.19	–.26*	–.14	–.08	–.17	–.16	–.08
KOI	–.04	–.42*	–.51*	–.51*	–.29*	–	–.10	.42*	–.41*	.50*	–.52*	.05
PP4	–.22	.08	–.16	.06	–.07	–.05	–	.37*	–.03	.00	.05	–.03
PP3	–.49*	–.46*	–.61*	–.53*	–.37*	.68*	.22	–	–.48*	.66*	–.48*	.04
CF2	.09	.19	.24*	.42*	.17	–.49*	.06	–.51*	–	–.48*	.48*	–.14
NG1	–.54*	–.42*	–.60*	–.56*	–.41*	.68*	.04	.85*	–.47*	–	–.40*	–.10
NG2	.09	.17	.17	.16	–.03	–.50*	–.09	–.54*	.50*	–.40*	–	–.11
SB8	–.32*	–.44*	–.44*	–.52*	–.46*	.00	–.32*	.08	–.33*	.08	–.16	–

**Table 2** . Pearson’s correlations on the coefficients from logistic regression (WCSR) for the *Billboard* 2013 test set with vocabulary V (lower triangle); Spearman’s correlations for the same data (upper triangle).  $N = 161$  songs per cell. Starred correlations are significant at  $\alpha = .005$ , controlling for the FDR. A set of algorithms (viz., KO1, PP3, NG1, and SB8) stands out for negative correlations with the top performers; in general, these algorithms did not attempt to recognise seventh chords.

logous to Friedman’s ANOVA) in the upper triangle, and in the lower triangle, as estimated from logistic regression with GEES. Significant correlations are marked, again controlling the FDR at  $\alpha = .005$ . Positive correlations do not necessarily imply that the algorithms perform similarly; rather it implies that they find the same songs relatively easy or difficult. Negative correlations imply that songs that one algorithm finds difficult are relatively easy for the other algorithm.

### 3.4 Outliers

To identify outliers, we considered all evaluations on the *Billboard* 2013 test set and examined the distribution of residuals. Chauvenet’s criterion for outliers in a sample of this size is to lie more than 4.09 standard deviations from the mean [16, §6.2]. Under Friedman’s ANOVA, Chauvenet’s criterion identified 7 extreme data points. These are all for algorithm SB8, a submission with a programming bug that erroneously returned alternating C- and B-major chords regardless of the song, on songs that were so difficult for most other algorithms that the essentially random approach of the bug did better. Under the logistic regression model, the criterion identified 26 extreme points. Here, the unexpected behaviour was primarily for songs that are tuned a quarter-tone off from standard tuning ( $A_4 = 440$  Hz). The ground truth necessarily is ‘rounded off’ to standard tuning in one direction or the other, but in cases where an otherwise high-performing algorithm happened to round off in the opposite direction, the performance is markedly low.

## 4. DISCUSSION

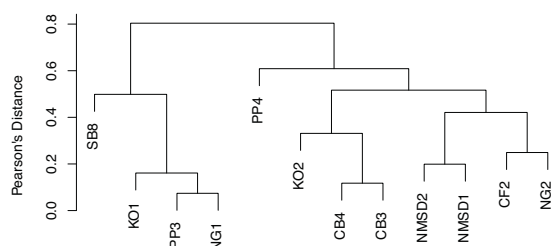
We were surprised to find that in terms of distinguishing between algorithms, Friedman’s ANOVA was in fact more powerful than logistic regression, detecting a few extra significant pairs. Nonetheless, the two approaches yield substantially equivalent broad conclusions: that a group of top performers – CB3, CB4, KO2, NMSDI, and NMSD2 – are statistically indistinguishable from each other, with KO1 also indistinguishable from the lower end of this group. Moreover, having now benefited from years of study, WCSR

is a reasonably intuitive and well-motivated measure of ACE performance, and it is awkward to have to work on the Friedman’s rank scale instead, especially since it ultimately ranks the algorithms’ overall performance in a slightly different order than the headline WCSR-based results.

Friedman’s ANOVA did exhibit less power for our question about interactions between algorithms and differing chord vocabularies. Again, WCSR as a unit and as a concept is highly meaningful for chord estimation, and there is a conceptual loss from rank transformation. Given the rank transformation, Friedman’s ANOVA can only be sensitive to reconfigurations of relative performance as the vocabularies become more difficult; logistic regression can also be sensitive to different effect sizes across algorithms even when their relative ordering remains the same.

It was encouraging to see that under either statistical model, there was a benefit to evaluating with multiple vocabularies. That encouraged us to examine the inter-correlations for the performance of the algorithms. Figure 2 summarises the original correlation matrix in Table 2 more visually by using the correlations from logistic regression as the basis of a hierarchical clustering. Two clear groups emerge, both from the clustering and from minding negative correlations in the original matrix: one relatively low-performing group including KO1, PP3, NG1, and SB8, and one relatively high-performing group including all others but for perhaps PP4, which does not seem to correlate strongly with any other algorithm. The shape of the equivalent tree based on Spearman’s correlations is similar but for joining PP4 with SB8 instead of the high-performing group. Table 1 uncovers the secret behind the low performers: KO1 excepted, none of the low-performing algorithms attempt to recognise seventh chords, which comprise 29 percent of all chords under Vocabulary V. Furthermore, we performed an additional evaluation of seventh chords only, in the style of [15] and using their software available online.<sup>4</sup> From the resulting low score of KO1, we can deduce that this algorithm is able to recognise seventh chords in theory, but that it was most likely trained on the relatively seventh-poor Isophon-

<sup>4</sup> <https://github.com/jpauwels/MusOEEvaluator>



**Figure 2** . Hierarchical clustering of algorithms based on WCSR for for the *Billboard* 2013 test set with vocabulary  $V$ , Pearson's distance as derived from the estimated correlation matrix under logistic regression, and complete linkage. The group of algorithms that is negatively correlated with the top performers appears at the left. PP4 stands out as the most idiosyncratic performer.

ics corpus (only 15 percent of all chords). KO2 is the same algorithm trained directly on the MIREX *Billboard* training corpus, and with that training, it becomes a top performer.

Our analysis of outliers again showed Friedman's ANOVA to be less powerful than logistic regression, as one would expect given the range restrictions on rank transformation. But here also the more important advantage of logistic regression is the ability to work on the WCSR scale. Outliers under the logistic regression model are also points that have an unusually strong effect on the reported results. In our analysis, they highlight the practical consequences of the well-known problem of atypically-tuned commercial recordings. Although we would not propose deleting outliers, it is sobering to know that tuning problems may be having an outsized effect on our headline evaluation figures. It might be worth considering allowing algorithms their best score in keys up to a semitone above or below the ground truth.

Overall, we have shown that as ACE becomes more established and its evaluation more thorough, it is useful to use a subtler statistical model for comparative analysis. We recommend that future MIREX ACE evaluations use logistic regression in preference to Friedman's ANOVA. It preserves the natural units and scales of WCSR and segmentation analysis, is more powerful for many (although not all) statistical tests, and when augmented with GEES, it allows for a detailed correlational analysis of which algorithms tend to have problems with the same songs as others and which have perhaps genuinely broken innovative ground. This is by no means to suggest that Friedman's test is a bad test in general – its near-universal applicability makes it an excellent choice in many circumstances, including many other MIREX evaluations – but for ACE, we believe that the extra understanding logistic regression can offer may help researchers predict which techniques are most promising for breaking the current performance plateau.

## 5. REFERENCES

- [1] A. Agresti. *Categorical Data Analysis*. Wiley, New York, 2nd edition, 2007.
- [2] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. Roy. Stat. Soc. B*, 1(57):289–300, 1995.
- [3] J. A. Burgoyne. *Stochastic Processes and Database-Driven Musicology*. PhD thesis, McGill U., Montréal, QC, 2012.
- [4] J. A. Burgoyne, J. Wild, and I. Fujinaga. An expert ground-truth set for audio chord recognition and music analysis. In *Proc. Int. Soc. Music Inf. Retr.*, pages 633–38, Miami, FL, 2011.
- [5] S. Ferrari and F. Cribari-Neto. Beta regression for modeling rates and proportions. *J. Appl. Stat.*, 31(7):799–815, 2004.
- [6] W. B. de Haas, J. P. Magalhães, D. ten Heggeler, G. Bekenkamp, and T. Ruizendaal. Chordify: Chord transcription for the masses. Demo at the Int. Soc. Music Inf. Retr. Conf., Curitiba, Brazil, 2012.
- [7] W. B. de Haas, J. P. Magalhães, R. C. Veltkamp, and F. Wiering. Harmtrace: Improving harmonic similarity estimation using functional harmony analysis. In *Proc. Int. Soc. Music Inf. Retr.*, pages 67–72, Miami, FL, 2011.
- [8] C. Harte. *Towards Automatic Extraction of Harmony Information from Music Signals*. PhD thesis, Queen Mary, U. London, 2010.
- [9] V. E. Johnson. Revised standards for statistical evidence. *P. Nat'l Acad. Sci. USA*, 110(48):19313–17, 2013.
- [10] M. Khadkevich and M. Omologo. Large-scale cover song identification using chord profiles. In *Proc. Int. Soc. Music Inf. Retr. Conf.*, pages 233–38, Curitiba, Brazil, 2013.
- [11] M. H. Kutner, C. J. Nachtsheim, J. Neter, and W. Li. *Applied Linear Statistical Models*. McGraw-Hill, Boston, MA, 5th edition, 2005.
- [12] M. Mauch. *Automatic Chord Transcription from Audio Using Computational Models of Musical Context*. PhD thesis, Queen Mary, U. London, 2010.
- [13] M. Mauch, S. Dixon, C. Harte, M. Casey, and B. Fields. Discovering chord idioms through Beatles and Real Book songs. In *Proc. Int. Soc. Music Inf. Retr. Conf.*, pages 255–58, Vienna, Austria, 2007.
- [14] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Chapman & Hall/CRC, Boca Raton, FL, 2nd edition, 1989.
- [15] J. Pauwels and G. Peeters. Evaluating automatically estimated chord sequences. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pages 749–53, Vancouver, British Columbia, 2013.
- [16] J. R. Taylor. *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. University Science Books, Sausalito, CA, 2nd edition, 1997.

# MERGED-OUTPUT HMM FOR PIANO FINGERING OF BOTH HANDS

**Eita Nakamura**

National Institute of Informatics  
Tokyo 101-8430, Japan  
eita.nakamura@gmail.com

**Nobutaka Ono**

National Institute of Informatics  
Tokyo 101-8430, Japan  
onono@nii.ac.jp

**Shigeki Sagayama**

Meiji University  
Tokyo 164-8525, Japan  
sagayama@meiji.ac.jp

## ABSTRACT

This paper discusses a piano fingering model for both hands and its applications. One of our motivations behind the study is automating piano reduction from ensemble scores. For this, quantifying the difficulty of piano performance is important where a fingering model of both hands should be relevant. Such a fingering model is proposed that is based on merged-output hidden Markov model and can be applied to scores in which the voice part for each hand is not indicated. The model is applied for decision of fingering for both hands and voice-part separation, automation of which is itself of great use and were previously difficult. A measure of difficulty of performance based on the fingering model is also proposed and yields reasonable results.

## 1. INTRODUCTION

Music arrangement is one of the most important musical activities, and its automation certainly has attractive applications. One common form is piano arrangement of ensemble scores, whose purposes are, among others, to enable pianists to enjoy a wider variety of pieces and to accompany other instruments by substituting the role of orchestra. While certain piano reductions have high technicality and musicality as in the examples by Liszt [8], those for vocal scores of operas and reduction scores of orchestra accompaniments are often faithful to the original scores in most parts. The most faithful reduction score is obtained by gathering every note in the original score, but the result can be too difficult to perform, and arrangement such as deleting notes is often in order.

In general, the difficulty of a reduction score can be reduced by arrangement, but then the fidelity also decreases. If one can quantify the performance difficulty and the fidelity to the original score, the problem of “minimal” piano reduction can be considered as an optimization problem of the fidelity given constraints on the performance difficulty. A method for guitar arrangement based on probabilistic model with a similar formalization is proposed in Ref. [5]. This paper is a step toward a realization of piano reduction algorithm based on the formalization.

The playability of piano passages is discussed in Refs. [3, 2] in connection with automatic piano arrangement. There, constraints such as the maximal number of notes in each hand, the maximal interval being played, say, 10th, and the minimal time interval of a repeated note are considered. Although these constraints are simple and effective to some extent, the actual situation is more complicated as manifested in the fact that, for example, the playability can change with tempos and players can arpeggiate chords that cannot be played simultaneously. In addition, the playability can depend on the technical level of players [3]. Given these problems, it seems appropriate to consider performance difficulty that takes values in a range.

There are various measures and causes of performance difficulty including player’s movements and notational complexity of the score [12, 1, 15]. Here we focus on the difficulty of player’s movements, particularly piano fingering, which is presumably one of the most important factors. The difficulty of fingering is closely related to the decision of fingering [4, 7, 13, 16]. Given the current situation that a method of determining the fingering costs from first principles is not established, however, it is also effective to take a statistical approach, and consider the naturalness of fingering in terms of probability obtained from actual fingering data. With a statistical model of fingering, the most natural fingering can be determined, and one can quantify the difficulty of fingering in terms of naturalness. This will be explained in Secs. 2 and 3. The practical importance of piano fingering and its applications are discussed in Ref. [17].

Since voice parts played by both hands are not a priori separated or indicated in the original ensemble score, a fingering model must be applicable in such a situation. Thus, a fingering model for both hands and an algorithm to separate voice parts are necessary. We propose such a model and an algorithm based on merged-output hidden Markov model (HMM), which is suited for modeling multi-voice-part structured phenomena [10, 11]. Since multi-voice-part structure of music is common and voice-part separation can be applied for a wide range of information processing, the results are itself of great importance.

## 2. MODEL FOR PIANO FINGERING FOR BOTH HANDS

### 2.1 Model for one hand

Before discussing the piano fingering model for both hands, let us discuss the fingering model for one hand. Piano



© Eita Nakamura, Nobutaka Ono, Shigeki Sagayama.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Eita Nakamura, Nobutaka Ono, Shigeki Sagayama. “Merged-Output HMM for Piano Fingering of Both Hands”, 15th International Society for Music Information Retrieval Conference, 2014.

fingering models and algorithms for decision of fingering have been studied in Refs. [13, 16, 4, 18, 19, 20, 7]. Here we extend the model in Ref. [19] to including chords.

Piano fingering for one hand, say, the right hand, is indicated by associating a finger number  $f_n = 1, \dots, 5$  (1 = thumb, 2 = the index finger,  $\dots$ , 5 = the little finger) to each note  $p_n$  in a score<sup>1</sup>, where  $n = 1, \dots, N$  indexes notes in the score and  $N$  is the number of notes. We consider the probability of a fingering sequence  $f_{1:N} = (f_n)_{n=1}^N$  given a score, or a pitch sequence,  $p_{1:N} = (p_n)_{n=1}^N$ , which is written as  $P(f_{1:N}|p_{1:N})$ . As explained in detail in Sec. 3.1, an algorithm for fingering decision can be obtained by estimating the most probable candidate  $\hat{f}_{1:N} = \operatorname{argmax}_{f_{1:N}} P(f_{1:N}|p_{1:N})$ . The fingering of a particular note

is more influenced by neighboring notes than notes that are far away in score position. Dependence on neighboring notes is most simply described by that on adjacent notes, and it can be incorporated with a Markov model. It also has advantages in efficiency in maximizing probability and setting model parameters. Although the probability of fingering may depend on inter-onset intervals between notes, the dependence is not considered here for simplicity.

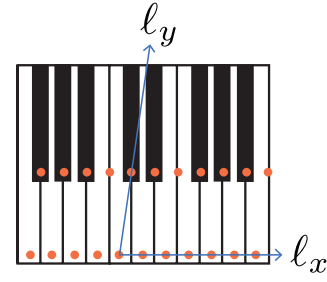
As proposed in Ref. [18, 19], the fingering model can be constructed with an HMM. Supposing that notes in score are generated by finger movements and the resulting performed pitches, their probability is represented with the probability that a finger would be used after another finger  $P(f_n|f_{n-1})$ , and the probability that a pitch would result from succeeding two used fingers. The former is called the transition probability, and the latter output probability. The output probability of pitch depends on the previous pitch in addition to the corresponding used fingers, and it is described with a conditional probability  $P(p_n|p_{n-1}, f_{n-1}, f_n)$ . In terms of these probabilities, the probability of notes and fingerings is given as

$$P(p_{1:N}, f_{1:N}) = \prod_{n=1}^N P(p_n|p_{n-1}, f_{n-1}, f_n)P(f_n|f_{n-1}), \quad (1)$$

where the initial probabilities are written as  $P(f_1|f_0) \equiv P(f_1)$  and  $P(p_1|p_0, f_0, f_1) \equiv P(p_1|f_1)$ . The probability  $P(f_{1:N}|p_{1:N})$  can also be given accordingly.

To train the model efficiently, we assume some reasonable constraints on the parameters. First we assume that the probability depends on pitches only through their geometrical positions on the keyboard which is represented as a two-dimensional lattice (Fig. 1). We also assume the translational symmetry in the  $x$ -direction and the time inversion symmetry for the output probability. If the coordinate on the keyboard is written as  $\ell(p) = (\ell_x(p), \ell_y(p))$ , the assumptions mean that the output probability has a form  $P(p'|p, f, f') = F(\ell_x(p') - \ell_x(p), \ell_y(p') - \ell_y(p); f, f')$ , and it satisfies  $F(\ell_x(p') - \ell_x(p), \ell_y(p') - \ell_y(p); f, f') = F(\ell_x(p) - \ell_x(p'), \ell_y(p) - \ell_y(p'); f', f)$ . A model for each hand can be obtained in this way, and it is written as  $F_\eta(\ell_x(p') - \ell_x(p), \ell_y(p') - \ell_y(p); f, f')$  with  $\eta = L, R$ .

<sup>1</sup> We do not consider the so-called finger substitution in this paper.



**Figure 1.** Keyboard lattice. Each key on a keyboard is represented by a point of a two-dimensional lattice.

It is further assumed that these probabilities are related by reflection in the  $x$ -direction, which yields  $F_L(\ell_x(p') - \ell_x(p), \ell_y(p') - \ell_y(p); f, f') = F_R(\ell_x(p') - \ell_x(p), \ell_y(p') - \ell_y(p); f, f')$ .

The above model can be extended to be applied for passages with chords, by converting a polyphonic passage to a monophonic passage by virtually arpeggiating the chords [7]. Here, notes in a chord are ordered from low pitch to high pitch. The parameter values can be obtained from fingering data.

## 2.2 Model for both hands

Now let us consider the fingering of both hands in the situation that it is unknown a priori which of the notes are to be played by the left or right hand. The problem can be stated as associating the fingering information  $(\eta_n, f_n)_{n=1}^N$  for the pitch sequence  $p_{1:N}$ , where  $\eta_n = L, R$  indicates the hand with which the  $n$ -th note is played.

One might think to build a model of both hands by simply extending the one-hand model and using  $(\eta_n, f_n)$  as a latent variable. However, this is not an effective model as far as it is a first-order Markov model since, for example, probabilistic constraints between two successive notes by the right hand cannot be directly incorporated when they are interrupted by other notes of the left hand. Using higher-order Markov models leads to the problem of increasing number of parameters that is hard to train as well as the increasing computational cost. The underlying problem is that the model cannot capture the structure of dependencies that is stronger among notes in each hand than those across hands.

Recently an HMM, called merged-output HMM, is proposed that is suited for describing such voice-part-structured phenomena [10, 11]. The basic idea is to construct a model for both hands by starting with two parallel HMMs, called part HMMs, each of which corresponds to the HMM for fingering of each hand, and then merging the outputs of the part HMMs. Assuming that only one of the part HMMs transits and outputs an observed symbol at each time, the state space of the merged-output HMM is given as a triplet  $k = (\eta, f_L, f_R)$  of the hand information  $\eta = L, R$  and fingerings of both hands:  $\eta$  indicate which of the HMMs transits, and  $f_L$  and  $f_R$  indicate the current states of the part HMMs. Let the transition and output probabilities

of the part HMMs be  $a_{ff'}^\eta = P_\eta(f'|f)$  and  $b_{ff'}^\eta(\ell) = F_\eta(\ell; f, f')$  ( $\eta = L, R$ ). Then the transition and output probabilities of the merged-output HMM are given as

$$a_{kk'} = \begin{cases} \alpha_L a_{f_L f'_L}^L \delta_{f_R f'_R}, & \eta' = L; \\ \alpha_R a_{f_R f'_R}^R \delta_{f_L f'_L}, & \eta' = R, \end{cases} \quad (2)$$

$$b_{kk'}(\ell) = \begin{cases} b_{f_L f'_L}^L(\ell), & \eta' = L; \\ b_{f_R f'_R}^R(\ell), & \eta' = R, \end{cases} \quad (3)$$

where  $\delta$  denotes Kronecker's delta. Here,  $\alpha_{L,R}$  represent the probability of choosing which of the hands to play the note, and practically, they satisfy  $\alpha_L \sim \alpha_R \sim 1/2$ . As shown in Ref. [11], certain interaction factors can be introduced to Eqs. (2) and (3). Although such interactions may be important in the future [14], we confine ourselves to the case of no interactions in this paper for simplicity.

By estimating the most probable sequence  $\hat{k}_{1:N}$ , both the optimal configuration of hands  $\hat{\eta}_{1:N}$ , which yields a voice-part separation, and that of fingers  $(\hat{f}_L, \hat{f}_R)_{1:N}$  are obtained. For details of inference algorithms and other aspects of merged-output HMM, see Ref. [11].

### 2.3 Model for voice-part separation

The model explained in the previous section involves both hands and the used hand and fingers are modeled simultaneously. We can alternatively consider the problem of associating fingerings of both hands as first separating voice parts for both hands, and then associating fingerings for notes in each voice part. In this subsection, a simple model that can be used for voice-part separation is given. The model is also based on a simpler merged-output HMM, and it yields more efficient algorithm for voice-part separation.

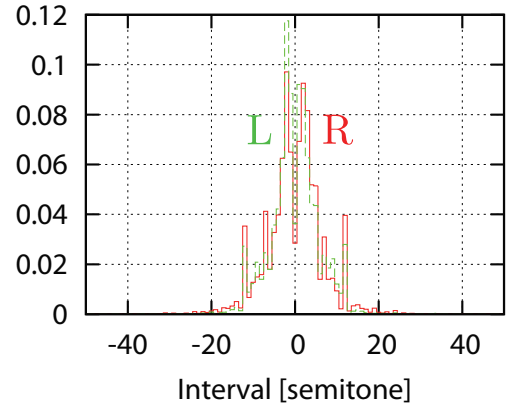
We consider a merged-output HMM with a hidden state  $x = (\eta, p_L, p_R)$ , where  $\eta = L, R$  indicates the voice part, and  $p_{L,R}$  describes the pitch played in each voice part. If the pitch sequence in the score is denoted by  $(y_n)_n$ , the transition and output probabilities are written as

$$a_{xx'} = \begin{cases} \alpha_L a_{p_L p'_L}^L \delta_{p_R p'_R}, & \eta' = L; \\ \alpha_R a_{p_R p'_R}^R \delta_{p_L p'_L}, & \eta' = R, \end{cases} \quad (4)$$

$$b_x(y) = \delta_{y, p_\eta}. \quad (5)$$

Here the transition probability  $a_{pp'}^{L,R}$  describes the pitch sequence in each voice part directly, without any information on fingerings. The corresponding distributions can be obtained from actual data of piano pieces, as shown in Fig. 2.

So far we have considered a model of pitches and horizontal intervals for voice-part separation. The voice-part-separation algorithm can be derived by applying the Viterbi algorithm to the above model. In fact, a voice part in the score played by one hand is also constrained by vertical intervals since it is physically difficult to play a chord containing an interval far wider than an octave by one hand. The constraint on the vertical intervals can also be introduced in terms of probability.



**Figure 2.** Histograms of pitch transitions in piano scores for each hand.

## 3. APPLICATIONS OF THE FINGERING MODEL

### 3.1 Algorithm for decision of fingering

A direct application of the model explained in Secs. 2.1 and 2.2 is the decision of fingering. The algorithm can be derived by applying the Viterbi algorithm. For one hand, the derived algorithm is similar as the one in Ref. [19], but we reevaluated the accuracy since the present model can be applied for polyphonic passages and the details of the models are different.

For evaluation, we prepared manually labeled fingerings of classical piano pieces and compared them to the one estimated with the algorithm. The test pieces were Nos. 1, 2, 3, and 8 of Bach's two-voice inventions, and the introduction and exposition parts from Beethoven's 8th piano sonata in C minor. The training and test of the algorithm was done with the leave-one-out cross validation method for each piece. To avoid zero frequencies in the training, we added a uniform count of 0.1 for every bin.

The averaged accuracy was 56.0% (resp. 55.4%) for the right (resp. left) hand where the number of notes was 5202 (resp. 5539). Since the training data was not big, and we had much higher rate of more than 70% for closed test, the accuracy may improve if a larger set of training data is given. The results were better than the reported values in Ref. [19]. The reason would be that the constraints of the model in the reference was too strong, which is relaxed in the present model. For detailed analysis of the estimation errors, see Ref. [19].

### 3.2 Voice-part separation

Voice-part separation between two hands can be done with the model described in Sec. 2.3, and the algorithm can be obtained by the Viterbi algorithm. In fact, we can derive a more efficient estimation algorithm which is effectively equivalent since the model has noiseless observations as in Eq. (5).

It is obtained by minimizing the following potential with respect to the variables  $\{(\eta_n, h_n)\}$ ,  $h_n = 0, 1, \dots, N_h$  for

**Table 1.** Error rates of the voice-part-separation algorithms. The 0-HMM (resp. 1-HMM, 2-HMM) indicates the algorithm with the zeroth-order (resp. first-order, second-order) HMM.

Pieces	# Notes	0-HMM [%]	1-HMM [%]	2-HMM [%]	Merged-output HMM [%]
Bach (15 pcs)	9638	5.1	5.3	6.1	1.9
Beethoven (2 pcs)	18144	13.0	11.1	11.5	9.28
Chopin (5 pcs)	8508	5.7	4.0	4.29	3.8
Debussy (3 pcs)	3360	17.8	14.8	14.8	18.7
Total	39650	9.9	8.5	8.9	7.1

each note:

$$V(\boldsymbol{\eta}, \mathbf{h}) = - \sum_n \ln Q(\eta_{n-1}, h_{n-1}; \eta_n, h_n), \quad (6)$$

$$Q(\eta_{n-1}, h_{n-1}; \eta_n, h_n) = \begin{cases} \alpha_{\eta_n} a_{y_{n-1}, y_n}^{(\eta_n)} \delta_{h_n, h_{n-1}+1}, & \eta_n = \eta_{n-1}; \\ \alpha_{\eta_n} a_{y_{n-2-h_{n-1}}, y_n}^{(\eta_n)} \delta_{h_n, 0}, & \eta_n \neq \eta_{n-1}. \end{cases} \quad (7)$$

Here  $h_n$  is necessary to memorize the current state of the voice part opposite of  $\eta_n$ . The minimization of the potential can be done with dynamic programming incrementally for each  $n$ . The estimation result is the same as the one with the Viterbi algorithm applied to the model when  $N_h$  is sufficiently large, and we confirmed that  $N_h = 50$  is sufficient to provide a good approximation.

The algorithm was evaluated by applying it to several classical piano pieces. The used pieces were all pieces of Bach's two-voice inventions, the first two piano sonatas by Beethoven, Chopin's Etude Op. 10 Nos. 1–5, and the first three pieces in the first book of Debussy's Préludes. For comparison, we also evaluated algorithms based on lower-order HMMs. The zeroth-order model with transition and output probabilities  $P(\eta)$  and  $P(p|\eta)$  is almost equivalent to the keyboard splitting method, the first-order model with  $P(\eta'|\eta)$  and  $P(\delta p|\eta, \eta')$  and the second-order model are simple applications of HMMs whose latent variables are hand informations  $\eta = L, R$ .

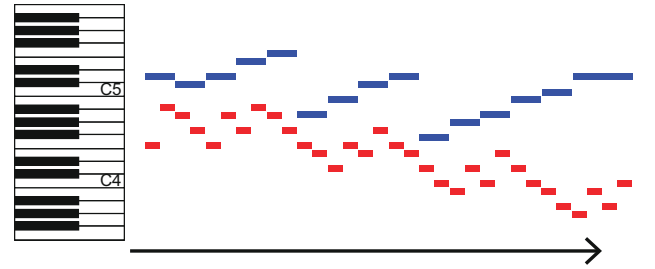
The results are shown in Table 1. In total, the merged-output HMM yielded the lowest error rate, with which relatively accurate voice part separation can be done. On the other hand, there were less changes in results for the lower-order HMMs, showing that the effectiveness of the merged-output HMM. In Debussy's pieces, the error rates were relatively high since the pieces necessitate complex fingerings with wide movements of the hands. An example of the voice-part separation result is shown in Fig. 3.

### 3.3 Quantitative measure of difficulty of performance

A measure of performance difficulty based on the naturalness of the fingerings can be obtained by the probabilistic fingering model. Although global structures in scores may influence the difficulty, we concentrate on the effect of local structures. It is supposed that the difficulty is additive with regard to performed notes and an increasing function of tempo. A quantity satisfying these conditions is the time rate of probabilistic cost. Let  $\mathbf{p}(t)$  denote the sequence of



(a) Passage in Bach's two-voice invention No. 1.



(b) Piano role representation of the voice-part separation result. Two voice parts are colored red and blue.

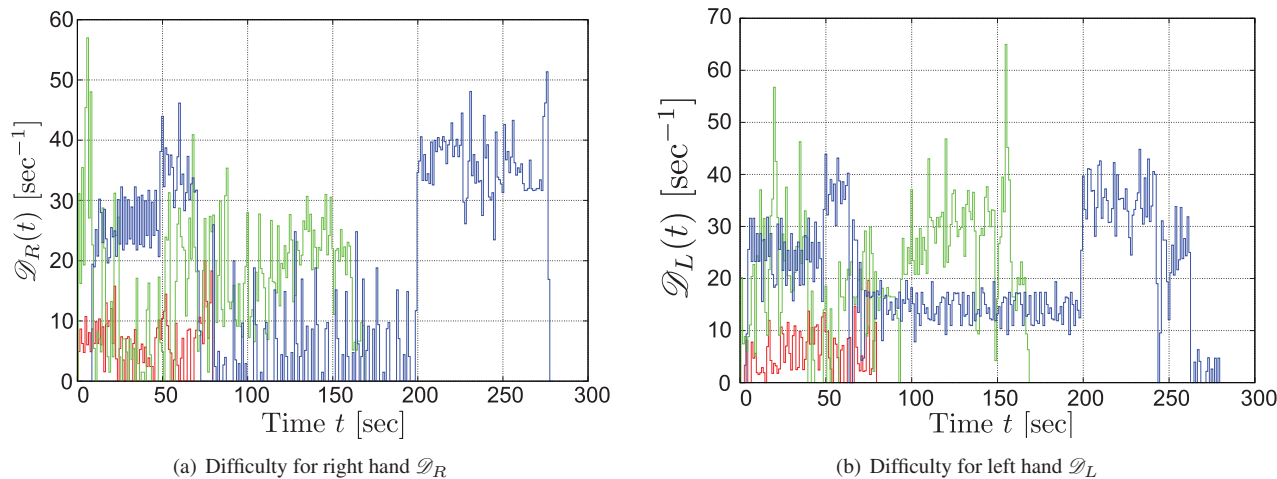
**Figure 3.** Example of a voice-part separation result.

notes in the time range of  $[t - \Delta t/2, t + \Delta t/2]$ , and  $\mathbf{f}(t)$  be the corresponding fingerings, where  $\Delta t$  is a width of the time range to define the time rate. Then it is given as

$$\mathcal{D}(t) = - \ln P(\mathbf{p}(t), \mathbf{f}(t)) / \Delta t. \quad (8)$$

Since the minimal time interval of successive notes are about a few 10 milli seconds and it is hard to imagine that difficulty is strongly influenced by notes that are separated more than 10 seconds, it is natural to set  $\Delta t$  within these extremes. The right-hand side is given by Eq. (1). It is possible to calculate  $\mathcal{D}(t)$  for a score without indicated fingerings by replacing  $\mathbf{f}(t)$  with the estimated fingerings  $\hat{\mathbf{f}}(t)$  with the model in Sec. 2. In addition to the difficulty for both hands, that for each hand  $\mathcal{D}_{L,R}(t)$  can also be defined similarly.

Fig. 4 shows some examples of  $\mathcal{D}_{L,R}(t)$  calculated for several piano pieces. Here  $\Delta t$  was set to 1 sec. Although it is not easy to evaluate the quantity in a strict way, the results seems reasonable and reflects generic intuition of difficulty. The invention by Bach that can be played by beginners yields  $\mathcal{D}_{L,R}$  that are less than about 10, the example of Beethoven's sonata which requires middle-level technicality has  $\mathcal{D}_{L,R}$  around 20 to 30, and Chopin's Fantasia Impromptu which involves fast passages and difficult fingerings has  $\mathcal{D}_{L,R}$  up to about 40. It is also worthy of noting that relatively difficult passages such as the fast chromatique passage of the right hand in the introduction of Beethoven's sonata and ornaments in the right hand of the



**Figure 4.** Examples of  $\mathcal{D}_R$  and  $\mathcal{D}_L$ . The red (resp. green, blue) line is for Bach’s two-voice invention No.=1, (resp. Introduction and exposition parts of the first movement of Beethoven’s eighth piano sonata, Chopin’s Fantasie Impromptu).

slow part of the Fantasie Impromptu are also captured in terms of  $\mathcal{D}_R$ .

#### 4. CONCLUSIONS

In this paper, we considered a piano fingering model of both hands and its applications especially toward a piano reduction algorithm. First we reviewed a piano fingering model for one hand based on HMM, and then constructed a model for both hands based on merged-output HMM. Next we applied the model for constructing an algorithm for fingering decision and voice-part-separation algorithm and obtained a measure of performance difficulty. The algorithm for fingering decision yielded better results than the previously proposed one by a modification in details of the model. The results of voice-part separation is quite good and encouraging. The proposed measure of performance difficulty successfully captures the dependence on tempos and complexity of pitches and finger movements.

The next step to construct a piano reduction algorithm according to the formalization mentioned in the Introduction is to quantify the fidelity of the arranged score to the original score and to integrate it with the constraints of performance difficulty. The fidelity can be described with edit probability, similarly as in Ref. [5], and an arrangement model can be obtained by integrating the fingering model with the edit probability. We are currently working on these issues and the results will be reported elsewhere.

#### 5. ACKNOWLEDGMENTS

This work is supported in part by Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science, No. 23240021, No. 26240025 (S.S. and N.O.), and No. 25880029 (E.N.).

#### 6. REFERENCES

- [1] S.-C. Chiu and M.-S. Chen, “A study on difficulty level recognition of piano sheet music,” *Proc. AdMIRe*, pp. 10–12, 2012.
- [2] S.-C. Chiu *et al.*, “Automatic system for the arrangement of piano reductions,” *Proc. AdMIRe*, 2009.
- [3] K. Fujita *et al.*, “A proposal for piano score generation that considers proficiency from multiple part (in Japanese),” *Tech. Rep. SIGMUS*, MUS-77, pp. 47–52, 2008.
- [4] M. Hart and E. Tsai, “Finding optimal piano fingerings,” *The UMAP Journal*, **21**(1), pp. 167–177, 2000.
- [5] G. Hori *et al.*, “Input-output HMM applied to automatic arrangement for guitars,” *J. Information Processing*, **21**(2), pp. 264–271, 2013.
- [6] Z. Ghahramani and M. Jordan, “Factorial Hidden Markov Models,” *Machine Learning*, **29**, pp. 245–273, 1997.
- [7] A. Al Kasimi *et al.*, “A simple algorithm for automatic generation of polyphonic piano fingerings,” *ISMIR*, pp. 355–356, 2007.
- [8] F. Liszt, *Musikalische Werke*, Serie IV, Breitkopf & Härtel, 1922.
- [9] J. Musafia, *The Art of Fingering in Piano Playing*, MCA Music, 1971.
- [10] E. Nakamura *et al.*, “Merged-output hidden Markov model and its applications to score following and hand separation of polyphonic keyboard music (in Japanese),” *Tech. Rep. SIGMUS*, 2013-EC-27, **15**, 2013.
- [11] E. Nakamura *et al.*, “Merged-output hidden Markov model for score following of MIDI performance with ornaments, desynchronized voices, repeats and skips,” to appear in *Proc. ICMC*, 2014.
- [12] C. Palmer, “Music performance,” *Ann. Rev. Psychol.*, **48**, pp. 115–138, 1997.

- [13] R. Parncutt *et al.*, “An ergonomic model of keyboard fingering for melodic fragments,” *Music Perception*, **14(4)**, pp. 341–382, 1997.
- [14] R. Parncutt *et al.*, “Interdependence of right and left hands in sight-read, written, and rehearsed fingerings of parallel melodic piano music,” *Australian J. of Psychology*, **51(3)**, pp. 204–210, 1999.
- [15] V. Sébastien *et al.*, “Score analyzer: Automatically determining scores difficulty level for instrumental e-learning,” *Proc. ISMIR*, 2012.
- [16] H. Sekiguchi and S. Eiho, “Generating and displaying the human piano performance,” **40(6)**, pp. 167–177, 1999.
- [17] Y. Takegawa *et al.*, “Design and implementation of a real-time fingering detection system for piano performance,” *Proc. ICMC*, pp. 67–74, 2006.
- [18] Y. Yonebayashi *et al.*, “Automatic determination of piano fingering based on hidden Markov model (in Japanese),” *Tech. Rep. SIGMUS*, 2006-05-13, pp. 7–12, 2006.
- [19] Y. Yonebayashi *et al.*, “Automatic decision of piano fingering based on hidden Markov models,” *IJCAI*, pp. 2915–2921, 2007.
- [20] Y. Yonebayashi *et al.*, “Automatic piano fingering decision based on hidden Markov models with latent variables in consideration of natural hand motions (in Japanese),” *Tech. Rep. SIGMUS*, MUS-71-29, pp. 179–184, 2007.



# MODELING RHYTHM SIMILARITY FOR ELECTRONIC DANCE MUSIC

**Maria Panteli**

University of Amsterdam,  
Amsterdam, Netherlands

m.x.panteli@gmail.com

**Niels Bogaards**

Elephantcandy,  
Amsterdam, Netherlands

niels@elephantcandy.com

**Aline Honingh**

University of Amsterdam,  
Amsterdam, Netherlands

a.k.honingh@uva.nl



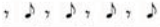

## ABSTRACT

A model for rhythm similarity in electronic dance music (EDM) is presented in this paper. Rhythm in EDM is built on the concept of a ‘loop’, a repeating sequence typically associated with a four-measure percussive pattern. The presented model calculates rhythm similarity between segments of EDM in the following steps. 1) Each segment is split in different perceptual rhythmic streams. 2) Each stream is characterized by a number of attributes, most notably: attack phase of onsets, periodicity of rhythmic elements, and metrical distribution. 3) These attributes are combined into one feature vector for every segment, after which the similarity between segments can be calculated. The stages of stream splitting, onset detection and downbeat detection have been evaluated individually, and a listening experiment was conducted to evaluate the overall performance of the model with perceptual ratings of rhythm similarity.

## 1. INTRODUCTION

Music similarity has attracted research from multidisciplinary domains including tasks of music information retrieval and music perception and cognition. Especially for rhythm, studies exist on identifying and quantifying rhythm properties [16, 18], as well as establishing rhythm similarity metrics [12]. In this paper, rhythm similarity is studied with a focus on Electronic Dance Music (EDM), a genre with various and distinct rhythms [2].

EDM is an umbrella term consisting of the ‘four on the floor’ genres such as techno, house, trance, and the ‘breakbeat-driven’ genres such as jungle, drum ‘n’ bass, breaks etc. In general, four on the floor genres are characterized by a four-beat steady bass-drum pattern whereas breakbeat-driven exploit irregularity by emphasizing the metrically weak locations [2]. However, rhythm in EDM exhibits multiple types of subtle variations and embellishments. The goal of the present study is to develop a rhythm similarity model that captures these embellishments and allows for a fine inter-song rhythm similarity.

Rhythm in Musical Notation	Attack Positions of Rhythm	Most Common Instrumental Associations
	1/5/9/13	Bass drum
	5/13	Snare drum; handclaps
	3/7/11/15	Hi-hat (open or closed); also snare drum or synth "stabs"
	All	Hi-hat (closed)

**Figure 1:** Example of a common (even) EDM rhythm [2].

The model focuses on content-based analysis of audio recordings. A large and diverse literature deals with the challenges of audio rhythm similarity. These include, amongst other, approaches to onset detection [1], tempo estimation [9, 25], rhythmic representations [15, 24], and feature extraction for automatic rhythmic pattern description and genre classification [5, 12, 20]. Specific to EDM, [4] study rhythmic and timbre features for automatic genre classification, and [6] investigate temporal and structural features for music generation.

In this paper, an algorithm for rhythm similarity based on EDM characteristics and perceptual rhythm attributes is presented. The methodology for extracting rhythmic elements from an audio segment and a summary of the features extracted is provided. The steps of the algorithm are evaluated individually. Similarity predictions of the model are compared to perceptual ratings and further considerations are discussed.

## 2. METHODOLOGY

Structural changes in an EDM track typically consist of an evolution of timbre and rhythm as opposed to a verse-chorus division. Segmentation is firstly performed to split the signal into meaningful excerpts. The algorithm developed in [21] is used, which segments the audio signal based on timbre features (since timbre is important in EDM structure [2]) and musical heuristics.

EDM rhythm is expressed via the ‘loop’, a repeating pattern associated with a particular (often percussive) instrument or instruments [2]. Rhythm information can be extracted by evaluating characteristics of the loop: First, the rhythmic pattern is often presented as a combination of instrument sounds (eg. Figure 1), thus exhibiting a certain ‘rhythm polyphony’ [3]. To analyze this, the signal is split into the so-called rhythmic streams. Then, to describe the underlying rhythm, features are extracted for each stream based on three attributes: a) The attack phase of the onsets is considered to describe if the pattern is performed on



© Maria Panteli, Niels Bogaards, Aline Honingh.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Maria Panteli, Niels Bogaards, Aline Honingh. “Modeling rhythm similarity for electronic dance music”, 15th International Society for Music Information Retrieval Conference, 2014.

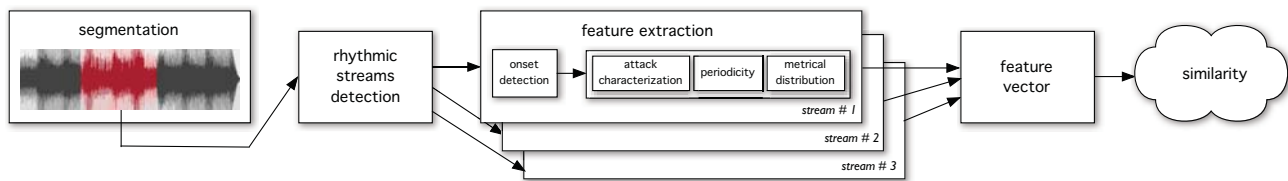


Figure 2: Overview of methodology.

percussive or non-percussive instruments. Although this is typically viewed as a timbre attribute, the percussiveness of a sound is expected to influence the perception of rhythm [16]. b) The repetition of rhythmic sequences of the pattern are described by evaluating characteristics of different levels of onsets' periodicity. c) The metrical structure of the pattern is characterized via features extracted from the metrical profile [24] of onsets. Based on the above, a feature vector is extracted for each segment and is used to measure rhythm similarity. Inter-segment similarity is evaluated with perceptual ratings collected via a specifically designed experiment. An overview of the methodology is shown in Figure 2 and details for each step are provided in the sections below. Part of the algorithm is implemented using the MIRTtoolbox [17].

## 2.1 Rhythmic Streams

Several instruments contribute to the rhythmic pattern of an EDM track. Most typical examples include combinations of bass drum, snare and hi-hat (eg. Figure 1). This is mainly a functional rather than a strictly instrumental division, and in EDM one finds various instrument sounds to take the role of bass, snare and hi-hat. In describing rhythm, it is essential to distinguish between these sources since each contributes differently to rhythm perception [11].

Following this, [15, 24] describe rhythmic patterns of latin dance music in two prefixed frequency bands (low and high frequencies), and [9] represents drum patterns as two components, the bass and snare drum pattern, calculated via non-negative matrix factorization of the spectrogram. In [20], rhythmic events are split based on their perceived loudness and brightness, where the latter is defined as a function of the spectral centroid.

In the current study, rhythmic streams are extracted with respect to the frequency domain and loudness pattern. In particular, the Short Time Fourier Transform of the signal is computed and logarithmic magnitude spectra are assigned to bark bands, resulting into a total of 24 bands for a 44.1 kHz sampling rate. Synchronous masking is modeled using the spreading function of [23], and temporal masking is modeled with a smoothing window of 50 ms. This representation is hereafter referred to as loudness envelope and denoted by  $L_b$  for bark bands  $b = 1, \dots, 24$ . A self-similarity matrix is computed from this 24-band representation indicating the bands that exhibit similar loudness pattern. The novelty approach of [8] is applied to the  $24 \times 24$  similarity matrix to detect adjacent bands that should be grouped to the same rhythmic stream. The peak

locations  $P$  of the novelty curve define the number of the bark band that marks the beginning of a new stream, i.e., if  $P = \{p_i \in \{1, \dots, 24\} | i = 1, \dots, I\}$  for total number of peaks  $I$ , then stream  $S_i$  consists of bark bands  $b$  given by,

$$S_i = \begin{cases} \{b | b \in [p_i, p_{i+1} - 1]\} & \text{for } i = 1, \dots, I - 1 \\ \{b | b \in [p_I, 24]\} & \text{for } i = I. \end{cases} \quad (1)$$

An upper limit of 6 streams is considered based on the approach of [22] that uses a total of 6 bands for onset detection and [14] that suggests a total of three or four bands for meter analysis.

The notion of rhythmic stream here is similar to the notion of 'accent band' in [14] with the difference that each rhythmic stream is formed on a variable number of adjacent bark bands. Detecting a rhythmic stream does not necessarily imply separating the instruments, since if two instruments play the same rhythm they should be grouped to the same rhythmic stream. The proposed approach does not distinguish instruments that lie in the same bark band. The advantage is that the number of streams and the frequency range for each stream do not need to be predetermined but are rather estimated from the spectral representation of each song. This benefits the analysis of electronic dance music by not imposing any constraints on the possible instrument sounds that contribute to the characteristic rhythmic pattern.

### 2.1.1 Onset Detection

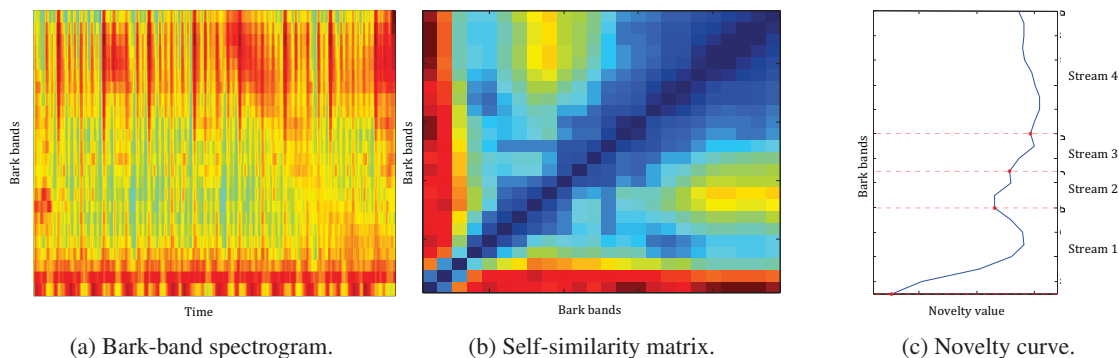
To extract onset candidates, the loudness envelope per bark band and its derivative are normalized and summed with more weight on loudness than its derivative, i.e.,

$$O_b(n) = (1 - \lambda)N_b(n) + \lambda N'_b(n) \quad (2)$$

where  $N_b$  is the normalized loudness envelope  $L_b$ ,  $N'_b$  the normalized derivative of  $L_b$ ,  $n = 1, \dots, N$  the frame number for a total of  $N$  frames, and  $\lambda < 0.5$  the weighting factor. This is similar to the approach described by Equation 3 in [14] with reduced  $\lambda$ , and is computed prior summation to the different streams as suggested in [14, 22]. Onsets are detected via peak extraction within each stream, where the (rhythmic) content of stream  $i$  is defined as

$$R_i = \sum_{b \in S_i} O_b \quad (3)$$

with  $S_i$  as in Equation 1 and  $O_b$  as in Equation 2. This onset detection approach incorporates similar methodological concepts with the positively evaluated algorithms for the task of audio onset detection [1] in MIREX 2012, and tempo estimation [14] in the review of [25].



**Figure 3:** Detection of rhythmic streams using the novelty approach; first a bark-band spectrogram is computed, then its self-similarity matrix, and then the novelty [7] is applied where the novelty peaks define the stream boundaries.

## 2.2 Feature Extraction

The onsets in each stream represent the rhythmic elements of the signal. To model the underlying rhythm, features are extracted from each stream, based on three attributes, namely, characterization of attack, periodicity, and metrical distribution of onsets. These are combined to a feature vector that serves for measuring inter-segment similarity. The sections below describe the feature extraction process in detail.

### 2.2.1 Attack Characterization

To distinguish between percussive and non-percussive patterns, features are extracted that characterize the attack phase of the onsets. In particular, the attack time and attack slope are considered, among other, essential in modeling the perceived attack time [10]. The attack slope was also used in modeling pulse clarity [16]. In general, onsets from percussive sounds have a short attack time and steep attack slope, whereas non-percussive sounds have longer attack time and gradually increasing attack slope.

For all onsets in all streams, the attack time and attack slope is extracted and split in two clusters; the ‘slow’ (non-percussive) and ‘fast’ (percussive) attack phase onsets. Here, it is assumed that both percussive and non-percussive onsets can be present in a given segment, hence splitting in two clusters is superior to, e.g., computing the average. The mean and standard deviation of the two clusters of the attack time and attack slope (a total of 8 features) is output to the feature vector.

### 2.2.2 Periodicity

One of the most characteristic style elements in the musical structure of EDM is repetition; the loop, and consequently the rhythmic sequence(s), are repeating patterns. To analyze this, the periodicity of the onset detection function per stream is computed via autocorrelation and summed across all streams. The maximum delay taken into account is proportional to the bar duration. This is calculated assuming a steady tempo and  $\frac{4}{4}$  meter throughout the EDM track [2]. The tempo estimation algorithm of [21] is used.

From the autocorrelation curve (cf. Figure 4), a total of 5 features are extracted:

**Lag duration of maximum autocorrelation:** The location (in time) of the second highest peak (the first being at lag 0) of the autocorrelation curve normalized by the bar duration. It measures whether the strongest periodicity occurs in every bar (i.e. feature value = 1), or every half bar (i.e. feature value = 0.5) etc.

**Amplitude of maximum autocorrelation:** The amplitude of the second highest peak of the autocorrelation curve normalized by the amplitude of the peak at lag 0. It measures whether the pattern is repeated in exactly the same way (i.e. feature value = 1) or somewhat in a similar way (i.e. feature value < 1) etc.

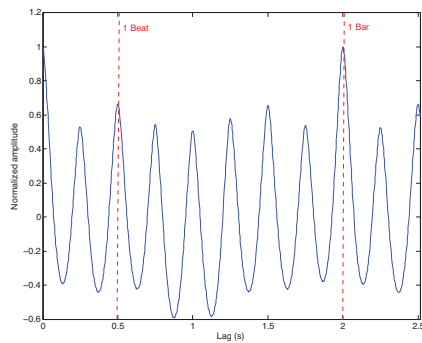
**Harmonicity of peaks:** This is the harmonicity as defined in [16] with adaptation to the reference lag  $l_0$  corresponding to the beat duration and additional weighting of the harmonicity value by the total number of peaks of the autocorrelation curve. This feature measures whether rhythmic periodicities occur in harmonic relation to the beat (i.e. feature value = 1) or inharmonic (i.e. feature value = 0).

**Flatness:** Measures whether the autocorrelation curve is smooth or spiky and is suitable for distinguishing between periodic patterns (i.e. feature value = 0), and non-periodic (i.e. feature value = 1).

**Entropy:** Another measure of the ‘peakiness’ of autocorrelation [16], suitable for distinguishing between ‘clear’ repetitions (i.e. distribution with narrow peaks and hence feature value close to 0) and unclear repetitions (i.e. wide peaks and hence feature value increased).

### 2.2.3 Metrical Distribution

To model the metrical aspects of the rhythmic pattern, the metrical profile [24] is extracted. For this, the downbeat is detected as described in Section 2.2.4, onsets per stream are quantized assuming a  $\frac{4}{4}$  meter and 16-th note resolution [2], and the pattern is collapsed to a total of 4 bars. The latter is in agreement with the length of a musical phrase in EDM being usually in multiples of 4, i.e., 4-bar, 8-bar, or 16-bar phrase [2]. The metrical profile of a given stream is thus presented as a vector of 64 bins (4 bars  $\times$  4 beats  $\times$  4 sixteenth notes per beat) with real values ranging between 0 (no onset) to 1 (maximum onset strength) as shown in Figure 5. For each rhythmic stream, a metrical pro-



**Figure 4:** Autocorrelation of onsets indicating high periodicities of 1 bar and 1 beat duration.

	Bar 1	Bar 2
	J J J J	J J J J
Stream 1	1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0	1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0
Stream 2	0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0	0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0
Stream 3	0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0	0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0
Stream 4	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

**Figure 5:** Metrical profile of the rhythm in Figure 1 assuming for simplicity a 2-bar length and constant amplitude.

file is computed and the following features are extracted. Features are computed per stream and averaged across all streams.

**Syncopation:** Measures the strength of the events lying on the weak locations of the meter. The syncopation model of [18] is used with adaptation to account for the amplitude (onset strength) of the syncopated note. Three measures of syncopation are considered that apply hierarchical weights with, respectively, sixteenth note, eighth note, and quarter note resolution.

**Symmetry:** Denotes the ratio of the number of onsets in the second half of the pattern that appear in exactly the same position in the first half of the pattern [6].

**Density:** Is the ratio of the number of onsets over the possible total number of onsets of the pattern (in this case 64).

**Fullness:** Measures the onsets' strength of the pattern. It describes the ratio of the sum of onsets' strength over the maximum strength multiplied by the possible total number of onsets (in this case 64).

**Centre of Gravity:** Denotes the position in the pattern where the most and strongest onsets occur (i.e., indicates whether most onsets appear at the beginning or at the end of the pattern etc.).

Aside from these features, the metrical profile (cf. Figure 5) is also added to the final feature vector. This was found to improve results in [24]. In the current approach, the metrical profile is provided per stream, restricted to a total of 4 streams, and output in the final feature vector in order of low to high frequency content streams.

#### 2.2.4 Downbeat Detection

The downbeat detection algorithm uses information from the metrical structure and musical heuristics. Two assump-

tions are made:

**Assumption 1:** Strong beats of the meter are more likely to be emphasized across all rhythmic streams.

**Assumption 2:** The downbeat is often introduced by an instrument in the low frequencies, i.e. a bass or a kick drum [2, 13].

Considering the above, the onsets per stream are quantized assuming a  $\frac{4}{4}$  meter, 16-th note resolution, and a set of downbeat candidates (in this case the onsets that lie within one bar length counting from the beginning of the segment). For each downbeat candidate, hierarchical weights [18] that emphasize the strong beats of the meter as indicated by Assumption 1, are applied to the quantized patterns. Note, there is one pattern for each rhythmic stream. The patterns are then summed by applying more weight to the pattern of the low-frequency stream as indicated by Assumption 2. Finally, the candidate whose quantized pattern was weighted most, is chosen as the downbeat.

### 3. EVALUATION

One of the greatest challenges of music similarity evaluation is the definition of a ground truth. In some cases, objective evaluation is possible, where a ground truth is defined on a quantifiable criterion, i.e., rhythms from a particular genre are similar [5]. In other cases, music similarity is considered to be influenced by the perception of the listener and hence subjective evaluation is more suitable [19]. Objective evaluation in the current study is not preferable since different rhythms do not necessarily conform to different genres or subgenres<sup>1</sup>. Therefore a subjective evaluation is used where predictions of rhythm similarity are compared to perceptual ratings collected via a listening experiment (cf. Section 3.4). Details of the evaluation of rhythmic stream, onset, and downbeat detection are provided in Sections 3.1 - 3.3. A subset of the annotations used in the evaluation of the latter is available online<sup>2</sup>.

#### 3.1 Rhythmic Streams Evaluation

The number of streams is evaluated with perceptual annotations. For this, a subset of 120 songs from a total of 60 artists (2 songs per artist) from a variety of EDM genres and subgenres was selected. For each song, segmentation was applied using the algorithm of [21] and a characteristic segment was selected. Four subjects were asked to evaluate the number of rhythmic streams they perceive in each segment, choosing between 1 to 6, where rhythmic stream was defined as a stream of unique rhythm.

For 106 of the 120 segments, the subjects' responses' standard deviation was significantly small. The estimated number of rhythmic streams matched the mean of the subject's response distribution with an accuracy of 93%.

<sup>1</sup> Although some rhythmic patterns are characteristic to an EDM genre or subgenre, it is not generally true that these are unique and invariant.

<sup>2</sup> [https://staff.fnwi.uva.nl/a.k.honingh/rhythm\\_similarity.html](https://staff.fnwi.uva.nl/a.k.honingh/rhythm_similarity.html)

### 3.2 Onset Detection Evaluation

Onset detection is evaluated with a set of 25 MIDI and corresponding audio excerpts, specifically created for this purpose. In this approach, onsets are detected per stream, therefore onset annotations should also be provided per stream. For a number of different EDM rhythms, MIDI files were created with the constraint that each MIDI instrument performs a unique rhythmic pattern therefore represents a unique stream, and were converted to audio.

The onsets estimated from the audio were compared to the annotations of the MIDI file using the evaluation measures of the MIREX Onset Detection task<sup>3</sup>. For this, no stream alignment is performed but rather onsets from all streams are grouped to a single set. For 25 excerpts, an  $F$ -measure of 85%, precision of 85%, and recall of 86% are obtained with a tolerance window of 50 ms. Inaccuracies in onset detection are due (on average) to doubled than merged onsets, because usually more streams (and hence more onsets) are detected.

### 3.3 Downbeat Detection Evaluation

To evaluate the downbeat the subset of 120 segments described in Section 3.1 was used. For each segment the annotated downbeat was compared to the estimated one with a tolerance window of 50 ms. An accuracy of 51% was achieved. Downbeat detection was also evaluated at the beat-level, i.e., estimating whether the downbeat corresponds to one of the four beats of the meter (instead of off-beat positions). This gave an accuracy of 59%, meaning that in the other cases the downbeat was detected on the off-beat positions. For some EDM tracks it was observed that high degree of periodicity compensates for a wrongly estimated downbeat. The overall results of the similarity predictions of the model (Section 3.4) indicate only a minor increase when the correct (annotated) downbeats are taken into account. It is hence concluded that the downbeat detection algorithm does not have great influence on the current results of the model.

### 3.4 Mapping Model Predictions to Perceptual Ratings of Similarity

The model's predictions were evaluated with perceptual ratings of rhythm similarity collected via a listening experiment. Pairwise comparisons of a small set of segments representing various rhythmic patterns of EDM were presented. Subjects were asked to rate the perceived rhythm similarity, choosing from a four point scale, and report also the confidence of their rating. From a preliminary collection of experiment data, 28 pairs (representing a total of 18 unique music segments) were selected for further analysis. These were rated from a total of 28 participants, with mean age 27 years old and standard deviation 7.3. The 50% of the participants received formal musical training, 64% was familiar with EDM and 46% had experience as EDM musician/producer. The selected pairs were rated between 3 to 5 times, with all participants reporting confidence in their

<sup>3</sup> www.MIREX.org

r	p	features
-0.17	0.22	attack characterization
0.48	0.00	periodicity
0.33	0.01	metrical distribution excl. metrical profile
0.69	0.00	metrical distribution incl. metrical profile
0.70	0.00	all

**Table 1:** Pearson's correlation  $r$  and  $p$ -values between the model's predictions and perceptual ratings of rhythm similarity for different sets of features.

rating, and all ratings being consistent, i.e., rated similarity was not deviating more than 1 point scale. The mean of the ratings was utilized as the ground truth rating per pair.

For each pair, similarity can be calculated via applying a distance metric to the feature vectors of the underlying segments. In this preliminary analysis, the cosine distance was considered. Pearson's correlation was used to compare the annotated and predicted ratings of similarity. This was applied for different sets of features as indicated in Table 1.

A maximum correlation of 0.7 was achieved when all features were presented. The non-zero correlation hypothesis was not rejected ( $p > 0.05$ ) for the attack characterization features indicating non-significant correlation with the (current set of) perceptual ratings. The periodicity features are correlated with  $r = 0.48$ , showing a strong link with perceptual rhythm similarity. The metrical distribution features indicate a correlation increase of 0.36 when the metrical profile is included in the feature vector. This is in agreement with the finding of [24].

As an alternative evaluation measure, the model's predictions and perceptual ratings were transformed to a binary scale (i.e., 0 being dissimilar and 1 being similar) and their output was compared. The model's predictions matched the perceptual ratings with an accuracy of 64%. Hence the model matches the perceptual similarity ratings at not only relative (i.e., Pearson's correlation) but also absolute way, when a binary scale similarity is considered.

## 4. DISCUSSION AND FUTURE WORK

In the evaluation of the model, the following considerations are made. High correlation of 0.69 was achieved when the metrical profile, output per stream, was added to the feature vector. An alternative experiment tested the correlation when considering the metrical profile as a whole, i.e., as a sum across all streams. This gave a correlation of only 0.59 indicating the importance of stream separation and hence the advantage of the model to account for this.

A maximum correlation of 0.7 was reported, taking into account the downbeat detection being 51% of the cases correct. Although regularity in EDM sometimes compensates for this, model's predictions can be improved with a more robust downbeat detection.

Features of periodicity (Section 2.2.2) and metrical distribution (Section 2.2.3) were extracted assuming a  $\frac{4}{4}$  meter, and 16-th note resolution throughout the segment. This is generally true for EDM, but exceptions do exist [2]. The

assumptions could be relaxed to analyze EDM with ternary divisions or no  $\frac{4}{4}$  meter, or expanded to other music styles with similar structure.

The correlation reported in Section 3.4 is computed from a preliminary set of experiment data. More ratings are currently collected and a regression analysis and tuning of the model is considered in future work.

## 5. CONCLUSION

A model of rhythm similarity for Electronic Dance Music has been presented. The model extracts rhythmic features from audio segments and computes similarity by comparing their feature vectors. A method for rhythmic stream detection is proposed that estimates the number and range of frequency bands from the spectral representation of each segment rather than a fixed division. Features are extracted from each stream, an approach shown to benefit the analysis. Similarity predictions of the model match perceptual ratings with a correlation of 0.7. Future work will fine-tune predictions based on a perceptual rhythm similarity model.

## 6. REFERENCES

- [1] S. Böck, A. Arzt, K. Florian, and S. Markus. Online real-time onset detection with recurrent neural networks. In *International Conference on Digital Audio Effects*, 2012.
- [2] M. J. Butler. *Unlocking the Groove*. Indiana University Press, Bloomington and Indianapolis, 2006.
- [3] E. Cambouropoulos. Voice and Stream: Perceptual and Computational Modeling of Voice Separation. *Music Perception*, 26(1):75–94, 2008.
- [4] D. Diakopoulos, O. Vallis, J. Hochenbaum, J. Murphy, and A. Kapur. 21st Century Electronica: MIR Techniques for Classification and Performance. In *ISMIR*, 2009.
- [5] S. Dixon, F. Gouyon, and G. Widmer. Towards Characterisation of Music via Rhythmic Patterns. In *ISMIR*, 2004.
- [6] A. Eigenfeldt and P. Pasquier. Evolving Structures for Electronic Dance Music. In *Genetic and Evolutionary Computation Conference*, 2013.
- [7] J. Foote and S. Uchihashi. The beat spectrum: a new approach to rhythm analysis. In *ICME*, 2001.
- [8] J. T. Foote. Media segmentation using self-similarity decomposition. In *Electronic Imaging*. International Society for Optics and Photonics, 2003.
- [9] D. Gärtner. Tempo estimation of urban music using tatum grid non-negative matrix factorization. In *ISMIR*, 2013.
- [10] J. W. Gordon. The perceptual attack time of musical tones. *The Journal of the Acoustical Society of America*, 82(1):88–105, 1987.
- [11] T. D. Griffiths and J. D. Warren. What is an auditory object? *Nature Reviews Neuroscience*, 5(11):887–892, 2004.
- [12] C. Guastavino, F. Gómez, G. Toussaint, F. Marandola, and E. Gómez. Measuring Similarity between Flamenco Rhythmic Patterns. *Journal of New Music Research*, 38(2):129–138, June 2009.
- [13] J. A. Hockman, M. E. P. Davies, and I. Fujinaga. One in the Jungle: Downbeat Detection in Hardcore, Jungle, and Drum and Bass. In *ISMIR*, 2012.
- [14] A. Klapuri, A. J. Eronen, and J. T. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech and Language Processing*, 14(1):342–355, January 2006.
- [15] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *ISMIR*, 2013.
- [16] O. Lartillot, T. Eerola, P. Toiviainen, and J. Fornari. Multi-feature Modeling of Pulse Clarity: Design, Validation and Optimization. In *ISMIR*, 2008.
- [17] O. Lartillot and P. Toiviainen. A Matlab Toolbox for Musical Feature Extraction From Audio. In *International Conference on Digital Audio Effects*, 2007.
- [18] H. C. Longuet-Higgins and C. S. Lee. The Rhythmic Interpretation of Monophonic Music. *Music Perception: An Interdisciplinary Journal*, 1(4):424–441, 1984.
- [19] A. Novello, M. M. F. McKinney, and A. Kohlrausch. Perceptual Evaluation of Inter-song Similarity in Western Popular Music. *Journal of New Music Research*, 40(1):1–26, March 2011.
- [20] J. Paulus and A. Klapuri. Measuring the Similarity of Rhythmic Patterns. In *ISMIR*, 2002.
- [21] B. Rocha, N. Bogaards, and A. Honingh. Segmentation and Timbre Similarity in Electronic Dance Music. In *Sound and Music Computing Conference*, 2013.
- [22] E. D. Scheirer. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103(1):588–601, January 1998.
- [23] M. R. Schroeder, B. S. Atal, and J. L. Hall. Optimizing digital speech coders by exploiting masking properties of the human ear. *The Journal of the Acoustical Society of America*, pages 1647–1652, 1979.
- [24] L. M. Smith. Rhythmic similarity using metrical profile matching. In *International Computer Music Conference*, 2010.
- [25] J. R. Zapata and E. Gómez. Comparative Evaluation and Combination of Audio Tempo Estimation Approaches. In *Audio Engineering Society Conference*, 2011.

# MUSE: A MUSIC RECOMMENDATION MANAGEMENT SYSTEM

Martin Przyjaciel-Zablocki, Thomas Hornung, Alexander Schätzle,  
Sven Gauß, Io Taxidou, Georg Lausen

Department of Computer Science, University of Freiburg

zablocki, hornungt, schaetzle, gauss, taxidou, lausen@informatik.uni-freiburg.de

## ABSTRACT

Evaluating music recommender systems is a highly repetitive, yet non-trivial, task. But it has the advantage over other domains that recommended songs can be evaluated immediately by just listening to them.

In this paper, we present MUSE – a music recommendation management system – for solving the typical tasks of an in vivo evaluation. MUSE provides the typical off-the-shelf evaluation algorithms, offers an online evaluation system with automatic reporting, and by integrating online streaming services also a legal possibility to evaluate the quality of recommended songs in real time. Finally, it has a built-in user management system that conforms with state-of-the-art privacy standards. New recommender algorithms can be plugged in comfortably and evaluations can be configured and managed online.

## 1. INTRODUCTION

One of the hallmarks of a good recommender system is a thorough and significant evaluation of the proposed algorithm(s) [6]. One way to do this is to use an offline dataset like *The Million Song Dataset* [1] and split some part of the data set as training data and run the evaluation on top of the remainder of the data. This approach is meaningful for features that are already available for the dataset, such as e.g. tag prediction for new songs. However, some aspects of recommending songs are inherently subjective, such as serendipity [12], and thus the evaluation of such algorithms can only be done in vivo, i.e. with real users not in an artificial environment.

When conducting an in vivo evaluation, there are some typical issues that need to be considered:

**User management.** While registering for evaluations, users should be able to provide some context information about them to guide the assignment in groups for A/B testing.

**Privacy & Security.** User data is highly sensitive, and high standards have to be met wrt. who is allowed to access

the data. Also, an evaluation framework needs to ensure that user data cannot be compromised.

**Group selection.** Users are divided into groups for A/B testing, e.g. based on demographic criteria like age or gender. Then, recommendations for group A are provided by a baseline algorithm, and for group B by the new algorithm.

**Playing songs.** Unlike other domains, e.g. books, users can give informed decisions by just listening to a song. Thus, to assess a recommended song, it should be possible to play the song directly during the evaluation.

**Evaluation monitoring.** During an evaluation, it is important to have an overview of how each algorithm performs so far, and how many and how often users participate.

**Evaluation metrics.** Evaluation results are put into graphs that contain information about the participants and the performance of the evaluated new recommendation algorithm.

**Baseline algorithms.** Results of an evaluation are often judged by improvements over a baseline algorithm, e.g. a collaborative filtering algorithm [10].

In this paper, we present MUSE – a music recommendation management system – that takes care of all the regular tasks that are involved in conducting an in vivo evaluation. Please note that MUSE can be used to perform in vivo evaluations of *arbitrary* music recommendation algorithms. An instance of MUSE that conforms with state-of-the-art privacy standards is accessible by using the link below, a documentation is available on the MUSE website<sup>2</sup>.

[muse.informatik.uni-freiburg.de](http://muse.informatik.uni-freiburg.de)

The remainder of the paper is structured as follows: After a discussion of related work in Section 2, we give an overview of our proposed music recommendation management system in Section 3 with some insights in our evaluation framework in Section 4. Included recommenders are presented in Section 5, and we conclude with an outlook on future work in Section 6.

## 2. RELATED WORK

The related work is divided in three parts: (1) music based frameworks for recommendations, (2) recommenders' evaluation, (3) libraries and platforms for developing and plugin recommenders.

Music recommendation has attracted a lot of interest from the scientific community since it has many real life applications and bears multiple challenges. An overview



© Martin Przyjaciel-Zablocki, Thomas Hornung, Alexander Schätzle, Sven Gauß, Io Taxidou, Georg Lausen.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Martin Przyjaciel-Zablocki, Thomas Hornung, Alexander Schätzle, Sven Gauß, Io Taxidou, Georg Lausen. "MuSe: A Music Recommendation Management System", 15th International Society for Music Information Retrieval Conference, 2014.

<sup>2</sup> MUSE - Music Sensing in a Social Context: [dbis.informatik.uni-freiburg.de/MuSe](http://dbis.informatik.uni-freiburg.de/MuSe)

of factors affecting music recommender systems and challenges that emerge both for the users' and the recommenders side are highlighted in [17]. Improving music recommendations has attracted equal attention. In [7, 12], we built and evaluated a weighted hybrid recommender prototype that incorporates different techniques for music recommendations. We used Youtube for playing songs but due to a complex process of identifying and matching songs, together with some legal issues, such an approach is no longer feasible. Music platforms are often combined with social media where users can interact with objects maintaining relationships. Authors in [2] leverage this rich information to improve music recommendations by viewing recommendations as a ranking problem.

The next class of related work concerns evaluation of recommenders. An overview of existing systems and methods can be found in [16]. In this study, recommenders are evaluated based on a set of properties relevant for different applications and evaluation metrics are introduced to compare algorithms. Both offline and online evaluation with real users are conducted, discussing how to draw valuable conclusion. A second review on collaborative recommender systems specifically can be found in [10]. It consists the first attempt to compare and evaluate user tasks, types of analysis, datasets, recommendation quality and attributes. Empirical studies along with classification of existing evaluation metrics and introduction of new ones provide insights into the suitability and biases of such metrics in different settings. In the same context, researchers value the importance of user experience in the evaluation of recommender systems. In [14] a model is developed for assessing the perceived recommenders quality of users leading to more effective and satisfying systems. Similar approaches are followed in [3, 4] where authors highlight the need for user-centric systems and high involvement of users in the evaluation process. Relevant to our study is the work in [9] which recognizes the importance for online user evaluation, while implementing such evaluations simultaneously by the same user in different systems.

The last class of related work refers to platforms and libraries for developing and selecting recommenders. The authors of [6] proposed LensKit, an open-source library that offers a set of baseline recommendation algorithms including an evaluation framework. MyMediaLite [8] is a library that offers state of the art algorithms for collaborative filtering in particular. The API offers the possibility for new recommender algorithm's development and methods for importing already trained models. Both provide a good foundation for comparing different research results, but without a focus on in vivo evaluations of music recommenders, thus they don't offer e.g. capabilities to play and rate songs or manage users. A patent in [13] describes a portal extension with recommendation engines via interfaces, where results are retrieved by a common recommendation manager. A more general purpose recommenders framework [5] which is close to our system, allows using and comparing different recommendation methods on provided datasets. An API offers the possibility to develop and

incorporate algorithms in the framework, integrate plugins, make configurations and visualize the results. However, our system offers additionally real-time online evaluations of different recommenders, while incorporating end users in the evaluation process. A case study of using Apache Mahout, a library for distributed recommenders based on MapReduce can be found in [15]. Their study provides insights into the development and evaluation of distributed algorithms based on Mahout.

To the best of our knowledge, this is the first system that incorporates such a variety of characteristics and offers a full solution for music recommenders development and evaluation, while highly involving the end users.

### 3. MUSE OVERVIEW

We propose MUSE: a web-based music recommendation management system, built around the idea of recommenders that can be plugged in. With this in mind, MUSE is based on three main system design pillars:

**Extensibility.** The whole infrastructure is highly extensible, thus new recommendation techniques but also other functionalities can be added as modular components.

**Reusability.** Typical tasks required for evaluating music recommendations (e.g. managing user accounts, playing and rating songs) are already provided by MUSE in accordance with current privacy standards.

**Comparability.** By offering one common evaluation framework we aim to reduce side-effects of different systems that might influence user ratings, improving both comparability and validity of in-vivo experiments.

A schematic overview of the whole system is depicted in Fig. 1. The MUSE Server is the core of our music recommendation management system enabling the communication between all components. It coordinates the interaction with pluggable recommenders, maintains the data in three different repositories and serves the requests from multiple MUSE clients. Next, we will give some insights in the architecture of MUSE by explaining the most relevant components and their functionalities.

#### 3.1 Web-based User Interface

Unlike traditional recommender domains like e-commerce, where the process of consuming and rating items takes up to several weeks, recommending music exhibits a highly dynamic nature raising new challenges and opportunities for recommender systems. Ratings can be given on the fly and incorporated immediately into the recommending process, just by listening to a song. However, this requires a reliable and legal solution for playing a large variety of songs. MUSE benefits from a tight integration of Spotify<sup>3</sup>, a music streaming provider that allows listening to millions of songs for free. Thus, recommended songs can be embedded directly into the user interface, allowing to listen and rate them in a user-friendly way as shown in Fig. 2.

<sup>3</sup> A Spotify account is needed to play songs



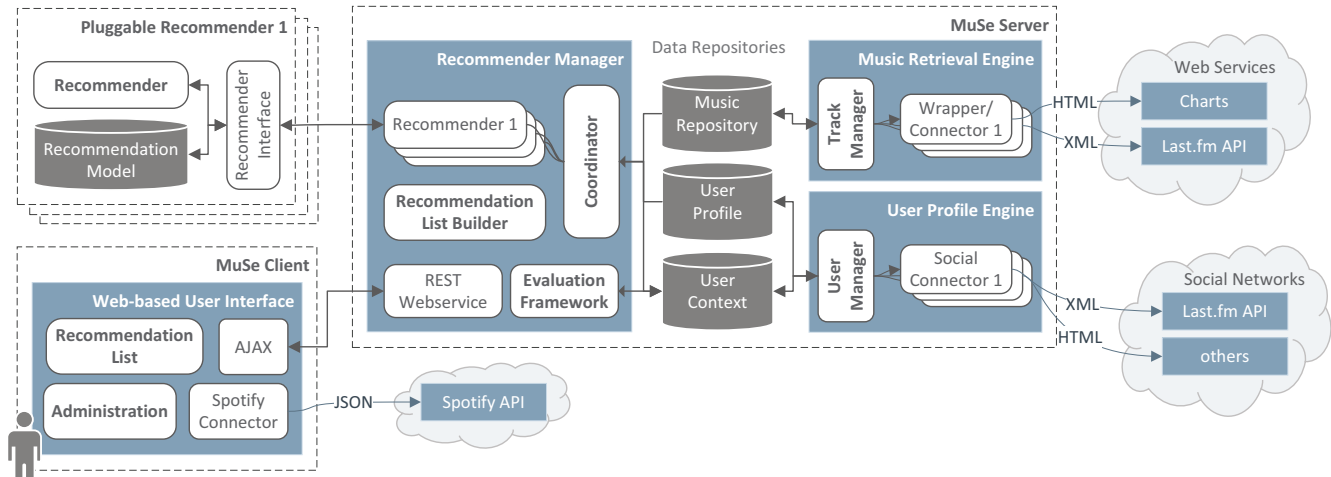


Figure 1. Muse – Music Recommendation Management System Overview

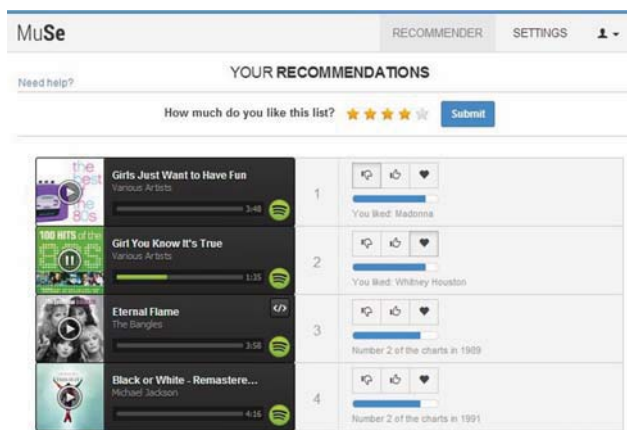


Figure 2. Songs can be played & rated

In order to make sure that users can obtain recommendations without having to be long-time MUSE users, we ask for some contextual information during the registration process. Each user has to provide coarse-grained demographic and preference information, namely the user's spoken languages, year of birth, and optionally a Last.fm user name. In Section 5, we will present five different approaches that utilize those information to overcome the cold start problem. Beyond that, these information is also exploited for dividing users into groups for A/B testing.

Fig. 3 shows the settings pane of a user. Note, that this window is available only for those users, who are not participating in an evaluation. It allows to browse all available recommenders and compare them based on meta data provided with each recommender. Moreover, it is also possible to control how recommendations from different recommenders are amalgamated to one list. To this end, a summary is shown that illustrates the interplay of novelty, accuracy, serendipity and diversity. Changes are applied and reflected in the list of recommendations directly.

### 3.2 Data Repositories

Although recommenders in MUSE work independently of each other and may even have their own recommendation model with additional data, all music recommenders have access to three global data structures.

The first one is the *Music Repository* that stores songs with their meta data. Only songs in this database can be recommended, played and rated. The *Music Retrieval Engine* periodically collects new songs and meta data from Web Services, e.g. chart lists or Last.fm. It can be easily extended by new sources of information like audio analysis features from the Million Song Dataset [1], that can be requested periodically or dynamically. Each recommender can access all data stored in the Music Repository.

The second repository stores the *User Profile*, hence it also contains personal data. In order to comply with German data privacy requirements only restricted access is granted for both, recommenders and evaluation analyses.

The last repository collects the *User Context*, e.g. which songs a user has listened to with the corresponding rating for the respective recommender.

Access with anonymized user IDs is granted for all recommenders and evaluation analyses. Finally, both user-related repositories can be enriched by the *User Profile Engine* that fetches data from other sources like social networks. Currently, the retrieval of listening profiles of publicly available data from Last.fm and Facebook is supported.

### 3.3 Recommender Manager

The Recommender Manager has to coordinate the interaction of recommenders with users and the access to the data. This process can be summarized as follows:

- It coordinates access to the repositories, forwards user request for new recommendations, and receives generated recommendations.
- It composes a list of recommendations by amalgamating recommendations from different recommenders into one list based on individual user settings.

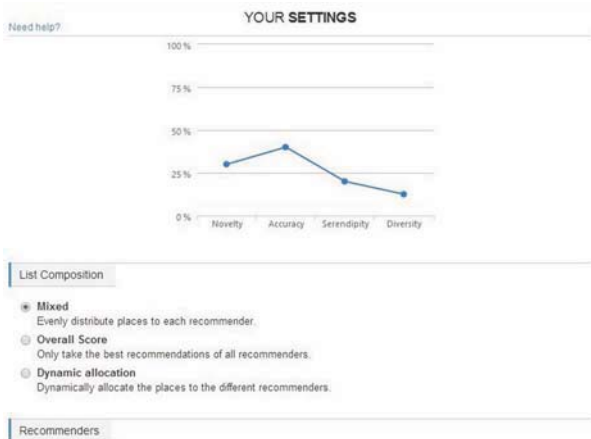


Figure 3. Users can choose from available recommenders

- A panel for administrative users allows enabling, disabling and adding of recommenders that implement the interface described in Section 3.4. Moreover, even composing hybrid recommenders is supported.

### 3.4 Pluggable Recommender

A cornerstone of MUSE is its support for plugging in recommenders easily. The goal was to design a rather simple and compact interface enabling other developers to implement new recommenders with enough flexibility to incorporate existing approaches as well. This is achieved by a predefined Java interface that has to be implemented for any new recommender. It defines the interplay between the MUSE Recommender Manager and its pluggable recommenders by (1) providing methods to access all three data repositories, (2) forwarding requests for recommendations and (3) receiving recommended items. Hence, new recommenders do not have to be implemented within MUSE in order to be evaluated, it suffices to use the interface to provide a mapping of inputs and outputs<sup>4</sup>.

## 4. EVALUATION FRAMEWORK

There are two types of experiments to measure the performance of recommenders: (1) offline evaluations based on historical data and (2) in vivo evaluations where users can evaluate recommendations online. Since music is of highly subjective nature with many yet unknown correlations, we believe that in vivo evaluations have the advantage of also capturing subtle effects on the user *during* the evaluation. Since new songs can be rated within seconds by a user, such evaluations are a good fit for the music domain. MUSE addresses the typical issues that are involved in conducting an in-vivo evaluation and thus allows researchers to focus on the actual recommendation algorithm.

This section gives a brief overview of how evaluations are created, monitored and analyzed.

<sup>4</sup> More details can be found on our project website.

## 4.1 Evaluation Setup

The configuration of an evaluation consists of three steps (cf. Fig. 4): (1) A new evaluation has to be scheduled, i.e. a start and end date for the evaluation period has to be specified. (2) The number and setup of groups for A/B testing has to be defined, where up to six different groups are supported. For each group an available recommender can be associated with the possibility of hybrid combinations of recommenders if desired. (3) The group placement strategy based on e.g. age, gender and spoken languages is required. As new participants might join the evaluation over time, an online algorithm maintains a uniform distribution with respect to the specified criteria. After the setup is completed, a preview illustrates how group distributions would resemble based on a sample of registered users.

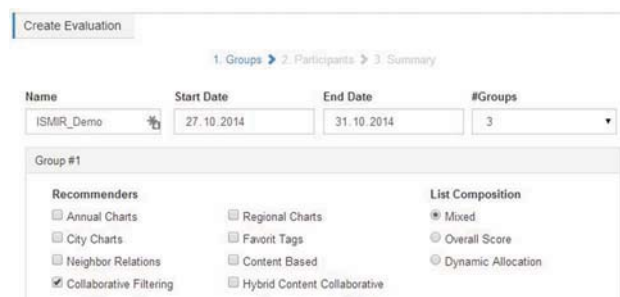


Figure 4. Evaluation setup via Web interface

While an evaluation is running, both registered users and new ones are asked to participate after they login to MUSE. If a user joins an evaluation, he will be assigned to a group based on the placement strategy defined during the setup and all ratings are considered for the evaluation. So far, the following types of ratings can be discerned:

**Song rating.** The user can provide three ratings for the quality of the recommended song (“love”, “like”, and “dislike”). Each of these three rating options is mapped to a numerical score internally, which is then used as basis for the analysis of each recommender.

**List rating.** The user can also provide ratings for the entire list of recommendations that is shown to him on a five-point Likert scale, visualized by stars.

**Question.** To measure other important aspects of a recommendation like its novelty or serendipity, an additional field with a question can be configured that contains either a yes/no button or a five-point Likert scale.

The user may also decide not to rate some of the recommendations. In order to reduce the number of non-rated recommendations in evaluations, the rating results can only be submitted when at least 50% of the recommendations are rated. Upon submitting the rating results, the user gets a new list with recommended songs.

## 4.2 Monitoring Evaluations

Running in vivo evaluations as a *black box* is undesirable, since potential issues might be discovered only after the

evaluation is finished. Also, it is favorable to have an overview of the current state, e.g. if there are enough participants, and how the recommenders perform so far. MUSE provides comprehensive insights via an administrative account into running evaluations as it offers an easy accessible visualization of the current state with plots. Thus, adjustments like adding a group or changing the runtime of the evaluation can be made while the evaluation is still running.

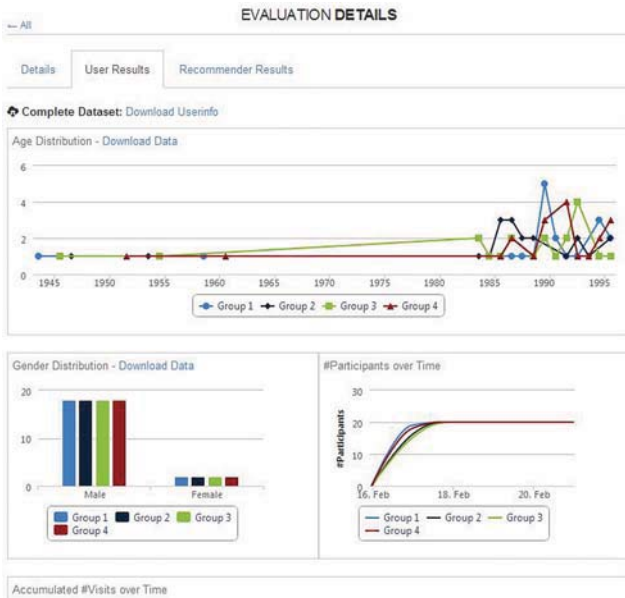


Figure 5. Evaluation results are visualized dynamically

### 4.3 Analyzing Evaluations

For all evaluations, including running and finished ones, a result overview can be accessed that shows results in a graphical way to make them easier and quicker to grasp (c.f. Fig. 5). The plots are implemented in a dynamic fashion allowing to adjust, e.g., the zoom-level or the displayed information as desired. They include a wide range of metrics like group distribution, number of participants over time, averaged ratings, mean absolute error, accuracy per recommender, etc. Additionally, the complete dataset or particular plotting data can be downloaded in CSV format.

## 5. RECOMMENDATION TECHNIQUES

MUSE comes with two types of recommenders out-of-the-box. The first type includes traditional algorithms, i.e. *Content Based* and *Collaborative Filtering* [10] that can be used as baseline for comparison. The next type of recommenders is geared towards overcoming the cold start problem by (a) exploiting information provided during registration (*Annual*, *Country*, and *City Charts* recommender), or (b) leveraging knowledge from social networks (*Social Neighborhood* and *Social Tags* recommender).

**Annual Charts Recommender.** Studies have shown, that the apex of evolving music taste is reached between the

age of 14 and 20 [11]. The Annual Charts Recommender exploits this insight and recommends those songs, which were popular during this time. This means, when a user indicates 1975 as his year of birth, he will be assigned to the music context of years 1989 to 1995, and obtain recommendations from that context. The recommendation ranking is defined by the charts position in the corresponding annual charts, where the following function is used to map the charts position to a score, with  $c_s$  as the position of song  $s$  in charts  $c$  and  $n$  is the maximum rank of charts  $c$ :

$$\text{score}(s) = -\log\left(\frac{1}{n}c_s\right) \quad (1)$$

**Country Charts Recommender.** Although music taste is subject to diversification across countries, songs that a user has started to listen to and appreciate oftentimes have peaked in others countries months before. This latency aspect as well as an inter-country view on songs provide a good foundation for serendipity and diversity. The source of information for this recommender is the spoken languages, provided during registration, which are mapped to a set of countries for which we collect the current charts. Suppose there is a user  $a$  with only one country  $A$  assigned to his spoken languages, and  $C_A$  the set of charts songs for  $A$ . Then, the set  $C_R$  of possible recommendations for  $a$  is defined as follows, where  $L$  is the set of all countries:

$$C_R = \left( \bigcup_{X \in L} C_X \right) \setminus C_A$$

The score for a song  $s \in C_R$  is defined by the average charts position across all countries, where Function (1) is used for mapping the charts position into a score.

**City Charts Recommender.** While music tastes differ across countries, they may likewise differ across cities in the same country. We exploit this idea by the City Charts Recommender, hence it can be seen as a more granular variant of the Country Charts Recommender. The set of recommendations  $C_R$  is now composed based on the city charts from those countries a user was assigned to. Hereby, the ranking of songs in that set is not only defined by the average charts position, but also by the number of cities where the song occurs in the charts: The fewer cities a song appears in, the more “exceptional” and thus relevant it is.

**Social Neighborhood Recommender.** Social Networks are, due to their growing rates, an excellent source for contextual knowledge about users, which in turn can be utilized for better recommendations. In this approach, we use the underlying social graph of Last.fm to generate recommendations based on user’s Last.fm neighborhood which can be retrieved by our *User Profile Engine*. To compute recommendations for a user  $a$ , we select his five closest neighbors, an information that is estimated by Last.fm internally. Next, for each of them, we retrieve its recent top 20 songs and thus get five sets of songs, namely  $N_1 \dots N_5$ . Since that alone would provide already known songs in general, we define the set  $N_R$  of possible recommendations as follows, where  $N_a$  is the set of at most 25 songs a

user  $a$  recently listened to and appreciated:

$$N_R = \left( \bigcup_{1 \leq i \leq 5} N_i \right) \setminus N_a$$

**Social Tags Recommender.** Social Networks collect an enormous variety of data describing not only users but also items. One common way of characterising songs is based on tags that are assigned to them in a collaborative manner. Our Social Tag Recommender utilizes such tags to discover new genres which are related to songs a user liked in the past. At first, we determine his recent top ten songs including their tags from Last.fm. We merge all those tags and filter out the most popular ones like “rock” or “pop” to avoid getting only obvious recommendations. By counting the frequency of the remaining tags, we determine the three most common thus relevant ones. For the three selected tags, we use again Last.fm to retrieve songs where the selected tags were assigned to most frequently.

To test our evaluation framework as well as to assess the performance of our five recommenders we conducted an in vivo evaluation with MUSE. As a result 48 registered users rated a total of 1567 song recommendations confirming the applicability of our system for in vivo evaluations. Due to space limitations, we decided to omit a more detailed discussion of the results.

## 6. CONCLUSION

MUSE puts the fun back in developing new algorithms for music recommendations by taking the burden from the researcher to spent cumbersome time on programming yet another evaluation tool. The module-based architecture offers the flexibility to immediately test novel approaches, whereas the web-based user-interface gives control and insight into running in vivo evaluations. We tested MUSE with a case study confirming the applicability and stability of our proposed music recommendation management system. As future work, we envision to increase the flexibility of setting up evaluations, add more metrics to the result overview, and to develop further connectors for social networks and other web services to enrich the user’s context while preserving data privacy.

## 7. REFERENCES

- [1] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *ISMIR*, 2011.
- [2] Jiajun Bu, Shulong Tan, Chun Chen, Can Wang, Hao Wu, Lijun Zhang 0005, and Xiaofei He. Music recommendation by unified hypergraph: combining social media information and music content. In *ACM Multimedia*, pages 391–400, 2010.
- [3] Li Chen and Pearl Pu. User evaluation framework of recommender systems. In *Workshop on Social Recommender Systems (SRS’10) at IUI*, volume 10, 2010.
- [4] Paolo Cremonesi, Franca Garzotto, Sara Negro, Alessandro Vittorio Papadopoulos, and Roberto Turrin. Looking for “good” recommendations: A comparative evaluation of recommender systems. In *INTERACT (3)*, pages 152–168, 2011.
- [5] Aviram Dayan, Guy Katz, Naseem Biasdi, Lior Rokach, Bracha Shapira, Aykan Aydin, Roland Schwaiger, and Radmila Fishel. Recommenders benchmark framework. In *RecSys*, pages 353–354, 2011.
- [6] Michael D. Ekstrand, Michael Ludwig, Joseph A. Konstan, and John Riedl. Rethinking the recommender research ecosystem: reproducibility, openness, and LensKit. In *RecSys*, pages 133–140, 2011.
- [7] Simon Franz, Thomas Hornung, Cai-Nicolas Ziegler, Martin Przyjaciel-Zablocki, Alexander Schätzle, and Georg Lausen. On weighted hybrid track recommendations. In *ICWE*, pages 486–489, 2013.
- [8] Zeno Gantner, Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Mymedialite: a free recommender system library. In *RecSys*, pages 305–308, 2011.
- [9] Conor Hayes and Pádraig Cunningham. An on-line evaluation framework for recommender systems. *Trinity College Dublin, Dep. of Computer Science*, 2002.
- [10] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John Riedl. Evaluating collaborative filtering recommender systems. In *ACM Trans. Inf. Syst.*, pages 5–53, 2004.
- [11] Morris B Holbrook and Robert M Schindler. Some exploratory findings on the development of musical tastes. *Journal of Consumer Research*, pages 119–124, 1989.
- [12] Thomas Hornung, Cai-Nicolas Ziegler, Simon Franz, Martin Przyjaciel-Zablocki, Alexander Schätzle, and Georg Lausen. Evaluating Hybrid Music Recommender Systems. In *WI*, pages 57–64, 2013.
- [13] Stefan Liesche, Andreas Nauerz, and Martin Welsch. Extendable recommender framework for web-based systems, 2008. US Patent App. 12/209,808.
- [14] Pearl Pu, Li Chen, and Rong Hu. A user-centric evaluation framework for recommender systems. In *RecSys ’11*, pages 157–164, New York, NY, USA, 2011. ACM.
- [15] Carlos E Seminario and David C Wilson. Case study evaluation of mahout as a recommender platform. In *RecSys*, 2012.
- [16] Guy Shani and Asela Gunawardana. Evaluating recommendation systems. In *Recommender Systems Handbook*, pages 257–297, 2011.
- [17] Alexandra L. Uitdenbogerd and Ron G. van Schyndel. A review of factors affecting music recommender success. In *ISMIR*, 2002.

# TEMPO- AND TRANSPOSITION-INVARIANT IDENTIFICATION OF PIECE AND SCORE POSITION

Andreas Arzt<sup>1</sup>, Gerhard Widmer<sup>1,2</sup>, Reinhard Sonnleitner<sup>1</sup>

<sup>1</sup>Department of Computational Perception, Johannes Kepler University, Linz, Austria

<sup>2</sup>Austrian Research Institute for Artificial Intelligence (OFAI), Vienna, Austria

andreas.arzt@jku.at

## ABSTRACT

We present an algorithm that, given a very small snippet of an audio performance and a database of musical scores, quickly identifies the piece and the position in the score. The algorithm is both tempo- and transposition-invariant. We approach the problem by extending an existing tempo-invariant symbolic fingerprinting method, replacing the absolute pitch information in the fingerprints with a relative representation. Not surprisingly, this leads to a big decrease in the discriminative power of the fingerprints. To overcome this problem, we propose an additional verification step to filter out the introduced noise. Finally, we present a simple tracking algorithm that increases the retrieval precision for longer queries. Experiments show that both modifications improve the results, and make the new algorithm usable for a wide range of applications.

## 1. INTRODUCTION

Efficient algorithms for content-based retrieval play an important role in many areas of music retrieval. A well known example are *audio fingerprinting* algorithms, which permit the retrieval of all audio files from the database that are (almost) *exact* replicas of a given example query (a short audio excerpt). For this task there exist efficient algorithms that are in everyday commercial use (see e.g. [4], [13]).

A related task, relevant especially in the world of classical music, is the following: given a short audio excerpt of a performance of a piece, identify both the piece (i.e. the musical score the performance is based on), and the position within the piece. For example, when presented with an audio excerpt of Vladimir Horowitz playing Chopin's Nocturne Op. 55 No. 1, the goal is to return the name and data of the piece (Nocturne Op. 55 No. 1 by Chopin) rather than identifying the exact audio recording. Hence, the database for this task does not contain audio recordings, but symbolic representations of musical scores. This is related to *version identification* (see [11] for an overview), where the

goal is to identify different versions of one and the same song, mostly in order to detect cover versions in popular music.

A common way to solve this task, especially for classical music, is to use an *audio matching* algorithm (see e.g. [10]). Here, all the scores are first transformed into audio files (or a suitable in-between representation), and then aligned to the query in question, most commonly with algorithms based on dynamic programming techniques. A limitation of this approach is that relatively large queries are needed (e.g. 20 seconds), to achieve good retrieval results. Another problem is computational cost. To cope with this, in [8] clever indexing strategies were presented that greatly reduce the computation time.

In [2] an approach is presented that tries to solve the task in the symbolic domain instead. First, the query is transformed into a symbolic list of note events via an *audio transcription* algorithm. Then, a globally tempo-invariant fingerprinting method is used to query the database and identify matching positions. In this way even for queries with lengths of only a few seconds very robust retrieval results can be achieved. A downside is that this method depends on automatic music transcription, which in general is an unsolved problem. In [2] a state of the art transcription system for piano music is used, thus limiting the approach to piano music only, at least for the time being.

In addition, we identified two other limitations of this algorithm, which we tackle in this paper. First, the approach depends on the performer playing the piece in the correct key and the correct octave (i.e. in the same key and octave as it is stored in the database). In music it is quite common to transpose a piece of music according to specific circumstances, e.g. a singer preferring to sing in a specific range. Secondly, while this algorithm works very well for small queries, larger queries with local tempo changes *within* the query tend to be problematic. Of course these limitations were already discussed in the literature for other approaches, see e.g. [10] for tempo- and transposition-invariant audio matching.

In this paper we present solutions to both problems by proposing (1) a *transposition-invariant fingerprinting* method for symbolic music representations which uses an additional verification step that largely compensates for the general loss in discriminative power, and (2) a simple but effective tracking method that essentially achieves not only global, but also *local invariance to tempo changes*.



© Andreas Arzt<sup>1</sup>, Gerhard Widmer<sup>1,2</sup>, Reinhard Sonnleitner<sup>1</sup>.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Andreas Arzt<sup>1</sup>, Gerhard Widmer<sup>1,2</sup>, Reinhard Sonnleitner<sup>1</sup>. "Tempo- and Transposition-invariant Identification of Piece and Score Position", 15th International Society for Music Information Retrieval Conference, 2014.

## 2. TEMPO-INVARIANT FINGERPRINTING

The basis of our algorithm is a fingerprinting method presented in [2] (which in turn is based on [13]) that is *invariant to the global tempo* of both the query and the entries in the database. In this section we will give a brief summary of this algorithm. Then we will show how to make it *transposition-invariant* (Section 3) and how to make it *invariant to local tempo changes* (Section 4).

### 2.1 Building the Score Database

In [2] a fingerprinting algorithm was introduced that is invariant to global tempo differences between the query and the scores in the database. Each score is represented as an ordered list of [ontime, pitch] pairs, which in turn are extracted from MIDI files with a suitable but constant tempo for the whole piece.

For each score, fingerprint tokens are generated and stored in a database. Tokens are created from triplets of note-on events according to some constraints to make them tempo invariant. A fixed event  $e$  is paired with the first  $n_1$  events with a distance of at least  $d$  seconds “in the future” of  $e$ . This results in  $n_1$  event pairs. For each of these pairs this step is repeated with the  $n_2$  future events with a distance of at least  $d$  seconds. This finally results in  $n_1 * n_2$  event triplets. In our experiments we used the values  $d = 0.05$  seconds and  $n_1 = n_2 = 5$  (i.e. for each event 25 tokens are created). The pair creation steps are constrained to notes which are at most 2 octaves apart.

Given such a triplet consisting of the events  $e_1$ ,  $e_2$  and  $e_3$ , the time difference  $td_{1,2}$  between  $e_1$  and  $e_2$  and the time difference  $td_{2,3}$  between  $e_2$  and  $e_3$  are computed. To get a tempo independent fingerprint token, the ratio of the time differences is computed:  $tdr = \frac{td_{2,3}}{td_{1,2}}$ . This finally leads to a fingerprint token  $dbtoken = [pitch_1 : pitch_2 : pitch_3 : tdr] : pieceID : time : td_{1,2}$ , with the hash key being  $[pitch_1 : pitch_2 : pitch_3 : tdr]$ ,  $pieceID$  the identifier of the piece, and  $time$  the onset time of  $e_1$ . The tokens in our database are unique, i.e. we only insert the generated token if an equivalent one does not exist yet.

### 2.2 Querying the Database

Before querying the database, the query (an audio snippet of a performance) has to be transformed into a symbolic representation. The algorithm we use to transcribe musical note onsets from an audio signal is based on the system described in [3]. The result of this step is a possibly very noisy list of [ontime, pitch] pairs.

This list is processed in exactly the same fashion as above, resulting in a list of tokens of the form  $qtoken = [qpitch_1 : qpitch_2 : qpitch_3 : qtdr] : qtime : qtd_{1,2}$ . Then, all the tokens which match hash keys of the query tokens are extracted from the database (we allow a maximal deviation of the ratio of the time differences of 15%). For querying, the general idea is to find regions in the database of scores which share a continuous sequence of tokens with the query. To quickly identify these regions we use the histogram approach presented in [2] and [13].

This is a computationally inexpensive way of finding these sequences by sorting the matched tokens into a histogram with a bin width of 1 second such that peaks appear at the start points of these regions (i.e. the start point where the query matches a database position). We also included the restriction that each query token can only be sorted at most once into each bin of the histogram, effectively preventing excessively high scores for sequences of repeated patterns in a brief period of time.

The matching score for each score position is computed as the number of tokens in the respective histogram bin. In addition, we can also compute a tempo estimate, i.e. the tempo of the performance compared to the tempo in the score, by taking the mean of the ratios of  $td_{1,2}$  and  $qtd_{1,2}$  of the respective matching query and database tokens that were sorted in the bin in question. We will use this information for the tracking approach presented in Section 4.

## 3. TRANSPOSITION-INVARIANT FINGERPRINTS

### 3.1 General Approach

In the algorithm described above, the pitches in the hash keys are represented as absolute values. Thus, if a performer decides to transpose a piece by an arbitrary number of semi-tones, any identification attempt by the algorithm must fail.

To overcome this problem, we suggest a simple, *relative* representation of the pitch values, which makes the algorithm invariant to linear transpositions. Instead of using 3 absolute pitch values, we replace them by 2 differences,  $pd_1 = pitch_2 - pitch_1$  and  $pd_2 = pitch_3 - pitch_2$ , resulting in a hash key  $[pd_1 : pd_2 : tdr]$ . For use in Section 3.2 below we additionally store  $pitch_1$ , the absolute pitch of the first note, in the token value.

In every other aspect the algorithm works in the same way as the purely tempo-invariant version described above. Of course this kind of transposition invariance cannot come for free as the resulting fingerprints will not be as discriminative as before. This has two important direct consequences: (1) the retrieval accuracy will suffer, and (2) for every query a lot more matching tokens are found in the database, thus the runtime for each query increases (see Section 5).

### 3.2 De-noising the Results: Token Verification

To compensate for the loss in discriminative power we propose an additional step before accepting a database token as a match to the query. The general idea is taken from [9] and was first used in a music context by [12]. It is based on a verification step for each returned token that looks at the context within the query and the context at the returned position in the database.

Each token  $dbtoken$  that was returned in response to a  $qtoken$  can be used to *project the query* (i.e. the notes identified from the query audio snippet by the transcription algorithm) to the possibly matching position in the score indicated by the  $dbtoken$ . The intuition then is that a

true matching positions we will find a majority of the notes from the query at their expected positions in the score. This will permit us to more reliably decide if the match of hash keys is a false positive or an actual match.

To do this, we need to compute the pitch shift and the tempo difference between the query and the potential position in the database. The pitch shift is computed as the difference of the  $pitch_1$  of  $qtoken$  and  $dbtoken$ . The difference in tempo is computed as the ratio of  $td_{1,2}$  of the two tokens. This information can now in turn be used to compute the expected time and pitch for each query note at the current score position hypothesis. We actually do not do this for the whole query, but only for a window of  $w = 10$  notes, centred at the event  $e_1$  of the query, and we exclude the notes  $e_1$ ,  $e_2$  and  $e_3$  from this list (as they were already used to come up with the match in the first place).

We now take these  $w$  notes and check if they appear in the database as would be expected. In this search we are strict on the pitch value, but allow for a window of  $\pm 100$  ms with regards to the actual time in the database. If we can confirm that a certain percentage of notes from the query appears in the database as expected (in the experiments we used 0.8), we finally accept the query token as an actual match.

As this approach is computationally expensive, we actually compute the results in two steps: we first do ‘normal’ fingerprinting without the verification step and only keep the top 5% of the results. We then perform the verification step on these results only and recompute the scores. On our dataset this effectively more than halves the computation time.

#### 4. PROCESSING LONGER QUERIES: MULTI-AGENT TRACKING

The fingerprinting method in [2] was mainly concerned with invariance regarding the *global tempo*. When applying this algorithm to our database with longer queries, *local tempo changes* (i.e. tempo changes within the query) prove to be problematic, because they break the ‘cheap’ histogram approach that is used to determine continuous regions of matching tokens.

Instead of using computationally much more expensive methods for determining these regions, we propose to split longer queries into shorter ones and track the results of these sub-queries over time. This is based on the assumption that in short queries the tempo is (quasi) stationary, and that a few exceptions will not break the tracking algorithm we use. In our implementation, we split each query into sub-queries with a window size of  $w = 15$  notes and a hop size of  $h = 5$  notes and then feed each sub-query to the fingerprinter individually.

Each result of a sub-query (but at most the top 100 positions that are returned) is in turn fed to an on-line position hypothesis tracking algorithm. In our current proof-of-concept implementation we use a simple on-line rule-based multi-agent approach, inspired by the beat-tracking algorithm described in [6]. For a purely off-line retrieval task a non-causal algorithm will lead to even better results.

The basic idea is to create virtual ‘agents’ for positions in the result sets. Each agent has a current hypothesis of the piece, the position within the piece and the tempo, and a score based on the results of the sub-queries. The agents are updated, if possible, with newly arriving data. In doing so, agents that represent positions that successively occur in result sets will accumulate higher scores than agents that represent positions that only occurred once or twice by chance, and are most probably false positives.

More precisely, we iterate over all sub-queries and perform the following steps in each iteration:

- **Normalise Scores:** First the scores of the positions in the result set of the sub-query are normalised by dividing them by their median. This makes sure that each iteration has approximately the same influence on the tracking process.
- **Update Agents:** For every agent, we look for a matching position in the result set of the sub-query (i.e. a position that approximately fits the extrapolated position of the agent, given the old position, the tempo, and the elapsed time). The position, the tempo and the score of the agent are updated with the new data from the matching result of the sub-query. If we do not find a matching position in the result set, we update the agent with a score of 0, and the extrapolated position is taken as the new hypothesis. If a matching position is found, the accumulated score is updated in a fashion such that scores from further in the past have a smaller impact than more recent ones. Each agent has a ring buffer  $s$  of size 50, in which the scores of the individual sub-queries are being stored. The accumulated score of the agent is then calculated as  $score_{acc} = \sum_{i=1}^{50} \frac{s_i}{1+\log i}$ , where  $s_1$  is the most recent score.
- **Create Agents:** Each sub-query result that was not used to update an existing agent is used to initialise a new agent at the respective score position (i.e. in the first iteration up to 100 agents are created).
- **Remove obsolete Agents:** Finally, agents with low scores are removed. In our implementation we simply remove agents that are older than 10 iterations and are not part of the current top 25 agents.

At each point in time the agents are ordered by  $score_{acc}$  and can be seen as hypotheses about the current position in the database of pieces. Thus, in the case of a single long query, the agents with the highest accumulated scores are returned in the end. In an on-line scenario, where an audio stream is constantly being monitored by the fingerprinting system, the current top hypotheses can be returned after each performed update (i.e. after each processed sub-query).

## 5. EVALUATION

### 5.1 Dataset Description

For the evaluation of the proposed algorithms a ground truth is needed. We need exact alignments of performances (recordings) of classical music to their respective scores such that we know exactly when each note given in the score is actually played in the performance. This data can either be generated by a computer program or by extensive manual annotation but both ways are prone to errors.

Luckily, we have access to two unique datasets where professional pianists played performances on a computer-controlled piano<sup>1</sup> and thus every action (e.g. key presses, pedal movements) was recorded. The first dataset (see [14]) consists of performances of the first movements of 13 Mozart piano sonatas by Roland Batik. The second, much larger, dataset consists of nearly the complete solo piano works by Chopin performed by Nikita Magaloff [7]. For the latter set we do not have the original audio files and thus replayed the symbolic performance data on a Yamaha N2 hybrid piano and recorded the resulting performances.

As we have both symbolic and audio information about the performances, we know the exact timing of each played note in the audio files. To build the score database we converted the sheet music to MIDI files with a constant tempo such that the overall duration of the file is similar to a ‘normal’ performance of the piece.

In addition to these two datasets the score database includes the complete Beethoven piano sonatas, two symphonies by Beethoven, and various other piano pieces. To this data we have no ground truth, but this is irrelevant since we do not actively query for them with performance data in our evaluation runs. See Table 1 for an overview of the complete dataset.

### 5.2 Results

For the evaluation we follow the procedure from [2]. A score position  $X$  is considered correct if it marks the beginning ( $\pm 1.5$  seconds) of a score section that is identical in note content, over a time span the length of the query (but at least 20 notes), to the note content of the ‘real’ score situation corresponding to the audio segment that the system was just listening to. We can establish this as we have the correct alignment between performance time and score positions — our *ground truth*). This complex definition is necessary because musical pieces may contain repeated sections or phrases, and it is impossible for the system (or anyone else, for that matter) to guess the ‘true’ one out of a set of identical passages matching the current performance snippet, given just that performance snippet as input. We acknowledge that a measurement of musical time in a score in terms of seconds is rather unusual. But as the MIDI tempos in our database generally are set in a meaningful way, this seemed the best decision to make errors comparable over different pieces, with different time signatures — it would not be very meaningful to, e.g. compare errors in bars or beats over different pieces.

We tested the algorithms with different query lengths: 10, 15, 20 and 25 notes (automatically transcribed from the audio query). For each of the query lengths, we generated 2500 queries by picking random points in the performances of our test database, and used them as input for the proposed algorithms. Duplicate retrieval results (i.e. positions that have the exact same note content; also, duplicate piece IDs for the experiments on piece-level) are removed from the result set.

Table 2 shows the results of the original tempo-invariant (but not pitch-invariant) algorithm on our dataset. Here, we present results for two categories: correctly identified pieces, and correctly identified piece and position in the score. For both categories we give the percentage of correct results at rank 1, and the mean reciprocal rank. This experiment basically confirms the results that were reported in [2] on a larger database (more than twice as large), for which a slight drop in performance is expected.

In addition, for the experiments with the transposition-invariant fingerprinting method, we transposed each score randomly by between -11 and +11 semitones — although strictly speaking this was not necessary, as the transposition-invariant algorithm returns *exactly* the same (large) set of tokens for un-transposed and transposed queries or scores.

Table 3 gives the results of the transposition-invariant method on these queries, both without (left) and with the verification step (right). As expected, the use of pitch-invariant fingerprints without additional verification causes a big decrease in retrieval precision (compare left half of Table 3 with Table 2). Furthermore, the loss in discriminative power of the fingerprint tokens also results in an increased number of tokens returned for every query, which has a direct influence on the runtime of the algorithm (last row in Table 3). The proposed verification step solves the precision problem, at least to some extent, and in our opinion makes the approach usable. Of course this does not come for free, as the runtime increases slightly.

We also tried to use the verification step with the original tempo-invariant algorithm but were not able to improve on the retrieval results. At least on our test data the tempo-invariant fingerprints are discriminative enough to mostly avoid false positives.

Finally, Table 4 gives the results on slightly longer queries for both the original tempo-invariant and the new tempo- and transposition-invariant algorithm. As can be seen, for the detection of the exact position in the score, using no tracking, the results based on queries with length 100 notes are worse than those for queries with only 50 notes, i.e. more information leads to worse results. This is caused by local tempo changes within the query, which break the histogram approach for finding sequences of matching tokens.

As shown on the right hand side for both fingerprinting types in Table 4, the approach of splitting longer queries into shorter ones and tracking the results takes care of this problem. Please note that for the tracking approach we check if the position hypotheses after the last tracking step match the correct position in the score. Thus, as this is an

<sup>1</sup> Bösendorfer SE 290



Data Description	Score Database		Testset	
	Number of Pieces	Notes in Score	Notes in Performance	Performance Duration
Chopin Corpus	154	325,263	326,501	9:38:36
Mozart Corpus	13	42,049	42,095	1:23:56
Additional Pieces	159	574,926	–	–
Total	326	942,238		

**Table 1.** Database and Testset Overview. In the database, all the pieces are included. As we only have performances aligned to the scores for the Chopin and the Mozart corpus, only these are included in the test set to query the database.

Query Length in Notes	10	15	20	25
Correct Piece as Top Match	0.6	0.82	0.88	<b>0.91</b>
Correct Piece Mean Reciprocal Rank (MRR)	0.68	0.86	0.91	<b>0.93</b>
Correct Position as Top Match	0.53	0.72	0.77	<b>0.79</b>
Correct Position Mean Reciprocal Rank (MRR)	0.60	0.79	0.83	<b>0.85</b>
Mean Query Length in Seconds	1.47	2.26	3.16	3.82
Mean Query Execution Time in Seconds	0.02	0.06	0.11	0.16

**Table 2.** Results for different query sizes of the original *tempo-invariant* piece and score position identification algorithm on the test database at the piece level (upper half) and on the score position level (lower half). Each estimate is based on 2500 random audio queries. For both categories the percentage of correct detections at rank 1 and the mean reciprocal rank (MRR) are given. Additionally, the mean length of the query in seconds and the mean execution time for a query is shown.

Query Length in Notes	Without Verification				With Verification			
	10	15	20	25	10	15	20	25
Correct Piece as Top Match	0.30	0.40	<b>0.41</b>	0.40	0.43	0.63	0.71	<b>0.75</b>
Correct Piece MRR	0.36	0.47	<b>0.50</b>	0.49	0.49	0.69	0.76	<b>0.79</b>
Correct Position as Top Match	0.23	<b>0.33</b>	0.32	0.32	0.33	0.51	0.57	<b>0.60</b>
Correct Position MRR	0.29	0.40	<b>0.41</b>	0.40	0.41	0.59	0.66	<b>0.69</b>
Mean Query Length in Seconds	1.47	2.26	3.16	3.82	1.47	2.26	3.16	3.82
Mean Query Execution Time in Seconds	0.10	0.32	0.62	0.91	0.12	0.38	0.72	1.09

**Table 3.** Results for different query sizes of the proposed *tempo- and transposition-invariant* piece and score position identification algorithm on the test database with (right) and without (left) the proposed verification step. Each estimate is based on 2500 random audio queries. The upper half shows recognition results on the piece level, the lower half on the score position level. For both categories the percentage of correct detections at rank 1 and the mean reciprocal rank (MRR) are given. Additionally, the mean length of the query in seconds and the mean execution time for a query is shown.

Query Length in Notes	Tempo-invariant				Tempo- and Pitch-invariant			
	No Tracking		Tracking		No Tracking		Tracking	
	50	100	50	100	50	100	50	100
Correct Piece as Top Match	0.95	0.96	0.98	<b>1</b>	0.81	0.79	0.92	<b>0.98</b>
Correct Piece MRR	0.97	0.98	0.99	<b>1</b>	0.85	0.82	0.94	<b>0.99</b>
Correct Position as Top Match	0.78	0.73	0.87	<b>0.88</b>	0.64	0.59	0.77	<b>0.83</b>
Correct Position MRR	0.85	0.81	0.89	<b>0.90</b>	0.72	0.66	0.82	<b>0.86</b>
Mean Query Length in Seconds	7.62	15.03	7.62	15.03	7.62	15.03	7.62	15.03
Mean Query Execution Time in Seconds	0.42	0.92	0.49	1.08	2.71	6.11	3.21	7.09

**Table 4.** Results of the proposed tracking algorithm on the test database for both the original *tempo-invariant algorithm* (left) and the new *tempo- and transposition-invariant approach* (right), including the verification step. For the category ‘No Tracking’, the query was fed directly to the fingerprinting algorithm. For ‘Tracking’, the queries were split into sub-queries with a window size of 15 notes and a hop size of 5 notes, and the individual results were tracked by our proof-of-concept *multi-agent* approach. Evaluation of the tracking approach is based on the finding the endpoint of a query (see text). Each estimate is based on 2500 random audio queries. The upper half shows recognition results on the piece level, the lower half on the score position level. For both categories the percentage of correct detections at rank 1 and the mean reciprocal rank (MRR) are given. Additionally, the mean length of the query in seconds and the mean execution time for a query is shown.

on-line algorithm, we are not interested in the start position of the query in the score, but in the endpoint, i.e. if the query was tracked successfully, and the correct *current* position is returned. Even the causal approach leads to a high percentage of correct results with both the original and the tempo- and pitch-invariant fingerprinting algorithm. Most of the remaining mistakes happen because (very) similar parts within one and the same piece are confused.

## 6. CONCLUSIONS

### 6.1 Applications

The proposed algorithm is useful in a wide range of applications. As a retrieval algorithm it enables fast and robust (inter- and intra-document) searching and browsing in large collections of musical scores and corresponding performances. Furthermore, we believe that the algorithm is not limited to retrieval tasks in classical music, but may be of use for cover version identification in general, and possibly many other tasks. For example, it was already successfully applied in the field of symbolic music processing to find repeating motifs and sections in complex musical scores [5].

Currently, the algorithm is mainly used in an on-line scenario (see [1]). In connection with a score following algorithm it can act as a ‘piano music companion’. The system is able to recognise arbitrary pieces of classical piano music, identify the position in the score and track the progress of the performer. This enables a wide range of applications for musicians and for consumers of classical music.

### 6.2 Future Work

In its current state the algorithm is able to recognise the correct piece and the score position even for very short queries of piano music. It is invariant to both tempo differences and transpositions and can be used in on-line contexts (i.e. to monitor audio streams and at any time report what it is listening to) and as an off-line retrieval algorithm. The main direction for future work is to lift the restriction to piano music and make it applicable to all kinds of classical music, even orchestral music. The limiting component at the moment is the transcription algorithm, which is only trained on piano sounds.

## 7. ACKNOWLEDGMENTS

This research is supported by the Austrian Science Fund (FWF) under project number Z159 and the EU FP7 Project PHENICX (grant no. 601166).

## 8. REFERENCES

- [1] A. Arzt, S. Böck, S. Flossmann, H. Frostel, M. Gasser, and G. Widmer. The complete classical music companion v0. 9. In *Proceedings of the 53rd AES Conference on Semantic Audio*, 2014.
- [2] A. Arzt, S. Böck, and G. Widmer. Fast identification of piece and score position via symbolic fingerprinting. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2012.
- [3] S. Böck and M. Schedl. Polyphonic piano note transcription with recurrent neural networks. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2012.
- [4] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. A review of algorithms for audio fingerprinting. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2002.
- [5] T. Collins, A. Arzt, S. Flossmann, and G. Widmer. Siarct-cfp: Improving precision and the discovery of inexact musical patterns in point-set representations. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2013.
- [6] S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39–58, 2001.
- [7] S. Flossmann, W. Goebel, M. Grachten, B. Niedermayer, and G. Widmer. The Magaloff project: An interim report. *Journal of New Music Research*, 39(4):363–377, 2010.
- [8] F. Kurth and M. Müller. Efficient index-based audio matching. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):382–395, 2008.
- [9] D. Lang, D. W. Hogg, K. Mierle, M. Blanton, and S. Roweis. Astrometry. net: Blind astrometric calibration of arbitrary astronomical images. *The Astronomical Journal*, 139(5):1782, 2010.
- [10] M. Müller, F. Kurth, and M. Clausen. Audio matching via chroma-based statistical features. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2005.
- [11] J. Serra, E. Gómez, and P. Herrera. Audio cover song identification and similarity: background, approaches, evaluation, and beyond. In Z. W. Ras and A. A. Wiczkowska, editors, *Advances in Music Information Retrieval*, pages 307–332. Springer, 2010.
- [12] R. Sonnleitner and G. Widmer. Quad-based audio fingerprinting robust to time and frequency scaling. In *Proceedings of the International Conference on Digital Audio Effects*, 2014.
- [13] A. Wang. An industrial strength audio search algorithm. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2003.
- [14] G. Widmer. Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. *Artificial Intelligence*, 146(2):129–148, 2003.

# GENDER IDENTIFICATION AND AGE ESTIMATION OF USERS BASED ON MUSIC METADATA

**Ming-Ju Wu**

Computer Science Department  
National Tsing Hua University  
Hsinchu, Taiwan  
brian.wu@mirlab.org

**Jyh-Shing Roger Jang**

Computer Science Department  
National Taiwan University  
Taipei, Taiwan  
roger.jang@mirlab.org

**Chun-Hung Lu**

Innovative Digitech-Enabled Applications  
& Services Institute (IDEAS),  
Institute for Information Industry,  
Taipei, Taiwan  
enricoghlu@iii.org.tw

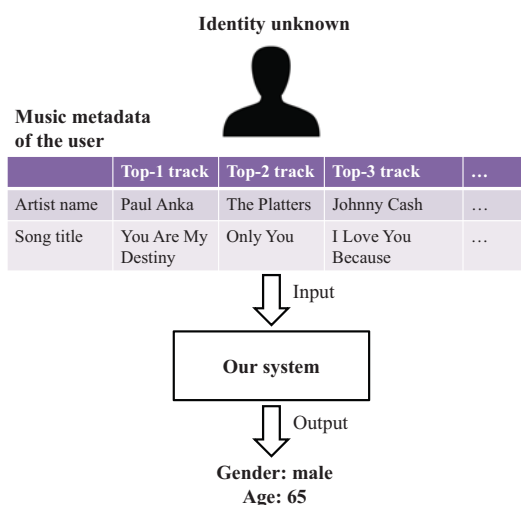
## ABSTRACT

Music recommendation is a crucial task in the field of music information retrieval. However, users frequently withhold their real-world identity, which creates a negative impact on music recommendation. Thus, the proposed method recognizes users' real-world identities based on music metadata. The approach is based on using the tracks most frequently listened to by a user to predict their gender and age. Experimental results showed that the approach achieved an accuracy of 78.87% for gender identification and a mean absolute error of 3.69 years for the age estimation of 48403 users, demonstrating its effectiveness and feasibility, and paving the way for improving music recommendation based on such personal information.

## 1. INTRODUCTION

Amid the rapid growth of digital music and mobile devices, numerous online music services (e.g., Last.fm, 7digital, Grooveshark, and Spotify) provide music recommendations to assist users in selecting songs. Most music-recommendation systems are based on content- and collaborative-based approaches [15]. For content-based approaches [2, 8, 9], recommendations are made according to the audio similarity of songs. By contrast, collaborative-based approaches involve recommending music for a target user according to matched listening patterns that are analyzed from massive users [1, 13].

Because music preferences of users relate to their real-world identities [12], several collaborative-based approaches consider identification factors such as age and gender for music recommendation [14]. However, online music services may experience difficulty obtaining such information. Conversely, music metadata (listening history) is generally available. This motivated us to recognize users' real-world identities based on music



**Figure 1.** Illustration of the proposed system using a real example.

metadata. Figure 1 illustrates the proposed system. In this preliminary study, we focused on predicting gender and age according to the most listened songs. In particular, gender identification was treated as a binary-classification problem, whereas age estimation was considered a regression problem. Two features were applied for both gender identification and age estimation tasks. The first feature, TF\*IDF, is a widely used feature representation in natural language processing [16]. Because the music metadata of each user can be considered directly as a document, gender identification can be viewed as a document categorization problem. In addition, TF\*IDF is generally applied with latent semantic indexing (LSI) to reduce feature dimension. Consequently, this serves as the baseline feature in this study.

The second feature, the Gaussian super vector (GSV) [3], is a robust feature representation for speaker verification. In general, the GSV is used to model acoustic features such as MFCCs. In this study, music metadata was translated into proposed hotness features (a bag-of-features representation) and could be modeled using the GSV. The concept of the GSV can be described as follows. First,



© Ming-Ju Wu, Jyh-Shing Roger Jang, Chun-Hung Lu.  
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Ming-Ju Wu, Jyh-Shing Roger Jang, Chun-Hung Lu. "Gender Identification and Age Estimation of Users Based on Music Metadata", 15th International Society for Music Information Retrieval Conference, 2014.

a universal background model (UBM) is trained using a Gaussian mixture model (GMM) to represent the global music preference of users. A user-specific GMM can then be obtained using the maximum a posteriori (MAP) adaptation from the UBM. Finally, the mean vectors of the user-specific GMM are applied as GSV features.

The remainder of this paper is organized as follows: Section 2 describes the related literature, and Section 3 introduces the TF\*IDF; the GSV is explained in Section 4, and the experimental results are presented in Section 5; finally, Section 6 provides the conclusion of this study.

## 2. RELATED LITERATURE

Machine learning has been widely applied to music information retrieval (MIR), a vital task of which is content-based music classification [5, 11]. For example, the annual Music Information Retrieval Evaluation eXchange (MIREX) competition has been held since 2004, at which some of the most popular competition tasks have included music genre classification, music mood classification, artist identification, and tag annotation. The purpose of content-based music classification is to recognize semantic music attributes from audio signals. Generally, songs are represented by features with different aspects such as timbre and rhythm. Classifiers are used to identify the relationship between low-level features and mid-level music metadata.

However, little work has been done on predicting personal traits based on music metadata [7]. Figure 2 shows a comparison of our approach and content-based music classification. At the top level, user identity provides a basic description of users. At the middle level, music metadata provides a description of music. A semantic gap exists between music metadata and user identity. Beyond content-based music classification, our approach serves as a bridge between them. This enables online music services to recognize unknown users more effectively and, consequently, improve their music recommendations.

## 3. TF\*IDF FEATURE REPRESENTATION

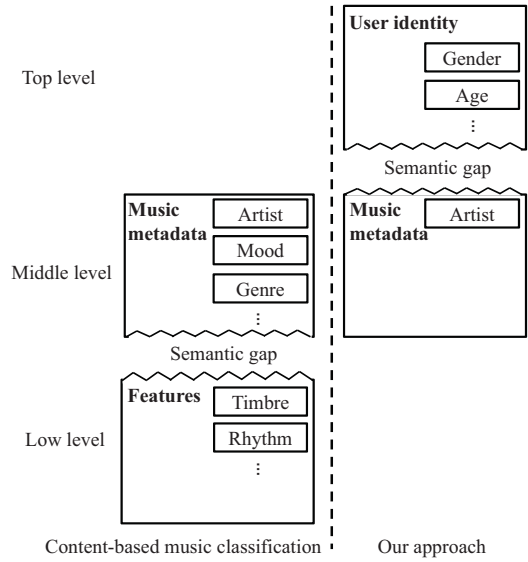
The music metadata of each user can be considered a document. The TF\*IDF describes the relative importance of an artist for a specific document. LSI is then applied for dimensionality reduction.

### 3.1 TF\*IDF

Let the document (music metadata) of each user in the training set be denoted as

$$d_i = \{t_1, t_2, \dots, t_n\}, d_i \in D \quad (1)$$

where  $t_n$  is the artist name of the top- $n$  listened to song of user  $i$ .  $D$  is the collection of all documents in the training set. The TF\*IDF representation is composed of the term frequency (TF) and inverse document frequency (IDF). TF indicates the importance of an artist for a particular document, whereas IDF indicates the discriminative power



**Figure 2.** Comparison of our approach and content-based music classification.

of an artist among documents. The TF\*IDF can be expressed as

$$tfidf_{i,n} = tf_{i,n} \times \log \left( \frac{|D|}{df_n} \right) \quad (2)$$

where  $tf_{i,n}$  is the frequency of  $t_n$  in  $d_i$ , and  $df_n$  represents the number of documents in which  $t_n$  appears.

$$df_n = |\{d : d \in D \text{ and } t_n \in d\}| \quad (3)$$

### 3.2 Latent Semantic Indexing

The TF\*IDF representation scheme leads to high feature dimensionality because the feature dimension is equal to the number of artists. Therefore, LSI is generally applied to transform data into a lower-dimensional semantic space. Let  $W$  be the TF\*IDF reorientation of  $D$ , where each column represents document  $d_i$ . The LSI performs singular value decomposition (SVD) as follows:

$$W \approx U \Sigma V^T \quad (4)$$

where  $U$  and  $V$  represent terms and documents in the semantic space, respectively.  $\Sigma$  is a diagonal matrix with corresponding singular values.  $\Sigma^{-1}U^T$  can be used to transform new documents into the lower-dimensional semantic space.

## 4. GSV FEATURE REPRESENTATION

This section introduces the proposed hotness features and explains how to generate the GSV features based on hotness features.

### 4.1 Hotness Feature Extraction

We assumed each artist  $t_n$  may exude various degrees of hotness to different genders and ages. For example, the

count (the number of times) of Justin Bieber that occurs in users' top listened to songs of the training set was 845, where 649 was from the female class and 196 was from the male class. We could define the hotness of Justin Bieber for females as 76.80% (649/845) and that for males as 23.20% (196/845). Consequently, a user tends to be a female if her top listened to songs related mostly to Justin Bieber. Consequently, the age and gender characteristics of a user can be obtained by computing the hotness features of relevant artists.

Let  $D$  be divided into classes  $C$  according to users' genders or ages:

$$\begin{cases} C_1 \cup C_2 \cup \dots \cup C_p = D \\ C_1 \cap C_2 \cap \dots \cap C_p = \emptyset \end{cases} \quad (5)$$

where  $p$  is the number of classes. Here,  $p$  is 2 for gender identification and 51 (the range of age) for age estimation. The hotness feature of each artist  $t_n$  is defined as  $h_n$ :

$$h_n = \begin{bmatrix} \frac{c_{n,1}}{\alpha} \\ \frac{c_{n,2}}{\alpha} \\ \vdots \\ \frac{c_{n,p}}{\alpha} \end{bmatrix} \quad (6)$$

where  $c_{n,p}$  is the count of artist  $t_n$  in  $C_p$ , and  $\alpha$  is the count of artist  $t_n$  in all classes.

$$\alpha = \sum_{l=1}^p c_{n,l} \quad (7)$$

Next, each document in (1) can be transformed to a  $p \times n$  matrix  $x$ , which describes the gender and age characteristics of a user:

$$x = [h_1, h_2, \dots, h_n] \quad (8)$$

Because the form of  $x$  can be considered a bag-of-features, the GSV can be applied directly.

## 4.2 GSV Feature Extraction

Figure 3 is a flowchart of the GSV feature extraction, which can be divided into offline and online stages. At the offline stage, the goal is to construct a UBM [10] to represent the global hotness features, which are then used as prior knowledge for each user at the online stage. First, hotness features are extracted for all music metadata in the training set. The UBM is then constructed through a GMM estimated using the EM (expectation-maximization) algorithm. Specifically, the UBM evaluates the likelihood of a given feature vector  $x$  as follows:

$$f(x|\theta) = \sum_{k=1}^K w_k \mathcal{N}(x|m_k, r_k) \quad (9)$$

where  $\theta = (w_1, \dots, w_K, m_1, \dots, m_K, r_1, \dots, r_K)$  is a set of parameters, with  $w_k$  denoting the mixture gain for the  $k$ th mixture component, subject to the constraint  $\sum_{k=1}^K w_k = 1$ , and  $\mathcal{N}(x|m_k, r_k)$  denoting the Gaussian density function with a mean vector  $m_k$  and a covariance

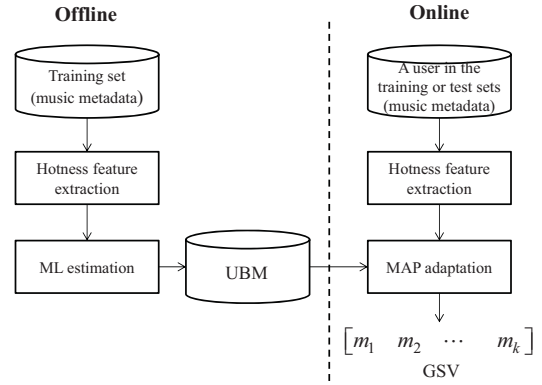


Figure 3. Flowchart of the GSV feature extraction.

matrix  $r_k$ . This bag-of-features model is based on the assumption that similar users have similar global artist characteristics.

At the online stage, the MAP adaptation [6] is used to produce an adapted GMM for a specific user. Specifically, MAP attempts to determine the parameter  $\theta$  in the parameter space  $\Theta$  that maximizes the posterior probability given the training data  $x$  and hyperparameter  $\omega$ , as follows:

$$\theta_{MAP} = \arg \max_{\theta} f(x|\theta) g(\theta|\omega) \quad (10)$$

where  $f(x|\theta)$  is the probability density function (PDF) for the observed data  $x$  given the parameter  $\theta$ , and  $g(\theta|\omega)$  is the prior PDF given the hyperparameter  $\omega$ .

Finally, for each user, the mean vectors of the adapted GMM are stacked to form a new feature vector called GSV. Because the adapted GMM is obtained using MAP adaptation over the UBM, it is generally more robust than directly modeling the feature vectors by using GMM without any prior knowledge.

## 5. EXPERIMENTAL RESULTS

This section describes data collection, experimental settings, and experimental results.

### 5.1 Data Collection

The Last.fm API was applied for data set collection, because it allows anyone to access data including albums, tracks, users, events, and tags. First, we collected user IDs through the *User.getFriends* function. Second, the *User.getInfo* function was applied to each user for obtaining their age and gender information. Finally, the *User.getTopTracks* function was applied to acquire at most top-50 tracks listened to by a user. The track information included song titles and artist names, but only artist names were used for feature extraction in this preliminary study.

The final collected data set included 96807 users, in which each user had at least 40 top tracks as well as complete gender and age information. According to the users' country codes, they were from 211 countries (or

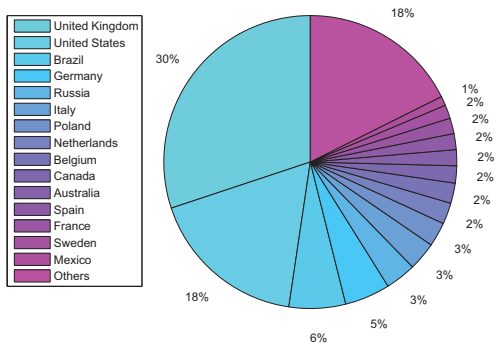


Figure 4. Ratio of countries of the collected data set.

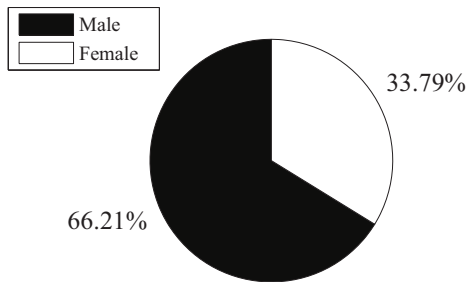


Figure 5. Gender ratio of the collected data set.

regions such as Hong Kong). The ratio of countries is shown in Figure 4. The majority were Western countries. The gender ratio is shown in Figure 5, in which approximately one-third of users (33.79%) were female and two-thirds (66.21%) were male. The age distribution of users is shown in Figure 6. The distribution was a skewed normal distribution and most users were young people.

Figure 7 shows the count of each artist that occurred in the users' top listened songs. Among 133938 unique artists in the data set, the ranking of popularity presents a pow-law distribution. This demonstrates that a few artists dominate the top listened songs. Although the majority of artists are not popular for all users, this does not indicate that they are unimportant, because their hotness could be discriminative over ages and gender.

## 5.2 Experimental Settings

The data set was equally divided into two subsets, the training (48404) and test (48403) sets. An open source tool of *Python*, *Gensim*, was applied for the TF\*IDF and LSI implementation. followed the default setting of *Gensim* that maintained 200 latent dimensions for the TF\*IDF. A support vector machine (SVM) tool, LIBSVM [4], was applied as the classifier. The SVM extension, support vector regression (SVR) was applied as the regressor, which has been observed in many cases to be superior

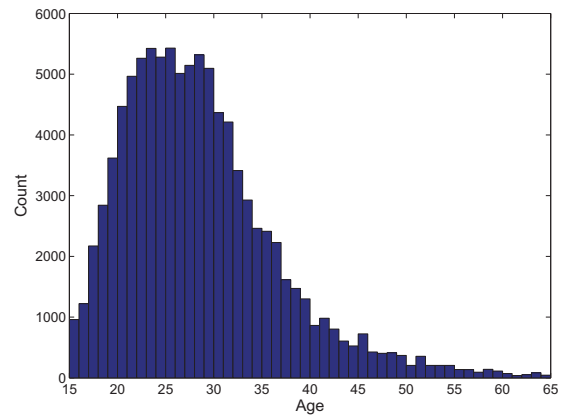


Figure 6. Age distribution of the collected data set.

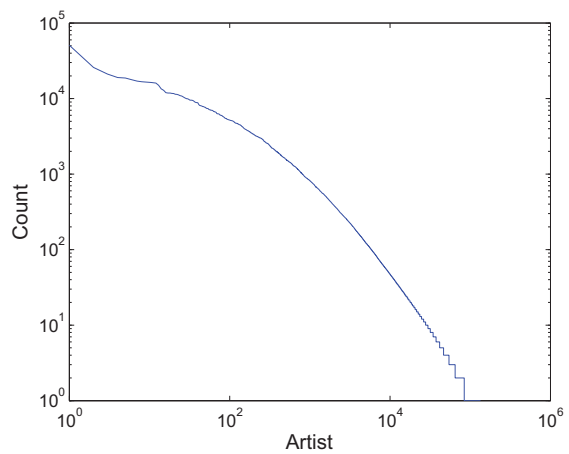


Figure 7. Count of artists of users' top listened songs. Ranking of popularity presents a pow-law distribution.

to existing regression approaches. The RBF kernel with  $\gamma = 8$  was applied to the SVM and SVR. For the UBM parameters, two Gaussian mixture components were experimentally applied (similar results can be obtained when using a different number of mixture components). Consequently, the numbers of dimensions of GSV features for gender identification and age estimation were 4 ( $2 \times 2$ ) and 102 ( $2 \times 51$ ), respectively.

## 5.3 Gender Identification

The accuracy was 78.87% and 78.21% for GSV and TF\*IDF + LSI features, respectively. This indicates that both features are adequate for such a task. Despite the low dimensionality of GSV (4), it was superior to the high dimensionality of TF\*IDF + LSI (200). This indicates the effectiveness of GSV use and the proposed hotness features. Figures 8 and 9 respectively show the confusion matrix of using GSV and TF\*IDF + LSI features. Both features yielded higher accuracies for the male class than for the female class. A possible explanation is that a portion of the females' were similar to the males'. The classifier tended to favor the majority class

(male), resulting in many female instances with incorrect predictions. The age difference can also be regarded for further analysis. Figure 10 shows the gender identification results of two features over various ages. Both features tended to have lower accuracies between the ages of 25 and 40 years, implying that a user whose age is between 25 and 40 years seems to have more blurred gender boundaries than do users below 25 years and above 40 years.

#### 5.4 Age Estimation

Table 1 shows the performance comparison for age estimation. The mean absolute error (MAE) was applied as the performance index. The range of the predicted ages of the SVR is between 15 and 65 years. The experimental results show that the MAE is 3.69 and 4.25 years for GSV and TF\*IDF + LSI, respectively. The GSV describes the age characteristics of a user and utilizes prior knowledge from the UBM; therefore, the GSV features are superior to

Method	MAE	MAE (male)	MAE (female)
GSV	3.69	4.31	2.48
TF*IDF+LSI	4.25	4.86	3.05

**Table 1.** Performance comparison for age estimation.

those of the TF\*IDF + LSI.

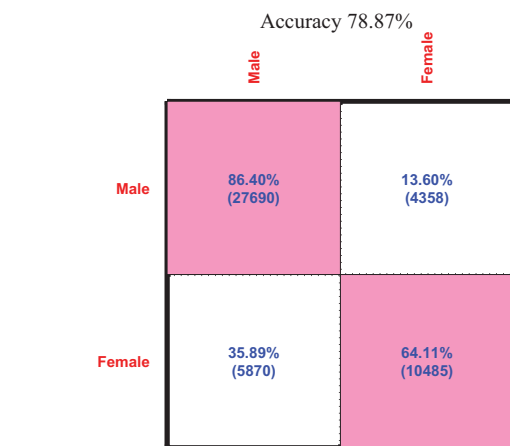
For further analysis, gender difference was also considered. Notably, the MAE of females is less than that of males for both GSV and TF\*IDF + LSI features. In particular, the MAE differences between males and females are approximately 1.8 for both features, implying that females have more distinct age divisions than males do.

#### 6. CONCLUSION AND FUTURE WORK

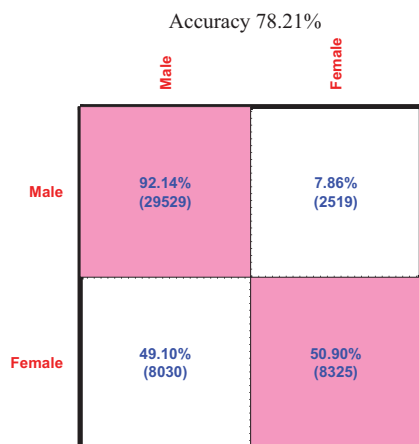
This study confirmed the possibility of predicting users' age and gender based on music metadata. Three of the findings are summarized as follows.

- GSV features are superior to those of TF\*IDF + LSI for both gender identification and age estimation tasks.
- Males tend to exhibit higher accuracy than females do in gender identification, whereas females are more predictable than males in age estimation.
- The experimental results indicate that gender identification is influenced by age, and vice versa. This suggests that an implicit relationship may exist between them.

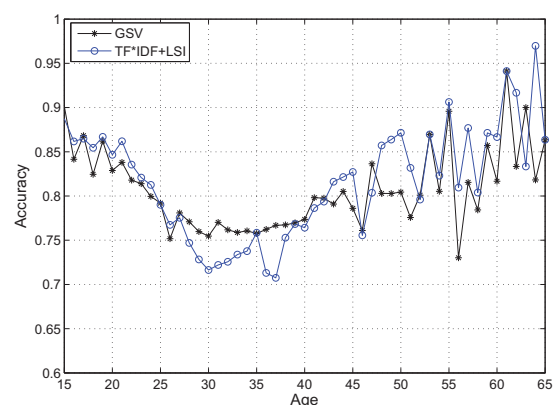
Future work could include utilizing the proposed approach to improve music recommendation systems. We will also explore the possibility of recognizing deeper social aspects of user identities, such as occupation and education level.



**Figure 8.** Confusion matrix of gender identification by using GSV features.



**Figure 9.** Confusion matrix of gender identification by using TF\*IDF + LSI features.



**Figure 10.** Gender identification results for various ages.

## 7. ACKNOWLEDGEMENT

This study is conducted under the "NSC 102-3114-Y-307-026 A Research on Social Influence and Decision Support Analytics" of the Institute for Information Industry which is subsidized by the National Science Council.

## 8. REFERENCES

- [1] L. Barrington, R. Oda, and G. Lanckriet. Smarter than genius? human evaluation of music recommender systems. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 357–362, 2009.
- [2] D. Bogdanov, M. Haro, F. Fuhrmann, E. Gomez, and P. Herrera. Content-based music recommendation based on user preference examples. In *Proceedings of the ACM Conf. on Recommender Systems. Workshop on Music Recommendation and Discovery*, 2010.
- [3] W. M. Campbell, D. E. Sturim, and D. A. Reynolds. Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters*, 13(5):308–311, May 2006.
- [4] C. C. Chang and C. J. Lin. Libsvm: A library for support vector machine, 2010.
- [5] Z. Fu, G. Lu, K. M. Ting, and D. Zhang. A survey of audio-based music classification and annotation. *IEEE Trans. Multimedia.*, 13(2):303–319, Apr. 2011.
- [6] J. L. Gauvain and C. H. Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *IEEE Trans. Audio, Speech, Lang. Process.*, 2(2):291–298, Apr. 1994.
- [7] Jen-Yu Liu and Yi-Hsuan Yang. Inferring personal traits from music listening history. In *Proceedings of the Second International ACM Workshop on Music Information Retrieval with User-centered and Multimodal Strategies*, MIRUM '12, pages 31–36, New York, NY, USA, 2012. ACM.
- [8] B. McFee, L. Barrington, and G. Lanckriet. Learning content similarity for music recommendation. *IEEE Trans. Audio, Speech, Lang. Process.*, 20(8):2207–2218, Oct. 2012.
- [9] A. V. D. Orrd, S. Dieleman, and B. Benjamin. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems*, 2013.
- [10] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Process*, 10(13):19–41, Jan. 2000.
- [11] B. L. Sturm. A survey of evaluation in music genre recognition. In *Proceedings of the Adaptive Multimedia Retrieval*, 2012.
- [12] A. Uitdenbogerd and R. V. Schnydel. A review of factors affecting music recommender success. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 204–208, 2002.
- [13] B. Xu, J. Bu, C. Chen, and D. Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web*, pages 21–30, 2012.
- [14] Billy Yapriady and AlexandraL. Uitdenbogerd. Combining demographic data with collaborative filtering for automatic music recommendation. In *Knowledge-Based Intelligent Information and Engineering Systems*, volume 3684 of *Lecture Notes in Computer Science*, pages 201–207. 2005.
- [15] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno. Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 296–301, 2006.
- [16] W. Zhang, T. Yoshida, and X. Tang. A comparative study of tf\*idf, lsi and multi-words for text classification. *Expert Systems with Applications*, 38(3):2758–2765, 2011.



# INFORMATION-THEORETIC MEASURES OF MUSIC LISTENING BEHAVIOUR

Daniel Boland, Roderick Murray-Smith

School of Computing Science, University of Glasgow, United Kingdom

daniel@dcs.gla.ac.uk; roderick.murray-smith@glasgow.ac.uk

## ABSTRACT

We present an information-theoretic approach to the measurement of users' music listening behaviour and selection of music features. Existing ethnographic studies of music use have guided the design of music retrieval systems however are typically qualitative and exploratory in nature. We introduce the *SPUD* dataset, comprising 10,000 hand-made playlists, with user and audio stream metadata. With this, we illustrate the use of entropy for analysing music listening behaviour, e.g. identifying when a user changed music retrieval system. We then develop an approach to identifying music features that reflect users' criteria for playlist curation, rejecting features that are independent of user behaviour. The dataset and the code used to produce it are made available. The techniques described support a quantitative yet user-centred approach to the evaluation of music features and retrieval systems, without assuming objective ground truth labels.

## 1. INTRODUCTION

Understanding how users interact with music retrieval systems is of fundamental importance to the field of Music Information Retrieval (MIR). The design and evaluation of such systems is conditioned upon assumptions about users, their listening behaviours and their interpretation of music. While user studies have offered guidance to the field thus far, they are mostly exploratory and qualitative [20]. The availability of quantitative metrics would support the rapid evaluation and optimisation of music retrieval. In this work, we develop an information-theoretic approach to measuring users' music listening behaviour, with a view to informing the development of music retrieval systems.

To demonstrate the use of these measures, we compiled 'Streamable Playlists with User Data' (*SPUD*) – a dataset comprising 10,000 playlists from Last.fm<sup>1</sup> produced by 3351 users, with track metadata including audio streams from Spotify.<sup>2</sup> We combine the dataset with the mood and genre classification of Syntonetic's Moodagent,<sup>3</sup> yielding a range of intuitive music features to serve as examples.

We identify the entropy of music features as a metric for characterising music listening behaviour. This measure can be used to produce time-series analyses of user behaviour, allowing for the identification of events where this behaviour changed. In a case study, the date when a user adopted a different music retrieval system is detected. These detailed analyses of listening behaviour can support user studies or provide implicit relevance feedback to music retrieval. More broad analyses are performed across the 10,000 playlists. A Mutual Information based feature selection algorithm is employed to identify music features relevant to how users create playlists. This user-centred feature selection can sanity-check the choice of features in MIR. The information-theoretic approach introduced here is applicable to any discretisable feature set and distinct in being based solely upon actual user behaviour rather than assumed ground-truth. With the techniques described here, MIR researchers can perform quantitative yet user-centred evaluations of their music features and retrieval systems.

### 1.1 Understanding Users

User studies have provided insights about user behaviour in retrieving and listening to music and highlighted the lack of consideration in MIR about actual user needs. In 2003, Cunningham et al. bemoaned that development of music retrieval systems relied on "anecdotal evidence of user needs, intuitive feelings for user information seeking behavior, and a priori assumptions of typical usage scenarios" [5]. While the number of user studies has grown, the situation has been slow to improve. A review conducted a decade later noted that approaches to system evaluation still ignore the findings of user studies [12]. This issue is stated more strongly by Schedl and Flexer, describing systems-centric evaluations that "completely ignore user context and user properties, even though they clearly influence the result" [15]. Even systems-centric work, such as the development of music classifiers, must consider the user-specific nature of MIR. Downie termed this the multi-experiential challenge, and noted that "Music ultimately exists in the mind of its perceiver" [6]. Despite all of this, the assumption of an objective ground truth for music genre, mood etc. is common [4], with evaluations focusing on these rather than considering users. It is clear that much work remains in placing the user at the centre of MIR.



© Daniel Boland, Roderick Murray-Smith.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Daniel Boland, Roderick Murray-Smith. "Information-Theoretic Measures of Music Listening Behaviour", 15th International Society for Music Information Retrieval Conference, 2014.

1. <http://www.last.fm>

2. <http://www.spotify.com>

3. <http://www.moodagent.com> Last accessed: 30/04/14

## 1.2 Evaluation in MIR

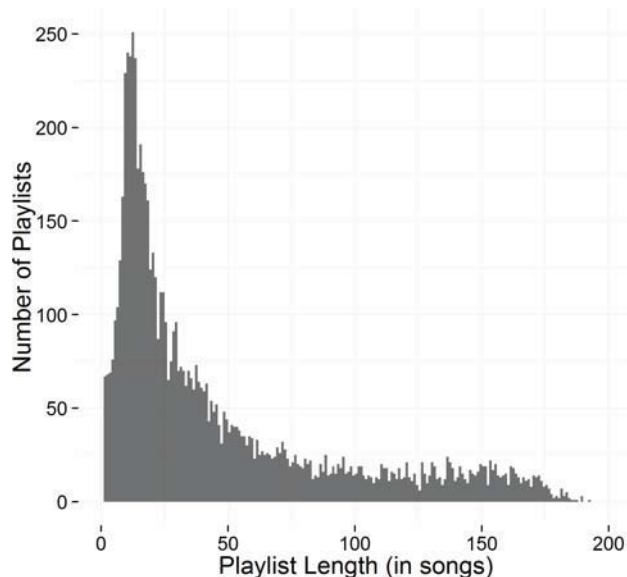
The lack of robust evaluations in the field of MIR was identified by Futrelle and Downie as early as 2003 [8]. They noted the lack of any standardised evaluations and in particular that MIR research commonly had an “emphasis on basic research over application to, and involvement with, users.” In an effort to address these failings, the Music Information Retrieval Evaluation Exchange (MIREX) was established [7]. MIREX provides a standardised framework of evaluation for a range of MIR problems using common metrics and datasets, and acts as the benchmark for the field. While the focus on this benchmark has done a great deal towards the standardisation of evaluations, it has distracted research from evaluations with real users.

A large amount of evaluative work in MIR focuses on the performance of classifiers, typically of mood or genre classes. A thorough treatment of the typical approaches to evaluation and their shortcomings is given by Sturm [17]. We note that virtually all such evaluations seek to circumvent involving users, instead relying on a ‘ground truth’ which is assumed to be objective. An example of a widely used ground truth dataset is *GTZAN*, a small collection of music with the author’s genre annotations. Even were the objectivity of such annotations to be assumed, such datasets can be subject to confounding factors and mislabellings as shown by Sturm [16]. Schedl et al. also observe that MIREX evaluations involve assessors’ own subjective annotations as ground truth [15].

## 1.3 User-Centred Approaches

There remains a need for robust, standardised evaluations featuring actual users of MIR systems, with growing calls for a more user-centric approach. Schedl and Flexer made the broad case for “putting the user in the center of music information retrieval”, concerning not only user-centred development but also the need for evaluative experiments which control independent variables that may affect dependent variables [14]. We note that there is, in particular, a need for quantitative dependent variables for user-centred evaluations. For limited tasks such as audio similarity or genre classification, existing dependent variables may be sufficient. If the field of MIR is to concern itself with the development of complete music retrieval systems, their interfaces, interaction techniques, and the needs of a variety of users, then additional metrics are required. Within the field of HCI it is typical to use qualitative methods such as the think-aloud protocol [9] or Likert-scale questionnaires such as the NASA Task Load Index (TLX) [10].

Given that the purpose of a Music Retrieval system is to support the user’s retrieval of music, a dependent variable to measure this ability is desirable. Such a measure cannot be acquired independently of users – the definition of musical relevance is itself subjective. Users now have access to ‘Big Music’ – online collections with millions of songs, yet it is unclear how to evaluate their ability to retrieve this music. The information-theoretic methodology introduced in this work aims to quantify the exploration, diversity and underlying mental models of users’ music retrieval.



**Figure 1.** Distribution of playlist lengths within the *SPUD* dataset. The distribution peaks around a playlist length of 12 songs. There is a long tail of lengthy playlists.

## 2. THE SPUD DATASET

The *SPUD* dataset of 10,000 playlists was produced by scraping from Last.fm users who were active throughout March and April, 2014. The tracks for each playlist are also associated with a Spotify stream, with scraped metadata, such as artist, popularity, duration etc. The number of unique tracks in the dataset is 271,389 from 3351 users. The distribution of playlist lengths is shown in Figure 1. We augment the dataset with proprietary mood and genre features produced by Syntonetic’s Moodagent. We do this to provide high-level and intuitive features which can be used as examples to illustrate the techniques being discussed. It is clear that many issues remain with genre and mood classification [18] and the results in this work should be interpreted with this in mind. Our aim in this work is not to identify which features are best for music classification but to contribute an approach for gaining an additional perspective on music features. Another dataset of playlists *AOTM-2011* is published [13] however the authors only give fragments of playlists where songs are also present in the Million Song Dataset (*MSD*) [1]. The *MSD* provides music features for a million songs but only a small fraction of songs in *AOTM-2011* were matched in *MSD*. Our *SPUD* dataset is distinct in maintaining complete playlists and having time-series data of songs listened to.

## 3. MEASURING MUSIC LISTENING BEHAVIOUR

When evaluating a music retrieval system, or performing a user study, it would be useful to quantify the music-listening behaviour of users. Studying this behaviour over time would enable the identification of how different music retrieval systems influence user behaviour. Quantifying listening behaviour would also provide a dependent variable for use in MIR evaluations. We introduce entropy as one such quantitative measure, capturing how a user’s music-listening relates to the music features of their songs.

### 3.1 Entropy

For each song being played by a user, the value of a given music feature can be taken as a random variable  $X$ . The entropy  $H(X)$  of this variable indicates the uncertainty about the value of that feature over multiple songs in a listening session. This entropy measure gives a scale from a feature's value never changing, through to every level of the feature being equally likely. The more a user constrains their music selection by a particular feature, e.g. mood or album, then the lower the entropy is over those features. The entropy for a feature is defined as:

$$H(X) = - \sum_{x \in X} p(x) \log_2[p(x)], \quad (1)$$

where  $x$  is every possible level of the feature  $X$  and the distribution  $p(x)$  is estimated from the songs in the listening session. The resulting entropy value is measured in bits, though can be normalised by dividing by the maximum entropy  $\log_2[|X|]$ . Estimating entropy in this way can be done for any set of features, though requires that they are discretised to an appropriate number of levels.

For example, if a music listening session is dominated by songs of a particular tempo, the distribution over values of a TEMPO feature would be very biased. The entropy  $H(\text{TEMPO})$  would thus be very low. Conversely, if users used shuffle or listened to music irrespective of tempo, then the entropy  $H(\text{TEMPO})$  would tend towards the average entropy of the whole collection.

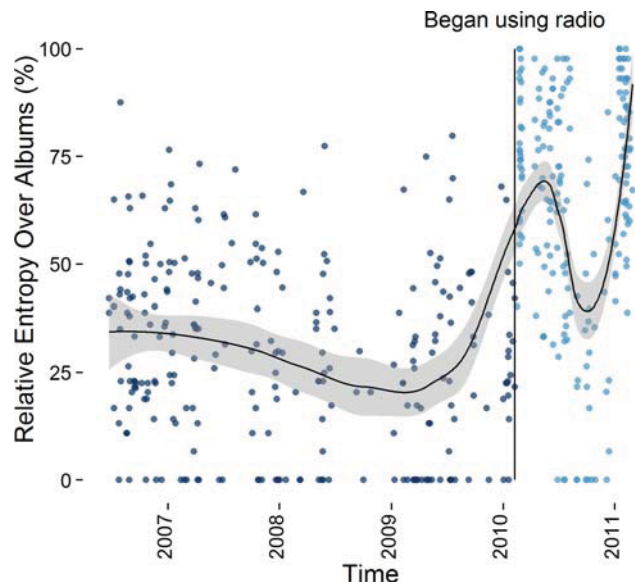
### 3.2 Applying a Window Function

Many research questions regarding a user's music listening behaviour concern the change in that behaviour over time. An evaluation of a music retrieval interface might hypothesise that users will be empowered to explore a more diverse range of music. Musicologists may be interested to study how listening behaviour has changed over time and which events precede such changes. It is thus of interest to extend Eqn (1) to define a measure of entropy which is also a function of time:

$$H(X, t) = H(w(X, t)), \quad (2)$$

where  $w(X, t)$  is a window function taking  $n$  samples of  $X$  around time  $t$ . In this paper we use a rectangular window function with  $n = 20$ , assuming that most albums will have fewer tracks than this. The entropy at any given point is limited to the maximum possible  $H(X, t) = \log_2[n]$  i.e. where each of the  $n$  points has a unique value.

An example of the change in entropy for a music feature over time is shown in Figure 2. In this case  $H(\text{ALBUM})$  is shown as this will be 0 for album-based listening and at maximum for exploratory or radio-like listening. It is important to note that while trends in mean entropy can be identified, the entropy of music listening is itself quite a noisy signal – it is unlikely that a user will maintain a single music-listening behaviour over a large period of time. Periods of album listening (low or zero entropy) can be seen through the time-series, even after the overall trend is towards shuffle or radio-like music listening.



**Figure 2.** Windowed entropy over albums shows a user's album-based music listening over time. Each point represents 20 track plays. The black line depicts mean entropy, calculated using locally weighted regression [3] with 95% CI of the mean shaded. A changepoint is detected around Feb. 2010, as the user began using online radio (light blue)

### 3.3 Changepoints in Music Retrieval

Having produced a time-series analysis of music-listening behaviour, we are now able to identify events which caused changes in this behaviour. In order to identify changepoints in the listening history, we apply the 'Pruned Exact Linear Time' (PELT) algorithm [11]. The time-series is partitioned in a way that reduces a cost function of changes in the mean and variance of the entropy. Changepoints can be of use in user studies, for example in Figure 2, the user explained in an interview that the detected changepoint occurred when they switched to using online radio. There is a brief return to album-based listening after the changepoint – users' music retrieval behaviour can be a mixture of different retrieval models. Changepoint detection can also be a user-centred dependent variable in evaluating music retrieval interfaces i.e. does listening behaviour change as the interface changes? Further examples of user studies are available with the *SPUD* dataset.

### 3.4 Identifying Listening Style

The style of music retrieval that the user is engaging in can be inferred using the entropy measures. Where the entropy for a given music feature is low, the user's listening behaviour can be characterised by that feature i.e. we can be certain about that feature's level. Alternately, where a feature has high entropy, then the user is not 'using' that feature in their retrieval. When a user opts to use shuffle-based playback i.e. the random selection of tracks, there is the unique case that entropy across all features will tend towards the maximum. In many cases, feature entropies have high covariance, e.g. songs on an album will have the same artist and similar features. We did not include other features in Figure 2 as the same pattern was apparent.

#### 4. SELECTING FEATURES FROM PLAYLISTS

Identifying which music features best describe a range of playlists is not only useful for playlist recommendation, but also provides an insight into how users organise and think about music. Music recommendation and playlist generation typically work on the basis of genre, mood and popularity, and we investigate which of these features is supported by actual user behaviour. As existing retrieval systems are based upon these features, there is a potential ‘chicken-and-egg’ effect where the features which best describe user playlists are those which users are currently exposed to in existing retrieval interfaces.

##### 4.1 Mutual Information

Information-theoretic measures can be used to identify to what degree a feature shares information with class labels. For a feature  $X$  and a class label  $Y$ , the mutual information  $I(X; Y)$  between these two can be given as:

$$I(X; Y) = H(X) - H(X|Y), \quad (3)$$

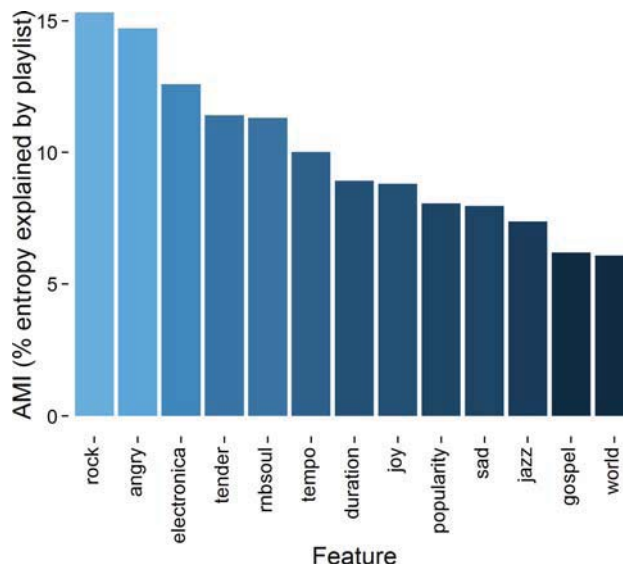
that is, the entropy of the feature  $H(X)$  minus the entropy of that feature if the class is known  $H(X|Y)$ . By taking membership of playlists as a class label, we can determine how much we can know about a song’s features if we know what playlist it is in. When using mutual information to compare clusterings in this way, care must be taken to account for random chance mutual information [19]. We adapt this approach to focus on how much the feature entropy is reduced, and normalise accordingly:

$$AMI(X; Y) = \frac{I(X; Y) - E[I(X; Y)]}{H(X) - E[I(X; Y)]}, \quad (4)$$

where  $AMI(X; Y)$  is the adjusted mutual information and  $E[I(X; Y)]$  is the expectation of the mutual information i.e. due to random chance. The AMI gives a normalised measure of how much of the feature’s entropy is explained by the playlist. When  $AMI = 1$ , the feature level is known exactly if the playlist is known, when  $AMI = 0$ , nothing about the feature is known if the playlist is known.

##### 4.2 Linking Features to Playlists

We analysed the AMI between the 10,000 playlists in the *SPUD* dataset and a variety of high level music features. The ranking of some of these features is given in Figure 3. Our aim is only to illustrate this approach, as any results are only as reliable as the underlying features. With this in mind, the features ROCK and ANGRY had the most uncertainty explained by playlist membership. While the values may seem small, they are calculated over many playlists, which may combine moods, genres and other criteria. As these features change most between playlists (rather than within them), they are the most useful for characterising the differences between playlists. The DURATION feature ranked higher than expected, further investigation revealed playlists that combined lengthy DJ mixes. It is perhaps unsurprising that playlists were not well characterised by whether they included WORLD music.



**Figure 3.** Features are ranked by their Adjusted Mutual Information with playlist membership. Playlists are distinguished more by whether they contain ROCK or ANGRY music than by whether they contain POPULAR or WORLD.

It is of interest that TEMPO was not one of the highest ranked features, illustrating the style of insights available when using this approach. Further investigation is required to determine whether playlists are not based on tempo as much as is often assumed or if this result is due to the peculiarities of the proprietary perceptual tempo detection.

##### 4.3 Feature Selection

Features can be selected using information-theoretic measures, with a rigorous treatment of the field given by Brown et al. [2]. They define a unifying framework within which to discuss methods for selecting a subset of features using mutual information. This is done by defining a  $J$  criterion for a feature:

$$J(f_n) = I(f_n; C | S). \quad (5)$$

This gives a measure of how much information the feature shares with playlists given some previously selected features, and can be used as a greedy feature selection algorithm. Intuitively, features should be selected that are relevant to the classes but that are also not redundant with regard to previously selected features. A range of estimators for  $I(f_n; C | S)$  are discussed in [2].

As a demonstration of the feature selection approach we have described, we apply it to the features depicted in Figure 3, selecting features to minimise redundancy. The selected subset of features in rank order is: ROCK, DURATION, POPULARITY, TENDER and JOY. It is notable that ANGRY had an AMI that was almost the same as ROCK, but it is redundant if ROCK is included. Unsurprisingly, the second feature selected is from a different source than the first – the duration information from Spotify adds to that used to produce the Syntonetic mood and genre features. Reducing redundancy in the selected features in this way yields a very different ordering, though one that may give a clearer insight into the factors behind playlist construction.

## 5. DISCUSSION

While we reiterate that this work only uses a specific set of music features and user base, we consider our results to be encouraging. It is clear that the use of entropy can provide a detailed time-series analysis of user behaviour and could prove a valuable tool for MIR evaluation. Similarly, the use of adjusted mutual information allows MIR researchers to directly link work on acquiring music features to the ways in which users interact with music. In this section we consider how the information-theoretic techniques described in this work can inform the field of MIR.

### 5.1 User-Centred Feature Selection

The feature selection shown in this paper is done directly from the user data. In contrast, feature selection is usually performed using classifier wrappers with ground truth class labels such as genre. The use of genre is based on the assumption that it would support the way users currently organise music and features are selected based on these labels. This has led to issues including classifiers being trained on factors that are confounded with these labels and that are not of relevance to genre or users [18]. Our approach selects features independently of the choice of classifier, in what is termed a ‘filter’ approach. The benefit of doing this is that a wide range of features can be quickly filtered at relatively little computational expense. While the classifier ‘wrapper’ approach may achieve greater performance, it is more computationally expensive and more likely to suffer from overfitting.

The key benefit of filtering features based on user behaviour is that it provides a perspective on music features that is free from assumptions about users and music ground truth. This user-centred perspective provides a sanity-check for music features and classification – if a feature does not reflect the ways in which users organise their music, then how useful is it for music retrieval?

### 5.2 When To Learn

The information-theoretic measures presented offer an implicit relevance feedback for music retrieval. While we have considered the entropy of features as reflecting user behaviour, this behaviour is conditioned upon the existing music retrieval interfaces being used. For example, after issuing a query and receiving results, the user selects relevant songs from those results. If the entropy of a feature for those selected songs is small relative to the result set, then this feature is implicitly relevant to the retrieval.

The identification of shuffle and explorative behaviour provides some context for this implicit relevance feedback. Music which is listened to in a seemingly random fashion may represent an absent or disengaged user, adding noise to attempts to weight recommender systems or build a user profile. At the very least, where entropy is high across all features, then those features do not reflect the user’s mental model for their music retrieval. The detection of shuffle or high-entropy listening states thus provides a useful data hygiene measure when interpreting listening data.

### 5.3 Engagement

The entropy measures capture how much each feature is being ‘controlled’ by the user when selecting their music. We have shown that it spans a scale from a user choosing to listen to something specific to the user yielding control to radio or shuffle. Considering entropy over many features in this way gives a high-dimensional vector representing the user’s engagement with music. Different styles of music retrieval occupy different points in this space, commonly the two extremes of listening to a specific album or just shuffling. There is an opportunity for music retrieval that has the flexibility to support users engaging and applying control over music features only insofar as they desire to. An example of this would be a shuffle mode that allowed users to bias it to varying degrees, or to some extent, the feedback mechanism in recommender systems.

### 5.4 Open Source

The SPUD dataset is made available for download at: <http://www.dcs.gla.ac.uk/~daniel/spud/> Example R scripts for importing data from *SPUD* and producing the analyses and plots in this paper are included. The code used to scrape this dataset is available under the MIT open source license, and can be accessed at: <http://www.github.com/dcboland/>

The MoodAgent features are commercially sensitive, thus not included in the *SPUD* dataset. At present, industry is far better placed to provide such large scale analyses of music data than academia. Even with user data and the required computational power, large-scale music analyses require licensing arrangements with content providers, presenting a serious challenge to academic MIR research. Our adoption of commercially provided features has allowed us to demonstrate our information-theoretic approach, and we distribute the audio stream links, however it is unlikely that many MIR researchers will have the resources to replicate all of these large scale analyses. The CoSound<sup>4</sup> project is an example of industry collaborating with academic research and state bodies to navigate the complex issues of music licensing and large-scale analysis.

## 6. CONCLUSION

This work introduces an information-theoretic approach to the study of users’ music listening behaviour. The case is made for a more user-focused yet quantitative approach to evaluation in MIR. We described the use of entropy to produce time-series analyses of user behaviour, and showed how changes in music-listening style can be detected. An example is given where a user started using online radio, having higher entropy in their listening. We introduced the use of adjusted mutual information to establish which music features are linked to playlist organisation. These techniques provide a quantitative approach to user studies and ground feature selection in user behaviour, contributing tools to support the user-centred future of MIR.

4. <http://www.cosound.dk/> Last accessed: 30/04/14

## ACKNOWLEDGEMENTS

This work was supported in part by Bang & Olufsen and the Danish Council for Strategic Research of the Danish Agency for Science Technology and Innovation under the CoSound project, case number 11-115328. This publication only reflects the authors' views.

## 7. REFERENCES

- [1] T Bertin-Mahieux, D. P Ellis, B Whitman, and P Lamere. The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval*, Miami, Florida, 2011.
- [2] G Brown, A Pocock, M.-J Zhao, and M Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The Journal of Machine Learning Research*, 13:27–66, 2012.
- [3] W. S Cleveland and S. J Devlin. Locally weighted regression: an approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610, 1988.
- [4] A Craft and G Wiggins. How many beans make five? the consensus problem in music-genre classification and a new evaluation method for single-genre categorisation systems. In *Proceedings of the 8th International Conference on Music Information Retrieval*, Vienna, Austria, 2007.
- [5] S. J Cunningham, N Reeves, and M Britland. An ethnographic study of music information seeking: implications for the design of a music digital library. In *Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, Houston, Texas, 2003.
- [6] J. S Downie. Music Information Retrieval. *Annual Review of Information Science and Technology*, 37(1):295–340, January 2003.
- [7] J. S Downie. The Music Information Retrieval Evaluation eXchange (MIREX). *D-Lib Magazine*, 12(12):795–825, 2006.
- [8] J Futrelle and J. S Downie. Interdisciplinary Research Issues in Music Information Retrieval: ISMIR 2000 - 2002. *Journal of New Music Research*, 32(2):121–131, 2003.
- [9] J. D Gould and C Lewis. Designing for usability: key principles and what designers think. *Communications of the ACM*, 28(3):300–311, 1985.
- [10] S. G Hart. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting*, San Francisco, California, 2006.
- [11] R Killick, P Fearnhead, and I. A Eckley. Optimal Detection of Change-points With a Linear Computational Cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.
- [12] J. H Lee and S. J Cunningham. The Impact (or Non-impact) of User Studies in Music Information Retrieval. In *Proceedings of the 13th International Conference for Music Information Retrieval*, Porto, Portugal, 2012.
- [13] B McFee and G Lanckriet. Hypergraph models of playlist dialects. In *Proceedings of the 13th International Conference for Music Information Retrieval*, Porto, Portugal, 2012.
- [14] M Schedl and A Flexer. Putting the User in the Center of Music Information Retrieval. In *Proceedings of the 13th International Conference on Music Information Retrieval*, Porto, Portugal, 2012.
- [15] M Schedl, A Flexer, and J Urbano. The neglected user in music information retrieval research. *Journal of Intelligent Information Systems*, 41(3):523–539, 2013.
- [16] B. L Sturm. An Analysis of the GTZAN Music Genre Dataset. In *Proceedings of the 2nd International ACM Workshop on Music Information Retrieval with User-centered and Multimodal Strategies*, MIRUM '12, New York, USA, 2012.
- [17] B. L Sturm. Classification accuracy is not enough. *Journal of Intelligent Information Systems*, 41(3):371–406, 2013.
- [18] B. L Sturm. A simple method to determine if a music information retrieval system is a horse. *IEEE Transactions on Multimedia*, 2014.
- [19] N. X Vinh, J Epps, and J Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854, 2010.
- [20] D. M Weigl and C Guastavino. User studies in the music information retrieval literature. In *Proceedings of the 12th International Conference for Music Information Retrieval*, Miami, Florida, 2011.

# EVALUATION FRAMEWORK FOR AUTOMATIC SINGING TRANSCRIPTION

**Emilio Molina, Ana M. Barbancho, Lorenzo J. Tardón, Isabel Barbancho**

Universidad de Málaga, ATIC Research Group, Andalucía Tech,

ETSI Telecomunicación, Campus de Teatinos s/n, 29071 Málaga, SPAIN

emm@ic.uma.es, abp@ic.uma.es, lorenzo@ic.uma.es, ibp@ic.uma.es

## ABSTRACT

In this paper, we analyse the evaluation strategies used in previous works on automatic singing transcription, and we present a novel, comprehensive and freely available evaluation framework for automatic singing transcription. This framework consists of a cross-annotated dataset and a set of extended evaluation measures, which are integrated in a Matlab toolbox. The presented evaluation measures are based on standard MIREX note-tracking measures, but they provide extra information about the type of errors made by the singing transcriber. Finally, a practical case of use is presented, in which the evaluation framework has been used to perform a comparison in detail of several state-of-the-art singing transcribers.

## 1. INTRODUCTION

Singing transcription refers to the automatic conversion of a recorded singing signal into a symbolic representation (e.g. a MIDI file) by applying signal-processing methods [1]. One of its renowned applications is query-by-humming [5], but other types of applications also are related to this task, like singing tutors [2], computer games (e.g. Singstar<sup>1</sup>), etc. In general, singing transcription is considered a specific case of melody transcription (also called note tracking), which is more general problem. However, singing transcription not only relates to melody transcription but also to speech recognition, and still nowadays it is a challenging problem even in the case of monophonic signals without accompaniment [3].

In the literature, various approaches for singing transcription can be found. A simple but commonly referenced approach was proposed by McNab in 1996 [4], and it relied on several handcrafted pitch-based and energy-based segmentation methods. Later, in 2001 Haus et al. used a similar approach with some rules to deal with intonation issues [5], and in 2002, Clarisse et al. [6] contributed with an auditory model, leading to later improved systems

<sup>1</sup> <http://www.singstar.com>



© Emilio Molina, Ana M. Barbancho, Lorenzo J. Tardón, Isabel Barbancho.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Emilio Molina, Ana M. Barbancho, Lorenzo J. Tardón, Isabel Barbancho. "Evaluation framework for automatic singing transcription", 15th International Society for Music Information Retrieval Conference, 2014.

such as [7] (later included in MAMI project<sup>2</sup> and today in SampleSumo products<sup>3</sup>). Additionally, other more recent approaches use hidden Markov models (HMM) to detect note-events in singing voice [8, 9, 11]. One of the most representative HMM-based singing transcribers was published by Ryyänen in 2004 [9]. More recently, in 2013, another probabilistic approach for singing transcription has been proposed in [3], also leading to relevant results. Regarding the evaluation methodologies used in these works (see Sections 2.1 and 3.1 for a review), there is not a standard methodology.

In this paper, we present a comprehensive evaluation framework for singing transcription. This framework consists of a cross-annotated dataset (Section 2) and a novel, compact set of evaluation measures (Section 3), which report information about the type of errors made by the singing transcriber. These measures have been integrated in a freely available Matlab toolbox (see Section 3.3). Then, we present a practical case in which the evaluation framework has been used to perform a comparison in detail of several state-of-the-art singing transcribers (Section 4). Finally, some relevant conclusions are presented in Section 5

## 2. DATASETS

In this section, we review the evaluation datasets used in prior works on singing transcription, and we describe the proposed evaluation dataset and our strategy for ground-truth annotation.

### 2.1 Datasets used in prior works

In Table 1, we present the datasets used in some relevant works on singing transcription. Note that none of the datasets fully represents the possible contexts in which singing transcription might be applied, since they are either too small (e.g. [5,6]), either very specific in style (e.g. [11] for opera and [3] for flamenco), or either they use an annotation strategy that may be subjective (e.g. [5,6]), or only valid for very good performances in rhythm and intonation (e.g. [8,9]). In addition, only the flamenco dataset used in [3] is freely available.

### 2.2 Proposed dataset

In this section we describe the music collection, as well as the annotation strategy used to build the ground-truth.

<sup>2</sup> <http://www.ipem.ugent.be/MAMI>

<sup>3</sup> <http://www.samplesumo.com>

Author	Year	Dataset size	Audio quality	Music style	Singing style	Ground-truth (GT) annotation strategy	Tuning devs. annotated in GT	Freely available
McNab [4]	1996	NONE						
Haus & Pollastri [5]	2001	20 short melodies	Low & moderate noise	Popular and scales	Syllables: 'na-na'...	Annotated by one musician	No	No
Clarisse et al. [6]	2002	22 short melodies	Low & moderate noise	Popular	Singing with & without lyrics	Annotation by one musician	No	No
Viitaniemi et al. [8]	2003	66 melodies (120 minutes)	High quality (studio conditions)	Folk songs & scales	Singing, humming & whistling	Original score used as ground-truth	No	No
Ryynänen et al. [9]	2004							
Mulder et al. [7]	2004	52 melo. (1354 notes)	Good & moderate noise	Popular songs	Syllables, singing & whistling	Team of musicologists	No	No
Kumar et al. [10]	2007	47 songs (2513 notes)	Good	Indian music	Syllables: /la/ /da/ /na/	Manual annot. of vowel onsets [REf]	No	No
Krige et al. [11]	2008	13842 notes	High quality but strong reverberation	Opera lessons & scales	Syllables	Time alignment using Viterbi	No	No
Gómez & Bonada [3]	2013	72 excerpts (2803 notes)	Good & slightly noisy	Flamenco songs	Lyrics & ornaments	Musicians team (cross-annotation)	Yes	Yes

**Table 1.** Review of the evaluation datasets used in prior works on singing transcription. Some details about the dataset are not provided in some cases, so certain fields can not be expressed in the same units (e.g. dataset size).

### 2.2.1 Music collection

The proposed dataset consists of 38 melodies sung by adult and child untrained singers, recorded in mono with a sample rate of 44100Hz and a resolution of 16 bits. Generally, the recordings are not clean and some background noise is present. The duration of the excerpts ranges from 15 to 86 seconds and the total duration of the whole dataset is 1154 seconds. This music collection can be broken down into three categories, according to the type of singer:

- Children (our own recordings<sup>4</sup>): 14 melodies of traditional children songs (557 seconds) sung by 8 different children (5-11 years old).
- Adult male: 13 pop melodies (315 seconds) sung by 8 different adult male untrained singers. These recordings were randomly chosen from the public dataset MTG-QBH<sup>5</sup> [12].
- Adult female: 11 pop melodies (281 seconds) sung by 5 different adult female untrained singers, also taken from MTG-QBH dataset.

Note that in this collection the pitch and the loudness can be unstable, and well performed vibratos are not frequent.

### 2.2.2 Ground-truth: annotation strategy

The described music collection has been manually annotated to build the ground truth<sup>4</sup>. First, we have transcribed the audio recordings with a baseline algorithm (Section 4.2), and then all the transcription errors have been corrected by an expert musician with more than 10 years of music training. Then, a second expert musician (with 7 years of music training) checked all the annotations until both musicians agreed in their correctness. The transcription errors were corrected by listening, at the same time, to the synthesized transcription and the original audio. The

<sup>4</sup> Available at <http://www.atc.uma.es/ismir2014singing>

<sup>5</sup> <http://mtg.upf.edu/download/datasets/mtg-qbh>

musicians were given a set of instructions about the specific criteria to annotate the singing melody:

- Ornaments such as pitch bending at the beginning of the notes or vibratos are not considered independent notes. This criterion is based on Vocaloid's<sup>6</sup> approach, where ornaments are not modelled with extra notes.
- Portamento between two notes does not produce an extra third note (again, this is the criteria used in Vocaloid).
- The onsets are placed at the beginning of voiced segments and in each clear change of pitch or phoneme. In the case of 'l', 'm', 'n' voiced consonants + vowel (e.g. 'la'), the onset is not placed at the beginning of the consonant but at the beginning of the vowel.
- The pitch of each note is annotated with cents resolution as perceived by the team of experts. Note that we annotate the tuning deviation for each independent note.

## 3. EVALUATION MEASURES

In this section, we describe the evaluation measures used in prior works on automatic singing transcription, and we present the proposed ones.

### 3.1 Evaluation measures used in prior works

In Table 2, we review the evaluation measures used in some relevant works on singing transcription. In some cases, only the note and/or frame error is provided as a compact, representative measure [5, 9], whereas other approaches provide extra information about the type of errors made by the system using dynamic time warping (DTW) [6] or Viterbi-based alignment [11]. In our case, we have taken the most relevant aspects of these approaches and we added some novel ideas in order to define a novel, compact and comprehensive set of evaluations.

<sup>6</sup> <http://www.vocaloid.com>

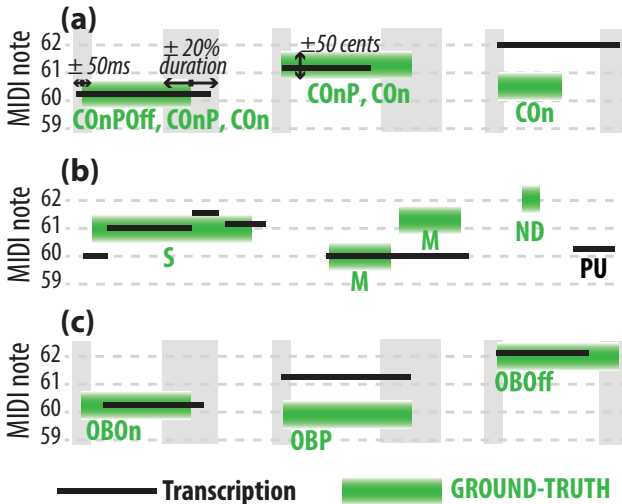


Author	Year	Evaluation measures
McNab	1996	NONE
Haus & Pollastri [5]	2001	Rate of note pitch errors (segmentation errors are not considered)
Clarisse et al. [6]	2002	DTW-based measurement of various note errors, e.g. insertions deletions and substitutions.
Viitaniemi et al. [8]	2003	Frame-based errors. Do not report information about type of errors made.
Ryynänen et al. [9]	2004	Note-based and frame-based errors. Do not report information about type of errors made.
Mulder et al. [7]	2004	DTW-based measurement of various note errors, e.g. insertions deletions and substitutions.
Kumar et al. [10]	2007	Onset detection errors (pitch and durations are ignored).
Krige et al. [11]	2008	Viterbi-based measurement of deletions, insertions and substitutions (typical evaluation in speech recognition).
Gómez & Bonada [3]	2013	MIREX measures for audio melody extraction and note-tracking. Do not report information about type of errors made.

**Table 2.** Evaluation measures used in prior works on singing transcription.

### 3.2 Proposed measures

In this section, we firstly present the notation and some needed definitions that are used in the rest of sections, and then we describe the evaluation measures used to quantify the proportion of correctly transcribed notes. Finally, we present a set of novel evaluation measures that independently report the importance of each type of error. In Figure 1 we show an example of the types of errors considered.



**Figure 1.** Examples of the different proposed measures.

#### 3.2.1 Notation

The  $i$ :th note of the ground-truth is noted as  $n_i^{GT}$ , and the  $j$ :th note of the transcription is noted as  $n_j^{TR}$ . The total number of notes in the ground-truth and the transcription

are  $N^{GT}$  and  $N^{TR}$ , respectively. Regarding the expressions used in the for correct notes, we have used Precision, Recall and F-measure, which are defined as follow:

$$CX_{\text{Precision}} = \frac{N_{CX}^{GT}}{N^{GT}} \quad (1)$$

$$CX_{\text{Recall}} = \frac{N_{CX}^{TR}}{N^{TR}} \quad (2)$$

$$CX_{\text{F-measure}} = 2 \cdot \frac{CX_{\text{Precision}} \cdot CX_{\text{Recall}}}{CX_{\text{Precision}} + CX_{\text{Recall}}} \quad (3)$$

where  $CX$  makes reference to the specific category of correct note: Correct Onset & Pitch & Offset ( $X = \text{COnPOff}$ ), Correct Onset & Pitch ( $X = \text{COnP}$ ) or Correct Onset ( $X = \text{COn}$ ). Finally,  $N_{CX}^{GT}$  and  $N_{CX}^{TR}$  are the total number of matching  $CX$  conditions in the ground-truth and the transcription, respectively.

Regarding the measures used for errors, we have computed the Error Rate with respect to  $N^{GT}$ , or with respect to  $N^{TR}$ , as follow:

$$X_{\text{RateGT}} = \frac{N_X^{GT}}{N^{GT}} \quad (4)$$

$$X_{\text{RateTR}} = \frac{N_X^{TR}}{N^{TR}} \quad (5)$$

Finally, in the case of segmentation errors (Section 3.2.5), we also compute the mean number of notes tagged as  $X$  in the transcription for each note tagged as  $X$  in the ground-truth. This magnitude has been expressed as a ratio:

$$X_{\text{Ratio}} = \frac{N_X^{TR}}{N_X^{GT}} \quad (6)$$

#### 3.2.2 Definition of correct onset/pitch/offset

The definitions of correctly transcribed notes (given in Section 3.2.3) consists of combinations of three independent conditions: correct onset, correct pitch and correct offset. We have defined these conditions according to MIREX (*Multiple F0 estimation and tracking and Audio Onset Detection tasks*), and so they are defined as follow:

- Correct Onset: If the note's onset of a transcribed note  $n_j^{TR}$  is within a  $\pm 50\text{ms}$  range of the onset of a ground-truth note  $n_i^{GT}$ , i.e.:

$$\text{onset}(n_j^{TR}) \in [\text{onset}(n_i^{GT}) - 50\text{ms}, \text{onset}(n_i^{GT}) + 50\text{ms}] \quad (7)$$

then we consider that  $n_i^{GT}$  has a correct onset with respect to  $n_j^{TR}$ .

- Correct Pitch: If the note's pitch of a transcribed note  $n_j^{TR}$  is within a  $\pm 0.5$  semitones range of the pitch of a ground-truth note  $n_i^{GT}$ , i.e.:

$$\text{pitch}(n_j^{TR}) \in [\text{pitch}(n_i^{GT}) - 0.5 \text{ st}, \text{pitch}(n_i^{GT}) + 0.5 \text{ st}] \quad (8)$$

then we consider that  $n_i^{GT}$  has a correct pitch with respect to  $n_j^{TR}$ .

- Correct Offset: If the offsets of the ground-truth note  $n_i^{GT}$  and the transcribed note  $n_j^{TR}$  are within a range of  $\pm 20\%$  of the duration of  $n_i^{GT}$  or  $\pm 50\text{ms}$ , whichever is larger, i.e.:

$$\text{offset}(n_j^{TR}) \in [\text{offset}(n_i^{GT}) - \text{OffRan}, \text{offset}(n_i^{GT}) + \text{OffRan}] \quad (9)$$

where  $\text{OffRan} = \max(50\text{ms}, \text{duration}(n_i^{GT}))$ , then we consider that  $n_i^{GT}$  has a correct offset with respect to  $n_j^{TR}$ .

### 3.2.3 Correctly transcribed notes

The definition of “correct note” should be useful to measure the suitability of a given singing transcriber for a specific application. However, different applications may require a different definition of correct note. Therefore, we have chosen three different definitions of correct note as defined in MIREX:

- **Correct onset, pitch and offset (COnPOff):** This is a standard correctness criteria, since it is used in MIREX (*Multiple F0 estimation and tracking* task), and it is the most restrictive one. The note  $n_i^{GT}$  is assumed to be correctly transcribed into the note  $n_j^{TR}$  if it has correct onset, correct pitch and correct offset (as defined in Section 3.2.2). In addition, one ground truth note  $n_i^{GT}$  can only be associated with one transcribed note  $n_j^{TR}$ . In our evaluation framework, we report Precision, Recall and F-measure as defined in Section 3.2.1:

$$\text{COnPOff}_{\text{Precision}}, \text{COnPOff}_{\text{Recall}} \text{ and } \text{COnPOff}_{\text{F-measure}}.$$

- **Correct Onset, Pitch (COnP):** This criteria is also used in MIREX, but it is less restrictive since it just considers onset and pitch, and ignores the offset value. Therefore, in COnP criteria, a note  $n_i^{GT}$  is assumed to be correctly transcribed into the note  $n_j^{TR}$  if it has correct onset and correct pitch. In addition, one ground truth note  $n_i^{GT}$  can only be associated with one transcribed note  $n_j^{TR}$ . In our evaluation framework, we report Precision, Recall and F-measure:

$$\text{COnP}_{\text{Precision}}, \text{COnP}_{\text{Recall}} \text{ and } \text{COnP}_{\text{F-measure}}.$$

- **Correct Onset (COn):** Additionally, we have included the evaluation criteria used in MIREX *Audio Onset Detection* task. In this case, a note  $n_i^{GT}$  is assumed to be correctly transcribed into the note  $n_j^{TR}$  if it has correct onset. In addition, one ground truth note  $n_i^{GT}$  can only be associated with one transcribed note  $n_j^{TR}$ . In our evaluation framework, we report Precision, Recall and F-measure:

$$\text{COnPOff}_{\text{Precision}}, \text{COnPOff}_{\text{Recall}} \text{ and } \text{COnPOff}_{\text{F-measure}}.$$

### 3.2.4 Incorrect notes with one single error

In addition, we have included some novel evaluation measures to identify the notes that are close to be correctly transcribed, but they fail in one single aspect. These measures are useful to identify specific weaknesses of a given singing transcriber. The proposed categories are:

- **Only-Bad-Onset (OBOn):** A ground-truth note  $n_i^{GT}$  is labelled as OBOn if it has been transcribed into a note  $n_j^{TR}$  with correct pitch and offset, but wrong onset. In order to detect them, firstly we find all ground-truth notes with correct pitch and offset, taking into account that one ground-truth note can only be associated with one transcribed note. Then, we remove all notes previously tagged as COnPOff (Section 3.2.3). The reported measure is the rate of OBOn notes in the ground-truth:

$$\text{OBOn}_{\text{RateGT}}$$

- **Only-Bad-Pitch (OBP):** A ground-truth note  $n_i^{GT}$  is labelled as OBP if it has been transcribed into a note  $n_j^{TR}$

with correct onset and offset, but wrong pitch. In order to detect them, firstly we find all ground-truth notes with correct onset and offset, taking into account that one ground-truth note can only be associated with one transcribed note. Then, we remove all notes previously tagged as COnPOff (Section 3.2.3). The reported measure is the rate of OBP notes in the ground-truth:

$$\text{OBP}_{\text{RateGT}}$$

- **Only-Bad-Offset (OBOff):** A ground-truth note  $n_i^{GT}$  is labelled as OBOn if it has been transcribed into a note  $n_j^{TR}$  with correct pitch and onset, but wrong offset. In order to detect them, firstly we find all ground-truth notes with correct pitch and onset, taking into account that one ground-truth note can only be associated with one transcribed note. Then, we remove all notes previously tagged as COnPOff (Section 3.2.3). The reported measure is the rate of OBOff notes in the ground-truth:

$$\text{OBOff}_{\text{RateGT}}$$

### 3.2.5 Incorrect notes with segmentation errors

Segmentation errors refer to the case in which sung notes are incorrectly split or merged during the transcription. Depending on the final application, certain types of segmentation errors may not be important (e.g. frame-based systems for query-by-humming are not affected by splits), but they can lead to problems in many other situations. Therefore, we have defined two evaluation measures which are informative about the segmentation errors made by the singing transcriber.

- **Split (S):** A split note is a ground truth note  $n_i^{GT}$  that is incorrectly segmented into different consecutive notes  $n_{j_1}^{TR}, n_{j_2}^{TR}, \dots, n_{j_n}^{TR}$ . Two requirements are needed in a split: (1) the set of transcribed notes  $n_{j_1}^{TR}, n_{j_2}^{TR}, \dots, n_{j_n}^{TR}$  must overlap at least the 40% of  $n_i^{GT}$  in time (pitch is ignored), and (2)  $n_i^{GT}$  must overlap at least the 40% of every note  $n_{j_1}^{TR}, n_{j_2}^{TR}, \dots, n_{j_n}^{TR}$  in time (again, pitch is ignored). These requirements are needed to ensure a consistent relationship between ground truth and transcribed notes. The specific reported measures are:

$$\text{S}_{\text{RateGT}} \text{ and } \text{S}_{\text{Ratio}}$$

Note that in this case  $\text{S}_{\text{Ratio}} > 1$ .

- **Merged (M):** A set of consecutive ground-truth notes  $n_{i_1}^{GT}, n_{i_2}^{GT}, \dots, n_{i_n}^{GT}$  are considered to be merged if they all are transcribed into the same note  $n_j^{TR}$ . This is the complementary case of split. Again, two requirements must be true to consider a group of merged notes: (1) the set of ground truth notes  $n_{i_1}^{GT}, n_{i_2}^{GT}, \dots, n_{i_n}^{GT}$  must overlap the 40% of  $n_j^{TR}$  in time (pitch is ignored), and (2)  $n_j^{TR}$  must overlap the 40% of every note  $n_{i_1}^{GT}, n_{i_2}^{GT}, \dots, n_{i_n}^{GT}$  in time (again, pitch is ignored). The specific reported measures are:

$$\text{M}_{\text{RateGT}} \text{ and } \text{M}_{\text{Ratio}}$$

Note that in this case  $\text{M}_{\text{Ratio}} < 1$ .

### 3.2.6 Incorrect notes with voicing errors

Voicing errors happen when an unvoiced sound produces a false transcribed note (spurious note), or when a sung note is not transcribed at all (non-detected note). This situation is commonly associated to a bad performance of the voicing stage within the singing transcriber. We have defined two categories:

- Spurious notes (PU): A spurious note is a transcribed note  $n_j^{TR}$  that does not overlap at all (neither in time nor in pitch) any note in the ground truth. The associated reported measure is:

$$PU_{\text{RateTR}}$$

- Non-detected notes (ND): A ground-truth note  $n_i^{GT}$  is non-detected if it does not overlap at all (neither in time nor in pitch) any transcribed note. The associated reported measure is:

$$ND_{\text{RateGT}}$$

### 3.3 Proposed Matlab toolbox

The presented evaluation measures have been implemented in a freely available Matlab toolbox<sup>4</sup>, which consists of a set of functions and structures, as well as a graphical user interface to visually analyse the performance of the evaluated singing transcriber.

The main function of our toolbox is `evaluation.m`, which receives the ground-truth and the transcription of an audio clip as inputs, and it outputs the results of all the evaluation measures. In addition, we have included a function called `listnotes.m`, which receives as inputs the ground-truth, the transcription and the category  $\mathbf{X}$  to be listed, and it outputs a list (in a two-columns format: onset time-offset time) of all the notes in the ground-truth tagged as  $\mathbf{X}$  category. This information is useful to isolate the problematic audio excerpts for further analysis.

Finally, we have implemented a graphical user interface, where the ground-truth and the transcription of a given audio clip can be compared using a piano-roll representation. This interface also allows the user to highlight notes tagged as  $\mathbf{X}$  (e.g. COnPOff, S, etc.).

## 4. PRACTICAL USE OF THE PROPOSED TOOLBOX

In this section, we describe a practical case of use in which the presented evaluation framework has been used to perform an improved comparative study of several state-of-the-art singing transcribers (presented in Section 4.1). In addition, a simple, easily reproducible baseline approach has been included in this comparative study. Finally, we show and discuss the obtained results.

### 4.1 Compared algorithms

We have compared three state-of-the-art algorithms for singing transcription:

- **Method (a):** Gómez & Bonada (2013) [3]. It consists of three main steps: tuning-frequency estimation, transcription into short notes, and an iterative process involving note consolidation and refinement of the tuning frequency. For

the experiment, we have used a binary provided by the authors of the algorithm.

- **Method (b):** Ryyänen (2008) [13]. We have used the method for automatic transcription of melody, bass line and chords in polyphonic music published by Ryyänen in 2008 [13], although we only focus on melody transcription. It is the last evolution of the original HMM-based monophonic singing transcriber [9]. For the experiment, we have used a binary provided by the authors of the algorithm.

- **Method (c):** Melotranscript<sup>4</sup> (based on Mulder 2004 [7]). It is the commercial version derived from the research carried out by Mulder et al. [7]. It is based on an auditory model. For the experiment, we have used the demo version available in SampleSumo website<sup>3</sup>.

### 4.2 Baseline algorithm

According to [8], the simplest possible segmentation consists of simply rounding a rough pitch estimate to the closest MIDI note  $n_i$  and taking all pitch changes as note boundaries. The proposed baseline method is based on such idea, and it uses Yin [14] to extract the F0 and aperiodicity at frame-level. A frame is classified as unvoiced if its aperiodicity is under  $< 0.4$ . Finally, all notes shorter than 100ms are discarded.

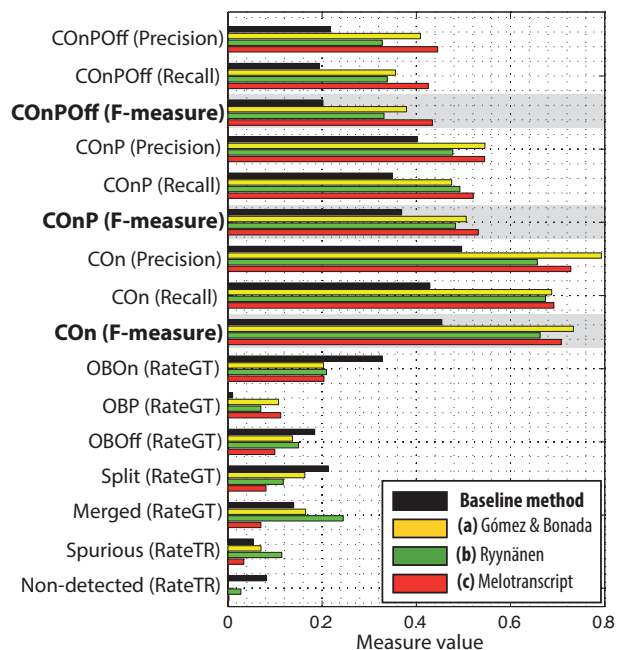
### 4.3 Results & discussion

In Figure 2 we show the results of our comparative analysis. Regarding the F-measure of correct notes (COnPOff, COnP and COn), methods (a) and (c) attains similar values, whereas method (b) performs slightly worse. In addition, it seems that method (a) is slightly superior to method (c) for onset detection, but method (c) is superior when pitch and offset values must be also estimated. In all cases, the baseline is clearly worse than the rest of methods.

In addition, we observed that the rate of notes with incorrect onset (OOn) is equally high (20%) in all methods. After analysing the specific recordings, we concluded that onset detection within a range of  $\pm 50$ ms is very restrictive in the case of singing voice with lyrics, since many onsets are not clear even for an expert musician (as proved during the ground-truth building). Moreover, we also observed that all methods, and especially method (a), have problems with pitch bendings at the beginning of the notes, since they tend to split them.

Regarding the segmentation and voicing errors, we realised that method (a) tends to split notes, whereas method (b) tends to merge notes. This information, easily provided by our evaluation framework, may be useful to improve specific weaknesses of the algorithms during the development stage. Finally, we also realised that method (b) is worse than method (a) and (c) in terms of voicing.

To sum up, method (c) seems to be the best one in most measures, mainly due to a better performance in segmentation and voicing. However, method (a) is very appropriate for onset detection. Finally, although method (b) works clearly better than the baseline, has a poor performance due to errors in segmentation (mainly merged notes) and voicing (mainly spurious).



**Figure 2.** Comparison in detail of several state-of-the-art singing transcription systems using the presented evaluation framework.

## 5. CONCLUSIONS

In this paper, we have presented an evaluation framework for singing transcription. It consists of a cross-annotated dataset of 1154 seconds and a novel set of evaluation measures, able to report the type of errors made by the system. Both the dataset, and a Matlab toolbox including the presented evaluation measures, are freely available<sup>4</sup>. In order to show the utility of the work presented in this paper, we have performed a detailed comparative study of three state-of-the-art singing transcribers plus a baseline method, leading to relevant information about the performance of each method. In the future, we plan to expand our evaluation dataset in order to make it comparable to other datasets<sup>7</sup> used in MIREX (e.g. MIR-1K or MIR-QBSH).

## 6. ACKNOWLEDGEMENTS

This work has been funded by the Ministerio de Economía y Competitividad of the Spanish Government under Project No. TIN2013-47276-C6-2-R and by the Junta de Andalucía under Project No. P11-TIC-7154. The work has been done at Universidad de Málaga. Campus de Excelencia Internacional Andalucía Tech.

## 7. REFERENCES

- [1] M. Ryyänen, "Singing transcription," in *Signal Processing Methods for Music Transcription* (A. Klapuri and M. Davy, eds.), pp. 361–390, Springer Science + Business Media LLC, 2006.
- [2] E. Molina, I. Barbancho, E. Gómez, A. M. Barbancho, and L. J. Tardón, "Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP*, pp. 744–748, 2013.
- [3] E. Gómez and J. Bonada, "Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a capella singing," *Computer Music Journal*, vol. 37, no. 2, pp. 73–90, 2013.
- [4] R. J. McNab, L. A. Smith, and I. H. Witten, "Signal Processing for Melody Transcription," *Proceedings of the 19th Australasian Computer Science Conference*, vol. 18, no. 4, pp. 301–307, 1996.
- [5] G. Haus and E. Pollastri, "An audio front end for query-by-humming systems," in *Proceedings of the 2nd International Symposium on Music Information Retrieval ISMIR*, pp. 65–72, sn, 2001.
- [6] L. P. Clarisse, J. P. Martens, M. Lesaffre, B. D. Baets, H. D. Meyer, and M. Leman, "An Auditory Model Based Transcriber of Singing Sequences," in *Proceedings of the 3rd International Conference on Music Information Retrieval ISMIR*, pp. 116–123, 2002.
- [7] T. De Mulder, J.P. Martens, M. Lesaffre, M. Leman, B. De Baets, H. De Meyer, "Recent improvements of an auditory model based front-end for the transcription of vocal queries", , *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP 2004)*, Montreal, Quebec, Canada, May 17–21, Vol. IV, pp. 257–260, 2004.
- [8] T. Viitaniemi, A. Klapuri, and A. Eronen, "A probabilistic model for the transcription of single-voice melodies," in *Proceedings of the 2003 Finnish Signal Processing Symposium FINSIG03*, pp. 59–63, 2003.
- [9] M. Ryyänen and A. Klapuri, "Modelling of note events for singing transcription," in *Proceedings of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing SAPA*, (Jeju, Korea), Oct. 2004.
- [10] P. Kumar, M. Joshi, S. Hariharan, and P. Rao, "Sung Note Segmentation for a Query-by-Humming System". In *Intl Joint Conferences on Artificial Intelligence (IJCAI)*, 2007.
- [11] W. Krige, T. Herbst, and T. Niesler, "Explicit transition modelling for automatic singing transcription," *Journal of New Music Research*, vol. 37, no. 4, pp. 311–324, 2008.
- [12] J. Salamon, J. Serrá and E. Gómez, "Tonal Representations for Music Retrieval: From Version Identification to Query-by-Humming", *International Journal of Multimedia Information Retrieval, special issue on Hybrid Music Information Retrieval*, vol. 2, no. 1, pp. 45–58, 2013.
- [13] M. P. Ryyänen and A. P. Klapuri, "Automatic Transcription of Melody, Bass Line, and Chords in Polyphonic Music," in *Computer Music Journal*, vol.32, no. 3, 2008.
- [14] A. De Cheveigné and H. Kawahara: "YIN, a fundamental frequency estimator for speech and music," *Journal of the Acoustic Society of America*, Vol. 111, No. 4, pp. 1917–1930, 2002.

<sup>7</sup> <http://mirlab.org/dataSet/public/>

# WHAT IS THE EFFECT OF AUDIO QUALITY ON THE ROBUSTNESS OF MFCCs AND CHROMA FEATURES?

**Julián Urbano, Dmitry Bogdanov, Perfecto Herrera, Emilia Gómez and Xavier Serra**  
Music Technology Group, Universitat Pompeu Fabra Barcelona, Spain

{julian.urbano,dmitry.bogdanov,perfecto.herrera,emilia.gomez,xavier.serra}@upf.edu

## ABSTRACT

Music Information Retrieval is largely based on descriptors computed from audio signals, and in many practical applications they are to be computed on music corpora containing audio files encoded in a variety of lossy formats. Such encodings distort the original signal and therefore may affect the computation of descriptors. This raises the question of the robustness of these descriptors across various audio encodings. We examine this assumption for the case of MFCCs and chroma features. In particular, we analyze their robustness to sampling rate, codec, bitrate, frame size and music genre. Using two different audio analysis tools over a diverse collection of music tracks, we compute several statistics to quantify the robustness of the resulting descriptors, and then estimate the practical effects for a sample task like genre classification.

## 1. INTRODUCTION

A significant amount of research in Music Information Retrieval (MIR) is based on descriptors computed from audio signals. In many cases, research corpora contain music files encoded in a lossless format. In some situations, datasets are distributed without their original music corpus, so researchers have to gather audio files themselves. In many other cases, audio descriptors are distributed instead of the audio files. In the end, MIR research is thus based on corpora that very well may use different audio encodings, all under the assumption that audio descriptors are robust to these variations and the final MIR algorithms are not affected. This possible lack of robustness poses serious questions regarding the reproducibility of MIR research and its applicability. For instance, whether algorithms trained with lossless audio files can generalize to lossy encodings; or whether a minimum audio bitrate should be required in datasets that distribute descriptors instead of audio files.

In this paper we examine the assumption of robustness of music descriptors across different audio encodings on the example of Mel-frequency cepstral coefficients (MFCCs) and chroma features. They are among the most popular music descriptors used in MIR research, as they respectively capture timbre and tonal information.

Many MIR tasks such as classification, similarity, autotagging, recommendation, cover identification and audio fingerprinting, audio-to-score alignment, audio segmentation, key and chord estimation, and instrument detection are at least partially based on them. As they pervade the literature on MIR, we analyzed the effect of audio encoding and signal analysis parameters on the robustness of MFCCs and chroma. To this end, we run two different audio analysis tools over a diverse collection of 400 music tracks. We then compute several indicators that quantify the robustness and stability of the resulting features and estimate the practical implications for a general task like genre classification.

## 2. DESCRIPTORS

### 2.1 Mel-Frequency Cepstrum Coefficients

MFCCs are inherited from the speech domain [18], and they have been extensively used to summarize the spectral content of music signals within an analysis frame. MFCCs are widely used in tasks like music similarity [1, 12], music classification [6] (in particular, genre), autotagging [13], preference learning for music recommendation [19, 24], cover identification and audio segmentation [17].

There is no standard algorithm to compute MFCCs, and a number of variants have been proposed [8] and adapted for MIR applications. MFCCs are commonly computed as follows. The first step consists in windowing the input signal and computing its magnitude spectrum with the Fourier transform. We then apply a filterbank with critical (mel) band spacing of the filters and bandwidths. Energy values are obtained for the output of each filter, followed by a logarithm transformation. We finally compute a discrete cosine transform to the set of log-energy values to obtain the final set of coefficients. The number of mel bands and the frequency interval on which they are computed may vary among implementations. The low order coefficients account for the slowly changing spectral envelope, while the higher order coefficients describe the fast variations of the spectrum shape, including pitch information. The first coefficient is typically discarded in MIR applications because it does not provide information about the spectral shape; it reflects the overall energy in mel bands.

### 2.2 Chroma

Chroma features represent the spectral energy distribution within an analysis frame, summarized into 12 semitones across octaves in equal-tempered scale. Chroma captures the pitch class distribution of an input signal, typically used



for key and chord estimation [7, 9], music similarity and cover identification [20], classification [6], segmentation and summarization [5, 17], and synchronization [16].

Several approaches exist for chroma feature extraction, including the following steps. The signal is first analyzed with a high frequency resolution in order to obtain its frequency domain representation. The main frequency components (e.g. spectral peaks) are mapped onto pitch classes according to an estimated tuning frequency. For most approaches, a frequency value partially contributes to a set of “sub-harmonic” fundamental frequency (pitch) candidates. The chroma vector is computed with a given interval resolution (number of bins per octave) and is finally post-processed to obtain the final chroma representation. Timbre invariance is achieved by different transformations such as spectral whitening [9] or cepstrum liftering [15].

### 3. EXPERIMENTAL DESIGN

#### 3.1 Factors Affecting Robustness

We identified several factors that could have an effect on the robustness of audio descriptors, from the perspective of their audio encoding (codec, bitrate and sampling rate), analysis parameters (frame/hop size and audio analysis tool) and the musical characteristics of the songs (genre).

**SRate.** The sampling rate at which an audio signal is encoded may affect robustness when using very high frequency rates. We study standard 44100 and 22050 Hz.

**Codec.** Perceptual audio coders may also affect descriptors because they introduce perturbations to the original audio signal, in particular by reducing high-frequency content, blurring the attacks, and smoothing the spectral envelope. In our experiments, we chose one lossless and two lossy audio codecs: WAV, MP3 CBR and MP3 VBR.

**BRate.** Different audio codecs allow different bitrates depending on the sampling rate, so we can not combine all codecs with all bitrates. The following combinations are permitted and used in our study:

- WAV: 1411 Kbps.
- MP3 CBR at 22050 Hz: 64, 96, 128 and 160 Kbps.
- MP3 CBR at 44100 Hz: 64, 96, 128, 160, 192, 256 and 320 Kbps.
- MP3 VBR: 6 (100-130 Kbps), 4 (140-185 Kbps), 2 (170-210 Kbps) and 0 (220-260 Kbps).

**FSize.** We considered a variety of frame sizes for spectral analysis: 23.2, 46.4, 92.9, 185.8, 371.5 and 743.0 ms. That is, we used frame sizes of 1024, 2048, 4096, 8192, 16384 and 32768 samples for signals with sampling rate of 44100 Hz, and the halved values (512, 1024, 2048, 4096, 8192 and 16384 samples) in the case of 22050 Hz.

**Audio analysis tool.** The specific software used to compute descriptors may have an effect on their robustness due to parameterizations (e.g. frequency ranges) and other implementation details. We use two state-of-the-art and open source tools publicly available online: *Essentia 2.0.1*<sup>1</sup> [2] and *QM Vamp Plugins 1.7 for Sonic Annotator 0.7*<sup>2</sup> [3].

<sup>1</sup><http://essentia.upf.edu>

<sup>2</sup><http://vamp-plugins.org/plugin-doc/qm-vamp-plugins.html>

Since our goal here is not to compare tools, we refer to them simply as Lib1 and Lib2 throughout the paper.

Lib1 and Lib2 provide by default two different implementations of MFCCs, both of which compute cepstral coefficients on 40 mel bands, resembling the MFCC FB-40 implementation [8, 22] but on different frequency intervals. Lib1 covers a wider frequency range of 0-11000 Hz with mel bin centers being equally spaced on the mel scale in this range, while Lib2 covers a frequency range of 66-6364 Hz. We compute the first 13 MFCCs in both systems and discard the first coefficient. In the case of chroma, Lib1 analyzes a frequency range of 40-5000 Hz based on Fourier transform and estimates tuning frequency. Lib2 uses a Constant Q Transform and analyzes the frequency range 65-2093 Hz assuming tuning frequency of 440 Hz, but it does not account for harmonics of the detected peaks. We compute 12-dimensional chroma features.

**Genre.** Robustness may depend as well on the music genre of songs. For instance, as the most dramatic change that perceptual coders introduce is that of filtering out high-frequency spectral content, genres that make use of very high-frequency sounds (e.g. cymbals and electronic tones) should show a more detrimental effect than genres not including them (e.g. country, blues and classical).

#### 3.2 Data

We created an ad-hoc corpus of music for this study, containing 400 different music tracks (30 seconds excerpts) by 395 different artists, uniformly covering 10 music genres (blues, classical, country, disco/funk/soul, electronic, jazz, rap/hip-hop, reggae, rock and rock’n’roll). All 400 tracks are encoded from their original CD at a 44100 Hz sampling rate using the lossless FLAC audio codec.

We converted all lossless tracks in our corpus into various audio formats in accordance with the factors identified above, taking into account all possible combinations of sampling rate, codec and bitrate. Audio conversion was done using the *FFmpeg 0.8.3*<sup>3</sup> converter, which includes the LAME codec for MP3 joint stereo mode (*Lavf53.21.1*). Afterwards, we analyzed the original lossless files and their lossy versions using both Lib1 and Lib2. In the case of Lib1, both MFCCs and chroma features were computed for all different frame sizes with the hop size equal to half the frame size. MFCCs were computed similarly in the case of Lib2, but chroma features only allow a fixed frame size of 16384 samples (we selected a hop size of 2048 samples). In all cases, we summarize the frame-wise feature vectors with the mean of each coefficient.

#### 3.3 Indicators of Robustness

We computed several indicators of the robustness of MFCCs and chroma, each measuring the difference between the descriptors computed with the original lossless audio clips and the descriptors computed with their lossy versions. We blocked by tool, sampling rate and frame size under the assumption that these factors are not mixed in practice within the same application. For two arbitrary

<sup>3</sup><http://www.ffmpeg.org>

vectors  $x$  and  $y$  (each containing  $n = 12$  MFCC or chroma values) from a lossless and a lossy version, we compute five indicators to measure how different they are.

**Relative error  $\delta$ .** It is computed as the average relative difference across coefficients. This indicator can be easily interpreted as the percentage error between coefficients, and it is of especial interest for tasks in which coefficients are used as features to train some model.

$$\delta(x, y) = \frac{1}{n} \sum \frac{|x_i - y_i|}{\max(|x_i|, |y_i|)}$$

**Euclidean distance  $\varepsilon$ .** The Euclidean distance between the two vectors, which is especially relevant for tasks that compute distances between pairs of songs, such as in music similarity or other tasks that use techniques like clustering.

**Pearson's  $r$ .** The common parametric correlation coefficient between the two vectors, ranging from -1 to 1.

**Spearman's  $\rho$ .** A non-parametric correlation coefficient, equal to the Pearson's  $r$  correlation after transforming all coefficients to their corresponding ranks in  $x \cup y$ .

**Cosine similarity  $\theta$ .** The angle between both vectors. It is similar to  $\varepsilon$ , but it is normalized between 0 and 1.

We have 400 tracks  $\times$  19 *BRate:Codec*  $\times$  6 *FSize* = 45600 datapoints for MFCCs with Lib1, MFCCs with Lib2, and chroma with Lib1. For chroma with Lib2 there is just one *FSize*, which yields 7600 datapoints. This adds up to 144400 datapoints for each indicators, 722000 overall.

### 3.4 Analysis

For simplicity, we followed a hierarchical analysis for each combination of sampling rate, tool, feature and robustness indicator. We are first interested in the mean of the score distributions, which tells us the expected *robustness* in each case (e.g. a low  $\varepsilon$  mean score suggests that the descriptor is robust because it does not differ much between the lossless and the lossy versions). But we are also interested in the *stability* of the descriptor, that is, the variance of the distribution. For instance, a descriptor might be robust on average but not below 192 Kbps, or robust only with a frame size of 2048.

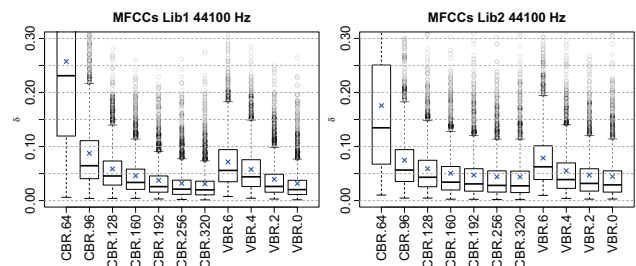
To gain a deeper understanding of the variations in the indicators, we fitted a random effects model to study the effects of codec, bitrate and frame size [14]. The specific models included the *FSize* and *Codec* main effects, and the bitrate was modeled as nested within the *Codec* effect (*BRate:Codec*); all interactions among them were also fitted. Finally, we included the *Genre* and *Track* main effects to estimate the specific variability due to inherent differences among the music pieces themselves. We did not consider any *Genre* or *Track* interactions because they can not be controlled in a real-world application, so their effects are all confounded with the *residual* effect. Note though that this residual does not account for any random error (in fact, there is no random error in this model); it accounts for high-order interactions associated with *Genre* and *Track* that are irrelevant for our purposes. This results in a Resolution V design for the factors of interest (main effects unconfounded with two- or three-factor interactions) and a Resolution III design for musical factors

related to genre (main effects confounded with two-factor interactions) [14]. We ran an ANOVA analysis on these models to estimate variance components, which indicate the contribution of each factor to the total variance, that is, their impact on the robustness of the audio descriptors.

## 4. RESULTS

Table 1 shows the results for MFCCs. As shown by the mean scores, the descriptors computed by Lib1 and Lib2 are similarly robust (note that  $\varepsilon$  scores are not directly comparable across tools because they are not normalized; actual MFCCs in Lib1 are orders of magnitude larger than in Lib2). Both correlation coefficients  $r$  and  $\rho$ , as well as cosine similarity  $\theta$ , are extremely high, indicating that the shape of the feature vectors is largely preserved. However, the average error across coefficients is as high  $\delta \approx 6.1\%$  at 22050 Hz and  $\delta \approx 6.7\%$  at 44100 Hz.

When focusing on the stability of the descriptors, we see that the implementation in Lib2 is generally more stable because the distributions have less variance, except for  $\delta$  and  $\rho$  at 22050 Hz. The decomposition in variance components indicates that the choice of frame size is irrelevant in general (low  $\hat{\sigma}_{FSize}^2$  scores), and that the largest part of the variability depends on the particular characteristics of the music pieces (very high  $\hat{\sigma}_{Track}^2 + \hat{\sigma}_{residual}^2$  scores). For Lib2 in particular, this means that controlling encodings or analysis parameters does not increase robustness significantly when the sampling rate is 22050 Hz; it depends almost exclusively on the specific music pieces. On the other hand, the combination of codec and bitrate has a quite large effect in Lib1. For instance, about 42% of the variability in Euclidean distances is due to the *BRate:Codec* interaction effect. This means that an appropriate selection of the codec and bitrate of the audio files leads to significantly more robust descriptors. At 44100 Hz both tools are clearly affected by the *BRate:Codec* effect as well, especially Lib1. Figure 1 compares the distributions of  $\delta$  scores for each tool. We can see that Lib1 has indeed large variance across groups, but small variance within groups, as opposed to Lib2. The robustness of Lib1 seems to converge to  $\delta \approx 3\%$  at 256 Kbps, and the descriptors are clearly more stable with larger bitrates (smaller within-group variance). On the other hand, the average robustness of Lib2 converges to  $\delta \approx 5\%$  at 160-192 Kbps, and stabil-



**Figure 1.** Distributions of  $\delta$  scores for different combinations of MP3 codec and bitrate at 44100 Hz, and for both audio analysis tools. Blue crosses mark the sample means. Outliers are rather uniformly distributed across genres.

		22050 Hz					44100 Hz				
		$\delta$	$\varepsilon$	$r$	$\rho$	$\theta$	$\delta$	$\varepsilon$	$r$	$\rho$	$\theta$
Lib1	$\hat{\sigma}_{FSize}^2$	1.08	3.03	1.73	0	1.74	0.21	0.09	0.01	0	0
	$\hat{\sigma}_{Codec}^2$	0	0	0	0	0	0	0	0	0	0
	$\hat{\sigma}_{BRate:Codec}^2$	31.25	42.13	21.61	8.38	21.49	46.98	41.77	22.52	24.03	21.51
	$\hat{\sigma}_{FSize \times Codec}^2$	0	0	0	0	0	0	0.20	0.07	0.05	0.06
	$\hat{\sigma}_{FSize \times (BRate:Codec)}^2$	4.87	11.71	12.36	1.23	13.21	7.37	18.25	17.98	10.85	18.02
	$\hat{\sigma}_{Genre}^2$	0.99	4.53	3.92	0.08	3.80	1.12	0.52	0.90	0.32	0.89
	$\hat{\sigma}_{Track}^2$	19.76	5.84	6.46	11.59	5.73	10.12	3.91	2.65	5.23	2.59
	$\hat{\sigma}_{residual}^2$	42.05	32.75	53.92	78.72	54.03	34.19	35.26	55.87	59.52	56.92
	Grand mean	0.0591	1.6958	0.9999	0.9977	0.9999	0.0682	1.8820	0.9998	0.9939	0.9998
	Total variance	0.0032	3.4641	1.8e-7	3.2e-5	1.5e-7	0.0081	11.44	1.6e-6	0.0005	1.4e-6
Standard deviation	0.0567	1.8612	0.0004	0.0056	0.0004	0.0897	3.3835	0.0013	0.0214	0.0012	
Lib2	$\hat{\sigma}_{FSize}^2$	1.17	0.32	0.16	0.24	0.18	0.25	0	0	0	0
	$\hat{\sigma}_{Codec}^2$	0	0	0	0	0	0	0	0	0	0
	$\hat{\sigma}_{BRate:Codec}^2$	4.91	6.01	2.32	0.74	3.14	23.46	24.23	14.27	13.31	15.02
	$\hat{\sigma}_{FSize \times Codec}^2$	0	0	0	0	0	0	0	0	0	0
	$\hat{\sigma}_{FSize \times (BRate:Codec)}^2$	0.96	0.43	0.03	0.04	0.09	7.17	8.09	10.35	6.34	10.86
	$\hat{\sigma}_{Genre}^2$	4.21	14.68	2.84	0.61	4.41	0.37	5.37	0.50	0	0.48
	$\hat{\sigma}_{Track}^2$	52.34	61.05	32.07	66.10	41.26	27.33	14.10	6.55	13.32	5.53
	$\hat{\sigma}_{residual}^2$	36.41	17.51	62.57	32.27	50.92	41.42	48.21	68.32	67.03	68.11
	Grand mean	0.0622	0.0278	0.9999	0.9955	0.9999	0.0656	0.0342	0.9998	0.9947	0.9999
	Total variance	0.0040	0.0015	8.9e-8	0.0002	3.5e-8	0.0055	0.0034	6.4e-7	0.0002	4.8e-7
Standard deviation	0.0631	0.0391	0.0003	0.0131	0.0002	0.0740	0.0587	0.0008	0.0150	0.0007	

**Table 1.** Variance components in the distributions of robustness of MFCCs for Lib1 (top) and Lib2 (bottom). Each component represents the percentage of total variance due to each effect (eg.  $\hat{\sigma}_{FSize}^2 = 3.03$  indicates that 3.03% of the variability in the robustness indicator is due to differences across frame sizes;  $\hat{\sigma}_x^2 = 0$  when the effect is so extremely small that the estimate is slightly below zero). All interactions with the *Genre* and *Track* main effects are confounded with the *residual* effect. The last rows show the grand mean, total variance and standard deviation of the distributions.

ity remains virtually the same beyond 96 Kbps. These plots confirm that the MFCC implementation in Lib1 is nearly twice as robust and stable when the encoding is homogeneous in the corpus, while the implementation in Lib2 is less robust but more stable with heterogeneous encodings.

The *FSize* effect is negligible, indicating that the choice of frame size does not affect the robustness of MFCCs in general. However, in several cases we can observe large  $\hat{\sigma}_{FSize \times (BRate:Codec)}^2$  scores, meaning that for some codec-bitrate combinations it does matter. An in-depth analysis shows that these differences only occur at 64 Kbps though (small frame sizes are more robust); differences are very small otherwise. Finally, the small  $\hat{\sigma}_{Genre}^2$  scores indicate that robustness is similar across music genres.

A similar analysis was conducted to assess the robustness and stability of chroma features. Even though the correlation indicators are generally high as well, Table 2 shows that chroma vectors do not preserve the shape as well as MFCCs do. When looking at individual coefficients, the relative errors are similarly  $\delta \approx 6\%$  in Lib1, but they are greatly reduced in Lib2, especially at 44100 Hz. In fact, the chroma implementation in Lib2 is more robust and stable according to all indicators<sup>4</sup>. For Lib1, virtually all the variability in the distributions is due to the *Track* and *residual* effects, meaning that chroma is similarly robust across encodings, analysis parameters and genre. For Lib2, we can similarly observe that errors in the correlation indicators depend almost entirely on the *Track* effect, but  $\delta$  and  $\varepsilon$  depend mostly on the codec-bitrate combination. This indicates that, despite chroma vectors preserve

their shape, the individual components vary significantly across encodings; we observed that increasing the bitrate leads to larger coefficients overall. This suggests that normalizing the chroma coefficients could dramatically improve the distributions of  $\delta$  and  $\varepsilon$ . We tried the parameter `normalization=2` to have Lib2 normalize chroma vectors to unit maximum. As expected, the effects of codec and bitrate are removed after normalization, and most of the variability is due to the *Track* effect. The correlation indicators are practically unaltered after normalization.

## 5. ROBUSTNESS IN GENRE CLASSIFICATION

The previous section provided indicators of robustness that can be easily understood. However, they can be hard to interpret because in the end we are interested in the robustness of the various algorithms that make use of these features; whether  $\delta = 5\%$  is large or not depends on how MFCCs and chroma are used in practice. To investigate this question we consider a music genre classification task. For each sampling rate, codec, bitrate and tool we trained one SVM model with radial basis kernel using MFCCs and another using chroma. For MFCCs we used a standard frame size of 2048, and for chroma we set 4096 in Lib1 and the fixed 16384 in Lib2. We did random sub-sampling validation with 100 random trials for each model, using 320 tracks for training and the remaining 80 for testing.

We first investigate whether a particular choice of encoding is likely to classify better when fixed across training and test sets. Table 3 shows the results for a selection of encodings at 44100 Hz. Within the same tool and descriptor, differences across encodings are quite small, approximately 0.02. In particular, for MFCCs and Lib1 an ANOVA analysis suggests that differences are signifi-

<sup>4</sup> Even though these distributions include all frame sizes in Lib1 but only 16384 in Lib2, the *FSize* effect is negligible in Lib1, meaning that these indicators are still comparable across implementations



		22050 Hz					44100 Hz				
		$\delta$	$\epsilon$	$r$	$\rho$	$\theta$	$\delta$	$\epsilon$	$r$	$\rho$	$\theta$
Lib1	$\hat{\sigma}_{FSize}^2$	1.68	2.77	0.20	0.15	0.38	2.37	2.42	0.24	0.34	0.50
	$\hat{\sigma}_{Genre}^2$	2.81	2.75	1.29	1.47	0.81	3.12	2.61	1.17	1.25	0.85
	$\hat{\sigma}_{Track}^2$	20.69	19.27	17.75	18.52	16.63	22.28	20.78	18.81	19.92	18.64
	$\hat{\sigma}_{residual}^2$	74.82	75.21	80.75	79.86	82.17	72.22	74.19	79.79	78.49	80.01
	Grand Mean	0.0610	0.0545	0.9554	0.9366	0.9920	0.0588	0.0521	0.9549	0.9375	0.9922
	Total variance	0.0046	0.0085	0.0276	0.0293	0.0014	0.0048	0.0082	0.0286	0.0298	0.0013
Standard deviation		0.0682	0.0924	0.1663	0.1713	0.0373	0.0695	0.0904	0.1691	0.1725	0.0355
Lib2	$\hat{\sigma}_{Codec}^2$	63.62	34.55	0	0	0	32.32	21.59	0	0	0
	$\hat{\sigma}_{BRate:Codec}^2$	0.71	0.23	0	0	0	61.80	39.51	0.01	0.03	0.04
	$\hat{\sigma}_{Genre}^2$	0.25	15.87	2.90	4.05	7.95	0.62	9.98	3.43	1.33	3.66
	$\hat{\sigma}_{Track}^2$	19.29	32.77	96.71	92.75	91.80	3.27	13.79	94.24	93.04	77.00
	$\hat{\sigma}_{residual}^2$	16.14	16.58	0.38	3.20	0.25	1.98	15.13	2.32	5.60	19.30
	Grand mean	0.0346	0.0031	0.9915	0.9766	0.9998	2.6e-2	2.2e-3	0.9989	0.9928	1
	Total variance	0.0004	5e-6	0.0002	0.0007	6.1e-8	4.6e-4	4.8e-6	3.7e-6	0.0001	1.8e-9
	Standard deviation		0.0195	0.0022	0.0135	0.0270	0.0002	0.0213	0.0022	0.0019	0.0122

**Table 2.** Variance components in the distributions of robustness of Chroma for Lib1 (top) and Lib2 (bottom), similar to Table 1. The *Codec* main effect and all its interactions are not shown for Lib1 because all variance components are estimated as 0. Note that the *FSize* main effect and all its interactions are omitted for Lib2 because it is fixed to 16384.

		64	96	128	160	192	256	320	WAV
Lib1	MFCCs	.383	.384	.401	.403	.395	.402	.394	.393
	Chroma	.275	.281	.288	.261	.278	.278	.284	.291
Lib2	MFCCs	.335	.329	.332	.341	.336	.336	.344	.335
	Chroma	.320	.325	.320	.323	.325	.319	.320	.313

**Table 3.** Mean classification accuracy over 100 trials when training and testing with the same encoding (MP3 CBR and WAV only) at 44100 Hz.

cant,  $F(7, 693) = 2.34, p = 0.023$ ; a multiple comparisons analysis reveals that 64 Kbps is significantly worse than the best (160 Kbps). In terms of chroma, differences are again statistically significant,  $F(7, 693) = 3.71, p < 0.001$ ; 160 Kbps is this time significantly worse than most of the others. With Lib2 differences are not significant for MFCCs,  $F(7, 693) = 1.07, p = 0.378$ . No difference is found for chroma either,  $F(7, 693) = 0.67, p = 0.702$ . Overall, despite some pairwise comparisons are significantly different, there is no particular encoding that clearly outperforms the others; the observed differences are probably just Type I errors. There is no clear correlation either between bitrate and accuracy.

We then investigate whether a particular choice of encoding for training is likely to produce better results when the target test set has a fixed encoding. For MFCCs and Lib1 there is no significant difference in any but one case (testing with 160 Kbps is worst when training with 64 Kbps). For chroma there are a few cases where 160 Kbps is again significantly worse than others, but we attribute these to Type I errors as well. Although not significantly so, the best result is always obtained when the training set has the same encoding as the target test set. With Lib2 there is no significant difference for MFCCs or chroma. Overall, we do not observe a correlation either between training and test encodings. Due to space constraints, we do not discuss results for VBR or 22050 Hz, but the same general conclusions can be drawn nonetheless.

## 6. DISCUSSION

Sigurdsson et al. [21] suggested that MFCCs are sensitive to the spectral perturbations that result from low bi-

trate compression, mostly due to distortions at high frequencies. They estimated squared Pearson's correlation between MFCCs computed on original lossless audio and its MP3 derivatives, using 4 different MFCC implementations. All implementations were found to be robust at bitrates of at least 128 Kbps, with  $r^2 > 0.95$ , but a significant loss in robustness was observed at 64 Kbps in some of the implementations. The most robust MFCC implementation had a highest frequency of 4600 Hz, while the least robust implementation included frequencies up to 11025 Hz. Their music corpus contained only 46 songs though, clearly limiting their results. In our experiments, all encodings show  $r^2 > 0.99$ . However, we note that Pearson's  $r$  is very sensitive to outliers with such small samples. This is the case of the first MFCC coefficients, which are orders of magnitude larger than the last coefficients. This makes  $r$  extremely large simply because the first coefficients are remotely similar; most of the variability between feature vectors is explained because of the first coefficient. This is clear in our Table 1, where  $r \approx 1$  and variance is nearly 0. To minimize this sensitivity to outliers, we also included the non-parametric Spearman's  $\rho$  correlation coefficient as well as the cosine similarity. In our case, the tool with the larger frequency range was shown to be more robust under homogeneous encodings, while the shorter range was more stable under heterogeneous conditions.

Hamawaki et al. [10] analyzed differences in the distribution of MFCCs for different bitrates using a corpus of 2513 MP3 files of Japanese and Korean pop songs with bitrates between 96 and 192 Kbps. Following a music similarity task, they compared differences in the top-10 ranked results when using MFCCs derived from WAV audio, its MP3 encoded versions, and the mixture of MFCCs from different sources. They found that the correlation of the results deteriorates smoothly as the bitrate decreases, while ranking on a set of MFCCs derived from different formats revealed uncorrelated results. We similarly observed that the differences between MFCCs of the original WAV files and its MP3 versions decrease smoothly with bitrate.

Jensen et al. [12] measured the effect of audio encoding on performance of an instrument classifier using MFCCs.

They compared MFCCs computed from MP3 files at only 32-64 Kbps, observing a decrease in performance when using a different encoder for training and test sets. In contrast, performance did not change significantly when using the same encoder. For genre classification with MFCCs, our results showed no differences in either case. We note though that the bitrates we considered are much larger. Uemura et al. [23] examined the effect of bitrate on chord recognition using chroma features with an SVM classifier. They observed no obvious correlation between encoding and estimation results; the best results were even obtained with very low bitrates for some codecs. Our results on genre classification with chroma largely agree in this case as well; the best results with Lib2 were also obtained by low bitrates. Casey et al. [4] evaluated the effect of lossy encodings on genre classification tasks using audio spectrum projection features. They found a small but statistically significant decrease in accuracy for bitrates of 32 and 96 Kbps. In our experiments, we do not observe these differences, although the lowest bitrate we consider is 64 Kbps. Jacobson et al. [11] also investigated the robustness of onset detection methods to lossy MP3 encoding. They found statistically significant changes in accuracy only at bitrates lower than 32 Kbps.

Our results showed that MFCCs and chroma features, as computed by Lib1 and Lib2, are generally robust and stable within reasonable limits. Some differences have been noted between tools though, largely attributable to the different frequency ranges they employ. Nonetheless, it is evident that certain combinations of codec and bitrate may require a re-parameterization of some descriptors to improve or even maintain robustness. In practice, these parameterizations affect the performance and applicability of algorithms, so a balance between performance, robustness and generalizability should be sought. These considerations are of major importance when collecting audio files for some dataset, as a minimum audio quality might be needed for some descriptors.

## 7. CONCLUSIONS

In this paper we have studied the robustness of two common audio descriptors used in Music Information Retrieval, namely MFCCs and chroma, to different audio encodings and analysis parameters. Using a varied corpora of music pieces and two different audio analysis tools we have confirmed that MFCCs are robust to frame/hop sizes and lossy encoding provided that a minimum bitrate of approximately 160 Kbps is used. Chroma features were shown to be even more robust, as the codec and bitrates had virtually no effect on the computed descriptors. This is somewhat expected given that chroma does not capture information as fine-grained as MFCCs do, and that lossy compression does not alter the perceived tonality. We did find subtle differences between implementations of these audio features, which call for further research on standardizing algorithms and parameterizations to maximize their robustness while maintaining their effectiveness in the various tasks they are used in. The immediate line for future work includes the analysis of other features and tools.

## 8. ACKNOWLEDGMENTS

This work is partially supported by an A4U postdoctoral grant and projects SIGMUS (TIN2012-36650), Comp-Music (ERC 267583), PHENICX (ICT-2011.8.2) and GiantSteps (ICT-2013-10).

## 9. REFERENCES

- [1] J.J. Aucouturier, F. Pachet, and M. Sandler. "The way it sounds": timbre models for analysis and retrieval of music signals. *IEEE Trans. Multimedia*, 2005.
- [2] D. Bogdanov, N. Wack, et al. ESSENTIA: an audio analysis library for music information retrieval. In *ISMIR*, 2013.
- [3] C. Cannam, M.O. Jewell, C. Rhodes, M. Sandler, and M. d'Inverno. Linked data and you: bringing music research software into the semantic web. *J. New Music Res.*, 2010.
- [4] M. Casey, B. Fields, et al. The effects of lossy audio encoding on genre classification tasks. In *AES*, 2008.
- [5] W. Chai. Semantic segmentation and summarization of music: methods based on tonality and recurrent structure. *IEEE Signal Processing Magazine*, 2006.
- [6] D. Ellis. Classifying music audio with timbral and chroma features. In *ISMIR*, 2007.
- [7] T. Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. In *ICMC*, 1999.
- [8] T. Ganchev, N. Fakotakis, and G. Kokkinakis. Comparative evaluation of various MFCC implementations on the speaker verification task. In *SPECOM*, 2005.
- [9] E. Gómez. *Tonal description of music audio signals*. PhD thesis, Universitat Pompeu Fabra, 2006.
- [10] S. Hamawaki, S. Funasawa, et al. Feature analysis and normalization approach for robust content-based music retrieval to encoded audio with different bit rates. In *MMM*, 2008.
- [11] K. Jacobson, M. Davies, and M. Sandler. The effects of lossy audio encoding on onset detection tasks. In *AES*, 2008.
- [12] J.H. Jensen, M.G. Christensen, D. Ellis, and S.H. Jensen. Quantitative analysis of a common audio similarity measure. *IEEE TASLP*, 2009.
- [13] B. McFee, L. Barrington, and G. Lanckriet. Learning content similarity for music recommendation. *IEEE TASLP*, 2012.
- [14] D.C. Montgomery. *Design and Analysis of Experiments*. Wiley & Sons, 2009.
- [15] M. Müller and S. Ewert. Towards timbre-invariant audio features for harmony-based music. *IEEE TASLP*, 2010.
- [16] M. Müller, H. Mattes, and F. Kurth. An efficient multiscale approach to audio synchronization. In *ISMIR*, 2006.
- [17] J. Paulus, M. Müller, and A. Klapuri. Audio-based music structure analysis. In *ISMIR*, 2010.
- [18] L.R. Rabiner and R.W. Schafer. *Introduction to Digital Speech Processing*. Foundations and Trends in Signal Processing. 2007.
- [19] J. Reed and C. Lee. Preference music ratings prediction using tokenization and minimum classification error training. *IEEE TASLP*, 2011.
- [20] J. Serrà, E. Gómez, and P. Herrera. Audio cover song identification and similarity: background, approaches, evaluation, and beyond. In Z. Raś and A.A. Wiczorkowska, editors, *Advances in Music Information Retrieval*. Springer, 2010.
- [21] S. Sigurdsson, K.B. Petersen, and T. Lehn-Schiler. Mel Frequency Cepstral Coefficients: an evaluation of robustness of MP3 encoded music. In *ISMIR*, 2006.
- [22] M. Slaney. Auditory toolbox. *Interval Research Corporation, Technical Report*, 1998. <http://engineering.purdue.edu/~malcolm/interval/1998-010/>.
- [23] A. Uemura, K. Ishikura, and J. Katto. Effects of audio compression on chord recognition. In *MMM*, 2014.
- [24] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and HG. Okuno. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model. *IEEE TASLP*, 2008.

# MUSIC INFORMATION BEHAVIORS AND SYSTEM PREFERENCES OF UNIVERSITY STUDENTS IN HONG KONG

**Xiao Hu**

University of Hong Kong  
xiaoxhu@hku.hk

**Jin Ha Lee**

University of Washington  
jinhalee@uw.edu

**Leanne Ka Yan Wong**

University of Hong Kong  
wkayan@connect.hku.hk

## ABSTRACT

This paper presents a user study on music information needs and behaviors of university students in Hong Kong. A mix of quantitative and qualitative methods was used. A survey was completed by 101 participants and supplemental interviews were conducted in order to investigate users' music information related activities. We found that university students in Hong Kong listened to music frequently and mainly for the purposes of entertainment, singing and playing instruments, and stress reduction. This user group often searches for music with multiple methods, but common access points like genre and time period were rarely used. Sharing music with people in their online social networks such as Facebook and Weibo was a common activity. Furthermore, the popularity of smartphones prompted the need for streaming music and mobile music applications. We also examined users' preferences on music services available in Hong Kong such as YouTube and KKBox, as well as the characteristics liked and disliked by the users. The results not only offer insights into non-Western users' music behaviors but also for designing online music services for young music listeners in Hong Kong.

## 1. INTRODUCTION AND RELATED WORK

Seeking music and music information is prevalent in our everyday life as music is an indispensable element for many people [1]. People in Hong Kong are not an exception. Hong Kong has the second highest penetration rate of broadband Internet access in Asia, following South Korea<sup>1</sup>. Consequently, Hong Kongers are increasingly using various online music information services to seek and listen to music, including iTunes, YouTube, Kugou, Sogou and Baidu<sup>2</sup>. However, our current understanding of their music information needs and behaviors are still lacking, as few studies explored user populations in Hong Kong, or in any non-Western cultures.

Hong Kong is a unique location that merges the West-

ern and Eastern cultures. Before the handover to the Chinese government in 1997, Hong Kong had been ruled by the British government for 100 years. This had resulted in a heavy influence of Western culture, although much of the Chinese cultural heritage has also been preserved well in Hong Kong. The cultural influences of Hong Kong to the neighboring regions in Asia were significant, especially in the pre-handover era. In fact, in the 80s and throughout the 90s, Cantopop (Cantonese popular music, sometimes referred to as HK-pop) was widely popular across many Asian countries, and produced many influential artists such as Leslie Cheung, Anita Mui, Andy Lau, and so on [2]. In the post-handover era, there has been an influx of cultural products from mainland China which is significantly affecting the popular culture of Hong Kong [8]. The cultural history and influences of Hong Kong, especially paired with the significance of Cantopop, makes it an interesting candidate to explore among many non-Western cultures.

Of the populations in Hong Kong, we specifically wanted to investigate young adults on their music information needs and behaviors. They represent a vibrant population who are not only heavily exposed to and fast adopters of new ideas, but also represent the future workforce and consumers. University students in Hong Kong are mostly digital natives (i.e., grew up with access to computers and the Internet from an early age) with rich experience of seeking and listening to digital music. Additionally the fact that they are influenced by both Western and Eastern cultures, and exposed to both global and local music make them worthy of exploring as a particular group of music users.

There have been a few related studies which investigated music information users in Hong Kong. Lai and Chan [5] surveyed information needs of users in an academic music library setting. They found that the frequencies of using score and multimedia were higher than using electronic journal databases, books, and online journals. Nettamo et al. [9] compared users in New York City and those in Hong Kong in using their mobile devices for music-related tasks. Their results showed that users' envi-



© Xiao Hu, Jin Ha Lee, Leanne Ka Yan Wong.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Xiao Hu, Jin Ha Lee, Leanne Ka Yan Wong. "Music Information Behaviors and System Preferences of University Students in Hong Kong", 15th International Society for Music Information Retrieval Conference, 2014.

<sup>1</sup> <http://www.itu.int/ITU-D/ICTEYE/Reporting/DynamicReportWizard.aspx>

<sup>2</sup> <http://hk.epochtimes.com/b5/11/10/20/145162.htm>

ronment and context greatly influenced their behaviors, and there were cultural differences in consuming and managing mobile music between the two user groups. Our study investigates everyday music information behaviors of university students in Hong Kong, and thus the scope is broader than these studies.

In addition to music information needs and behaviors, this study also examines the characteristics of popular music services adopted by university students in Hong Kong, in order to investigate their strengths and weaknesses. Recommendations for designing music services are proposed based on the results. This study will improve our understanding on music information behaviors of the target population and contribute to the design of music services that can better serve target users.

## 2. METHODS

A mix of quantitative and qualitative methods was used in order to triangulate our results. We conducted a survey in order to collect general information about target users' music information needs, seeking behaviors, and opinions on commonly used music services. Afterwards, follow-up face-to-face interviews of a smaller user group were conducted to collect in-depth explanations on the themes and patterns discovered in the survey results. Prior to the formal survey and interviews, pilot tests were carried out with a smaller group of university students to ensure that the questions were well-constructed and students were able to understand and answer them without major issues.

### 2.1 Survey

The survey was conducted as an online questionnaire. The questionnaire instrument was adapted from the one used in [6] and [7], with modifications to fit the multilingual and multicultural environment. Seventeen questions about the use of popular music services were added to the questionnaire. The survey was implemented with LimeSurvey, an open-source survey application, and consisted of five parts: demographic information, music preference, music seeking behaviors, music collection management, and opinions on preferred music services. Completing the survey took approximately 30 minutes, and each participant was offered a chance to enter his/her name for a raffle to win one of the three supermarket gift coupons of HKD50, if they wished.

The target population was students (both undergraduate and graduate) from the eight universities sponsored by the government of Hong Kong Special Administrative Region. The sample was recruited using Facebook due to its popularity among university students in Hong Kong. Survey invitations were posted on Facebook, initially through the list of friends of the authors, and then further disseminated by chain-referrals.

### 2.2 Interviews

Semi-structured interviews were conducted after the survey data were collected and analyzed, in order to seek in-

depth explanations to support the survey findings. Face-to-face interviews were carried out individually with five participants from three different universities. The interviews were conducted in Cantonese, the mother tongue of the interviewees, and were later transcribed and translated to English. Each interview lasted up to approximately 20 minutes.

## 3. SURVEY DATA ANALYSIS

Of the 167 survey responses collected, 101 complete responses were analyzed in this study. All the survey participants were university students in Hong Kong. Among them, 58.4% of were female and 41.6% of them were male. They were all born between 1988 and 1994, and most of them (88.1%) were born between 1989 and 1992. Therefore, they were in their early 20s when the survey was taken in 2013. Nearly all of them (98.0%) were undergraduates majoring Science/Engineering (43.6%), Social Sciences/Humanities (54.0%) and Other (2.0%).

### 3.1 Music Preferences

In order to find out participants' preferred music genres, they were asked to select and rank up to five of their favorite music genres from a list of 25 genres covering most Western music genres. To ensure that the participants understand the different genres, titles and artist names of example songs representative of each genre were provided. The results are shown in Table 1 where each cell represents the number of times each genre was mentioned with the rank corresponding in the column. Pop was the most preferred genre among the participants, followed by R&B/Soul and Rock. We also aggregated the results by assigning reversely proportional weights to the ranks (1st: 5 points, and 5th: 1 point). The most popular music genres among the participants were Pop (311 pts), R&B/Soul (204 pts), Rock (109 pts), Gospel (88 pts) and Jazz (86 pts).

	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>	5 <sup>th</sup>	Total	Total (%)
Pop	43	14	9	4	5	75	74.2%
R&B	7	29	11	7	6	60	59.4%
Rock	6	9	10	3	7	35	34.7%
Gospel	9	6	2	4	5	26	25.7%
Jazz	6	8	3	5	5	27	26.7%

**Table 1.** Preferences on music genres

Moreover, as both Chinese and English are official languages of Hong Kong, participants were also asked to rank their preferences on languages of lyrics. The five options were English, Cantonese, Mandarin, Japanese and Korean. The last three were included due to popularity of songs from nearby countries/regions in Hong Kong, including mainland China and Taiwan (Mandarin), Japan (Japanese), and Korea (Korean). As shown in Table 2, English was in fact highly preferred, followed by Cantonese. Mandarin was mostly ranked at the second or third

place, while Korean and Japanese were ranked lower. We also aggregated the answers and found that the most popular languages in songs are English (394 points), Cantonese (296 points), and Mandarin (223 points).

	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>	5 <sup>th</sup>	Total	Total (%)
English	46	27	16	3	2	94	93.1%
Cantonese	31	20	15	7	2	75	74.3%
Mandarin	13	23	20	2	2	60	59.4%
Korean	6	15	6	16	14	57	56.4%
Japanese	5	5	10	16	16	52	51.5%

**Table 2.** Preferences on languages of song lyrics

### 3.2 Music Seeking Behaviors

When asked about the type of music information they have ever searched, most participants indicated preferences on audio: MP3s and music videos (98.0%), music recordings (e.g., CDs, vinyl records, tapes) (94.1%), and music multimedia in other formats (e.g., Blue-ray, DVD, VHS) (88.1%). Written forms of music information were sought by fewer respondents: books on music (73.2%), music magazines (69.3%), and academic music journals (63.4%). Approximately one out of three participants even responded that they have never sought music magazines (30.7%) or academic music journals (36.6%).

As for the frequency of search, 41.6% of respondents indicated that they sought MP3s and music videos at least a few times a week, compared to only 18.8% for music recordings (e.g., CDs, vinyl records, tapes) and 24.8% for music multimedia in other formats (e.g., Blue-ray, DVD).

Moreover, 98.0% of participants responded that they had searched for music information on the Internet. Among them, almost all (99.0%) answered that they had downloaded free music online, and 95.0% responded that they had listened to streaming music or online radio. This clearly indicates that participants sought digital music more often through online channels than offline or physical materials. However, even though 77.8% of respondents had visited online music store, only 69.7% of them had purchased any electronic music files or albums. Not surprisingly, participants preferred free music resources.

Music was certainly a popular element of entertainment in the lives of the participants. When asked why they sought music, all participants included entertainment in their answers. Also, a large proportion (83.0%) indicated that they sought music for entertainment at least a few times a week. Furthermore, 97.0% of respondents search for music information for singing or playing a musical instrument for fun. This proportion is significantly higher than the results from the previous survey of university population in the United States (32.8% for singing and 31.9% for playing a musical instrument) [6]. In addition, 78.2% of our respondents do this at least two or three times a month. We conjecture that this is most likely due to the popularity of karaoke in Hong Kong.

Known-item search was the most common type of music information seeking; nearly all respondents (95.1%) sought music information for the identification/verification of musical works, artist and lyrics, and about half of them do so at least a few times a week. Obtaining background information was also a strong reason; over 90% of the participants sought music to learn more about music artists (97.0%) as well as music (94.1%), and approximately half of them (53.5% and 40.6%, respectively) sought this kind of music information at least two or three times a month.

When asked which sources stimulated or influenced their music information needs, all 101 participants acknowledged online video clips (e.g. YouTube) and TV shows/movies. This suggests that the influence of other media using music is quite significant which echoes the finding that associative metadata in music seeking was important for the university population in the United States [6]. Also over 70% of the participants' music needs were influenced by music heard in public places, advertisement/commercial, radio show, or family members'/friends' home.

As for the metadata used in searching for music, performer was the most popular access point with 80.2% of positive responses, followed by the title of work(s) (65.3%) and some words of lyrics (62.4%). Other common types of metadata such as genre and time period were only used by a few respondents (33.7% and 29.7%, respectively). Particularly for genre, the proportion is significantly lower than 62.7% as found in the prior survey of university population in the United States [6]. This is perhaps related to the exposure to different music genres in Hong Kong, and the phenomenon that Hong Kongers music listeners tend to emphasize an affinity with friends while Americans (New Yorkers) are more likely to use music to highlight their individual personalities [9]. Moreover, participants responded that they would also seek music based on other users' opinions: 57.4% by recommendations from other people and 52.5% by popularity. The proportion for popularity is also fairly larger than the 31% in [6]. This shows that the social aspect is a crucial factor affecting participants' music seeking behaviors.

Of the different types of people, friends and family members (91.1%) and people on their social network websites (e.g. Facebook, Weibo) (89.1%) were the ones whom they most likely ask for help when searching for music. In addition, they turned to the Internet more frequently than friends and family members. Thirty-nine percent of them sought help on social network websites at least a few times a week while only 23.8% turned to friends/family members at least a few times a week.

On the other hand, when asked which physical places they go to in order to search for music or music information, 82.18% said that they would find music in family members' or friends' home, which was higher than going to record stores (75.3%), libraries (70.3%), and academic

institutions (64.4%). Overall, these data show that users' social networks, and especially online networks are important for their music searching process.

### 3.3 Music Collection Management

More participants were managing a digital collection (40.6%) than a physical one (25.7%). On average, each respondent estimated that he/she managed 900 pieces of digital music and 94 pieces of music in physical formats. This shows that managing digital music is more popular among participants, although the units that they typically associate with digital versus physical items might differ (e.g., digital file vs. physical album).

We also found that students tended to manage their music collections with simple methods. Over half of the respondents (50.0% for music in physical formats and 56.1% for digital music) manage their music collection by artist name. Participants sometimes also organized their digital collections by album title (17.7%), but rarely by format type (3.9%) and never by record label. More participants indicated they did not organize their music at all for their physical music collection (19.2%) than their digital music collection (2.4%). When they did organize their physical music collection, they would use album title (11.5%) and genre (11.5%). Overall, organizing the collection did not seem to be one of the users' primary activities related to music information.

### 3.4 Preferred Music Services

Respondents gave a variety of responses regarding their most frequently visited music services: YouTube (51.5%), KKBox (26.7%), and iTunes (14.9%) were the most popular ones. KKBox is a large cloud-based music service provider founded in Taiwan, very popular in the region and sometimes referred to as "Asian Spotify." YouTube, which provides free online streaming music video, was almost twice as popular as the second most favored music service, KKBox. The popularity of YouTube was also observed in Lee and Waterman's survey of 520 music users in 2012 [7]. Their respondents ranked Pandora as the most preferred service, followed by YouTube as the second.

The participants were also asked to evaluate their favorite music services. Specifically, they were asked to indicate their level of satisfaction using a 5-point Likert scale on 15 different aspects on search function, search results and system utility. Table 3 shows the percentage of positive (aggregation of "somewhat satisfied" and "very satisfied") and negative (aggregation of "somewhat unsatisfied" and "very unsatisfied") ratings among users who chose each of the three services as their most favored one.

For those who selected YouTube as their most frequently used service, they indicated that they were especially satisfied with its keyword search function (74.5%), recommendation of keywords (70.6%), variety of available music information (60.8%) and attractive interface

(56.9%). Only a few respondents (9.8%) were unsatisfied with certain features of YouTube such as advanced search, relevance of search results, and navigation. It is surprising to see that five respondents rated YouTube negatively on the aspect of price. We suspect they might have associated this aspect with the price of purchasing digital music from certain music channels on YouTube, or the indirect cost of having to watch ads. However, we did not have the means to identify these respondents to verify the reasons behind their ratings.

		YouTube		KKBox		iTunes	
		P	N	P	N	P	N
search function	keyword search	74.5	7.8	29.6	7.4	13.3	0.0
	advanced search	54.9	9.8	44.4	18.5	46.7	6.7
	content-based search	51.0	7.8	44.4	29.6	66.7	13.3
	auto-correction	49.0	7.8	29.6	29.6	20.0	33.3
	keywords suggestion	70.6	3.9	40.7	25.9	20.0	53.3
search results	number of results	52.9	7.8	40.7	22.2	6.7	33.3
	relevance	47.1	9.8	48.1	18.5	13.3	33.3
	accuracy	49.0	7.8	44.4	18.5	33.3	26.7
utility	price of the service	39.2	9.8	25.9	25.9	33.3	20.0
	accessibility	52.9	7.8	22.2	37.0	26.7	20.0
	navigation	52.9	9.8	18.5	29.6	6.7	20.0
	variety of available music information	60.8	7.8	22.2	22.2	26.7	13.3
	music recommendation	52.9	7.8	33.3	22.2	53.3	20.0
	interface attractiveness	56.9	3.9	33.3	7.4	40.0	20.0
	music sharing	47.1	3.9	40.7	7.4	40.0	20.0

**Table 3** User ratings of three most preferred music services ("P": positive; "N": negative, in percentage)

The level of satisfaction for KKBox was lower than that of YouTube. Nearly half of the participants who use KKBox were satisfied with its relevance of results (48.1%), advanced search function (44.4%) and content-based search function (44.4%). The aspects of KKBox that participants did not like included the lack of accessibility (37.0%), content-based search function (29.6%), and auto-correction (29.6%). Interestingly, the content-based search function in KKBox was controversial among the participants. Some participants liked it probably because it was a novel feature that few music services had; while others were not satisfied with it, perhaps due to fact that current performance of audio content-based technologies have yet to meet users' expectation.

Only 15 participants rated iTunes as their most frequently used music service. Their opinions on iTunes were mixed. Its content-based search function and music recommendations were valued by 66.7% and 53.3% of the 15 participants, respectively. The data seem to suggest that audio content-based technologies in iTunes performed better than KKBox, but this must be verified with a larger sample in future work. On the other hand, over

half of the respondents gave negative response to the keyword suggestion function in iTunes. Moreover, the auto-correction, number of search results, and relevance of search results also received negative responses by one third of the respondents. These functions are related to the content of music collection in iTunes, and thus we suspect that the coverage of iTunes perhaps did not meet the expectations of young listeners in Hong Kong, as much as the other two services did.

#### 4. THEMES/TRENDS FROM INTERVIEWS

##### 4.1 Multiple Music Information Searching Strategies

Interviewees searched for music using not only music services like YouTube or KKBox, but also general-purpose search engines, such as Google and Yahoo!. Most often, a simple keyword search with the song title or artist name was conducted when locating music in these music services. However, more complicated searches such as those using lyrics and the name of composer are not supported by most existing music services. In this case, search engines had to be used. For example, if the desired song title and artist name are unknown or inaccurate, interviewees would search for them on Google or Yahoo! with any information they know about the song. The search often directed them to the right piece of metadata which then allowed them to conduct a search in YouTube or other music services. As expected, this does not always lead to successful results; one participant said *“when I did not know the song title or artist name, I tried singing the song to Google voice search, but the result was not satisfactory.”*

##### 4.2 Use of Online Social Networks

Online social network services are increasingly popular among people in Hong Kong. According to an online survey conducted with 387 Hong Kong residents in March 2011<sup>3</sup>, the majority of the respondents visited Facebook (92%), read blogs (77%) and even wrote blog posts (52%). Social media provides a convenient way for people to connect with in Hong Kong where maintaining a work-life balance can be quite challenging.

University students in Hong Kong are also avid social media users. They prefer communicating and sharing information with others using online social networks for the efficiency and flexibility. Naturally, it also serves as a convenient channel for sharing music recommendations and discussing music-related topics. Relying on others was considered an important way to search for music: *“Normally, I will consider others’ opinions first. There are just way too many songs, so it helps find good music much more easily.”*, *“I love other people’s comments, especially when they have the same view as me!”*

<sup>3</sup> Hong Kong social media use higher than United States: <http://travel.cnn.com/hong-kong/life/hong-kong-social-media-use-higher-united-states-520745>.

##### 4.3 24/7 Online Music Listening

Participants in this study preferred listening to or watching streaming music services rather than downloading music. Downloading an mp3 file of a song usually takes about a half minute with a broadband connection and slightly longer with a wireless connection. Interviewees commented that downloading just added an extra step which was inconvenient to them.

Apart from the web, smart mobile devices are becoming ubiquitous which is also affecting people’s mode of music listening. According to Mobilezine<sup>4</sup>, 87% of Hong Kongers aged between 15 and 64 own a smart device. According to Phneah [10], 55% of Hong Kong youths think that the use of smartphones dominates their lives as they are unable to stop using smartphones even in restrooms, and many sleep next to it. As expected, university students in Hong Kong are accustomed to having 24/7 access to streaming music on their smartphones.

#### 5. IMPLICATIONS FOR MUSIC SERVICES

##### 5.1 Advanced Search

A simple keyword search may not be sufficient to accommodate users who want to search for music with various metadata, not only with song titles, but also performer’s names, lyrics, and so on. For example, if a user wants to locate songs with the word “lotus” in the lyrics, they would simply use “lotus” as the search keyword. However, the search functions in various music services generally are not intelligent enough to understand the semantic differences among the band named Lotus and the word “lotus” in lyrics, not to mention which role the band Lotus might have played (e.g., performer, composer, or both). As a result, users have to conduct preliminary searches in web search engines as an extra step when attempting to locate the desired song. Many users will appreciate having an advanced search function with specific fields in music services that allow them to conduct lyric search with “lotus” rather than a general keyword search.

##### 5.2 Mood Search

Participants showed great interests in the feeling or emotion in music, as they perceived the meaning of songs were mostly about particular emotions. Terms such as “positive”, “optimistic”, and “touching” were used to describe the meaning of music during the interviews. Therefore, music services that can support searching by mood terms may be useful.

Music emotion or mood has been recognized as an important access point for music [3]. A cross-cultural study by Hu and Lee [4] points out that listeners from different cultural backgrounds have different music mood judgments and they tend to agree more with users from the

<sup>4</sup> Hong Kong has the second highest smartphone penetration in the world: <http://mobilezine.asia/2013/01/hong-kong-has-the-second-highest-smartphone-penetration-in-the-world/>.

same cultural background than users from other cultures. This cultural difference must be taken into account when establishing mood metadata for music services.

### 5.3 Connection with Social Media

Social media play a significant role in sharing and discussing music among university students in Hong Kong. YouTube makes it easy for people to share videos in various online social communities such as Facebook, Twitter and Google Plus. Furthermore, users can view the shared YouTube videos directly on Facebook which makes it even more convenient. This is one of the key reasons our participants preferred YouTube. However, music services like iTunes have yet to adopt this strategy. For our study population, linking social network to music services would certainly enhance user experience and help promote music as well.

### 5.4 Smartphone Application

Many participants are listening to streaming music with their smartphones, and thus naturally, offering music apps for smart devices will be critical for music services. Both YouTube and iTunes offer smartphone apps. Moreover, instant messaging applications, such as WhatsApp, is found as the most common reason for using smartphones among Hong Kongers [10]. To further improve the user experience, music-related smartphone apps may consider incorporating online instant messaging capabilities.

## 6. CONCLUSION

Music is essential for many university students in Hong Kong. They listen to music frequently for the purpose of entertainment and relaxation, to help reduce stress in their extremely tense daily lives. Currently, there does not exist a single music service that can fulfill all or most of their music information needs, and thus they often use multiple tools for specific searches. Furthermore, sharing and acquiring music from friends and acquaintances was a key activity, mainly done on online social networks. Comparing our findings to those of previous studies revealed some cultural differences between Hong Kongers and Americans, such as Hong Kongers relying more on popularity and significantly less on genres in music search.

With the prevalence of smartphones, students are increasingly becoming “demanding” as they get accustomed to accessing music anytime and anywhere. Streaming music and music apps for smartphones are becoming increasingly common. The most popular music service among university students in Hong Kong was YouTube due to its convenience, user-friendly interface, and requiring no payment to use their service. In order to further improve the design of music services, we recommended providing an advanced search function, emotion/mood-based search, social network connection, smartphone apps as well as access to high quality digital music which will help fulfill users’ needs.

## 7. ACKNOWLEDGEMENT

The study was partially supported by a seed basic research project in University of Hong Kong. The authors extend special thanks to Patrick Ho Ming Chan for assisting in data collection.

## 8. REFERENCES

- [1] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney: “Content-Based Music Information Retrieval: Current Directions and Future Challenges,” *Proceedings of the IEEE*, 96 (4), pp. 668-696, 2008.
- [2] S. Y. Chow: “Before and after the Fall: Mapping Hong Kong Cantopop in the Global Era,” *LEWI Working Paper Series*, 63, 2007.
- [3] X. Hu: “Music and mood: Where theory and reality meet,” *Proceedings of iConference*. 2010.
- [4] X. Hu and J. H. Lee: “A Cross-cultural Study of Music Mood Perception between American and Chinese Listeners,” *Proceedings of the ISMIR*, pp.535-540, 2012.
- [5] K. Lai and K. Chan: “Do you know your music users' needs? A library user survey that helps enhance a user-centered music collection.” *The Journal of Academic Librarianship*, 36(1), pp.63-69, 2010.
- [6] J. H. Lee and S. J. Downie: “Survey of music information needs, uses, and seeking behaviours: Preliminary findings,” *Proceedings of the ISMIR*, pp. 441-446, 2004.
- [7] J. H. Lee and M. N. Waterman: “Understanding user requirements for music information services,” *Proceedings of the ISMIR*, pp. 253-258, 2012.
- [8] B. T. McIntyre, C. C. W. Sum, and Z. Weiyu: “Cantopop: The voice of Hong Kong,” *Journal of Asian Pacific Communication*, 12 (2), pp. 217-243, 2002.
- [9] E. Nettamo, M. Norhamo, and J. Häkkinä: “A cross-cultural study of mobile music: Retrieval, management and consumption,” *Proceedings of OzCHI 2006*, pp. 87-94, 2006.
- [10] J. Phneah: “Worrying signals as smartphone addiction soars,” *The Standard*. Retrieved from [http://www.thestandard.com.hk/news\\_detail.asp?pp\\_cat=30&art\\_id=132763&sid=39444767&con\\_type=1](http://www.thestandard.com.hk/news_detail.asp?pp_cat=30&art_id=132763&sid=39444767&con_type=1), 2013.
- [11] V. M. Steelman: “Intraoperative music therapy: Effects on anxiety, blood pressure,” *Association of Operating Room Nurses Journal*, 52(5), pp. 1026-1034, 1990.



# LYRICSRADAR: A LYRICS RETRIEVAL SYSTEM BASED ON LATENT TOPICS OF LYRICS

Shoto Sasaki<sup>\*1</sup> Kazuyoshi Yoshii<sup>\*\*2</sup> Tomoyasu Nakano<sup>\*\*\*3</sup> Masataka Goto<sup>\*\*\*4</sup> Shigeo Morishima<sup>\*5</sup>

<sup>\*</sup>Waseda University <sup>\*\*</sup>Kyoto University

<sup>\*\*\*</sup>National Institute of Advanced Industrial Science and Technology (AIST)

<sup>1</sup>joudanjanai-ss[at]akane.waseda.jp <sup>2</sup>yoshii[at]kuis.kyoto-u.ac.jp

<sup>3,4</sup>(t.nakano, m.goto)[at]aist.go.jp <sup>5</sup>shigeo[at]waseda.jp

## ABSTRACT

This paper presents a lyrics retrieval system called *LyricsRadar* that enables users to interactively browse song lyrics by visualizing their topics. Since conventional lyrics retrieval systems are based on simple word search, those systems often fail to reflect user's intention behind a query when a word given as a query can be used in different contexts. For example, the word "tears" can appear not only in sad songs (e.g., feel heartrending), but also in happy songs (e.g., weep for joy). To overcome this limitation, we propose to automatically analyze and visualize topics of lyrics by using a well-known text analysis method called latent Dirichlet allocation (LDA). This enables *LyricsRadar* to offer two types of topic visualization. One is the topic radar chart that visualizes the relative weights of five latent topics of each song on a pentagon-shaped chart. The other is radar-like arrangement of all songs in a two-dimensional space in which song lyrics having similar topics are arranged close to each other. The subjective experiments using 6,902 Japanese popular songs showed that our system can appropriately navigate users to lyrics of interests.

## 1. INTRODUCTION

Some listeners regard lyrics as essential when listening to popular music. It was, however, not easy for listeners to find songs with their favorite lyrics on existing music information retrieval systems. They usually happen to find songs with their favorite lyrics while listening to music. The goal of this research is to assist listeners who think the lyrics are important to encounter songs with unfamiliar but interesting lyrics.

Although there were previous lyrics-based approaches for music information retrieval, they have not provided an interface that enables users to interactively browse lyrics of many songs while seeing latent topics behind those lyrics. We call these latent topics *lyrics topics*. Several

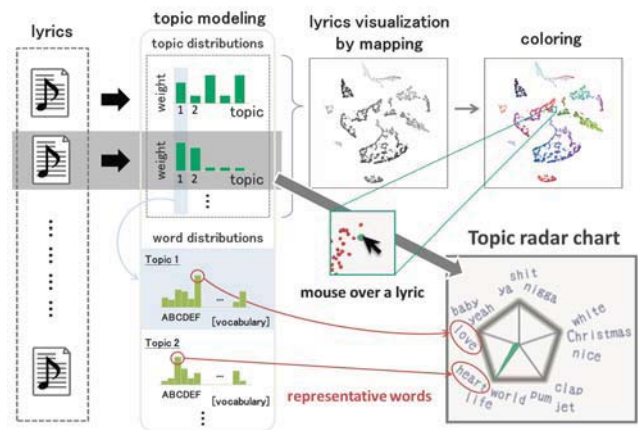


Figure 1. Overview of topic modeling of *LyricsRadar*.

approaches analyzed the text of lyrics by using natural language processing to classify lyrics according to emotions, moods, and genres [2, 3, 11, 19]. Automatic topic detection [6] and semantic analysis [1] of song lyrics have also been proposed. Lyrics can be used to retrieve songs [5] [10], visualize music archives [15], recommend songs [14], and generate slideshows whose images are matched with lyrics [16]. Some existing web services for lyrics retrieval are based on social tags, such as "love" and "graduation". Those services are useful, but it is laborious to put appropriate tags by hands and it is not easy to find a song whose tags are also put to many other songs. Macrae *et al.* showed that online lyrics are inaccurate and proposed a ranking method that considers their accuracy [13]. Lyrics are also helpful for music interfaces: LyricSynchronizer [8] and VocaRefiner [18], for example, show the lyrics of a song so that a user can click a word to change the current playback position and the position for recording, respectively. Latent topics behind lyrics, however, were not exploited to find favorite lyrics.

We therefore propose a lyrics retrieval system, *LyricsRadar*, that analyzes the lyrics topics by using a machine learning technique called latent Dirichlet allocation (LDA) and visualizes those topics to help users find their favorite lyrics interactively (Fig.1). A single word could have different topics. For example, "diet" may at least have two



© Shoto Sasaki, Kazuyoshi Yoshii, Tomoyasu Nakano, Masataka Goto, Shigeo Morishima.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Shoto Sasaki, Kazuyoshi Yoshii, Tomoyasu Nakano, Masataka Goto, Shigeo Morishima. *LyricsRadar: A Lyrics Retrieval System Based on Latent Topics of Lyrics*, 15th International Society for Music Information Retrieval Conference, 2014.

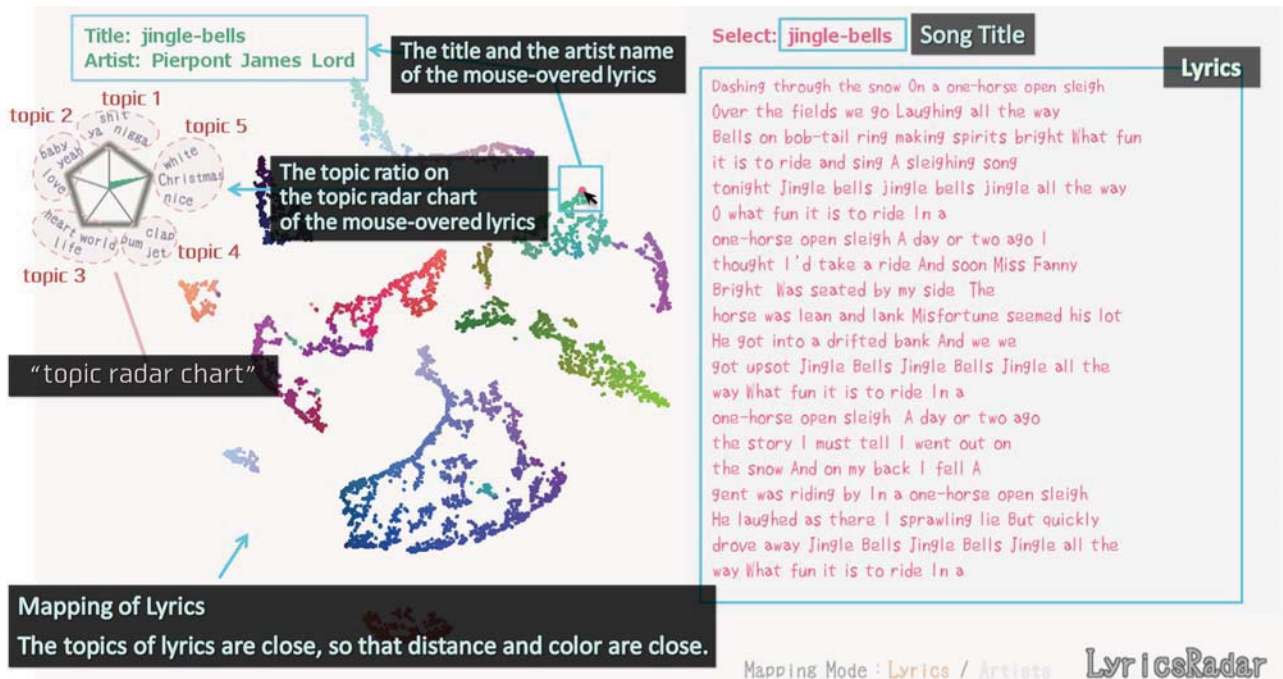


Figure 2. Example display of *LyricsRadar*.

lyrics topics. When it is used with words related to meal, vegetables, and fat, its lyrics topic “food and health” could be estimated by the LDA. On the other hand, when it is used with words like government, law, and elections, “politics” could be estimated. Although the LDA can estimate various lyrics topics, five typical topics common to all lyrics in a given database were chosen. The lyrics of each song are represented by the unique ratios of these five topics, which are displayed as pentagon-shaped chart called as a *topic radar chart*. This chart makes it easy to guess the meaning of lyrics before listening to its song. Furthermore, users can directly change the shape of this chart as a query to retrieve lyrics having a similar shape.

In *LyricsRadar*, all the lyrics are embedded in a two-dimensional space, mapped automatically based on the ratios of the five lyrics topics. The position of lyrics is such that lyrics in close proximity have similar ratios. Users can navigate in this plane by mouse operation and discover some lyrics which are located very close to their favorite lyrics.

## 2. FUNCTIONALITY OF LYRICSRADAR

*LyricsRadar* enables to bring a graphical user interface assisting users to navigate in a two dimensional space intuitively and interactively to come across the target song. This space is generated automatically by analysis of the topics which appear in common with the lyrics of many musical pieces in database using LDA. Also a latent meaning of lyrics is visualized by the topic radar chart based on the combination of topics ratios. Lyrics that are similar to a user’s preference (*target*) can be intuitively discovered by clicking of the topic radar chart or lyrics representing

by dots. So this approach cannot be achieved at all by the conventional method which directly searches for a song by the keywords or phrases appearing in lyrics. Since linguistic expressions of the topic are not necessary, user can find a target song intuitively even when user does not have any knowledge about lyrics.

### 2.1 Visualization based on the topic of lyrics

*LyricsRadar* has the following two visualization functions: (1) the topic radar chart; and (2) a mapping to the two-dimensional plane. Figure 2 shows an example display of our interface. The topic radar chart shown in upper-left corner of Figure 2 is a pentagon-shape chart which expresses the ratio of five topics of lyrics. Each colored dot displayed in two dimensional plane shown in Figure 2 means the relative location of lyrics in a database. We call these colored dot representations of lyrics *lyrics dot*. User can see lyrics, its title and artist name, and the topic ratio by clicking the lyrics dot placed on the 2D space, this supports to discover lyrics interactively. While the lyrics mapping assists user to understand the lyrics topic by the relative location in the map, the topic radar chart helps to get the lyrics image intuitively by the shape of chart. We explain each of these in the following subsections.

#### 2.1.1 Topic radar chart

The values of the lyrics topic are computed and visualized as the topic radar chart which is pentagon style. Each vertex of the pentagon corresponds to a distinct topic, and predominant words of each topic (e.g., “heart”, “world”, and “life” for the topic 3) are also displayed at the five corner of pentagon shown in Figure 2. The predominant words help user to guess the meaning of each topic. The center

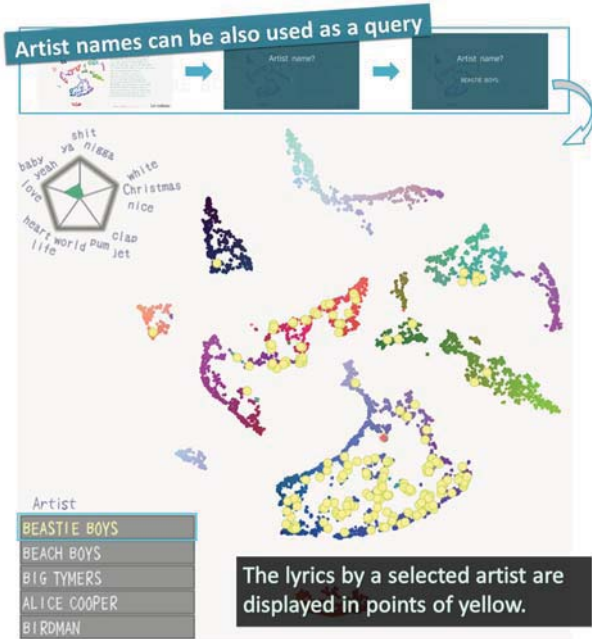


Figure 3. An example display of lyrics by a selected artist.

of the topic radar chart indicates 0 value of a ratio of the lyrics topic in the same manner as the common radar chart. Since the sum of the five components is a constant value, if the ratio of a topic stands out, it will clearly be seen by the user. It is easy to grasp the topic of selected lyrics visually and to make an intuitive comparison between lyrics.

Furthermore, the number of topics in this interface is set to five to strike a balance between the operability of interface and the variety of topics<sup>1</sup>.

### 2.1.2 Plane-mapped lyrics

The lyrics of musical pieces are mapped onto a two-dimensional plane, in which musical pieces with almost the same topic ratio can get closer to each other. Each musical piece is expressed by colored dot whose RGB components are corresponding to 3D compressed axis for five topics' values. This space can be scalable so that the local or global structure of each musical piece can be observed. The distribution of lyrics about a specific topic can be recognized by the color of the lyrics. The dimension compression in mapping and coloring used t-SNE [9]. When a user mouseovers a point in the space, it is colored pink and meta-information about the title, artist, the topic radar chart appears simultaneously.

By repeating mouseover, lyrics and names of its artist and songwriter are updated continuously. Using this approach, other lyrics with the similar topics to the input lyrics can be discovered. The lyrics map can be moved and zoomed by dragging the mouse or using a specific keyboard operation. Furthermore, it is possible to visualize the lyrics map specialized to artist and songwriter, which are

<sup>1</sup> If the number of topics was increased, a more subdivided and exacting semantic content could have been represented; however, the operation for a user will be getting more complicated.

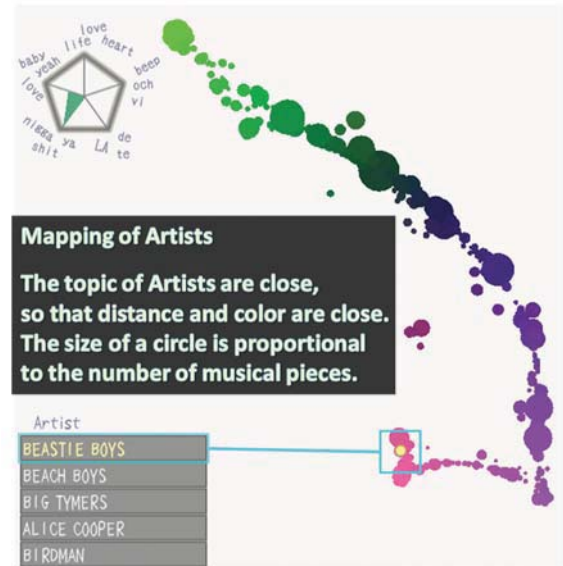


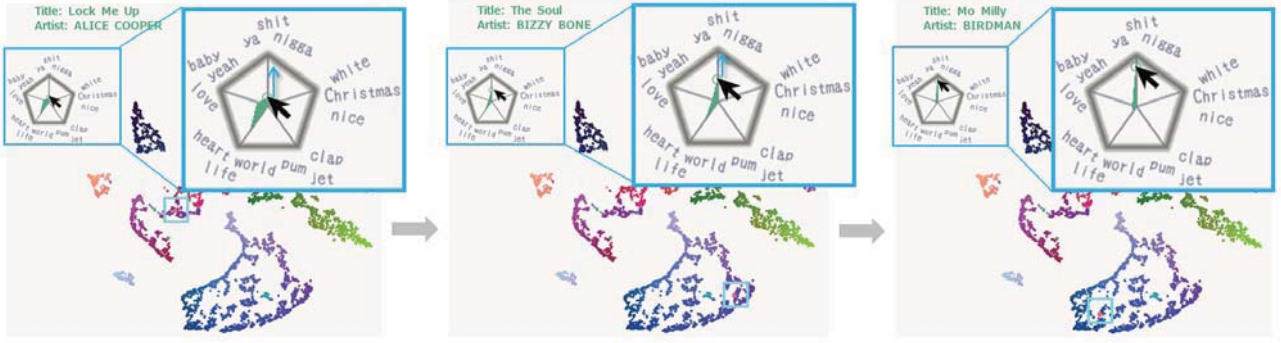
Figure 4. Mapping of 487 English artists.

associated with lyrics as metadata. When an artist name is chosen, as shown in the right side of Figure 3, the point of the artist's lyrics will be getting yellow; similarly, when a songwriter is chosen, the point of the songwriter's lyrics will be changed to orange. While this is somewhat equivalent to lyrics retrieval using the artist or songwriter as a query, it is our innovative point in the sense that a user can intuitively grasp how artists and songwriters are distributed based on the ratio of the given topic. Although music retrieval by artist is very popular in a conventional system, a retrieval by songwriter is not focused well yet. However, in the meaning of lyrics retrieval, it is easier for search by songwriter to discover songs with one's favorite lyrics because a songwriter has his own lyrics vocabulary.

Moreover, we can make a topic analysis depending on a specific artist in our system. Intuitively similar artists are also located and colored closer in a topic chart depending on topic ratios. The artist is colored based on a topic ratio in the same way as that of the lyrics. In Figure 4, the size of a circle is proportional to the number of musical pieces each artist has. In this way, other artists similar to one's favorite artist can be easily discovered.

## 2.2 Lyrics retrieval using topic of lyrics

In *LyricsRadar*, in addition to the ability to traverse and explore a map to find lyrics, we also propose a system to directly enter a topic ratio as an intuitive expression of one's latent feeling. More specifically, we consider the topic radar chart as an input interface and provide a means by which a user can give topic ratios for five elements directly to search for lyrics very close to one's latent image. This interface can satisfy the search query in which a user would like to search for lyrics that contain more of the same topics using the representative words of each topic. Figure 5 shows an example in which one of the five topics is increased by mouse drag, then the balance of five topics ratio



**Figure 5.** An example of the direct manipulation of the topic ratio on the topic radar chart. Each topic ratio can be increased by dragging the mouse.

has changed because the sum of five components is equal to 1.0. A user can repeat these processes by updating topics ratios or navigating the point in a space interactively until finding interesting lyrics. As with the above subsections, we have substantiated our claims for a more intuitive and exploratory lyrics retrieval system.

### 3. IMPLEMENTATION OF LYRICSRADAR

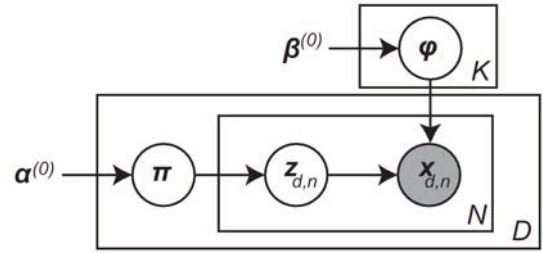
*LyricsRadar* used LDA [4] for the topic analysis of lyrics. LDA is a typical topic modeling method by machine learning. Since LDA assigns each word which constitutes lyrics to a different topic independently, the lyrics include a variety of topics according to the variation of words in the lyrics. In our system,  $K$  typical topics which constitute many lyrics in database are estimated and a ratio to each topic is calculated for lyrics with unsupervised learning. As a result, appearance probability of each word in every topic can be calculated. The typical representative word to each topic can be decided at the same time.

#### 3.1 LDA for lyrics

The observed data that we consider for LDA are  $D$  independent lyrics  $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_D\}$ . The lyrics  $\mathbf{X}_d$  consist of  $N_d$  word series  $\mathbf{X}_d = \{\mathbf{x}_{d,1}, \dots, \mathbf{x}_{d,N_d}\}$ . The size of all vocabulary that appear in the lyrics is  $V$ ,  $\mathbf{x}_{d,n}$  is a  $V$ -dimensional “1-of- $K$ ” vector (a vector with one element containing 1 and all other elements containing 0). The latent variable (i.e., the topics series) of the observed lyrics  $\mathbf{X}_d$  is  $\mathbf{Z}_d = \{\mathbf{z}_{d,1}, \dots, \mathbf{z}_{d,N_d}\}$ . The number of topics is  $K$ , so  $\mathbf{z}_{d,n}$  indicates a  $K$ -dimensional 1-of- $K$  vector. Hereafter, all latent variables of lyrics  $D$  are indicated  $\mathbf{Z} = \{\mathbf{Z}_1, \dots, \mathbf{Z}_D\}$ . Figure 6 shows a graphical representation of the LDA model used in this paper. The full joint distribution is given by

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\phi}) = p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\phi})p(\mathbf{Z}|\boldsymbol{\pi})p(\boldsymbol{\pi})p(\boldsymbol{\phi}) \quad (1)$$

where  $\boldsymbol{\pi}$  indicates the mixing weights of the multiple topics of lyrics ( $D$  of the  $K$ -dimensional vector) and  $\boldsymbol{\phi}$  indicates the unigram probability of each topic ( $K$  of the  $V$ -dimensional vector). The first two terms are likelihood



**Figure 6.** Graphical representation of the latent Dirichlet allocation (LDA).

functions, whereas the other two terms are prior distributions. The likelihood functions themselves are defined as

$$p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\phi}) = \prod_{d=1}^D \prod_{n=1}^{N_d} \prod_{v=1}^V \left( \prod_{k=1}^K \phi_{k,v}^{z_{d,n,k}} \right)^{x_{d,n,v}} \quad (2)$$

$$p(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{d=1}^D \prod_{n=1}^{N_d} \prod_{k=1}^K \pi_{d,k}^{z_{d,n,k}} \quad (3)$$

We then introduce conjugate priors as

$$p(\boldsymbol{\pi}) = \prod_{d=1}^D \text{Dir}(\boldsymbol{\pi}_d | \boldsymbol{\alpha}^{(0)}) = \prod_{d=1}^D C(\boldsymbol{\alpha}^{(0)}) \prod_{k=1}^K \pi_{d,k}^{\alpha_{d,k}^{(0)} - 1} \quad (4)$$

$$p(\boldsymbol{\phi}) = \prod_{k=1}^K \text{Dir}(\boldsymbol{\phi}_k | \boldsymbol{\beta}^{(0)}) = \prod_{k=1}^K C(\boldsymbol{\beta}^{(0)}) \prod_{v=1}^V \phi_{k,v}^{\beta_v^{(0)} - 1} \quad (5)$$

where  $p(\boldsymbol{\pi})$  and  $p(\boldsymbol{\phi})$  are products of Dirichlet distributions,  $\boldsymbol{\alpha}^{(0)}$  and  $\boldsymbol{\beta}^{(0)}$  are hyperparameters, and  $C(\boldsymbol{\alpha}^{(0)})$  and  $C(\boldsymbol{\beta}^{(0)})$  are normalization factors calculated as follows:

$$C(\mathbf{x}) = \frac{\Gamma(\hat{x})}{\Gamma(x_1) \cdots \Gamma(x_I)}, \quad \hat{x} = \sum_{i=1}^I x_i \quad (6)$$

Also note that  $\boldsymbol{\pi}$  is the topic mixture ratio of lyrics used as the topic radar chart by normalization. The appearance probability  $\boldsymbol{\phi}$  of the vocabulary in each topic was used to evaluate the high-representative word that is strongly correlated with each topic of the topic radar chart.

### 3.2 Training of LDA

The lyrics database contains 6902 Japanese popular songs (J-POP) and 5351 English popular songs. Each of these songs includes more than 100 words. J-POP songs are selected from our own database and English songs are from *Music Lyrics Database v.1.2.7*<sup>2</sup>. J-POP database has 1847 artists and 2285 songwriters and English database has 398 artists. For the topic analysis per artist, 2484 J-POP artists and 487 English artists whose all songs include at least 100 words are selected. 26229 words in J-POP and 35634 words in English which appear more than ten times in all lyrics is used for the value  $V$  which is the size of vocabulary in lyrics. In J-POP lyrics, MeCab [17] was used for the morphological analysis of J-POP lyrics. The noun, verb, and adjective components were extracted and then the original and the inflected form were counted as one word. In English lyrics, we use stopwords using *Full-Text Stopwords in MySQL*<sup>3</sup> to remove commonly-used words. However, words which appeared often in many lyrics were inconvenient to analyze topics. To lower the importance of such words in the topic analysis, they were weighted by inverse document frequency (idf).

In the training of LDA, the number of topics ( $K$ ) is set to 5. All initial values of hyperparameters  $\alpha^{(0)}$  and  $\beta^{(0)}$  were set to 1.

## 4. EVALUATION EXPERIMENTS

To verify the validity of the topic analysis results (as related to the topic radar chart and mapping of lyrics) in *LyricsRadar*, we conducted a subjective evaluation experiment. There were 17 subjects (all Japanese speakers) with ages from 21 to 32. We used the results of LDA for the lyrics of the 6902 J-POP songs described in Section 3.2.

### 4.1 Evaluation of topic analysis

Our evaluation here attempted to verify that the topic ratio determined by the topic analysis of LDA could appropriately represent latent meaning of lyrics. Furthermore, when the lyrics of a song are selected, relative location to other lyrics of the same artist or songwriter in the space is investigated.

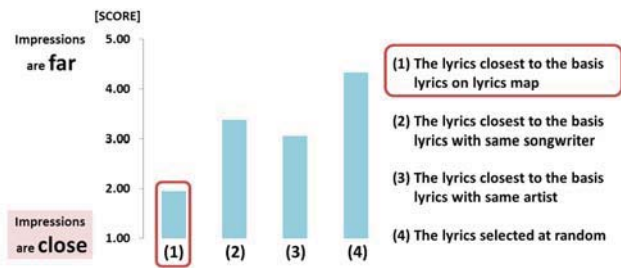
#### 4.1.1 Experimental method

In our experiment, the lyrics of a song are selected at random in the space as *basis* lyrics and also *target* lyrics of four songs are selected to be compared according to the following conditions.

- (1) The lyrics closest to the *basis* lyrics on lyrics map
- (2) The lyrics closest to the *basis* lyrics with same songwriter
- (3) The lyrics closest to the *basis* lyrics with same artist

<sup>2</sup>“Music Lyrics Database v.1.2.7,” <http://www.odditysoftware.com/page-datasales1.htm>.

<sup>3</sup>“Full-Text Stopwords in MySQL,” <http://dev.mysql.com/doc/refman/5.5/en/fulltext-stopwords.html>.



**Figure 7.** Results of our evaluation experiment to evaluate topic analysis; the score of (1) was the closest to 1.0, showing our approach to be effective.

#### (4) The lyrics selected at random

Each subject evaluated the similarity of the impression received from the two lyrics using a five-step scale (1: closest, 2: somehow close, 3: neutral, 4: somehow far, and 5: most far), comparing the *basis* lyrics and one of the *target* lyrics after seeing the *basis* lyrics. Presentation order to subjects was random. Furthermore, each subject described the reason of evaluation score.

#### 4.1.2 Experimental results

The average score of the five-step evaluation results for the four *target* lyrics by all subjects is shown in the Figure 7. As expected, lyrics closest to the *basis* lyrics on the lyrics map were evaluated as the closest in terms of the impression of the *basis* lyrics, because the score of (1) was closest to 1.0. Results of *target* lyrics (2) and (3) were both close to 3.0. The lyrics closest to the *basis* lyrics of the same songwriter or artist as the selected lyrics were mostly judged as “3: neutral.” Finally, the lyrics selected at random (4) were appropriately judged to be far.

As the subjects’ comments about the reason of decision, we obtained such responses as a sense of the season, positive-negative, love, relationship, color, light-dark, subjective-objective, and tension. Responses differed greatly from one subject to the next. For example, some felt the impression only by the similarity of a sense of the season of lyrics. Trial usage of *LyricsRadar* has shown that it is a useful tool for users.

### 4.2 Evaluation of the number of topics

The perplexity used for the quality assessment of a language model was computed for each number of topics. The more the model is complicated, the higher the perplexity becomes. Therefore, we can estimate that the performance of language model is good when the value of perplexity is low. We calculated perplexity as

$$\text{perplexity}(\mathbf{X}) = \exp\left(-\frac{\sum_{d=1}^D \log p(X_d)}{\sum_{d=1}^D N_d}\right) \quad (7)$$

In case the number of topics ( $K$ ) is five, the perplexity is 1150 which is even high.

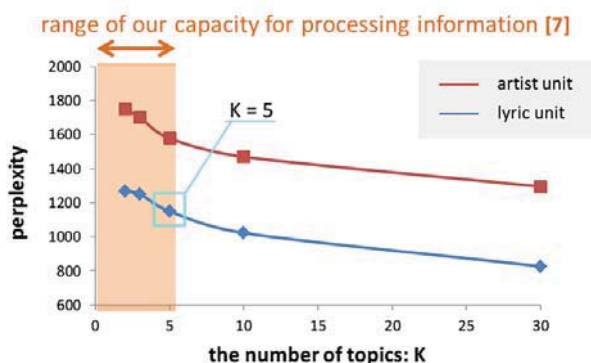


Figure 8. Perplexity for the number of topics.

On the other hand, because Miller showed that the number of objects human can hold in his working memory is  $7 \pm 2$  [7], the number of topics should be 1 to 5 in order to obtain information naturally. So we decided to show five topics in the topic radar chart.

Figure 8 shows calculation results of perplexity for each topic number. Blue points represent perplexity for LDA applied to lyrics and red points represent perplexity for LDA applied to each artist. Orange bar indicates the range of human capacity for processing information. Since there exists a tradeoff between the number of topics and operability, we found that five is appropriate number of topics.

## 5. CONCLUSIONS

In this paper, we propose *LyricsRadar*, an interface to assist a user to come across favorite lyrics interactively. Conventionally lyrics were retrieved by titles, artist names, or keywords. Our main contribution is to visualize lyrics in the latent meaning level based on a topic model by LDA. By seeing the pentagon-style shape of Topic Radar Chart, a user can intuitively recognize the meaning of given lyrics. The user can also directly manipulate this shape to discover *target* lyrics even when the user does not know any keyword or any query. Also the topic ratio of focused lyrics can be mapped to a point in the two dimensional space which visualizes the relative location to all the lyrics in our lyrics database and enables the user to navigate similar lyrics by controlling the point directly.

For future work, user adaptation is inevitable task because every user has an individual preference, as well as improvements to topic analysis by using hierarchical topic analysis [12]. Furthermore, to realize the retrieval interface corresponding to a minor topic of lyrics, a future challenge is to consider the visualization method that can reflect more numbers of topics by keeping an easy-to-use interactivity.

**Acknowledgment:** This research was supported in part by OngaCREST, CREST, JST.

## 6. REFERENCES

- [1] B. Logan *et al.*: “Semantic Analysis of Song Lyrics,” *Proceedings of IEEE ICME 2004* Vol.2, pp. 827–830, 2004.
- [2] C. Laurier *et al.*: “Multimodal Music Mood Classification Using Audio and Lyrics,” *Proceedings of ICMLA 2008*, pp. 688–693, 2008.
- [3] C. McKay *et al.*: “Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features,” *Proceedings of ISMIR 2008*, pp. 213–218, 2008.
- [4] D. M. Blei *et al.*: “Latent Dirichlet Allocation,” *Journal of Machine Learning Research* Vol.3, pp. 993–1022, 2003.
- [5] E. Brochu and N. de Freitas: ““Name That Song!”: A Probabilistic Approach to Querying on Music and Text,” *Proceedings of NIPS 2003*, pp. 1505–1512, 2003.
- [6] F. Kleedorfer *et al.*: “Oh Oh Oh Whoah! Towards Automatic Topic Detection In Song Lyrics,” *Proceedings of ISMIR 2008*, pp. 287–292, 2008.
- [7] G. A. Miller: “The magical number seven, plus or minus two: Some limits on our capacity for processing information,” *Journal of the Psychological Review* Vol.63(2), pp. 81–97, 1956.
- [8] H. Fujihara *et al.*: “LyricSynchronizer: Automatic Synchronization System between Musical Audio Signals and Lyrics,” *Journal of IEEE Selected Topics in Signal Processing*, Vol.5, No.6, pp. 1252–1261, 2011.
- [9] L. Maaten and G. E. Hinton: “Visualizing High-Dimensional Data Using t-SNE,” *Journal of Machine Learning Research*, Vol.9, pp. 2579–2605, 2008.
- [10] M. Müller *et al.*: “Lyrics-based Audio Retrieval and Multimodal Navigation in Music Collections,” *Proceedings of ECDL 2007*, pp. 112–123, 2007.
- [11] M. V. Zaanen and P. Kanthers: “Automatic Mood Classification Using TF\*IDF Based on Lyrics,” *Proceedings of ISMIR 2010*, pp. 75–80, 2010.
- [12] R. Adams *et al.*: “Tree-Structured Stick Breaking Processes for Hierarchical Data,” *Proceedings of NIPS 2010*, pp. 19–27, 2010.
- [13] R. Macrae and S. Dixon: “Ranking Lyrics for Online Search,” *Proceedings of ISMIR 2012*, pp. 361–366, 2012.
- [14] R. Takahashi *et al.*: “Building and combining document and music spaces for music query-by-webpage system,” *Proceedings of Interspeech 2008*, pp. 2020–2023, 2008.
- [15] R. Neumayer and A. Rauber: “Multi-modal Music Information Retrieval: Visualisation and Evaluation of Clusterings by Both Audio and Lyrics,” *Proceedings of RAO 2007*, pp. 70–89, 2007.
- [16] S. Funasawa *et al.*: “Automated Music Slideshow Generation Using Web Images Based on Lyrics,” *Proceedings of ISMIR 2010*, pp. 63–68, 2010.
- [17] T. Kudo: “MeCab: Yet Another Part-of-Speech and Morphological Analyzer,” <http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html>.
- [18] T. Nakano and M. Goto: “VocaRefiner: An Interactive Singing Recording System with Integration of Multiple Singing Recordings,” *Proceedings of SMC 2013*, pp. 115–122, 2013.
- [19] Y. Hu *et al.*: “Lyric-based Song Emotion Detection with Affective Lexicon and Fuzzy Clustering Method,” *Proceedings of ISMIR 2009*, pp. 122–128, 2009.

# JAMS: A JSON ANNOTATED MUSIC SPECIFICATION FOR REPRODUCIBLE MIR RESEARCH

Eric J. Humphrey<sup>1,\*</sup>, Justin Salamon<sup>1,2</sup>, Oriol Nieto<sup>1</sup>, Jon Forsyth<sup>1</sup>,  
Rachel M. Bittner<sup>1</sup>, and Juan P. Bello<sup>1</sup>

<sup>1</sup>Music and Audio Research Lab, New York University, New York

<sup>2</sup>Center for Urban Science and Progress, New York University, New York

## ABSTRACT

The continued growth of MIR is motivating more complex annotation data, consisting of richer information, multiple annotations for a given task, and multiple tasks for a given music signal. In this work, we propose JAMS, a JSON-based music annotation format capable of addressing the evolving research requirements of the community, based on the three core principles of simplicity, structure and sustainability. It is designed to support existing data while encouraging the transition to more consistent, comprehensive, well-documented annotations that are poised to be at the crux of future MIR research. Finally, we provide a formal schema, software tools, and popular datasets in the proposed format to lower barriers to entry, and discuss how now is a crucial time to make a concerted effort toward sustainable annotation standards.

## 1. INTRODUCTION

Music annotations—the collection of observations made by one or more agents about an acoustic music signal—are an integral component of content-based Music Information Retrieval (MIR) methodology, and are necessary for designing, evaluating, and comparing computational systems. For clarity, we define the scope of an annotation as corresponding to time scales at or below the level of a complete song, such as semantic descriptors (tags) or time-aligned chords labels. Traditionally, the community has relied on plain text and custom conventions to serialize this data to a file for the purposes of storage and dissemination, collectively referred to as “lab-files”. Despite a lack of formal standards, lab-files have been, and continue to be, the preferred file format for a variety of MIR tasks, such as beat or onset estimation, chord estimation, or segmentation.

\*Please direct correspondence to [ejhumphrey@nyu.edu](mailto:ejhumphrey@nyu.edu)



© Eric J. Humphrey, Justin Salamon, Oriol Nieto, Jon Forsyth, Rachel M. Bittner, Juan P. Bello.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Eric J. Humphrey, Justin Salamon, Oriol Nieto, Jon Forsyth, Rachel M. Bittner, Juan P. Bello. “JAMS: A JSON Annotated Music Specification for Reproducible MIR Research”, 15th International Society for Music Information Retrieval Conference, 2014.

Meanwhile, the interests and requirements of the community are continually evolving, thus testing the practical limitations of lab-files. By our count, there are three unfolding research trends that are demanding more of a given annotation format:

- **Comprehensive annotation data:** Rich annotations, like the Billboard dataset [2], require new, content-specific conventions, increasing the complexity of the software necessary to decode it and the burden on the researcher to use it; such annotations can be so complex, in fact, it becomes necessary to document how to understand and parse the format [5].
- **Multiple annotations for a given task:** The experience of music can be highly subjective, at which point the notion of “ground truth” becomes tenuous. Recent work in automatic chord estimation [8] shows that multiple reference annotations should be embraced, as they can provide important insight into system evaluation, as well as into the task itself.
- **Multiple concepts for a given signal:** Although systems are classically developed to accomplish a single task, there is ongoing discussion toward integrating information across various musical concepts [12]. This has already yielded measurable benefits for the joint estimation of chords and downbeats [9] or chords and segments [6], where leveraging multiple information sources for the same input signal can lead to improved performance.

It has long been acknowledged that lab-files cannot be used to these ends, and various formats and technologies have been previously proposed to alleviate these issues, such as RDF [3], HDF5 [1], or XML [7]. However, none of these formats have been widely embraced by the community. We contend that the weak adoption of any alternative format is due to the combination of several factors. For example, new tools can be difficult, if not impossible, to integrate into a research workflow because of compatibility issues with a preferred development platform or programming environment. Additionally, it is a common criticism that the syntax or data model of these alternative formats is non-obvious, verbose, or otherwise confusing. This is especially problematic when researchers must handle for-

mat conversions. Taken together, the apparent benefits to conversion are outweighed by the tangible costs.

In this paper, we propose a JSON Annotated Music Specification (JAMS) to meet the changing needs of the MIR community, based on three core design tenets: simplicity, structure, and sustainability. This is achieved by combining the advantages of lab-files with lessons learned from previously proposed formats. The resulting JAMS files are human-readable, easy to drop into existing workflows, and provide solutions to the research trends outlined previously. We further address classical barriers to adoption by providing tools for easy use with Python and MATLAB, and by offering an array of popular datasets as JAMS files online. The remainder of this paper is organized as follows: Section 2 identifies three valuable components of an annotation format by considering prior technologies; Section 3 formally introduces JAMS, detailing how it meets these design criteria and describing the proposed specification by example; Section 4 addresses practical issues and concerns in an informal FAQ-style, touching on usage tools, provided datasets, and some practical shortcomings; and lastly, we close with a discussion of next steps and perspectives for the future in Section 5.

## 2. CORE DESIGN PRINCIPLES

In order to craft an annotation format that might serve the community into the foreseeable future, it is worthwhile to consolidate the lessons learned from both the relative success of lab-files and the challenges faced by alternative formats into a set of principles that might guide our design. With this in mind, we offer that usability, and thus the likelihood of adoption, is a function of three criteria:

### 2.1 Simplicity

The value of simplicity is demonstrated by lab-files in two specific ways. First, the contents are represented in a format that is intuitive, such that the document model clearly matches the data structure and is human-readable, i.e. uses a lightweight syntax. This is a particular criticism of RDF and XML, which can be verbose compared to plain text. Second, lab-files are conceptually easy to incorporate into research workflows. The choice of an alternative file format can be a significant hurdle if it is not widely supported, as is the case with RDF, or the data model of the document does not match the data model of the programming language, as with XML.

### 2.2 Structure

It is important to recognize that lab-files developed as a way to serialize tabular data (i.e. arrays) in a language-independent manner. Though lab-files excel at this particular use case, they lack the structure required to encode complex data such as hierarchies or mix different data types, such as scalars, strings, multidimensional arrays, etc. This is a known limitation, and the community has devised a variety of ad hoc strategies to cope with it: folder trees and naming conventions, such as “{X}/{Y}/{Z}.lab”,

where X, Y, and Z correspond to “artist”, “album”, and “title”, respectively<sup>1</sup>; parsing rules, such as “lines beginning with ‘#’ are to be ignored as comments”; auxiliary websites or articles, decoupled from the annotations themselves, to provide critical information such as syntax, conventions, or methodology. Alternative representations are able to manage more complex data via standardized markup and named entities, such as fields in the case of RDF or JSON, or IDs, attributes and tags for XML.

### 2.3 Sustainability

Recently in MIR, a more concerted effort has been made toward sustainable research methods, which we see positively impacting annotations in two ways. First, there is considerable value to encoding methodology and metadata directly in an annotation, as doing so makes it easier to both support and maintain the annotation while also enabling direct analyses of this additional information. Additionally, it is unnecessary for the MIR community to develop every tool and utility ourselves; we should instead leverage well-supported technologies from larger communities when possible.

## 3. INTRODUCING JAMS

So far, we have identified several goals for a music annotation format: a data structure that matches the document model; a lightweight markup syntax; support for multiple annotations, multiple tasks, and rich metadata; easy workflow integration; cross-language compliance; and the use of pre-existing technologies for stability. To find our answer, we need only to look to the web development community, who have already identified a technology that meets these requirements. JavaScript Object Notation (JSON)<sup>2</sup> has emerged as *the* serialization format of the Internet, now finding native support in almost every modern programming language. Notably, it was designed to be maximally efficient and human readable, and is capable of representing complex data structures with little overhead.

JSON is, however, only a syntax, and it is necessary to define formal standards outlining how it should be used for a given purpose. To this end, we define a specification on top of JSON (JAMS), tailored to the needs of MIR researchers.

### 3.1 A Walk-through Example

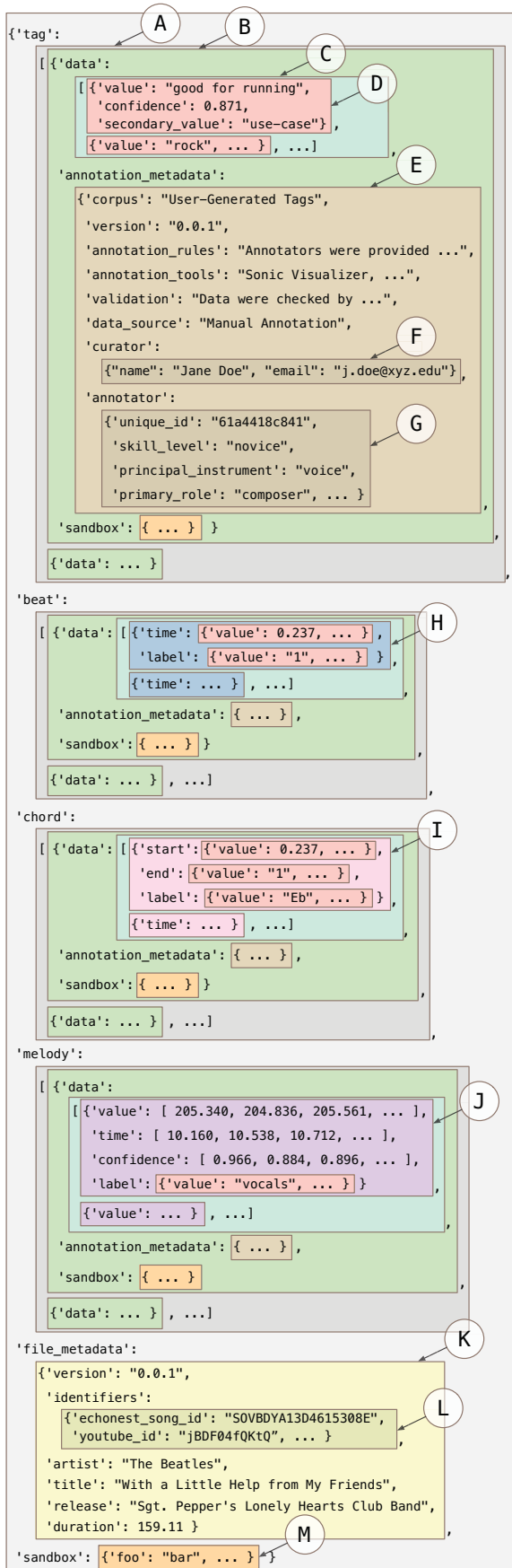
Perhaps the clearest way to introduce the JAMS specification is by example. Figure 1 provides the contents of a hypothetical JAMS file, consisting of nearly valid<sup>3</sup> JSON syntax and color-coded by concept. JSON syntax will be familiar to those with a background in C-style languages, as it uses square brackets (“[ ]”) to denote arrays (alternatively, lists or vectors), and curly brackets (“{ }”) to denote

<sup>1</sup> <http://www.isophonics.net/content/reference-annotations>

<sup>2</sup> <http://www.json.org/>

<sup>3</sup> The sole exception is the use of ellipses (“...”) as continuation characters, indicating that more information could be included.





**Figure 1.** Diagram illustrating the structure of the JAMS specification.

objects (alternatively, dictionaries, structs, or hash maps). Defining some further conventions for the purpose of illustration, we use single quotes to indicate field names, italics when referring to concepts, and consistent colors for the same data structures. Using this diagram, we will now step through the hierarchy, referring back to relevant components as concepts are introduced.

### 3.1.1 The JAMS Object

A JAMS file consists of one top-level object, indicated by the outermost bounding box. This is the primary container for all information corresponding to a music signal, consisting of several task-array pairs, an object for `file_metadata`, and an object for `sandbox`. A task-array is a list of *annotations* corresponding to a given task name, and may contain zero, one, or many annotations for that task. The format of each array is specific to the kind of annotations it will contain; we will address this in more detail in Section 3.1.2.

The `file_metadata` object (K) is a dictionary containing basic information about the music signal, or file, that was annotated. In addition to the fields given in the diagram, we also include an unconstrained `identifiers` object (L), for storing unique identifiers in various namespaces, such as the EchoNest or YouTube. Note that we purposely do not store information about the recording's audio encoding, as a JAMS file is format-agnostic. In other words, we assume that any sample rate or perceptual codec conversions will have no effect on the annotation, within a practical tolerance.

Lastly, the JAMS object also contains a `sandbox`, an unconstrained object to be used as needed. In this way, the specification carves out such space for any unforeseen or otherwise relevant data; however, as the name implies, no guarantee is made as to the existence or consistency of this information. We do this in the hope that the specification will not be unnecessarily restrictive, and that commonly “sandboxed” information might become part of the specification in the future.

### 3.1.2 Annotations

An *annotation* (B) consists of all the information that is provided by a single annotator about a single task for a single music signal. Independent of the task, an annotation comprises three sub-components: an array of objects for data (C), an `annotation_metadata` object (E), and an annotation-level `sandbox`. For clarity, a task-array (A) may contain multiple annotations (B).

Importantly, a `data` array contains the primary annotation information, such as its chord sequence, beat locations, etc., and is the information that would normally be stored in a lab-file. Though all `data` containers are functionally equivalent, each may consist of only one object type, specific to the given task. Considering the different types of musical attributes annotated for MIR research, we divide them into four fundamental categories:

1. Attributes that exist as a single *observation* for the entire music signal, e.g. tags.

2. Attributes that consist of sparse *events* occurring at specific times, e.g. beats or onsets.
3. Attributes that span a certain time *range*, such as chords or sections.
4. Attributes that comprise a dense *time series*, such as discrete-time fundamental frequency values for melody extraction.

These four types form the most atomic data structures, and will be revisited in greater detail in Section 3.1.3. The important takeaway here, however, is that data arrays are not allowed to mix fundamental types.

Following [10], an `annotation_metadata` object is defined to encode information about what has been annotated, who created the annotations, with what tools, etc. Specifically, `corpus` provides the name of the dataset to which the annotation belongs; `version` tracks the version of this particular annotation; `annotation_rules` describes the protocol followed during the annotation process; `annotation_tools` describes the tools used to create the annotation; `validation` specifies to what extent the annotation was verified and is reliable; `data_source` details how the annotation was obtained, such as manual annotations, online aggregation, game with a purpose, etc.; `curator` (F) is itself an object with two subfields, `name` and `email`, for the contact person responsible for the annotation; and `annotator` (G) is another unconstrained object, which is intended to capture information about the source of the annotation. While complete metadata are strongly encouraged in practice, currently only `version` and `curator` are mandatory in the specification.

### 3.1.3 Datatypes

Having progressed through the JAMS hierarchy, we now introduce the four atomic data structures, out of which an annotation can be constructed: *observation*, *event*, *range* and *time series*. For clarity, the `data` array (A) of a `tag` annotation is a list of *observation* objects; the `data` array of a `beat` annotation is a list of *event* objects; the `data` array of a `chord` annotation is a list of *range* objects; and the `data` array of a `melody` annotation is a list of *time series* objects. The current space of supported tasks is provided in Table 1.

Of the four types, an *observation* (D) is the most atomic, and used to construct the other three. It is an object that has one primary field, `value`, and two optional fields, `confidence` and `secondary_value`. The `value` and `secondary_value` fields may take any simple primitive, such as a string, numerical value, or boolean, whereas the `confidence` field stores a numerical confidence estimate for the observation. A secondary value field is provided for flexibility in the event that an observation requires an additional level of specificity, as is the case in hierarchical segmentation [11].

An *event* (H) is useful for representing musical attributes that occur at sparse moments in time, such as beats or onsets. It is a container that holds two *observations*, `time` and `label`. Referring to the first beat annotation in the

<i>observation</i>	<i>event</i>	<i>range</i>	<i>time series</i>
tag	beat	chord	melody
genre	onset	segment	pitch
mood		key	pattern
		note	
		source	

**Table 1.** Currently supported tasks and types in JAMS.

diagram, the value of `time` is a scalar quantity (0.237), whereas the value of `label` is a string ('1'), indicating metrical position.

A *range* (I) is useful for representing musical attributes that span an interval of time, such as chords or song segments (e.g. intro, verse, chorus). It is an object that consists of three *observations*: `start`, `end`, and `label`.

The *time series* (J) atomic type is useful for representing musical attributes that are continuous in nature, such as fundamental frequency over time. It is an object composed of four elements: `value`, `time`, `confidence` and `label`. The first three fields are arrays of numerical values, while `label` is an *observation*.

## 3.2 The JAMS Schema

The description in the previous sections provides a high-level understanding of the proposed specification, but the only way to describe it without ambiguity is through formal representation. To accomplish this, we provide a JSON schema<sup>4</sup>, a specification itself written in JSON that uses a set of reserved keywords to define valid data structures. In addition to the expected contents of the JSON file, the schema can specify which fields are required, which are optional, and the type of each field (e.g. numeric, string, boolean, array or object). A JSON schema is concise, precise, and human readable.

Having defined a proper JSON schema, an added benefit of JAMS is that a validator can verify whether or not a piece of JSON complies with a given schema. In this way, researchers working with JAMS files can easily and confidently test the integrity of a dataset. There are a number of JSON schema validator implementations freely available online in a variety of languages, including Python, Java, C, JavaScript, Perl, and more. The JAMS schema is included in the public software repository (cf. Section 4), which also provides a static URL to facilitate directly accessing the schema from the web within a workflow.

## 4. JAMS IN PRACTICE

While we contend that the use and continued development of JAMS holds great potential for the many reasons outlined previously, we acknowledge that specifications and standards are myriad, and it can be difficult to ascertain the benefits or shortcomings of one's options. In the interest of encouraging adoption and the larger discussion of

<sup>4</sup> <http://json-schema.org/>

standards in the field, we would like to address practical concerns directly.

#### 4.1 How is this any different than *X*?

The biggest advantage of JAMS is found in its capacity to consistently represent rich information with no additional effort from the parser and minimal markup overhead. Compared to XML or RDF, JSON parsers are extremely fast, which has contributed in no small part to its widespread adoption. These efficiency gains are coupled with the fact that JAMS makes it easier to manage large data collections by keeping all annotations for a given song in the same place.

#### 4.2 What kinds of things can I do with JAMS that I can't already do with *Y*?

JAMS can enable much richer evaluation by including multiple, possibly conflicting, reference annotations and directly embedding information about an annotation's origin. A perfect example of this is found in the Rock Corpus Dataset [4], consisting of annotations by two expert musicians: one, a guitarist, and the other, a pianist. Sources of disagreement in the transcriptions often stem from differences of opinion resulting from familiarity with their principal instrument, where the voicing of a chord that makes sense on piano is impossible for a guitarist, and vice versa. Similarly, it is also easier to develop versatile MIR systems that combine information across tasks, as that information is naturally kept together.

Another notable benefit of JAMS is that it can serve as a data representation for algorithm outputs for a variety of tasks. For example, JAMS could simplify MIREX submissions by keeping all machine predictions for a given team together as a single submission, streamlining evaluations, where the annotation sandbox and annotator metadata can be used to keep track of algorithm parameterizations. This enables the comparison of many references against many algorithmic outputs, potentially leading to a deeper insight into a system's performance.

#### 4.3 So how would this interface with my workflow?

Thanks to the widespread adoption of JSON, the vast majority of languages already offer native JSON support. In most cases, this means it is possible to go from a JSON file to a programmatic data structure in your language of choice in a single line of code using tools you didn't have to write. To make this experience even simpler, we additionally provide two software libraries, for Python and MATLAB. In both instances, a lightweight software wrapper is provided to enable a seamless experience with JAMS, allowing IDEs and interpreters to make use of autocomplete and syntax checking. Notably, this allows us to provide convenience functionality for creating, populating, and saving JAMS objects, for which examples and sample code are provided with the software library<sup>5</sup>.

<sup>5</sup><https://github.com/urinieto/jams>

#### 4.4 What datasets are already JAMS-compliant?

To further lower the barrier to entry and simplify the process of integrating JAMS into a pre-existing workflow, we have collected some of the more popular datasets in the community and converted them to the JAMS format, linked via the public repository. The following is a partial list of converted datasets: Isophonics (beat, chord, key, segment); Billboard (chord); SALAMI (segment, pattern); RockCorpus (chord, key); tmc323 (chords); Cal500 (tag); Cal10k (tag); ADC04 (melody); and MIREX05 (melody).

#### 4.5 Okay, but *my* data is in a different format – now what?

We realize that it is impractical to convert every dataset to JAMS, and provide a collection of Python scripts that can be used to convert lab-files to JAMS. In lieu of direct interfaces, alternative formats can first be converted to lab-files and translated to JAMS thusly.

#### 4.6 My MIR task doesn't really fit with JAMS.

That's not a question, but it is a valid point and one worth discussing. While this first iteration of JAMS was designed to be maximally useful across a variety of tasks, there are two broad reasons why JAMS might not work for a given dataset or task. One, a JAMS annotation only considers information at the temporal granularity of a single audio file and smaller, independently of all other audio files in the world. Therefore, extrinsic relationships, such as cover songs or music similarity, won't directly map to the specification because the concept is out of scope.

The other, more interesting, scenario is that a given use case requires functionality we didn't plan for and, as a result, JAMS doesn't yet support. To be perfectly clear, the proposed specification is exactly that –a proposal– and one under active development. Born out of an internal need, this initial release focuses on tasks with which the authors are familiar, and we realize the difficulty in solving a global problem in a single iteration. As will be discussed in greater detail in the final section, the next phase on our roadmap is to solicit feedback and input from the community at large to assess and improve upon the specification. If you run into an issue, we would love to hear about your experience.

#### 4.7 This sounds promising, but nothing's perfect. There must be shortcomings.

Indeed, there are two practical limits that should be mentioned. Firstly, JAMS is not designed for features or signal level statistics. That said, JSON is still a fantastic, cross-language syntax for serializing data, and may further serve a given workflow. As for practical concerns, it is a known issue that parsing large JSON objects can be slow in MATLAB. We've worked to make this no worse than reading current lab-files, but speed and efficiency are not touted benefits of MATLAB. This may become a bigger issue as JAMS files become more complete over time, but we are

actively exploring various engineering solutions to address this concern.

## 5. DISCUSSION AND FUTURE PERSPECTIVES

In this paper, we have proposed a JSON format for music annotations to address the evolving needs of the MIR community by keeping multiple annotations for multiple tasks alongside rich metadata in the same file. We do so in the hopes that the community can begin to easily leverage this depth of information, and take advantage of ubiquitous serialization technology (JSON) in a consistent manner across MIR. The format is designed to be intuitive and easy to integrate into existing workflows, and we provide software libraries and pre-converted datasets to lower barriers to entry.

Beyond practical considerations, JAMS has potential to transform the way researchers approach and use music annotations. One of the more pressing issues facing the community at present is that of dataset curation and access. It is our hope that by associating multiple annotations for multiple tasks to an audio signal with retraceable metadata, such as identifiers or URLs, it might be easier to create freely available datasets with better coverage across tasks. Annotation tools could serve music content found freely on the Internet and upload this information to a common repository, ideally becoming something like a Freebase<sup>6</sup> for MIR. Furthermore, JAMS provides a mechanism to handle multiple concurrent perspectives, rather than forcing the notion of an objective truth.

Finally, we recognize that any specification proposal is incomplete without an honest discussion of feasibility and adoption. The fact remains that JAMS arose from the combination of needs within our group and an observation of wider applicability. We have endeavored to make the specification maximally useful with minimal overhead, but appreciate that community standards require iteration and feedback. This current version is not intended to be the definitive answer, but rather a starting point from which the community can work toward a solution as a collective. Other professional communities, such as the IEEE, convene to discuss standards, and perhaps a similar process could become part of the ISMIR tradition as we continue to embrace the pursuit of reproducible research practices.

## 6. REFERENCES

- [1] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proc. of the 12th International Society for Music Information Retrieval Conference*, pages 591–596, 2011.
- [2] John Ashley Burgoyne, Jonathan Wild, and Ichiro Fujinaga. An expert ground truth set for audio chord recognition and music analysis. In *Proc. of the 12th International Society for Music Information Retrieval Conference*, pages 633–638, 2011.
- [3] Chris Cannam, Christian Landone, Mark B Sandler, and Juan Pablo Bello. The sonic visualiser: A visualisation platform for semantic descriptors from musical signals. In *Proc. of the 7th International Society for Music Information Retrieval Conference*, pages 324–327, 2006.
- [4] Trevor De Clercq and David Temperley. A corpus analysis of rock harmony. *Popular Music*, 30(1):47–70, 2011.
- [5] W Bas de Haas and John Ashley Burgoyne. Parsing the billboard chord transcriptions. *University of Utrecht, Tech. Rep*, 2012.
- [6] Matthias Mauch, Katy Noland, and Simon Dixon. Using musical structure to enhance automatic chord transcription. In *Proc. of the 10th International Society for Music Information Retrieval Conference*, pages 231–236, 2009.
- [7] Cory McKay, Rebecca Fiebrink, Daniel McEnnis, Beinan Li, and Ichiro Fujinaga. Ace: A framework for optimizing music classification. In *Proc. of the 6th International Society for Music Information Retrieval Conference*, pages 42–49, 2005.
- [8] Yizhao Ni, Matthew McVicar, Raul Santos-Rodriguez, and Tijl De Bie. Understanding effects of subjectivity in measuring chord estimation accuracy. *Audio, Speech, and Language Processing, IEEE Transactions on*, 21(12):2607–2615, 2013.
- [9] H el ene Papadopoulou and Geoffroy Peeters. Joint estimation of chords and downbeats from an audio signal. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(1):138–152, 2011.
- [10] G. Peeters and K. Fort. Towards a (better) definition of annotated MIR corpora. In *Proc. of the 13th International Society for Music Information Retrieval Conference*, pages 25–30, Porto, Portugal, Oct. 2012.
- [11] Jordan Bennett Louis Smith, John Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J Stephen Downie. Design and creation of a large-scale database of structural annotations. In *Proc. of the 12th International Society for Music Information Retrieval Conference*, pages 555–560, 2011.
- [12] Emmanuel Vincent, Stanislaw A Raczynski, Nobutaka Ono, Shigeki Sagayama, et al. A roadmap towards versatile mir. In *Proc. of the 11th International Society for Music Information Retrieval Conference*, pages 662–664, 2010.

<sup>6</sup> <http://www.freebase.com>

# ON THE CHANGING REGULATIONS OF PRIVACY AND PERSONAL INFORMATION IN MIR

**Pierre Saurel**  
Université Paris-Sorbonne  
[pierre.saurel@paris-sorbonne.fr](mailto:pierre.saurel@paris-sorbonne.fr)

**Francis Rousseaux**  
IRCAM  
[francis.rousseau@ircam.fr](mailto:francis.rousseau@ircam.fr)

**Marc Danger**  
ADAMI  
[mdanger@adami.fr](mailto:mdanger@adami.fr)

## ABSTRACT

In recent years, MIR research has continued to focus more and more on user feedback, human subjects data, and other forms of personal information. Concurrently, the European Union has adopted new, stringent regulations to take effect in the coming years regarding how such information can be collected, stored and manipulated, with equally strict penalties for being found in violation of the law.

Here, we provide a summary of these changes, consider how they relate to our data sources and research practices, and identify promising methodologies that may serve researchers well, both in order to be in compliance with the law and conduct more subject-friendly research. We additionally provide a case study of how such changes might affect a recent human subjects project on the topic of style, and conclude with a few recommendations for the near future.

This paper is not intended to be legal advice: our personal legal interpretations are strictly mentioned for illustration purpose, and reader should seek proper legal counsel.

## 1. INTRODUCTION

The International Society for Music Information Retrieval addresses a wide range of scientific, technical and social challenges, dealing with processing, searching, organizing and accessing music-related data and digital sounds through many aspects, considering real scale use-cases and designing innovative applications, exceeding its academic-only initiatory aims.

Some recent Music Information Retrieval tools and algorithms aim to attribute authorship and to characterize the structure of style, to reproduce the user's style and to manipulate one's style as a content [8], [1]. They deal for instance with active listening, authoring or personalised reflexive feedback. These tools will allow identification of users in the big data: authors, listeners, performers.

As the emerging MIR scientific community leads to industrial applications of interest to the international business (start-up, Majors, content providers, platforms) and to experimentations involving many users in living

labs (for MIR teaching, for multicultural emotion comparisons, or for MIR user requirement purposes) the identification of legal issues becomes essential or strategic.

Legal issues related to copyright and Intellectual Property have already been identified and expressed into Digital Rights Management by the MIR community [2], [7], when those related to security, business models and right to access have been expressed by Information Access [4], [11]. Privacy is another important legal issue. To address it properly one needs first to classify the personal data and processes. A naive classification appears when you quickly look at the kind of personal data MIR deals with:

- User's comments, evaluation, annotation and music recommendations are obvious personal data as long as they are published under their name or pseudo;
- Addresses allowing identification of a device or an instrument and Media Access Control addresses are linked to personal data;
- Any information allowing identification of a natural person, as some MIR processes do, shall be qualified as personal data and processing of personal data.

But the legal professionals do not unanimously approve this classification. For instance the Court of Appeal in Paris judged in two decisions (2007/04/27 and 2007/05/15) that the Internet Protocol address is not a personal data.

## 2. WHAT ARE PROCESSES OF PERSONAL DATA AND HOW THEY ARE REGULATED

A careful consideration of the applicable law of personal data is necessary to elaborate a proper classification of MIR personal data processes taking the different international regulations into account.

### 2.1 Europe vs. United States: two legal approaches

Europe regulates data protection through one of the highest State Regulations in the world [3], [9] when the United States lets contractors organize data protection through agreements supported by consideration and entered into voluntarily by the parties. These two approaches are deeply divergent. United States lets companies specify their own rules with their consumers while Europe enforces a unique regulated framework on all companies providing services to European citizens. For instance any company in the United States can define how long they keep the personal data, when the regulations in Europe would specify a maximum length of time the personal



data is to be stored. And this applies to any company offering the same service.

A prohibition is at the heart of the European Commission's Directive on Data Protection (95/46/CE – The Directive) [3]. The transfer of personal data to non-European Union countries that do not meet the European Union adequacy standard for privacy protection is strictly forbidden [3, article 25]<sup>1</sup>. The divergent legal approaches and this prohibition alone would outlaw the proposal by American companies of many of their IT services to European citizens. In response the U.S. Department of Commerce and the European Commission developed the Safe Harbor Framework (SHF) [6], [14]. Any non-European organization is free to self-certify with the SHF and join.

A new Proposal for a Regulation on the protection of individuals with regard to the processing of personal data was adopted the 12 March 2014 by the European Parliament [9]. The Directive allows adjustments from one European country to another and therefore diversity of implementation in Europe when the regulation is directly enforceable and should therefore be implemented directly and in the same way in all countries of the European Union. This regulation should apply in 2016. This regulation enhances data protection and sanctions to anyone who does not comply with the obligations laid down in the Regulation. For instance [9, article 79] the supervisory authority will impose, as a possible sanction, a fine of up to one hundred million Euros or up to 5% of the annual worldwide turnover in case of an enterprise.

## 2.2 Data protection applies to any information concerning an identifiable natural person

Until French law applied the 95/46/CE European Directive, personal data was only defined considering sets of data containing the name of a natural person. This definition has been extended; the 95/46/CE European Directive (ED) defines 'personal data' [3, article 2] as: "*any information relating to an identified or identifiable natural person ('data subject'); an identifiable person is one who can be identified, directly or indirectly, in particular by reference to an identification number or to one or more factors specific to his physical, physiological, mental, economic, cultural or social identity*".

For instance the identification of an author through the structure of his style as depending on his mental, cultural or social identity is a process that must comply with the European data privacy principles.

## 2.3 Safe Harbor is the Framework ISMIR affiliates need not to pay a fine up to hundreds million Euros

<sup>1</sup> Argentina, Australia, Canada, State of Israel, New Zealand, United States – Transfer of Air Passenger Name Record (PNR) Data, United States – Safe Harbor, Eastern Republic of Uruguay are, to date, the only non-European third countries ensuring an adequate level of protection: [http://ec.europa.eu/justice/data-protection/document/international-transfers/adequacy/index\\_en.htm](http://ec.europa.eu/justice/data-protection/document/international-transfers/adequacy/index_en.htm)

Complying with Safe Harbor is the easiest way for an organization using MIR processing to fulfill the high level European standard about personal data, to operate worldwide and to avoid prosecution regarding personal data. As explained below any non-European organization may enter the US – EU SHF's requirement and publicly declare that they do so. In that case the organization must develop a data privacy policy that conforms to the seven Safe Harbor Principles (SHP) [14].

First of all organizations must identify personal data and personal data processes. Then they apply the SHP to these data and processes. By joining the SHF, organizations must implement procedures and modify their own information system whether paper or electronic.

Organizations must notify (P1) individuals about the purposes for which they collect and use information about them, to whom the information can be disclosed and the choices and means offered for limiting its disclosure. Organizations must explain how they can be contacted with any complaints. Individuals should have the choice (P2) (opt out) whether their personal information is disclosed or not to a third party. In case of sensitive information explicit choice (opt in) must be given. A transfer to a third party (P3) is only possible if the individual made a choice and if the third party subscribed to the SHP or was subject to any adequacy finding regarding to the ED. Individuals must have access (P4) to personal information about them and be able to correct, amend or delete this information. Organizations must take reasonable precautions (P5) to prevent loss, misuse, disclosure, alteration or destruction of the personal information. Personal information collected must be relevant (P6: data integrity) for the purpose for which it is to be used. Sanctions (P7 enforcement) ensure compliance by the organization. There must be a procedure for verifying the implementation of the SHP and the obligation to remedy problems arising out of a failure to comply with the SHP.

## 3. CLASSIFICATION FOR MIR PERSONAL DATA PROCESSING

Considering the legal definition of personal data we can now propose a less naive classification of MIR processes and data into three sets: (i) nominative data, (ii) data leading to an easy identification of a natural person and (iii) data leading indirectly to the identification of a natural person through a complex process.

### 3.1 Nominative data and data leading easily to the identification of a natural person

The first set of processes deals with all the situations giving the name of a natural person directly. The second set deals with the cases of a direct or an indirect identification easily done for instance through devices.

In these two sets we find that the most obvious set of data concerns the "Personal Music Libraries" and "recommendations". Looking at the topics that characterize

ISMIR papers from year 2000 to 2013, we find more than 30 papers and posters dealing with those topics as their main topic. Can one recommend music to a user or analyze their personal library without tackling privacy?

### 3.2 Data leading to the identification of a natural person through a complex process

The third set of personal data deals with cases when a natural person is indirectly identifiable using a complex process, like some of the MIR processes.

Can one work on “Classification” or “Learning”, producing 130 publications (accepted contributions at ISMIR from year 2000 to year 2013) without considering users throughout their tastes or style? The processes used under these headings belong for the most part to this third set. Looking directly at the data without any sophisticated tool does not allow any identification of the natural person. On the contrary, using some MIR algorithms or machine learning can lead to indirect identifications [12].

Most of the time these non-linear methods use inputs to build new data which are outputs or data stored inside the algorithm, like weights for instance in a neural net.

### 3.3 The legal criteria of the costs and the amount of time required for identification

This third set of personal data is not as homogeneous as it seems to be at first glance. Can we compare sets of data that lead to an identification of a natural person through a complex process?

The European Proposal for a Regulation specifies the concept of “identifiability”. It tries to define legal criteria to decide if an identifiable set of data is or is not personal data. It considers the identification process [9, recital 23] as a relative one depending on the means used for that identification: *“To determine whether a person is identifiable, account should be taken of all the means reasonably likely to be used either by the controller or by any other person to identify or single out the individual directly or indirectly. To ascertain whether means are reasonably likely to be used to identify the individual, account should be taken of all objective factors, such as the costs of and the amount of time required for identification, taking into consideration both available technology at the time of the processing and technological development.”*

But under what criteria should we, as MIR practitioners, specify when a set of data allows an easy identification and belongs to the second set or, on the contrary, is too complex or reaches a too uncertain identification so that we would not legally say that these are personal data? To answer these questions, we must be able to compare MIR processes with new criteria.

## 4. MANAGING THE TWO FIRST SETS

On an example chosen to be problematic (but increasingly common in the industry), we show how to manage per-

sonal data in case of a simple direct or indirect identification process.

### 4.1 Trends in terms of use and innovative technology

Databases of personal data are no more clearly identified. We can view the situation as combining five aspects, which lead to new scientific problems concerning MIR personal data processing.

**Data Sources Explosion.** The number of databases for retrieving information is growing dramatically. Applications are also data sources. Spotify for instance provides a live flow of music consumption information from millions of users. Data from billions of sensors will soon be added. This profusion of data does not mean quality. Accessible does not mean legal or acceptable for a user. Those considerations are essential to build reliable and sustainable systems.

**Crossing & Reconciling Data.** Data sources are no longer isolated islands. Once the user can be identified (cookie, email, customer id), it is possible to match, aggregate and remix data that was previously isolated.

**Time Dimension.** The web has a good memory that humans are generally not familiar with. Data can be public one day and be considered as very private 3 years later. Many users forget they posted a picture after a student party. And the picture has the misfortune to crop up again when you apply for a job. And it is not only a question of human memory: Minute traces collected one day can be exploited later and provide real information.

**Permanent Changes.** The general instability of the data sources, technical formats and flows, applications and use is another strong characteristic of the situation. The impact on personal data is very likely. If the architecture of the systems changes a lot and frequently, the social norms also change. Users today publicly share information that they would have considered totally private a few years earlier. And the opposite could be the case.

**User Understandability and Control.** Because of the complexity of changing systems and complex interactions users will less and less control over their information. This lack of control is caused by the characteristics of the systems and by the mistakes and the misunderstandings of human users. The affair of the private Facebook messages appearing suddenly on timeline (Sept. 2012) is significant. Facebook indicates that there was no bug. Those messages were old wall posts that are now more visible with the new interface. This is a combination of bad user understanding and fast moving systems.

### 4.2 The case of an Apache Hadoop File System (AHFS) on which some machine learning is applied

Everyone produces data and personal data without being always aware that they provide data revealing their identification. When a user tags / rates musical items [13], he gives personal information. If a music recommender ex-

exploits this user data without integrating privacy concepts, he faces legal issues and strong discontent from the users.

The data volume has increased faster than “Moore’s law”: This is what is meant by “Big Data”. New data is generally unstructured and traditional database systems such as Relational Database Management Systems cannot handle the volume of data produced by users & machines & sensors. This challenge was the main drive for Google to define a new technology: the Apache Hadoop File System (AHFS). Within this framework, data and computational activities are distributed on a very large number of servers. Data is not loaded for computation, nor the results stored. Here, the algorithm is close to the data. This situation leads to the epistemological problem of separability into the field of MIR personal data processing: are all MIR algorithms (and for instance the authorship attribution algorithms) separable into data and processes? An answer to this question is required for any algorithm to be able to identify the set of personal data it deals with.

Now, let us consider a machine learning classifier/recommender trained on user data. In this sense, the algorithm is inseparable from the data it uses to function. And, if the machine is internalizing identifiable information from a set of users in a certain state (let say EU), it is then in violation to share the resulting function in a non-adequate country (let say Brazil) the EU if it was trained in, say, the US.

#### 4.3 Analyzing the multinational AHFS case

Regarding to the European regulation rules [3, art. 25], you may not transfer personal data collected in Europe to a non-adequate State (*see* list of adequate countries above). If you build a multinational AHFS system, you may collect data in Europe and in US depending on the way you localized the AHFS servers. The European data may not be transferred to Brazil. Even the classifier would not legally be used in Brazil as long as it internalizes some identifiable European personal information.

In practice one should then localize the AHFS files and machine-learning processes to make sure no identifiable data will be transferred from one country with a specific regulation to another with another regulation about personal data. We call these systems “heterarchical” due to the blended situation of a hierarchical system (the global AHFS management) and the need of a heterogeneous local regulation.

To manage properly the global AHFS system we need a first analysis of the system dispatching the different files on the right legal places. Privacy by Design (PbD) is a useful methodology to do so.

#### 4.4 Foundations Principals of Privacy by Design

PbD was first developed by Ontario’s Information and Privacy Commissioner, Dr. Ann Cavoukian, in the 1990s, at the very birth of the future big data phenomenon. This

solution has gained widespread international recognition, and was recently recognized as a global privacy standard.

According to its Canadian inventor<sup>1</sup>, is PbD based on seven Foundation Principles (FP): PbD “*is an approach to protect privacy by embedding it into the design specifications of technologies, business practices, and physical infrastructures. That means building in privacy up front – right into the design specifications and architecture of new systems and processes. PbD is predicated on the idea that, at the outset, technology is inherently neutral. As much as it can be used to chip away at privacy, it can also be enlisted to protect privacy. The same is true of processes and physical infrastructure*”:

- Proactive not Reactive (FP1): the PbD approach is based on proactive measures anticipating and preventing privacy invasive events before they occur;
- Privacy as the Default Setting (FP2): the default rules seek to deliver the maximum degree of privacy;
- Privacy embedded into Design (FP3): Privacy is embedded into the architecture of IT systems and business practices;
- Full Functionality – Positive Sum, not Zero-Sum (FP4): PbD seeks to accommodate all legitimate interests and objectives (security, etc.) in a “win-win” manner;
- End-to-End Security – Full Lifecycle Protection (FP5): security measures are essential to privacy, from start to finish;
- Visibility and Transparency — Keep it Open (FP6): PbD is subject to independent verification. Its component parts and operations remain visible and transparent, to users and providers alike;
- Respect for User Privacy — Keep it User-Centric (FP7): PbD requires architects and operators to keep the interests of the individual uppermost.

At the time of digital data exchange through networks, PbD is a key-concept in legacy [10]. In Europe, where this domain has been directly inspired by the Canadian experience, the EU<sup>2</sup> affirms: “*PbD means that privacy and data protection are embedded throughout the entire life cycle of technologies, from the early design stage to their deployment, use and ultimate disposal*”.

#### 4.5 Prospects for a MIR Privacy by Design

PbD is a reference for designing systems and processing involving personal data, enforced by the new European proposal for a Regulation [9, art. 23]. It becomes a method for these designs whereby it includes signal analysis methods and may interest MIR developers.

This proposal leads to new questions, such as the following: Is PbD a universal methodological solution about personal data for all MIR projects? Most of ISMIR contributions are still research oriented which doesn’t mean

<sup>1</sup> <http://www.ipc.on.ca/images/Resources/7foundationalprinciples.pdf>

<sup>2</sup> “Safeguarding Privacy in a Connected World – A European Data Protection Framework for the 21st Century” COM (2012) 9 final.



that they fulfill the two specific exceptions [9, art. 83]<sup>1</sup>. To say more about that intersection, we need to survey the ISMIR scientific production, throughout the main FPs. FP6 (transparency) and FP7 (user-centric) are usually respected among the MIR community as source code and processing are often (i) delivered under GNU like licensing allowing audit and traceability (ii) user-friendly. However, as long as PbD is not embedded, FP3 cannot be fulfilled and accordingly FP2 (default setting), FP5 (end-to-end), FP4 (full functionality) and FP1 (proactive) cannot be fulfilled even. Without any PbD embedded into Design, there are no default settings (FP2), you cannot follow an end-to-end approach (FP5), you cannot define full functionality regarding to personal data (FP4) nor be proactive. Principle of pro-activity (FP1) is the key. Fulfilling FP1 you define the default settings (FP2), be fully functional (FP4) and define an end-to-end process (FP5).

In brief is PbD useful to MIR developers even if it is not the definitive martingale!

## 5. EXPLORING THE THIRD SET

“Identifiability” is the potentiality of a set of data to lead to the identification of its source. A set of data should be qualified as being personal data if the cost and the amount of time required for identification are reasonable. These new criteria are a step forward since the qualification is not an absolute one anymore and depends specifically on the state of the art.

### 5.1 Available technology and technological development to take into account at this present moment

Changes in Information Technology lead to a shift in the approach of data management: from computational to data exploration. The main question is “What to look for?” Many companies build new tools to “make the data speak”. This is the case considering the trend of personalized marketing. Engineers using big data build systems that produce new personal dataflow.

Is it possible to stabilize these changes through standardization of metadata? Is it possible to develop a standardization of metadata which could ease the classification of MIR processing of personal data into identifying and non-identifying processes.

Many of the MIR methods are stochastic, probabilistic or designed to cost and more generally non-deterministic. On the contrary the European legal criteria [9, recital 23] (see above § 3.3) to decide whether a data is personal or not (the third set) seem to be much to deterministic to fit the effective new practices about machine learning on personal data.

<sup>1</sup> (i) these processing cannot be fulfilled otherwise and (ii) data permitting the identification are kept separately from the other information, or when the bodies conducting these data respect three conditions: (i) consent of the data subject, (ii) publication of personal data is necessary and (iii) data are made public

This situation leads to a new scientific problem: Is there an absolute criterion about the identifiability of personal data extracted from a set of data with a MIR process? What characterizes a maximal subset from the big data that could not ever be computed by any Turing machine to identify a natural person with any algorithm?

### 5.2 What about the foundational separation in computer science between data and process?

Computer science is based on a strict separation between data and process (dual as these two categories are interchangeable at any time; data can be activated as a process and a process can be treated as a data).

We may wonder about the possibility of maintaining the data/process separation paradigm if i) the data stick to the process and ii) the legal regulation leads to a location of the data in the legal system in which those data were produced.

## 6. CONCLUSION

### 6.1 When some process lead to direct or indirect personal data identification

**Methodological Recommendations.** MIR researchers could first audit their algorithm and data, and check if they are able to identify a natural person (two first sets of our classification). If so they could use the SHF which could already be an industrial challenge for instance regarding Cyber Security (P5). Using the PbD methodology certainly leads to operational solutions in these situations.

### 6.2 When some process may lead to indirect personal data identification through some complex process

In many circumstances, the MIR community develops new personal data on the fly, using the whole available range of data analysis and data building algorithm. Then researchers could apply the PbD methodology, to insure that no personal data is lost during the system design.

Here PbD is not a universal solution because the time when data (on the one hand) and processing (on the other hand) were functionally independent, formally and semantically separated, has ended. Nowadays, MIR researchers currently use algorithms that support effective decision, supervised or not, without introducing ‘pure’ data or ‘pure’ processing, but building up acceptable solutions together with machine learning [5] or heuristic knowledge that cannot be reduced to data or processing: The third set of personal data may appear, and raise theoretical scientific problems.

**Political Opportunities.** The MIR community has a political role to play in the data privacy domain, by explaining to lawyers —joining expert groups in the US, UE or elsewhere— what we are doing and how we overlap with the tradition in style description, turning it into a computed style genetic, which radically questions the analysis of data privacy traditions, cultures and tools.

**Future Scientific Works.** In addition to methodological and political ones, we face purely scientific challenges, which constitute our research program for future works. Under what criteria should we, as MIR practitioners, specify when a set of data allows an easy identification and belongs to the second set or on the contrary is too complex or allows a too uncertain identification so that we would say that these are not personal data? What characterizes a maximal subset from the big data that could not ever be computed by any Turing machine to identify a natural person with any algorithm?

## 7. REFERENCES

- [1] S. Argamon, K. Burns, S. Dubnov (Eds): *The Structure of Style*, Springer-Verlag, 2010.
- [2] C. Barlas: “Beating Babel - Identification, Metadata and Rights”, Invited Talk, *Proceedings of the International Symposium on Music Information Retrieval*, 2002.
- [3] Directive (95/46/EC) of 24 October 1995 *Official Journal L 281, 23/11/1995 P. 0031 - 0050*: <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML>
- [4] J.S. Downie, J. Futrelle, D. Tchong: “The International Music Information Retrieval Systems Evaluation Laboratory: Governance, Access and Security”, *Proceedings of the International Symposium on Music Information Retrieval*, 2004.
- [5] A. Gkoulalas-Divanis, Y. Saygin, Vassilios S. Verykios: “Special Issue on Privacy and Security Issues in Data Mining and Machine Learning”, *Transactions on Data Privacy*, Vol. 4, Issue 3, pp. 127-187, December 2011.
- [6] D. Greer: “Safe Harbor - A Framework that Works”, *International Data Privacy Law*, Vol.1, Issue 3, pp. 143-148, 2011.
- [7] M. Levering: “Intellectual Property Rights in Musical Works: Overview, Digital Library Issues and Related Initiatives”, Invited Talk, *Proceedings of the International Symposium on Music Information Retrieval*, 2000.
- [8] F. Pachet, P. Roy: “Hit Song Science is Not Yet a Science”, *Proceedings of the International Symposium on Music Information Retrieval*, 2008.
- [9] Proposal for a Regulation on the protection of individuals with regard to the processing of personal data was adopted the 12 March 2014 by the European Parliament: <http://www.europarl.europa.eu/sides/getDoc.do?type=TA&reference=P7-TA-2014-0212&language=EN>
- [10] V. Reding: “The European Data Protection Framework for the Twenty-first century”, *International Data Privacy Law*, volume 2, issue 3, pp.119-129, 2012.
- [11] A. Seeger: “I Found It, How Can I Use It? - Dealing With the Ethical and Legal Constraints of Information Access”, *Proceedings of the International Symposium on Music Information Retrieval*, 2003.
- [12] A.B. Slavkovic, A. Smith: “Special Issue on Statistical and Learning-Theoretic Challenges in Data Privacy”, *Journal of Privacy and Confidentiality*, Vol. 4, Issue 1, pp. 1-243, 2012.
- [13] P. Symeonidis, M. Ruxanda, A. Nanopoulos, Y. Manolopoulos: “Ternary Semantic Analysis of Social Tags for Personalized Music Recommendation”, *Proceedings of the International Symposium on Music Information Retrieval*, 2008.
- [14] U.S. – EU Safe Harbor: [http://www.export.gov/safeharbor/eu/eg\\_main\\_018365.asp](http://www.export.gov/safeharbor/eu/eg_main_018365.asp)

# A MULTI-MODEL APPROACH TO BEAT TRACKING CONSIDERING HETEROGENEOUS MUSIC STYLES

Sebastian Böck, Florian Krebs and Gerhard Widmer

Department of Computational Perception  
Johannes Kepler University, Linz, Austria

sebastian.boeck@jku.at

## ABSTRACT

In this paper we present a new beat tracking algorithm which extends an existing state-of-the-art system with a multi-model approach to represent different music styles. The system uses multiple recurrent neural networks, which are specialised on certain musical styles, to estimate possible beat positions. It chooses the model with the most appropriate beat activation function for the input signal and jointly models the tempo and phase of the beats from this activation function with a dynamic Bayesian network. We test our system on three big datasets of various styles and report performance gains of up to 27% over existing state-of-the-art methods. Under certain conditions the system is able to match even human tapping performance.

## 1. INTRODUCTION AND RELATED WORK

The automatic inference of the metrical structure in music is a fundamental problem in the music information retrieval field. In this line, *beat tracking* deals with finding the most salient level of this metrical grid, the *beat*. The beat consists of a sequence of regular time instants which usually invokes human reactions like foot tapping. During the last years, beat tracking algorithms have considerably improved in performance. But still they are far from being considered on par with human beat tracking abilities – especially for music styles which do not have simple metrical and rhythmic structures.

Most methods for beat tracking extract some features from the audio signal as a first step. As features, commonly low-level features such as amplitude envelopes [20] or spectral features [2], mid-level features like onsets either in discretised [8, 12] or continuous form [6, 10, 16, 18], chord changes [12, 18] or combinations thereof with higher level features such as rhythmic patterns [17] or metrical relations [11] are used. The feature extraction is usually followed by a stage that determines periodicities within the extracted features sequences. Autocorrelation [2, 9, 12] and comb filters [6, 20] are commonly used techniques for

this task. Most systems then determine the most predominant tempo from these periodicities and subsequently determine the beat times using *multiple agents* approaches [8, 12], *dynamic programming* [6, 10], *hidden Markov models (HMM)* [7, 16, 18], or *recurrent neural networks (RNN)* [2]. Other systems operate directly on the input features and jointly determine the tempo and phase of the beats using *dynamic Bayesian networks (DBN)* [3, 14, 17, 21].

One of the most common problems of beat tracking systems are “octave errors”, meaning that a system detects beats at double or half the rate of the ground truth tempo. For human tappers this generally does not constitute a problem, as can be seen when comparing beat tracking results at different metrical levels [6]. Hainsworth and Macleod stated that beat tracking systems will have to be style specific in the future in order to improve the state-of-the-art [14]. This is consistent with the finding of Krebs et al. [17] who showed on a dataset of Ballroom music that the beat tracking performance can be improved by incorporating style-specific knowledge, especially by resolving the octave error. While approaches have been proposed which combined multiple existing features for beat tracking [22], no one has so far combined several models specialised on different musical styles to improve the overall performance.

In this paper, we propose a multi-model approach to fuse information of different models that have been specialised on heterogeneous music styles. The model is based on the *recurrent neural network (RNN)* beat tracking system proposed in [2] and can be easily adapted to any music style without further parameter tweaking, only by providing a corresponding beat-annotated dataset. Further, we propose an additional *dynamic Bayesian network* stage based on the work of Whiteley et al. [21] which jointly infers the tempo and the beat phase from the beat activations of the RNN stage.

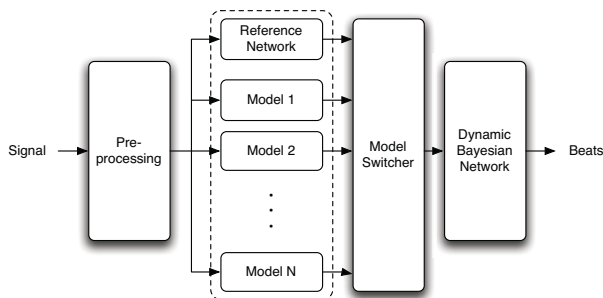
## 2. PROPOSED METHOD

The new beat tracking algorithm is based on the state-of-the-art approach presented by Böck and Schedl in [2]. We extend their system to be able to better deal with heterogeneous music styles and combine it with a dynamic Bayesian network similar to the ones presented in [21] and [17].

The basic structure is depicted in Figure 1 and consists of the following elements: first the audio signal is pre-processed and fed into multiple *neural network* beat track-



ing modules. Each of the modules is trained on different audio material and outputs a different beat activation function when activated with a musical signal. These functions are then fed into a module which chooses the most appropriate model and passes its activation function to a *dynamic Bayesian network* to infer the actual beat positions.



**Figure 1.** Overview of the new multi-model beat tracking system.

Theoretically, a single network large enough should be able to model all the different music styles simultaneously, but unfortunately this optimal solution is hardly achievable. The main reason for this is the difficulty to choose an absolutely balanced training set with an evenly distributed set of beats over all the different dimensions relevant for detecting beats. These include rhythmic patterns [17, 20], harmonic aspects and many other features. To overcome this limitation, we split the available training data into multiple parts. Each part should represent a more homogeneous subset than the whole set so that the networks are able to specialise on the dominant aspects of this subset.

It seems reasonable to assume that humans do something similar when tracking beats [4]. Depending on the style of the music, the rhythmic patterns present, the instrumentation, the timbre, they apply their musical knowledge to choose one of their “learned” models and then decide which musical events are beats or not. Our approach mimics this behaviour by learning multiple distinct models.

## 2.1 Signal pre-processing

All neural networks share the same signal pre-processing step, which is very similar to the work in [2]. As inputs to the different neural networks, the logarithmically filtered and scaled spectrograms of three parallel *Short Time Fourier Transforms (STFT)* obtained for different window lengths and their positive first order differences are used. The system works with a constant frame rate  $f_r$  of 100 frames per second. Window lengths of 23.2 ms, 46.4 ms and 92.9 ms are used and the resulting spectrogram bins of the discrete Fourier transforms are filtered with overlapping triangular filters to have a frequency resolution of three bands per octave. To put all resulting magnitude values into a positive range we add 1 before taking the logarithm.

## 2.2 Multiple parallel neural networks

At the core of the new approach, multiple neural networks are used to determine possible beat locations in the audio signal. As outlined previously, these networks are trained on material with different music styles to be able to better detect the beats in heterogeneous music styles.

As networks we chose the same *recurrent neural network (RNN)* topology as in [2] with three bidirectional hidden layers with 25 *long short-term memory (LSTM)* units per layer. For training of the networks, standard gradient descent with error backpropagation and a learning rate of  $1e^{-4}$  is used. We initialise the network weights with a Gaussian distribution with mean 0 and standard deviation of 0.1. We use early stopping with a disjoint validation set to stop training if no improvement over 20 epochs can be observed.

One reference network is trained on the complete dataset until the stopping criterion is reached for the first time. We use this point during the training phase to diverge the specialised models from the reference network.

Afterwards, all networks are fine-tuned with a reduced learning rate of  $1e^{-5}$  on either the complete set or the individual subsets (cf. Section 3.1) with the above mentioned stopping criterion. Given  $N$  subsets,  $N + 1$  models are generated.

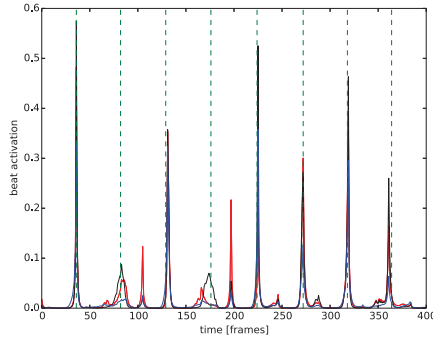
The output functions of the network models represent the beat probability at each time frame. Instead of tracking the beats with an autocorrelation function as described in the original work, the beat activation functions of the different models are fed into the next model-selection stage.

## 2.3 Model selection

The purpose of this stage is to select a model which outputs a better beat activation function than the reference model when activated with a signal. Compared to the reference model, the specialised models produce better predictions on input data which is similar to that used for fine-tuning, but worse predictions on signals dissimilar to the training data. This behaviour can be seen in Figure 2, where the specialised model produces higher beat activation values at the beat locations and lower values elsewhere.

Table 1 illustrates the impact on the *Ballroom* subset, where the relative gain of the best specialised model compared to the reference model (+1.7%) is lower than the penalties of the other models (−2.3% to −6.3%). The fact that the performance degradation of the unsuitable specialised models is greater than the gain of the most suitable model allows us to use a very simple but effective method to choose the best model.

To select the best performing model, all network outputs of the fine-tuned networks are compared with the output of the reference network (which was trained on the whole training set) and the one yielding the lowest *mean squared difference* is selected as the final one and its output is fed into the final beat tracking stage.



**Figure 2.** Example beat activations for a 4 seconds ballroom snippet. Red is the reference network’s activations, black the selected model and blue a discarded one. Green dashed vertical lines denote the annotated beat positions.

	F-measure	Cemgil	AMLc	AMLt
SMC *	0.834	0.807	0.664	0.767
Hainsworth *	0.867	0.839	0.694	0.793
Ballroom *	0.904	0.872	0.777	0.853
Reference	0.887	0.855	0.748	0.831
Multi-model	0.897	0.866	0.759	0.841

**Table 1.** Performance of differently specialised models (marked with asterisks, fine-tuned on the *SMC*, *Hainsworth* and *Ballroom* subsets) on the *Ballroom* subset compared to the reference model and the network selected by the multi-model selection stage.

## 2.4 Dynamic Bayesian network

Independent of whether only one or multiple neural networks are used, the approach of Böck and Schedl [2] has a fundamental shortcoming: the final peak-picking stage does not try to find a global optimum when selecting the final locations of the beats. It rather determines the dominant tempo of the piece (or a segment of certain length) and then aligns the beat positions according to this tempo by simply choosing the best start position and then progressively locating the beats at positions with the highest activation function values in a certain region around the pre-determined position. To allow a greater responsiveness to tempo changes, this chosen region must not be too small. However, this also introduces a weakness to the algorithm, because the tracking stage can easily get distracted by a few misaligned beats and needs some time to recover from this fault. The activation function depicted in Figure 2 has two of these spurious detections around frames 100 and 200.

To circumvent this problem, we feed the output of the chosen neural network model into a *dynamic Bayesian network (DBN)* which jointly infers tempo and phase of a beat sequence. Another advantage of this new method is that we are able to model both beat and non-beat states, which was shown to perform superior to the case where only beat states are modelled [7].

The DBN we use is closely related to the one proposed in [21], adapted to our specific needs. Instead of modelling whole bars, we only model one beat period which reduces the size of the search space. Additionally we do not model rhythmic patterns explicitly and leave this higher level analysis to the neural networks. This finally leads to a DBN which consists of two hidden variables, the tempo  $\omega$  and the position  $\phi$  inside a beat period. In order to infer the hidden variables from an audio signal, we have to specify three entities: A *transition model* which describes the transitions between the hidden variables, an *observation model* which takes the beat activations from the neural network and transforms them into probabilities suitable for the DBN, and the *initial distribution* which encodes prior knowledge about the hidden variables. For computational ease we discretise the tempo-beat space to be able to use standard hidden Markov model (HMM) [19] algorithms for inference.

### 2.4.1 Transition model

The beat period is discretised into  $\Phi = 640$  equidistant cells and  $\phi \in \{1, \dots, \Phi\}$ . We refer to the unit of the variable  $\phi$  (position inside a beat period) as *pib*.  $\phi_k$  at audio frame  $k$  is then computed by

$$\phi_k = (\phi_{k-1} + \omega_{k-1} - 1) \bmod \Phi + 1. \quad (1)$$

The tempo space is discretised into  $\Omega = 23$  equidistant cells, which cover the tempo range up to 215 beats per minute (BPM). The unit of the tempo variable  $\omega$  is *pib per audio frame*. As we want to restrict  $\omega$  to integer values (to stay within the  $\phi$  grid at transitions), we need a high resolution of  $\phi$  in order to get a high resolution of  $\omega$ . Based on experiments with the training set, we set the tempo space to  $\omega \in \{6, \dots, \Omega\}$ , where  $\omega = 6$  is equivalent to a minimum tempo of  $6 \times 60 \times f_r / \Phi \approx 56$  BPM. As in [21] we only allow for three tempo transitions at time frame  $k$ : It stays constant, it accelerates, or it decelerates.

$$\omega_k = \begin{cases} \omega_{k-1}, & P(\omega_k | \omega_{k-1}) = 1 - p_\omega \\ \omega_{k-1} + 1, & P(\omega_k | \omega_{k-1}) = \frac{p_\omega}{2} \\ \omega_{k-1} - 1, & P(\omega_k | \omega_{k-1}) = \frac{p_\omega}{2} \end{cases} \quad (2)$$

Transitions to tempi outside of the allowed range are not allowed by setting the corresponding transition probabilities to zero. The probability of a tempo change  $p_\omega$  was set to 0.002.

### 2.4.2 Observation model

Since the beat activation function  $a$  produced by the neural network is limited to the range  $[0, 1]$  and shows high values at beat positions and low values at non-beat positions, we use the activation function directly as state-conditional observation distributions (similar to [7]). We define the observation likelihood as

$$P(a_k | \phi_k) = \begin{cases} a_k, & 1 \leq \phi_k \leq \frac{\Phi}{\lambda} \\ \frac{1-a_k}{\lambda-1}, & \text{otherwise.} \end{cases} \quad (3)$$

$\lambda \in [\frac{\Phi}{\lambda-1}, \Phi]$  is a parameter that controls the proportion of the beat interval which is considered as beat and non-beat

location. Smaller values of  $\lambda$  (a higher proportion of beat locations and a smaller proportion of non-beat locations) are especially important for higher tempi, as the DBN visits only a few position states of a beat interval and could possibly miss the beginning of a beat. On the other hand, higher values of  $\lambda$  (a smaller proportion of beat locations) lead to less accurate beat tracking, as the activations are blurred in the state domain of the DBN. On our training set we achieved the best results with the value  $\lambda = 16$ .

### 2.4.3 Initial state distribution

The initial state distribution is normally used to incorporate any prior knowledge about the hidden states, such as tempo distributions. In this paper, we use a uniform distribution over all states, for simplicity and ease of generalisation.

### 2.4.4 Inference

We are interested in the sequence of hidden variables  $\phi_{1:K}$  and  $\omega_{1:K}$ , that maximise the posterior probability of the hidden variables given the observations (activations  $a_{1:K}$ ). Combining the discrete states of  $\phi$  and  $\omega$  into one state vector  $\mathbf{x}_k = [\phi_k, \omega_k]$ , we can compute the maximum a-posteriori state sequence  $\mathbf{x}_{1:K}^*$  by

$$\mathbf{x}_{1:K}^* = \arg \max_{\mathbf{x}_{1:K}} p(\mathbf{x}_{1:K} | a_{1:K}). \quad (4)$$

Equation 4 can be computed efficiently using the well-known Viterbi algorithm [19]. Finally the set of beat times  $\mathcal{B}$  are determined by the set of time frames  $k$  which were assigned to a beat position ( $\mathcal{B} = \{k : \phi_k < \phi_{k-1}\}$ ). In our experiments we found that the beat detection becomes less accurate if the part of the beat interval which is considered as beat-state is too large (i.e. smaller values of  $\lambda$ ). Therefore we determine the final beat times by looking for the highest beat activation value inside the beat-state window  $\mathcal{W} = \{k : \phi_k \leq \frac{\phi}{\lambda}\}$ .

## 3. EVALUATION

For the development and evaluation of the algorithm we used some well-known datasets. This allows for highest comparability with previously published results of state-of-the-art algorithms.

### 3.1 Datasets

As training material for our system, the datasets introduced in [13–15] are used. They are called *Ballroom*, *Hainsworth* and *SMC* respectively. To show the ability of our new algorithm to adapt to various music styles, a very simple approach of splitting the complete dataset into multiple subsets according to the original source was chosen. Although far from optimal – both the *SMC* and *Hainsworth* datasets contain heterogeneous music styles – we still consider this a valid choice, since any “better” splitting would allow the system to adapt even further to heterogeneous styles and in turn lead to better results. At least the three sets have a somehow different focus regarding the music styles present.

### 3.2 Performance measures

In line with almost all other publications on the topic of beat tracking, we report the following scores:

**F-measure** : counts the number of true positive (correctly located beats within a tolerance window of  $\pm 70$  ms), false positive and negative detections;

**P-score** : measures the tracking accuracy by the correlation of the detections and the annotations, considering deviations within 20% of the annotated beat interval as correct;

**Cemgil** : places a Gaussian function with a standard deviation of 40 ms around the annotations and then measures the tracking accuracy by summing up the scores of the detected beats on this function normalising it by the overall length of the annotations or detections, whichever is greater;

**CMLc & CMLt** : measure the longest continuously segment (CMLc) or all correctly tracked beats (CMLt) at the correct metrical level. A beat is considered correct if it is reported within a 17.5% tempo and phase tolerance, and the same applies for the previously detected beat;

**AMLc & AMLt** : like CMLc & CMLt, but additionally allow offbeat and double/half as well as triple/third tempo variations of the annotated beats;

**D & D<sub>g</sub>** : the information gain (D) and global information gain (D<sub>g</sub>) are phase agnostic measures comparing the annotations with the detections (and vice-versa) building an error histogram and then calculating the Kullback-Leibler divergence w.r.t. a uniform histogram.

A more detailed description of the evaluation methods can be found in [5]. However, since we only investigate offline algorithms, we do not skip the first five seconds for evaluation.

### 3.3 Results & Discussion

Table 2 lists the performance results of the reference implementation, Böck’s *BeatTracker.2013*, and the various extensions proposed in this paper for all datasets. All results are obtained with 8-fold cross validation with previously defined splittings, ensuring that no pieces are used both for training or parameter tuning and testing purposes. Additionally, we compare our new approach to published state-of-the-art results on the *Hainsworth* and *Ballroom* datasets.

#### 3.3.1 Multi-model extension

As can be seen, the use of the *multi-model* extension almost always improves the results over the implementation it is based on, especially on the *SMC* set. The gain in performance on the *Ballroom* set was expected, since Krebs et al. already showed that modelling rhythmic patterns helps to increase the overall detection accuracy [17]. Although we did not split the set according to the individual rhythmic patterns, the overall style of ballroom music can be considered unique enough to be distinct from the other music

	F-measure	P-score	Cemgil	CMLc	CMLt	AMLc	AMLt	D	D <sub>g</sub>
<i>Ballroom</i>									
BeatTracker.2013 [1, 2]	0.887	0.863	0.855	0.719	0.795	0.748	0.831	3.404	2.596
— Multi-Model	0.897	0.875	<b>0.866</b>	0.740	0.814	0.759	0.841	<b>3.480</b>	<b>2.674</b>
— DBN	0.903	0.876	0.838	0.792	0.825	0.873	0.915	3.427	2.275
— Multi-Model + DBN	<b>0.910</b>	<b>0.881</b>	0.845	<b>0.800</b>	<b>0.830</b>	<b>0.885</b>	<b>0.924</b>	3.469	2.352
Krebs et al. [17]	0.855	0.839	0.772	0.745	0.786	0.818	0.865	2.499	1.681
Zapata et al. [22] †	0.767	0.735	0.672	0.586	0.607	0.824	0.860	2.750	1.187
<i>Hainsworth</i>									
BeatTracker.2013 [1, 2]	0.832	0.843	0.712	0.618	0.756	0.655	0.807	2.167	1.468
— Multi-Model	0.832	0.847	<b>0.716</b>	0.617	0.761	0.652	0.809	2.171	<b>1.490</b>
— DBN	<b>0.843</b>	<b>0.867</b>	0.711	<b>0.696</b>	<b>0.808</b>	0.759	<b>0.883</b>	2.251	1.481
— Multi-Model + DBN	0.840	0.865	0.707	<b>0.696</b>	0.803	<b>0.760</b>	0.881	<b>2.268</b>	1.466
Zapata et al. [22] †	0.710	0.732	0.589	0.569	0.642	0.709	0.824	2.057	0.880
Davies et al. [6]	-	-	-	0.548	0.612	0.681	0.789	-	-
Peeters & Papadopoulos [18]	-	-	-	0.547	0.628	0.703	0.831	-	-
Degara et al. [7]	-	-	-	0.561	0.629	0.719	0.815	-	-
Human tapper [6] ‡	-	-	-	0.528	0.812	0.575	0.874	-	-
<i>SMC</i>									
BeatTracker.2013 [1, 2]	0.497	0.598	0.402	0.238	0.360	0.279	0.436	1.263	0.416
— Multi-Model	0.514	0.617	<b>0.415</b>	0.257	0.389	0.296	0.467	1.324	0.467
— DBN	0.516	0.622	0.404	0.294	0.415	0.378	0.550	1.426	0.504
— Multi-Model + DBN	<b>0.529</b>	<b>0.630</b>	<b>0.415</b>	<b>0.296</b>	<b>0.428</b>	<b>0.383</b>	<b>0.567</b>	<b>1.460</b>	<b>0.531</b>
Zapata et al. [22] †	0.369	0.460	0.285	0.115	0.158	0.239	0.397	0.879	0.126

**Table 2.** Performance of the proposed algorithm on the *Ballroom* [13], *Hainsworth* [14] and *SMC* [15] datasets. *BeatTracker* is the reference implementation our *Multi-Model* and *dynamic Bayesian network (DBN)* extensions are built on. The results marked with † are obtained with Essentia’s implementation of the multi-feature beat tracker. <sup>1</sup> ‡ denotes causal (i.e. online) processing, all listed algorithms use non-causal analysis (i.e. offline processing) with the best results in bold.

styles present in the other sets and the salient features can be exploited successfully by the multi-model approach.

### 3.3.2 Dynamic Bayesian network extension

As already indicated in the original paper [2] (and described earlier in Section 2.4), the original *BeatTracker* can be easily distracted by some misaligned beats and then needs some time to recover from any failure. The newly adapted dynamic Bayesian network beat tracking stage does not suffer from this shortcoming by searching for the globally best beat locations. The use of the DBN boosts the performance on all datasets for almost all evaluation measures. Interestingly, the Cemgil accuracy is degraded by using the DBN stage. This might be explained by the fact that the discretisation grid of the beat period beat positions becomes too coarse for low tempi (cf. Section 2.4.4) and therefore yields inaccurate beat detections, which especially affect the Cemgil accuracy. This is one of the issues that needs to be resolved in the future, especially for lower tempi where the penalty is the highest.

### 3.3.3 Comparison with other methods

Our new system set side by side with other state-of-the-art algorithms draws a clear picture. It outperforms all of them considerably – independently of the dataset and evaluation measure chosen. Especially the high performance boosts of the CMLc and CMLt scores on the *Hainworth* dataset highlight the ability to track the beats at the correct metrical level significantly more often than any other method.

Davies et al. [6] also list performance results of a human tapper on the same dataset. However it must be noted that these were obtained by online real-time tapping, hence they cannot be compared directly to the system presented. However, the system of Davies et al. can also be switched to causal mode (and thus being comparable to a human tapper). In this mode it achieved performance reduced by approximately 10% [6]. Adding the same amount to the reported tapping results of 0.528 CMLc and 0.575 AMLc suggests that our system is capable of performing as good as humans when continuous tapping is required.

On the *Ballroom* set we achieve higher results than the particularly specialised system of Krebs et al. [17]. Since our DBN approach is a simplified variant of their model, it can be assumed that the relatively low scores of the Cemgil accuracy and the information gain are due to the same reason – the coarse discretisation of the beat or bar states. Nonetheless, comparing the continuity scores (which have higher tolerance thresholds) we can still report an average increase in performance of more than 5%.

## 4. CONCLUSIONS & OUTLOOK

In this paper we have presented a new beat tracking system which is able to improve over existing algorithms by incorporating multiple models which were trained on different music styles and combining it with a dynamic Bayesian

<sup>1</sup> <http://essentia.upf.edu>, v2.0.1

network for the final inference of the beats. The combination of these two extensions yields a performance boost – depending on the dataset and evaluation measures chosen – of up to 27% relative, matching human tapping results under certain conditions. It outperforms other state-of-the-art algorithms in tracking the beats at the correct metrical level by 20%.

We showed that the specialisation on a certain musical style helps to improve the overall performance, although the method for splitting the available data into sets of different styles and then selecting the most appropriate model is rather simple. For the future we will investigate more advanced techniques for the selection of suitable data for the creation of the specialised models, e.g. splitting the datasets according to dance styles as performed by Krebs et al. [17] or applying unsupervised clustering techniques. We also expect better results from more advanced model selection methods. One possible approach could be to feed the individual model activations to the dynamic Bayesian network and let it choose among them.

Finally, the Bayesian network could be tuned towards using a finer beat positions grid and thus reporting the beats at more appropriate times than just selecting the position of the highest activation reported by the neural network model.

## 5. ACKNOWLEDGMENTS

This work is supported by the European Union Seventh Framework Programme FP7 / 2007-2013 through the GiantSteps project (grant agreement no. 610591) and the Austrian Science Fund (FWF) project Z159.

## 6. REFERENCES

- [1] MIREX 2013 beat tracking results. [http://nema.lis.illinois.edu/nema\\_out/mirex2013/results/abt/](http://nema.lis.illinois.edu/nema_out/mirex2013/results/abt/), 2013.
- [2] S. Böck and M. Schedl. Enhanced Beat Tracking with Context-Aware Neural Networks. In *Proceedings of the 14th International Conference on Digital Audio Effects (DAFx-11)*, pages 135–139, Paris, France, September 2011.
- [3] A. T. Cemgil, H. Kappen, P. Desain, and H. Honing. On tempo tracking: Tempogram Representation and Kalman filtering. *Journal of New Music Research*, 28:4:259–273, 2001.
- [4] N. Collins. Towards a style-specific basis for computational beat tracking. In *Proceedings of the 9th International Conference on Music Perception and Cognition (ICMPC9)*, pages 461–467, Bologna, Italy, 2006.
- [5] M. E. P. Davies, N. Degara, and M. D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. Technical Report C4DM-TR-09-06, Centre for Digital Music, Queen Mary University of London, 2009.
- [6] M. E. P. Davies and M. D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1009–1020, March 2007.
- [7] N. Degara, E. Argones-Rúa, A. Pena, S. Torres-Guijarro, M. E. P. Davies, and M. D. Plumbley. Reliability-informed beat tracking of musical signals. *IEEE Transactions on Audio, Speech and Language Processing*, 20(1):290–301, January 2012.
- [8] S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30:39–58, 2001.
- [9] D. Eck. Beat tracking using an autocorrelation phase matrix. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007)*, volume 4, pages 1313–1316, Honolulu, Hawaii, USA, April 2007.
- [10] D. P. W. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 2007:51–60, 2007.
- [11] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stafylakis. Music tempo estimation and beat tracking by applying source separation and metrical relations. In *Proceedings of the 37th International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, pages 421–424, Kyoto, Japan, March 2012.
- [12] M. Goto and Y. Muraoka. Beat tracking based on multiple-agent architecture a real-time beat tracking system for audio signals. In *Proceedings of the International Conference on Multiagent Systems*, pages 103–110, 1996.
- [13] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, September 2006.
- [14] S. Hainsworth and M. Macleod. Particle filtering applied to musical tempo tracking. *EURASIP J. Appl. Signal Process.*, 15:2385–2395, January 2004.
- [15] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2539–2548, November 2012.
- [16] A. Klapuri, A. Eronen, and J. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, January 2006.
- [17] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, pages 227–232, Curitiba, Brazil, November 2013.
- [18] G. Peeters and H. Papadopoulos. Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1754–1769, 2011.
- [19] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, pages 257–286, 1989.
- [20] E. D. Scheirer. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103(1):588–601, 1998.
- [21] N. Whiteley, A. Cemgil, and S. Godsill. Bayesian modelling of temporal structure in musical audio. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, pages 29–34, Victoria, BC, Canada, October 2006.
- [22] J. R. Zapata, M. E. P. Davies, and E. Gómez. Multi-feature beat tracking. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(4):816–825, April 2014.





Oral Session 8  
**Source Separation**

This Page Intentionally Left Blank

# EXTENDING HARMONIC-PERCUSSIVE SEPARATION OF AUDIO SIGNALS

Jonathan Driedger<sup>1</sup>, Meinard Müller<sup>1</sup>, Sascha Disch<sup>2</sup><sup>1</sup>International Audio Laboratories Erlangen<sup>2</sup>Fraunhofer Institute for Integrated Circuits IIS, Erlangen, Germany

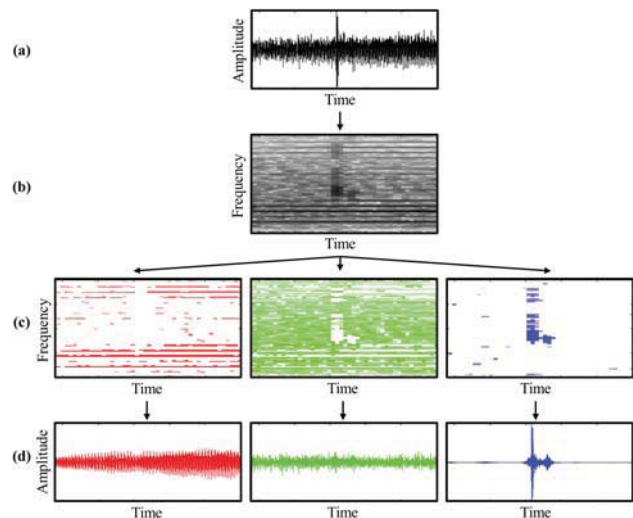
{jonathan.driedger,meinard.mueller}@audiolabs-erlangen.de, sascha.disch@iis.fraunhofer.de

## ABSTRACT

In recent years, methods to decompose an audio signal into a harmonic and a percussive component have received a lot of interest and are frequently applied as a processing step in a variety of scenarios. One problem is that the computed components are often not of purely harmonic or percussive nature but also contain noise-like sounds that are neither clearly harmonic nor percussive. Furthermore, depending on the parameter settings, one often can observe a leakage of harmonic sounds into the percussive component and vice versa. In this paper we present two extensions to a state-of-the-art harmonic-percussive separation procedure to target these problems. First, we introduce a *separation factor* parameter into the decomposition process that allows for tightening separation results and for enforcing the components to be clearly harmonic or percussive. As second contribution, inspired by the classical sines+transients+noise (STN) audio model, this novel concept is exploited to add a third *residual* component to the decomposition which captures the sounds that lie *in between* the clearly harmonic and percussive sounds of the audio signal.

## 1. INTRODUCTION

The task of decomposing an audio signal into its harmonic and its percussive component has received large interest in recent years. This is mainly because for many applications it is useful to consider just the harmonic or the percussive portion of an input signal. Harmonic-percussive separation has been applied, for example, for audio remixing [9], improving the quality of chroma features [14], tempo estimation [6], or time-scale modification [2, 4]. Several decomposition algorithms have been proposed. In [3], the percussive component is modeled by detecting portions in the input signal which have a rather noisy phase behavior. The harmonic component is then computed by the difference of the original signal and the computed percussive component. In [10], the crucial observation is that



**Figure 1.** (a): Input audio signal  $x$ . (b): Spectrogram  $X$ . (c): Spectrogram of the harmonic component  $X_h$  (left), the residual component  $X_r$  (middle) and the percussive component  $X_p$  (right). (d): Waveforms of the harmonic component  $x_h$  (left), the residual component  $x_r$  (middle) and the percussive component  $x_p$  (right).

harmonic sounds have a horizontal structure in a spectrogram representation of the input signal, while percussive sounds form vertical structures. By iteratively diffusing the spectrogram once in horizontal and once in vertical direction, the harmonic and percussive elements are enhanced, respectively. The two enhanced representations are then compared, and entries in the original spectral representation are assigned to either the harmonic or the percussive component according to the dominating enhanced spectrogram. Finally, the two components are transformed back to the time-domain. Following the same idea, Fitzgerald [5] replaces the diffusion step by a much simpler median filtering strategy, which turns out to yield similar results while having a much lower computational complexity.

A drawback of the aforementioned approaches is that the computed decompositions are often not very *tight* in the sense that the harmonic and percussive components may still contain some non-harmonic and non-percussive residues, respectively. This is mainly because of two reasons. First, sounds that are neither of clearly harmonic nor of clearly percussive nature such as applause, rain, or the sound of a heavily distorted guitar are often more or less



randomly distributed among the two components. Second, depending on the parameter setting, harmonic sounds often leak into the percussive component and the other way around. Finding suitable parameters which yield satisfactory results often involves a delicate trade-off between a leakage in one or the other direction.

In this paper, we propose two extensions to [5] that lead towards more flexible and refined decompositions. First, we introduce the concept of a *separation factor* (Section 2). This novel parameter allows for *tightening* decomposition results by enforcing the harmonic and percussive component to contain just the clearly harmonic and percussive sounds of the input signal, respectively, and therefore to attenuate the aforementioned problems. Second, we exploit this concept to add a third *residual* component that captures all sounds in the input audio signal which are neither clearly harmonic nor percussive (see Figure 1). This kind of decomposition is inspired by the classical *sines+transients+noise* (STN) audio model [8, 11] which aims at resynthesizing a given audio signal in terms of a parameterized set of sine waves, transient sounds, and shaped white noise. While a first methodology to compute such a decomposition follows rather straightforward from the concept of a separation factor, we also propose a more involved iterative decomposition procedure. Building on concepts proposed in [13], this procedure allows for a more refined adjustment of the decomposition results (Section 3.3). Finally, we evaluate our proposed procedures based on objective evaluation measures as well as subjective listening tests (Section 4). Note that this paper has an accompanying website [1] where you can find all audio examples discussed in this paper.

## 2. TIGHTENED HARMONIC-PERCUSSIVE SEPARATION

The first steps of our proposed decomposition procedure for tightening the harmonic and the percussive component are the same as in [5], which we now summarize. Given an input audio signal  $x$ , our goal is to compute a harmonic component  $x_h$  and a percussive component  $x_p$  such that  $x_h$  and  $x_p$  contain the clearly harmonic and percussive sounds of  $x$ , respectively. To achieve this goal, first a spectrogram  $X$  of the signal  $x$  is computed by applying a short-time Fourier transform (STFT)

$$X(t, k) = \sum_{n=0}^{N-1} w(n) x(n + tH) \exp(-2\pi i kn/N)$$

with  $t \in [0 : T-1]$  and  $k \in [0 : K]$ , where  $T$  is the number of frames,  $K = N/2$  is the frequency index corresponding to the Nyquist frequency,  $N$  is the frame size and length of the discrete Fourier transform,  $w$  is a sine-window function and  $H$  is the hopsize (we usually set  $H = N/4$ ). A crucial observation is that looking at one frequency band in the magnitude spectrogram  $Y = |X|$  (one row of  $Y$ ), harmonic components stay rather constant, while percussive structures show up as peaks. Contrary, in one frame (one column of  $Y$ ), percussive components tend to be equally

distributed, while the harmonic components stand out. By applying a median filter to  $Y$  once in horizontal and once in vertical direction, we get a harmonically enhanced magnitude spectrogram  $\tilde{Y}_h$  and a magnitude spectrogram  $\tilde{Y}_p$  with enhanced percussive content

$$\begin{aligned} \tilde{Y}_h(t, k) &:= \text{median}(Y(t - \ell_h, k), \dots, Y(t + \ell_h, k)) \\ \tilde{Y}_p(t, k) &:= \text{median}(Y(t, k - \ell_p), \dots, Y(t, k + \ell_p)) \end{aligned}$$

for  $\ell_h, \ell_p \in \mathbb{N}$  where  $2\ell_h + 1$  and  $2\ell_p + 1$  are the lengths of the median filters, respectively.

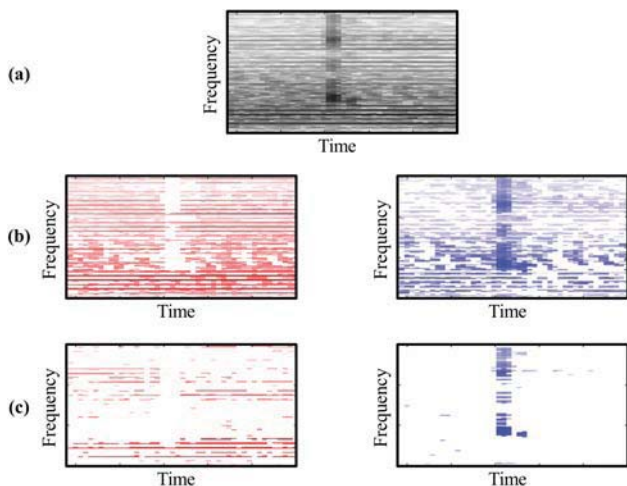
Now, extending [5], we introduce an additional parameter  $\beta \in \mathbb{R}$ ,  $\beta \geq 1$ , called the *separation factor*. We assume an entry of the original spectrogram  $X(t, k)$  to be part of the clearly harmonic or percussive component if  $\tilde{Y}_h(t, k)/\tilde{Y}_p(t, k) > \beta$  or  $\tilde{Y}_p(t, k)/\tilde{Y}_h(t, k) \geq \beta$ , respectively. Intuitively, for a sound to be included in the harmonic component it is required to stand out from the percussive portion of the signal by at least a factor of  $\beta$ , and vice versa for the percussive component. Using this principle, we can define binary masks  $M_h$  and  $M_p$

$$\begin{aligned} M_h(t, k) &:= \left( \tilde{Y}_h(t, k)/(\tilde{Y}_p(t, k) + \epsilon) \right) > \beta \\ M_p(t, k) &:= \left( \tilde{Y}_p(t, k)/(\tilde{Y}_h(t, k) + \epsilon) \right) \geq \beta \end{aligned}$$

where  $\epsilon$  is a small constant to avoid division by zero, and the operators  $\geq$  and  $>$  yield a binary result from  $\{0, 1\}$ . Applying these masks to the original spectrogram  $X$  yields the spectrograms for the harmonic and the percussive component

$$\begin{aligned} X_h(t, k) &:= X(t, k) \cdot M_h(t, k) \\ X_p(t, k) &:= X(t, k) \cdot M_p(t, k) . \end{aligned}$$

These spectrograms can then be brought back to the time-domain by applying an “inverse” short-time Fourier transform, see [7]. This yields the desired signals  $x_h$  and  $x_p$ . Choosing a separation factor  $\beta > 1$  tightens the separation result of the procedure by preventing sounds which are neither clearly harmonic nor percussive to be included in the components. In Figure 2a, for example, you see the spectrogram of a sound mixture of a violin (clearly harmonic), castanets (clearly percussive), and applause (noise-like, and neither harmonic nor percussive). The sound of the violin manifests itself as clear horizontal structures, while one clap of the castanets is visible as a clear vertical structure in the middle of the spectrogram. The sound of the applause however does not form any kind of directed structure and is spread all over the spectrum. When decomposing this audio signal with a separation factor of  $\beta=1$ , which basically yields the procedure proposed in [5], the applause is more or less equally distributed among the harmonic and the percussive component, see Figure 2b. However, when choosing  $\beta=3$ , only the clearly horizontal and vertical structures are preserved in  $X_h$  and  $X_p$ , respectively, and the applause is no longer contained in the two components, see Figure 2c.



**Figure 2.** (a): Original spectrogram  $X$ . (b): Spectrograms  $X_h$  (left) and  $X_p$  (right) for  $\beta = 1$ . (c): Spectrograms  $X_h$  (left) and  $X_p$  (right) for  $\beta = 3$ .

### 3. HARMONIC-PERCUSSIVE-RESIDUAL SEPARATION

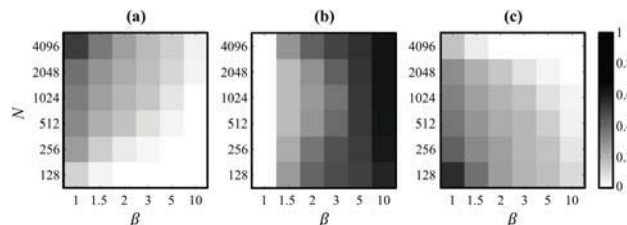
In Section 3.1 we show how harmonic-percussive separation can be extended with a third *residual* component. Afterwards, in Section 3.2, we show how the parameters of the proposed procedure influence the decomposition results. Finally, in Section 3.3, we present an iterative decomposition procedure which allows for a more flexible adjustment of the decomposition results.

#### 3.1 Basic Procedure and Related Work

The concept presented in Section 2 allows us to extend the decomposition procedure with a third component  $x_r$ , called the *residual component*. It contains the portion of the input signal  $x$  that is neither part of the harmonic component  $x_h$  nor the percussive components  $x_p$ . To compute  $x_r$ , we define the binary mask

$$M_r(t, k) := 1 - (M_h(t, k) + M_p(t, k)),$$

apply it to  $X$ , and transform the resulting spectrogram  $X_r$  back to the time-domain (note that the masks  $M_h$  and  $M_p$  are disjoint). This decomposition into three components is inspired by the STN audio model. Here, an audio signal is analyzed to yield parameters for sinusoidal, transient, and noise components which can then be used to approximately resynthesize the original signal [8, 11]. While the main application of the STN model lies in the field of low bitrate audio coding, the estimated parameters can also be used to synthesize just the sinusoidal, the transient, or the noise component of the approximated signal. The harmonic, the percussive, and the residual component resulting from our proposed decomposition procedure are often perceptually similar to the STN components. However, our proposed procedure is conceptually different. STN modeling aims for a *parametrization* of the given audio signal. While the estimated parameters constitute a compact approximation of the input signal, this approximation and



**Figure 3.** Energy distribution between the harmonic, residual, and percussive components for different frame sizes  $N$  and separation factors  $\beta$ . (a): Harmonic components. (b): Residual components. (c): Percussive components.

the original signal are not necessarily equal. Our proposed approach yields a *decomposition* of the signal. The three components always add up to the original signal again. The separation factor  $\beta$  hereby constitutes a flexible handle to adjust the sound characteristics of the components.

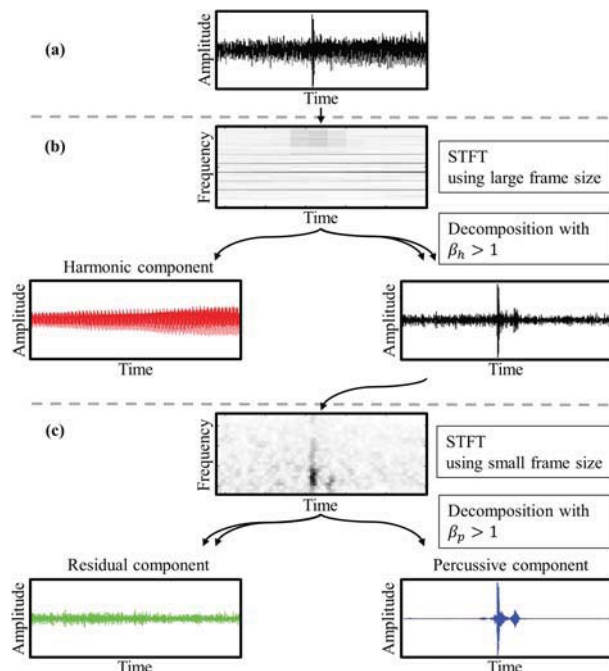
#### 3.2 Influence of the Parameters

The main parameters of our decomposition procedure are the length of the median filters, the frame size  $N$  used to compute the STFT, and the separation factor  $\beta$ . Intuitively, the length of the filters specify the minimal sizes of horizontal and vertical structures which should be considered as harmonic and percussive sounds in the STFT of  $x$ , respectively. Our experiments have shown that the filter lengths actually do not influence the decomposition too much as long as no extreme values are chosen, see also [1]. The frame size  $N$  on the other hand pushes the overall energy of the input signal towards one of the components. For large frame sizes, the short percussive sounds lose influence in the spectral representation and more energy is assigned to the harmonic component. This results in a leakage of some percussive sounds to the harmonic component. Vice versa, for small frame sizes the low frequency resolution often leads to a blurring of horizontal structures, and harmonic sounds tend to leak into the percussive component. The separation factor  $\beta$  shows a different behavior to the previous parameters. The larger its value, the clearer becomes the harmonic and percussive nature of the components  $x_h$  and  $x_p$ . Meanwhile, also the portion of the signal that is assigned to the residual component  $x_r$  increases. To illustrate this behavior, let us consider a first synthetic example where we apply our proposed procedure to the mixture of a violin (clearly harmonic), castanets (clearly percussive), and applause (neither harmonic nor percussive), all sampled at 22050 Hertz and having the same energy. In Figure 3, we visualized the relative energy distribution of the three components for varying frame sizes  $N$  and separation factors  $\beta$ , while fixing the length of the median filters to be always equivalent to 200 milliseconds in horizontal direction and 500 Hertz in vertical direction, see also [1]. Since the energy of all three signals is normalized, potential leakage between the components is indicated by components that have either more or less than a third of the overall energy assigned. Considering Fitzgerald's procedure [5] as a baseline ( $\beta=1$ ), we can investigate

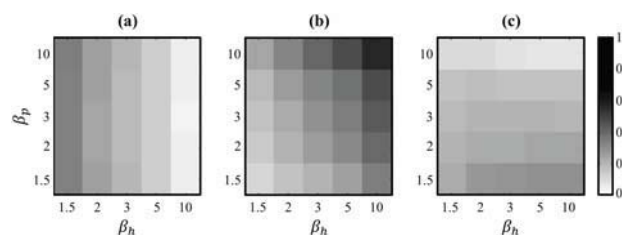
its behavior by looking at the first columns of the matrices in Figure 3. While the residual component has zero energy in this setting, one can observe by listening that the applause is more or less equally distributed between the harmonic and the percussive component for medium frame sizes. This is also reflected in Figure 3a/c by the energy being split up roughly into equal portions. For very large  $N$ , most of the signal's energy moves towards the harmonic component (value close to one in Figure 3a for  $\beta=1, N=4096$ ), while for very small  $N$ , the energy is shifted towards the percussive component (value close to one in Figure 3c for  $\beta=1, N=128$ ). With increasing  $\beta$ , one can observe how the energy gathered in the harmonic and the percussive component flows towards the residual component (decreasing values in Figure 3a/c and increasing values in Figure 3b for increasing  $\beta$ ). Listening to the decomposition results shows that the harmonic and the percussive component thereby become more and more extreme in their respective characteristics. For medium frame sizes, this allows us to find settings that lead to decompositions in which the harmonic component contains the violin, the percussive component contains the castanets, and the residual contains the applause. This is reflected by Figure 3, where for  $N=1024$  and  $\beta=2$  the three sound components all hold roughly one third of the overall energy. For very large or very small frame sizes it is not possible to get such a good decomposition. For example, considering  $\beta=1$  and  $N=4096$ , we already observed that the harmonic component holds most of the signal's energy and also contains some of the percussive sounds. However, already for small  $\beta > 1$  these percussive sounds are shifted towards the residual component (see the large amount of energy assigned to the residual in Figure 3b for  $\beta=1.5, N=4096$ ). Furthermore, also the energy from the percussive component moves towards the residual. The large frame size therefore results in a very clear harmonic component while the residual holds both the percussive as well as all other non-harmonic sounds, leaving the percussive component virtually empty. For very small  $N$  the situation is exactly the other way around. This observation can be exploited to define a refined decomposition procedure which we discuss in the next section.

### 3.3 Iterative Procedure

In [13], Tachibana et al. described a method for the extraction of human singing voice from music recordings. In this algorithm, the singing voice is estimated by iteratively applying the harmonic-percussive decomposition procedure described in [9] first to the input signal and afterwards again to one of the resulting components. This yields a decomposition of the input signal into three components, one of which containing the estimate of the singing voice. The core idea of this algorithm is to perform the two harmonic-percussive separations on spectrograms with two different time-frequency resolutions. In particular, one of the spectrograms is based on a large frame size and the other on a small frame size. Using this idea, we now extend our proposed harmonic-percussive-residual separation procedure



**Figure 4.** Overview of the refined procedure. (a): Input signal  $x$ . (b): First run of the decomposition procedure using a large frame size  $N_h$  and a separation factor  $\beta_h$ . (c): Second run of the decomposition procedure using a small frame size  $N_p$  and a separation factor  $\beta_p$ .



**Figure 5.** Energy distribution between the harmonic, residual, and percussive components for different separation factors  $\beta_h$  and  $\beta_p$ . (a): Harmonic components. (b): Residual components. (c): Percussive components.

presented in Section 3.1. So far, although it is possible to find a good combination of  $N$  and  $\beta$  such that both the harmonic as well as the percussive component represent the respective characteristics of the input signal well (see Section 3.2), the computation of the two components is still coupled. It is therefore not clear how to adjust the content of the harmonic and the percussive component independently. Having made the observation that large  $N$  lead to good harmonic but poor percussive/residual components for  $\beta > 1$ , while small  $N$  lead to good percussive components but poor harmonic/residual components for  $\beta > 1$ , we build on the idea from Tachibana et al. [13] and compute the decomposition in two iterations. Here, the goal is to decouple the computation of the harmonic component from the computation of the percussive component. First, the harmonic component is extracted by applying our basic procedure with a large frame size  $N_h$  and a separation factor  $\beta_h > 1$ , yielding  $x_h^{\text{first}}$ ,  $x_r^{\text{first}}$  and  $x_p^{\text{first}}$ . In a second run,

	SDR						SIR						SAR					
	BL	HP	HP-I	HPR	HPR-I	HPR-IO	BL	HP	HP-I	HPR	HPR-I	HPR-IO	BL	HP	HP-I	HPR	HPR-I	HPR-IO
<b>Violin</b>	-3.10	-5.85	0.08	8.23	7.65	8.85	-3.10	-5.09	1.08	17.69	14.58	21.65	274.25	8.33	9.44	8.82	8.78	9.11
<b>Castanets</b>	-2.93	3.58	2.86	8.29	9.14	9.28	-2.93	6.06	10.45	22.34	20.66	24.41	274.25	8.14	4.07	8.49	9.50	9.44
<b>Applause</b>	-3.04	-	-7.03	4.25	4.93	5.00	-3.04	-	14.69	8.41	12.80	9.04	274.25	-	-6.85	6.95	5.93	7.69

**Table 1.** Objective evaluation measures. All values are given in dB.

the procedure is applied again to the sum  $x_r^{\text{first}} + x_p^{\text{first}}$ , this time using a small frame size  $N_p$  and a second separation factor  $\beta_p > 1$ . This yields the components  $x_h^{\text{second}}$ ,  $x_r^{\text{second}}$  and  $x_p^{\text{second}}$ . Finally, we define the output components of the procedure to be

$$x_h := x_h^{\text{first}}, x_r := x_h^{\text{second}} + x_r^{\text{second}}, x_p := x_p^{\text{second}}.$$

For an overview of the procedure see Figure 4. While fixing the values of  $N_h$  and  $N_p$  to a small and a large frame size, respectively (in our experiments we chose  $N_h=4096$  and  $N_p=256$ ), the separation factors  $\beta_h$  and  $\beta_p$  yield handles that give simple and independent control over the harmonic and percussive component. Figure 5, which is based on the same audio example as Figure 3, shows the energy distribution among the three components for different combinations of  $\beta_h$  and  $\beta_p$ , see also [1]. For the harmonic components (Figure 5a) we see that the portion of the signals energy contained in this component is independent of  $\beta_p$  and can be controlled purely by  $\beta_h$ . This is a natural consequence from the fact that in our proposed procedure the harmonic component is always computed directly from the input signal  $x$  and  $\beta_p$  does not influence its computation at all. However, we can also observe that the energy contained in the percussive component (Figure 5c) is fairly independent of  $\beta_h$  and can be controlled almost solely by  $\beta_p$ . Listening to the decomposition results confirms these observations. Our proposed iterative procedure therefore allows to adjust the harmonic and the percussive component almost independently what significantly simplifies the process of finding an appropriate parameter setting for a given input signal. Note that in principle it would also be possible to choose  $\beta_h=\beta_p=1$ , resulting in an iterative application of Fitzgerald’s method [5]. However, as discussed in Section 3.2, Fitzgerald’s method suffers from component leakage when using very large or small frame sizes. Therefore, most of the input signal’s energy will be assigned to the harmonic component in the first iteration of the algorithm, while most of the remaining portion of the signal is assigned to the percussive component in the second iteration. This leads to a very weak, although not empty, residual component.

#### 4. EVALUATION

In a first experiment, we applied objective evaluation measures to our running example. Assuming that the violin,

the castanets, and the applause signal represent the characteristics that we would like to capture in the harmonic, the percussive, and the residual components, respectively, we treated the decomposition task of this mixture as a source separation problem. In an optimal decomposition the harmonic component would contain the original violin signal, the percussive component the castanets signal, and the residual component the applause. To evaluate the decomposition quality, we computed the *source to distortion ratios* (SDR), the *source to interference ratios* (SIR), and the *source to artifacts ratios* (SAR) [15] for the decomposition results of the following procedures.

As a baseline (BL), we simply considered the original mixture as an estimate for all three sources. Furthermore, we applied the standard harmonic-percussive separation procedure by Fitzgerald [5] (HP) with the frame size set to  $N=1024$ , the HP method applied iteratively (HP-I) with  $N_h=4096$  and  $N_p=256$ , the proposed basic harmonic-percussive-residual separation procedure (HPR) as described in Section 3.1 with  $N=1024$  and  $\beta=2$ , and the proposed iterative harmonic-percussive-residual separation procedure (HPR-I) as described in Section 3.3 with  $N_h=4096$ ,  $N_p=256$ , and  $\beta_h=\beta_p=2$ . As a final method, we also considered HPR-I with separation factor  $\beta_h=3$  and  $\beta_p=2.5$ , which were optimized manually for the task at hand (HPR-IO). The filter lengths in all procedures were always fixed to be equivalent to 200 milliseconds in time direction and 500 Hertz in frequency direction. Decomposition results for all procedures can be found at [1].

The results are listed in Table 1. All values are given in dB and higher values indicate better results. As expected, BL yields rather low SDR and SIR values for all components, while the SAR values are excellent since there are no artifacts present in the original mixture. The method HP yields low evaluation measures as well. However, these values are to be taken with care since HP decomposes the input mixture in just a harmonic and a percussive component. The applause is therefore not estimated explicitly and, as also discussed in Section 2, randomly distributed among the harmonic and percussive component. It is therefore clear that especially the SIR values are low in comparison to the other procedures since the applause heavily interferes with the remaining two sources in the computed components. When looking at HP-I, the benefit of having a third component becomes clear. Although here the residual component does not capture the applause very well (SDR of  $-7.03$  dB) this already suf-

Item name	Description
CastanetsViolinApplause	Synthetic mixture of a violin, castanets and applause.
Heavy	Recording of heavily distorted guitars, a bass and drums.
Stepdad	Excerpt from <i>My Leather, My Fur, My Nails</i> by the band <i>Stepdad</i> .
Bongo	Regular beat played on bongos.
Glockenspiel	Monophonic melody played on a glockenspiel.
Winterreise	Excerpt from “Gute Nacht” by Franz Schubert which is part of the <i>Winterreise</i> song cycle. It is a duet of a male singer and piano.

**Table 2.** List of audio excerpts.

fices to yield SDR and SIR values clearly above the baseline for the estimates of the violin and the castanets. The separation quality further improves when considering the results of our proposed method HPR. Here the evaluation yields high values for all measures and components. The very high SIR values are particularly noticeable since they indicate that the three sources are separated very clearly with very little leakage between the components. This confirms our claim that our proposed concept of a separation factor allows for *tightening* decomposition results as described in Section 2. The results of HPR-I are very similar to the results for the basic procedure HPR. However, listening to the decomposition reveals that the harmonic and the percussive component still contain some slight residue sounds of the applause. Slightly increasing the separation factors to  $\beta_h=3$  and  $\beta_p=2.5$  (HPR-IO) eliminates these residues and further increases the evaluation measures. This straight-forward adjustment is possible since the two separation factors  $\beta_h$  and  $\beta_p$  constitute independent handles to adjust the content of the harmonic and percussive component, what demonstrates the flexibility of our proposed procedure.

The above described experiment constitutes a first case study for the objective evaluation of our proposed decomposition procedures, based on an artificially mixed example. To also evaluate these procedures on real-world audio data, we additionally performed an informal subjective listening tests with several test participants. To this end, we applied our procedures to the set of audio excerpts listed in Table 2. Among the excerpts are complex sound mixtures as well as purely percussive and harmonic signals, see also [1]. Raising the question whether the computed harmonic and percussive components meet the expectation of representing the clearly harmonic or percussive portions of the audio excerpts, respectively, the performed listening test confirmed our hypothesis. It furthermore turned out that  $\beta_h=\beta_p=2$ ,  $N_h=4096$  and  $N_p=256$  seems to be a setting for our iterative procedure which robustly yields good decomposition results, rather independent of the input signal. Regarding the residual component, it was often described to sound like a *sound texture* by the test participants, which is a very interesting observation. Although there is no clear definition of what a sound texture exactly is, literature states “sound texture is like wallpaper: it can have local structure and randomness, but the characteris-

tics of the fine structure must remain constant on the large scale” [12]. In our opinion this is not a bad description of what one can hear in residual components.

### Acknowledgments:

This work has been supported by the German Research Foundation (DFG MU 2686/6-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer IIS.

## 5. REFERENCES

- [1] J. Driedger, M. Müller, and S. Disch. Accompanying website: Extending harmonic-percussive separation of audio signals. <http://www.audiolabs-erlangen.de/resources/2014-ISMIR-ExtHPSep/>.
- [2] J. Driedger, M. Müller, and S. Ewert. Improving time-scale modification of music signals using harmonic-percussive separation. *Signal Processing Letters, IEEE*, 21(1):105–109, 2014.
- [3] C. Duxbury, M. Davies, and M. Sandler. Separation of transient information in audio using multiresolution analysis techniques. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01)*, Limerick, Ireland, 12 2001.
- [4] C. Duxbury, M. Davies, and M. Sandler. Improved time-scaling of musical audio using phase locking at transients. In *Audio Engineering Society Convention 112*, 4 2002.
- [5] D. Fitzgerald. Harmonic/percussive separation using medianfiltering. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 246–253, Graz, Austria, 2010.
- [6] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stafylakis. Music tempo estimation and beat tracking by applying source separation and metrical relations. In *ICASSP*, pages 421–424, 2012.
- [7] D. W. Griffin and J. S. Lim. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 32(2):236–243, 1984.
- [8] S. N. Levine and J. O. Smith III. A sines+transients+noise audio representation for data compression and time/pitch scale modications. In *Proceedings of the 105th Audio Engineering Society Convention*, 1998.
- [9] N. Ono, K. Miyamoto, H. Kameoka, and S. Sagayama. A real-time equalizer of harmonic and percussive components in music signals. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 139–144, Philadelphia, Pennsylvania, USA, 2008.
- [10] N. Ono, K. Miyamoto, J. LeRoux, H. Kameoka, and S. Sagayama. Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram. In *European Signal Processing Conference*, pages 240–244, Lausanne, Switzerland, 2008.
- [11] A. Petrovsky, E. Azarov, and A. Petrovsky. Hybrid signal decomposition based on instantaneous harmonic parameters and perceptually motivated wavelet packets for scalable audio coding. *Signal Processing*, 91(6):1489–1504, 2011.
- [12] N. Saint-Arnaud and K. Popat. Computational auditory scene analysis. chapter Analysis and synthesis of sound textures, pages 293–308. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1998.
- [13] H. Tachibana, N. Ono, and S. Sagayama. Singing voice enhancement in monaural music signals based on two-stage harmonic/percussive sound separation on multiple resolution spectrograms. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):228–237, January 2013.
- [14] Y. Ueda, Y. Uchiyama, T. Nishimoto, N. Ono, and S. Sagayama. HMM-based approach for automatic chord detection using refined acoustic features. In *ICASSP*, pages 5518–5521, 2010.
- [15] E. Vincent, R. Gribonval, and C. Févotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006.



# SINGING VOICE SEPARATION USING SPECTRO-TEMPORAL MODULATION FEATURES

Frederick Yen    Yin-Jyun Luo

Master Program of SMIT  
National Chiao-Tung University, Taiwan  
{fredyen.smt01g, fredom.smt02g}  
@nctu.edu.tw

Tai-Shih Chi

Dept. of Elec. & Comp. Engineering  
National Chiao-Tung University, Taiwan  
tschi@mail.nctu.edu.tw

## ABSTRACT

An auditory-perception inspired singing voice separation algorithm for monaural music recordings is proposed in this paper. Under the framework of computational auditory scene analysis (CASA), the music recordings are first transformed into auditory spectrograms. After extracting the spectral-temporal modulation contents of the time-frequency (T-F) units through a two-stage auditory model, we define modulation features pertaining to three categories in music audio signals: *vocal*, *harmonic*, and *percussive*. The T-F units are then clustered into three categories and the singing voice is synthesized from T-F units in the vocal category via time-frequency masking. The algorithm was tested using the MIR-1K dataset and demonstrated comparable results to other unsupervised masking approaches. Meanwhile, the set of novel features gives a possible explanation on how the auditory cortex analyzes and identifies singing voice in music audio mixtures.

## 1. INTRODUCTION

Over the past decade, the task of singing voice separation has gained much attention due to improvements in digital audio technologies. In the research field of music information retrieval (MIR), separated vocal signals or accompanying music signals can be of great use in many applications, such as singer identification, pitch extraction, and music genre classification. During the past few years, many algorithms have been proposed for this challenging task. These algorithms can be categorized into unsupervised and supervised approaches.

The unsupervised approaches do not contain any training mechanism in the algorithms. For instance, Durrieu et al. used a source/filter signal model with non-negative matrix factorization (NMF) to perform source separation [5] and Fitzgerald et al. used median filtering and factorization techniques to separate harmonic and percussive components in audio signals [7]. Some other unsupervised methods considered structural characteristics of vocals and music accompaniments in several domains for separation. For example, Pardo and Rafii proposed REPET which views the accompaniments as repeating background signals and vocals as the varying information lying on top of them [16]. Tachibana et al. pro-

posed the separation technique, HPSS, to remove the harmonic and percussive instruments sequentially in a two-stage framework by considering the nature of fluctuations of audio signals [19]. Huang et al. used RPCA to present accompaniments in low-rank subspace and vocal in sparse representation [8]. In addition, some unsupervised CASA-based systems were proposed for singing voice separation by finding singing dominant regions on the spectrograms using pitch and harmonic information. For instance, Li and Wang proposed a CASA system obtaining binary masks using pitch-based inference [13]. Hsu and Jang extended the work and proposed a system for separating both voiced and unvoiced singing segments from the music mixtures [9]. Although training mechanisms were seen in these two systems, they were only for detecting voiced and unvoiced segments, but not for separation.

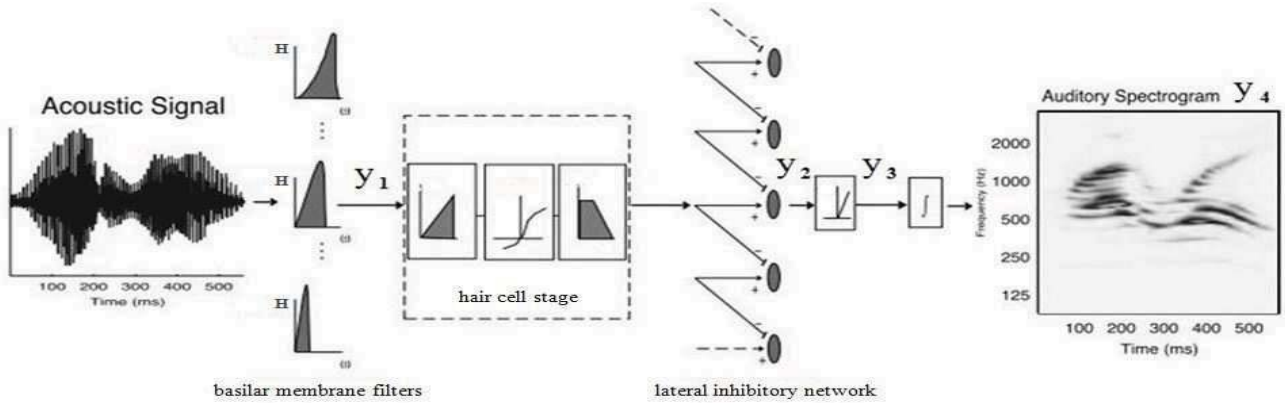
In contrast, there were approaches based on supervised learning techniques. For example, Vembu et al. used vocal/non-vocal SVM and neural-network (NN) classifiers for vocal-nonvocal segmentation [20]. Ozerov et al. used a vocal/non-vocal classifier based on Bayesian modeling [15]. Another group of methods combined RPCA with training mechanisms. For instance, Yang's low-rank representation method decomposed vocals and accompaniments using pre-trained low-rank matrices [22] and Sprechmann et al. proposed a real-time method using low-rank modeling with neural networks [17]. Although these supervised learning methods demonstrated very high performance, they usually offer a weaker conception of generality.

Music instruments produce signals with various kinds of fluctuations such that they can be briefly categorized into two groups, *percussive* and *harmonic*. Signals produced by percussive instruments are more consistent along the spectral axis and by harmonic instruments are more consistent along the temporal axis with little or no fluctuations. These two categories occupy a large proportion of a spectrogram with mainly vertical and horizontal lines. To extend this sense into a more general form, the fluctuations can be viewed as a sum of sinusoid modulations along the spectral axis and the temporal axis. If a signal has nearly zero modulation along one of the two axes, its energy is smoothly distributed along that axis. Conversely, if a signal has a high frequency of modulation along one axis, then its energy becomes scattered along that axis. Therefore, if one can decipher the modulation status of a signal, one may be able to identify the instrument type of the signal. An algorithm utilizing mo-



© Frederick Yen, Yin-Jyun Luo, Tai-Shih Chi.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Frederick Yen, Yin-Jyun Luo, Tai-Shih Chi. "Singing Voice Separation using Spectro-Temporal Modulation Features", 15th International Society for Music Information Retrieval Conference, 2014.



**Figure 1.** Stages of the cochlear module, adopted from [2].

ulation information can be seen in [1], where Barker et al. combined the modulation spectrogram (MS) with non-negative tensor factorization (NTF) to perform speech separation from mixtures of speech and music.

Although the above mentioned engineering approaches produce promising results, human's tremendous ability in sound streams separation makes a biomimetic approach interesting to investigate. Based on neurophysiological evidences, it is suggested that neurons of the auditory cortex (A1) respond to both spectral modulations and temporal modulations of the input sounds. Accordingly, a computational auditory model was proposed to model A1 neurons as spectro-temporal modulation filters [2]. This concept of spectro-temporal modulation decomposition has inspired many approaches in various engineering topics, such as using spectro-temporal modulation features for speaker recognition [12], robust speech recognition [18], voice activity detection [10], and sound segregation [6].

Since modulations are important for music signal categorization, this modulation-decomposition auditory model is used as a pre-processing stage for singing voice separation in this paper. Our proposed unsupervised algorithm adapts this two-stage auditory model, which decodes the spectro-temporal modulations of a T-F unit, to extract modulation based features and performs singing voice separation under the CASA framework. This paper is organized as follows. A brief review of the auditory model is presented in Section 2. Section 3 describes the proposed method. Section 4 shows evaluation and results. Lastly, Section 5 draws the conclusion.

## 2. SPECTRO-TEMPORAL AUDITORY MODEL

A neuro-physiological auditory model is used to extract the modulation features. The model consists of an early cochlear (ear) module and a central auditory cortex (A1) module.

### 2.1 Cochlear Module

As shown in Figure 1, the input sound goes through 128 overlapping asymmetric constant-Q band-pass filters ( $Q_{3dB} \gg 4$ ) whose center frequencies are uniformly distributed

over 5.3 octaves with the 24 filters/octave frequency resolution. These constant-Q filters mimic the frequency selectivity of the cochlea. Outputs of these filters are then transformed through a non-linear compression stage, a lateral inhibitory network (LIN), and a half-wave rectifier cascaded with a low-pass filter. The non-linear compression stage models the saturation caused by inner hair cells, the LIN models the spectral masking effect, and the following stage serves as an envelope extractor to model the temporal dynamic reduction along the auditory pathway to the midbrain. Outputs of the module from different stages are formulated below:

$$y_1(t, \omega) = s(t) *_{\tau} h(t; \omega) \quad (1)$$

$$y_2(t, \omega) = g(\partial_t y_1(t, \omega)) *_{\tau} \ell(t) \quad (2)$$

$$y_3(t, \omega) = \max(\partial_{\omega} y_2(t, \omega), 0) \quad (3)$$

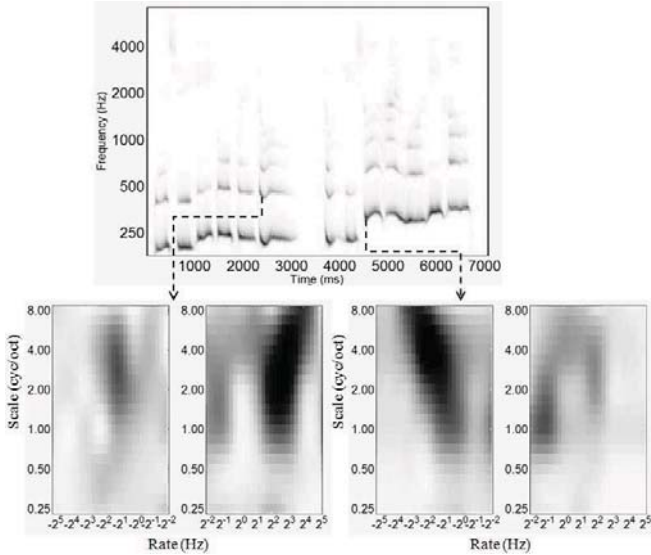
$$y_4(t, \omega) = y_3(t, \omega) *_{\tau} \mu(t; \tau) \quad (4)$$

where  $s(t)$  is the input signal;  $h(t; \omega)$  is the impulse response of the cochlear filter with center frequency  $\omega$ ;  $*_{\tau}$  denotes convolution in time;  $g(\cdot)$  is the nonlinear compression function;  $\partial_t$  is the partial derivative of  $t$ ;  $\ell(t)$  is the membrane leakage low-pass filter;  $\mu(t; \tau) = e^{-t/\tau} \cdot u(t)$  is the integration window with the time constant  $\tau$  to model current leakage of the midbrain;  $u(t)$  is the step function. Detailed descriptions of the cochlear module can be found in [2].

The output  $y_4(t, \omega)$  of the module is the auditory spectrogram, which represents the neuron activities along time and log-frequency axis. In this work, we bypass the non-linear compression stage by assuming input sounds are properly normalized without triggering the high-volume saturation effect of the inner hair cells.

### 2.2 Cortical Module

The second module simulates the neural responses of the auditory cortex (A1). The auditory spectrogram  $y_4(t, \omega)$  is analyzed by cortical neurons which are modeled by two-dimensional filters tuned to different spectro-temporal modulations. The rate parameter (in Hz) characterizes the velocity of local spectro-temporal envelope



**Figure 2.** Rate-scale outputs of the cortical module to two T-F units of the auditory spectrogram of the 'Ani\_2\_03.wav' vocal track in MIR-1K [9].

variation along the temporal axis. The scale parameter (in cycle/octave) characterizes the density of the local spectro-temporal envelope variation along the log-frequency axis. Furthermore, the cortical neurons are found sensitive to the direction of the spectro-temporal envelope. It is characterized by the sign of the rate parameter in this model, with negative for the upward direction and positive for the downward direction.

From functional point of view, this module performs a spectro-temporal multi-resolution analysis on the input auditory spectrogram in various rate-scale combinations. Outputs of various cortical neurons to a single T-F unit of the spectrogram demonstrate the local spectro-temporal modulation contents of the unit in terms of the rate, scale and directionality parameters.

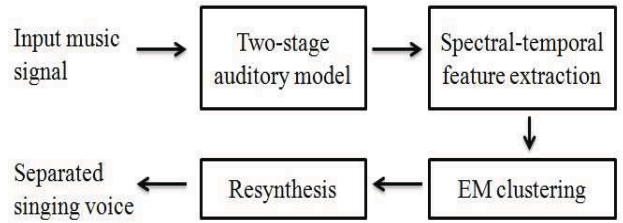
Figure 2 shows rate-scale outputs of two T-F units in an auditory spectrogram of a vocal clip. The rate-scale output is referred to as the rate-scale plot in this paper. The rate and scale indices are  $\pm 2^{-2} \sim \pm 2^5$  and  $2^{-2} \sim 2^3$ , respectively. The strong responses of the plots correspond to the variations of singing pitch envelopes resolved by the rate and scale parameters and the moving direction of the pitch. Detailed description of the cortical module is available in [3].

### 3. PROPOSED METHOD

A schematic diagram of the proposed algorithm is shown in Figure 3. The following sections will discuss each part in details.

#### 3.1 Feature Extraction

According to the spectral and temporal behaviors observed on the auditory spectrogram, components of a musical piece are characterized into three categories,

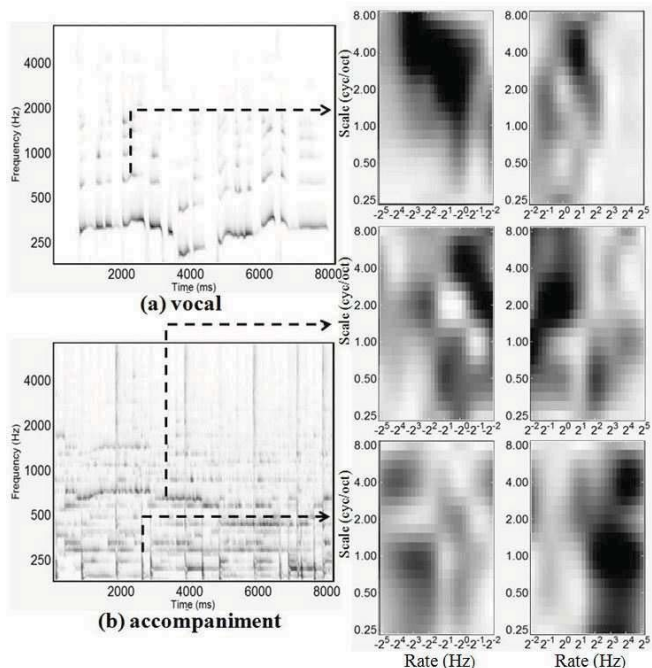


**Figure 3.** Block diagram of the proposed algorithm.

*harmonic*, *percussive* and *vocal*. Harmonic components have steady energy distributions over time and have clear formant structures over frequency. Each percussive component has impulsive energy concentrated in a short period of time and has no obvious harmonic structure. Vocal components possess harmonic structure and their energy is distributed along various time periods. Interpreting the above statements from the rate-scale perspective, several general properties can be drawn. Harmonic components can be usually regarded as having low rate and high scale modulations. It means that they have relatively slow energy change along time and rapid energy change along the log-frequency axis due to the harmonic structures. In contrast, percussive components typically show quick energy change along time and energy spreading along the whole log-frequency axis, such that they possess high rate and low scale modulations. Vocal components are often recognized as a mix version of the harmonic and percussive components with characteristics sometimes considered more similar to harmonics. Different types of singing or vocal expression can result in various values of rate and scale. Figure 4 shows some examples of rate-scale plots of components from the three categories.

Given an auditory spectrogram  $y_4 \in \mathcal{R}^{m \times n}$  transformed from an input music signal  $s(t)$ , the rate-scale plots of the T-F units are generated. As a pre-process, in order to prevent extracting trivial data from nearly inaudible T-F units of the auditory spectrogram, we leave out the T-F units that have energy less than 1% of the maximum energy of the whole auditory spectrogram. With the rest of the T-F units, we obtain the rate-scale plot of each unit and proceed to the feature extraction stage.

For each rate-scale plot, the total energies of the negative and positive rate side are compared. The side with greater energy is determined as the dominant plot. From the dominant plot, we extract 11 features as shown in Table 1. The features are selected by observing the rate-scale plots with some intuitive assumptions of the physical properties which distinguish between harmonic, percussive and vocal. The first 10 features are obtained by computing the energy ratio of two different areas on the rate-scale plot. For example, as shown in Table 1, the first feature is the ratio of the total modulation energy of scale = 1 to the total modulation energy of scale = 0.25. The low scales, such as 0.25 and 0.5, capture the degree of the



**Figure 4.** (a) Rate-scale plot from the vocal track of ‘Ani\_4\_07’ in MIR-1K. The modulation energy is mostly concentrated in the middle and high scales for a unit with a clear harmonic structure. (b) Rate-scale plots from the accompanying music track of ‘Ani\_4\_07’. The upper plot shows energy concentrating at low rates for a sustained unit. The lower plot shows energy concentrating at high rates for a transient unit.

flatness of the formant structure while the high scales, such as 1, 2, 4 and 8, capture the harmonicity with different frequency spacing between harmonics. Therefore, the first four features can be thought as descriptors which distinguish harmonic from percussive using spectral information. The fifth to the seventh features capture temporal information which can distinguish sustained units from transient units.

The feature values are saved as feature vectors and then grouped as a feature matrix  $F \in \mathcal{R}^{\ell \times i}$  for clustering, where  $\ell$  is the number of features and  $i$  is the number of total valid units in the auditory spectrogram.

### 3.2 Unsupervised Clustering

In the unsupervised clustering stage, a spectrogram is divided into three parts and clustering is performed for each part. Based on hearing perception, the frequency resolution is higher at lower frequencies while the temporal resolution is higher at higher frequencies [14]. Due to the frequency resolution of the constant-Q cochlear filters/channels in the auditory model, the auditory spectrogram can only resolve about ten harmonics [11]. To handle different resolutions, the spectrogram is separated into three sub-spectrograms with overlapped frequency ranges. The three sub-spectrograms consist of channel 1 to channel 60, channel 46 to channel 75, and channel 61 to channel 128, respectively, with overlaps of 15 channels.

Scale	Rate
1 : 0.25	all
2 : 0.25	all
4 : 0.25	all
8 : 0.25	all
(0.25, 2, 4)	(1, 2) : (0.25, 0.5, 1, 2, 16, 32)
(0.25, 2, 4)	(0.25, 0.5) : (0.25, 0.5, 1, 2, 16, 32)
(0.25, 2, 4)	(16, 32) : (0.25, 0.5, 1, 2, 16, 32)
(0.25, 0.5) : all	all
(1, 2) : all	all
(4, 8) : all	all
(0.25)	all

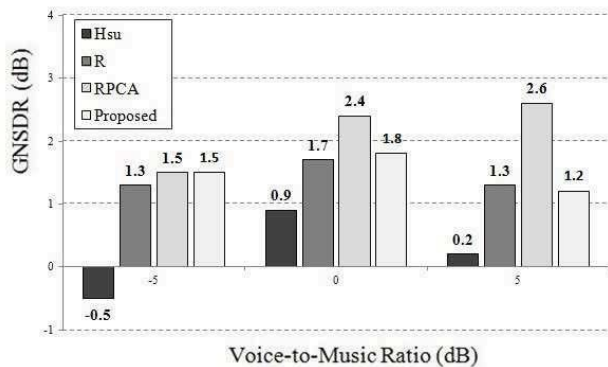
**Table 1.** Eleven extracted modulation energy features

The clustering step is performed using the EM algorithm to group data into three unlabelled clusters. The EM algorithm assigns a probability set to each T-F unit showing its likelihood of belonging to each cluster. Note that in spectrogram representations, the sound sources are superimposed on top of each other. It implies that one T-F unit may contain energy from more than one source. Therefore, in this work, if one T-F unit has a probability set in which the second highest probability is higher than 5%, that particular T-F unit will also be labelled to the second high probability cluster. It means one unit may eventually appear in more than one cluster. The parameter 5% was empirically determined. Each of the three sub-spectrograms is clustered into three groups. Total of nine groups are generated and merged back into three whole spectrograms by comparing the correlations of the overlapped channels between different groups. Each of the three whole spectrograms represents the extracted harmonic, percussive, and vocal part of the music mixture. With no prior information about the labels of the three whole spectrograms, the effective mean rate-scale plot of each spectrogram is examined. The effective mean rate-scale plot is the mean of rate-scale plots of the T-F units with energy higher than 20% of the maximum energy in that spectrogram. The total modulation energy of rate = 1, 2 Hz and scale = 0.25, 2, 4 cycle/octave is calculated from the effective mean rate-scale plot and referred to as  $E_v$ , which is used as the criterion to select the vocal spectrogram. The one with the maximum  $E_v$  value is picked as the vocal spectrogram since  $E_v$  catches modulations related to the formant structure (scale = 0.25), the harmonic structure (scale = 2 and 4) and the singing rate (rate = 1 and 2) of singing voices.

The vocal spectrogram is then synthesized to an estimated signal using the auditory model toolbox [24]. The nonlinear operation of the envelope extractor in the cochlear module makes perfect synthesis impossible, thus causing a general result of loss of higher frequencies of the signal. Detailed computations are shown in [2].

## 4. EVALUATION RESULTS

The MIR-1K [9] is used as the evaluation dataset. It cont-



**Figure 5.** GNSDR comparison at voice-to-music ratio of -5, 0, and 5 dB with existing methods.

ains 1000 WAV files of karaoke clips sung by amateur singers. The length of each clip is around 4~13 seconds. The vocal and music accompaniment parts were recorded in the right and the left channels separately. In this experiment, we mixed two channels in -5, 0, 5 dB SNR (signal to noise ratio, i.e., vocal to music accompaniment ratio) for test. To assess the quality of separation, the source-to-distortion ratio (SDR) [21] is used as the objective measure. The ratios are computed by the BSS Eval toolbox v3.0 [23]. Following [9], we compute the normalized SDR (NSDR) and the weighted average of NSDR, the global NSDR (GNSDR), with the weighting proportional to the length of each file. To have a fair comparison, we compare our method with other unsupervised methods, which extract vocal clips only through one major stage. The compared algorithms are listed below:

- I. **Hsu**: the approach proposed in [9] that performs unvoiced sound separation combined with the pitch-based inference method in [13].
- II. **R** (REPET with soft masking): the approach proposed in [16] that computes a repeating background structure and extract vocal with soft time-frequency masking.
- III. **RPCA**: a matrix decomposition method applying robust principal component analysis proposed by Huang et al. [8].

From Figure 5, we can observe that the proposed method has the highest performance tied with RPCA in the -5 dB SNR condition. In 0 and 5 dB SNR conditions, the performance of the proposed method is comparable to the performance of REPET.

## 5. CONCLUSION

In this paper, we propose a singing voice separation method utilizing the spectral-temporal modulations as clustering features. Based on the energy distributions on the rate-scale plots of T-F units, the vocal signal is extracted from the auditory spectrogram and the separation performance is evaluated using the MIR-1K dataset. Our proposed CASA-based masking method outperforms the CASA-based system in [9] and has comparable perfor-

mance to the masking-based REPET in all SNR conditions. When compared with the subspace RPCA method, our proposed method has comparable performance only in the -5 dB SNR condition. These results demonstrate the effectiveness of the spectral-temporal modulation features for analyzing music mixtures. As this proposed method only applies a simple EM algorithm for clustering, harmonic mismatches and artificial noises are yet to be discussed.

The future work will be focused on applying more advanced classifiers for more accurate separations and adopting a two-stage mechanism like HPSS to discard percussive and harmonic components sequentially. The other potential work is to implement the proposed spectro-temporal modulation based method in the Fourier spectrogram domain [4] to mitigate synthesis errors injected by the projection-based reconstruction process of the auditory model.

## 6. ACKNOWLEDGEMENTS

This research is supported by the National Science Council, Taiwan under Grant No NSC 102-2220-E-009-049 and the Biomedical Electronics Translational Research Center, NCTU.

## 7. REFERENCES

- [1] T. Barker and T. Virtanen, "Non-negative tensor factorization of modulation spectrograms for monaural sound source separation," *Proc. of Interspeech*, pp. 827-831, 2013.
- [2] T. Chi, P. Ru, and S. A. Shamma, "Multiresolution spectrotemporal analysis of complex sounds," *J. Acoust. Soc. Am.*, Vol. 118, No. 2, pp. 887-906, 2005.
- [3] T. Chi, Y. Gao, M. C. Guyton, P. Ru, and S. Shamma, "Spectro-temporal modulation transfer functions and speech intelligibility," *J. Acoust. Soc. Am.*, Vol. 106, No. 5, pp. 2719-2732, 1999.
- [4] T.-S. Chi and C.-C. Hsu, "Multiband analysis and synthesis of spectro-temporal modulations of Fourier spectrogram," *J. Acoust. Soc. Am.*, Vol. 129, No. 5, pp. EL190-EL196, 2011.
- [5] J.-L. Durrieu, B. David, and G. Richard, "A musically motivated mid-level representation for pitch estimation and musical audio source separation," *IEEE J. of Selected Topics on Signal Process.*, Vol. 5, No. 6, pp. 1180-1191, 2011.
- [6] M. Elhilali and S. A. Shamma, "A cocktail party with a cortical twist: how cortical mechanisms contribute to sound segregation," *J. Acoust. Soc. Am.*, Vol. 124, No. 6, pp. 3751-3771, 2008.
- [7] D. FitzGerald and M. Gainza, "Single channel vocal separation using median filtering and factorization techniques," *ISAST Trans. on Electron. and Signal Process.*, Vol. 4, No. 1, pp. 62-73 (ISSN 1797-2329), 2010.

- [8] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Process.*, pp. 57-60, 2012.
- [9] C.-L. Hsu and J.-S. R. Jang, "On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset," *IEEE Trans. on Audio, Speech, and Language Process.*, Vol. 18, No. 2, pp. 310-319, 2010.
- [10] C.-C. Hsu, T.-E. Lin, J.-H. Chen, and T.-S. Chi, "Voice activity detection based on frequency modulation of harmonics," *IEEE Int. Conf. on Acoust., Speech and Signal Process.*, pp. 6679-6683, 2013.
- [11] D. Klein, and S. A. Shamma, "The case of the missing pitch templates: how harmonic templates emerge in the early auditory system," *J. Acoust. Soc. Am.*, Vol. 107, No. 5, pp. 2631-2644, 2000.
- [12] H. Lei, B. T. Meyer, and N. Mirghafori, "Spectro-temporal Gabor features for speaker recognition," *IEEE Int. Conf. on Acoust., Speech and Signal Process.*, pp. 4241-4244, 2012.
- [13] Y. Li and D. Wang, "Separation of singing voice from music accompaniment for monaural recordings," *IEEE Trans. on Audio, Speech, and Language Process.*, Vol. 15, No. 4, pp. 1475-1487, 2007.
- [14] B. C. J. Moore: *An Introduction to the Psychology of Hearing 5<sup>th</sup> Ed.*, Academic Press, 2003.
- [15] A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval, "Adaptation of Bayesian models for single channel source separation and its application to voice / music separation in popular songs," *IEEE Trans. on Audio, Speech, and Language Process.*, special issue on Blind Signal Proc. for Speech and Audio Applications, Vol. 15, No. 5, pp. 1564-1578, 2007.
- [16] Z. Rafii and B. Pardo, "Repeating Pattern Extraction Technique (REPET): A Simple Method for Music/Voice Separation," *IEEE Trans. on Audio, Speech, and Language Process.*, Vol. 21, No. 1, pp. 73-84, 2013.
- [17] P. Sprechmann, A. Bronstein, and G. Sapiro, "Real-time online singing voice separation from monaural recordings using robust low-rank modeling," *Proc. of the Int. Soc. for Music Inform. Retrieval Conf.*, pp. 67-72, 2012.
- [18] R. M. Stern and N. Norgan, "Hearing is believing: biologically inspired methods for robust automatic speech recognition," *IEEE Signal Process. Mag.*, Vol. 29, No. 6, pp. 34-43, 2012.
- [19] H. Tachibana, N. Ono, and S. Sagayama, "Singing Voice Enhancement in Monaural Music Signals Based on Two-stage Harmonic/Percussive Sound Separation on Multiple Resolution Spectrograms," *IEEE/ACM Trans. on Audio, Speech, and Language Process.*, Vol. 22, No. 1, pp. 228-237, 2014.
- [20] S. Vembu and S. Baumann, "Separation of vocals from polyphonic audio recordings," *Proc. of the Int. Soc. for Music Inform. Retrieval Conf.*, pp. 337-344, 2005.
- [21] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. on Audio, Speech, and Language Process.*, Vol. 14, No. 4, pp. 1462-1469, 2006.
- [22] Y. Yang, "Low-rank representation of both singing voice and music accompaniment via learned dictionaries," *Proc. of the Int. Soc. for Music Inform. Retrieval Conf.*, pp. 427-432, 2013.
- [23] [http://bass-db.gforge.inria.fr/bss\\_eval/](http://bass-db.gforge.inria.fr/bss_eval/)
- [24] <http://www.isr.umd.edu/Labs/NSL/nsl.html>

# HARMONIC-TEMPORAL FACTOR DECOMPOSITION INCORPORATING MUSIC PRIOR INFORMATION FOR INFORMED MONAURAL SOURCE SEPARATION

Tomohiko Nakamura<sup>†</sup>, Kotaro Shikata<sup>†</sup>, Norihiro Takamune<sup>†</sup>, Hirokazu Kameoka<sup>†‡</sup>

<sup>†</sup>Graduate School of Information Science and Technology, The University of Tokyo.

<sup>‡</sup>NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation.  
{nakamura,k-shikata,takamune,kameoka}@hil.t.u-tokyo.ac.jp

## ABSTRACT

For monaural source separation two main approaches have thus far been adopted. One approach involves applying non-negative matrix factorization (NMF) to an observed magnitude spectrogram, interpreted as a non-negative matrix. The other approach is based on the concept of computational auditory scene analysis (CASA). A CASA-based approach called the “harmonic-temporal clustering (HTC)” aims to cluster the time-frequency components of an observed signal based on a constraint designed according to the local time-frequency structure common in many sound sources (such as harmonicity and the continuity of frequency and amplitude modulations). This paper proposes a new approach for monaural source separation called the “Harmonic-Temporal Factor Decomposition (HTFD)” by introducing a spectrogram model that combines the features of the models employed in the NMF and HTC approaches. We further describe some ideas how to design the prior distributions for the present model to incorporate musically relevant information into the separation scheme.

## 1. INTRODUCTION

Monaural source separation is a process in which the signals of concurrent sources are estimated from a monaural polyphonic signal and is one of fundamental objectives offering a wide range of applications such as music information retrieval, music transcription and audio editing.

While we can use spatial cues for blind source separation with multichannel inputs, for monaural source separation we need other cues instead of the spatial cues. For monaural source separation two main approaches have thus far been adopted. One approach is based on the concept of computational auditory scene analysis (e.g., [7]). The auditory scene analysis process described by Bregman [1] involves grouping elements that are likely to have originated from the same source into a perceptual structure called an auditory stream. In [8, 10], an attempt has been made to imitate this process by clustering time-frequency components based on a constraint designed according to the auditory grouping cues (such as the har-

monicity and the coherences and continuities of amplitude and frequency modulations). This method is called “harmonic-temporal clustering (HTC).”

The other approach involves applying non-negative matrix factorization (NMF) to an observed magnitude spectrogram (time-frequency representation) interpreted as a non-negative matrix [19]. The idea behind this approach is that the spectrum at each frame is assumed to be represented as a weighted sum of a limited number of common spectral templates. Since the spectral templates and the mixing weights should both be non-negative, this implies that an observed spectrogram is modeled as the product of two non-negative matrices. Thus, factorizing an observed spectrogram into the product of two non-negative matrices allows us to estimate the unknown spectral templates constituting the observed spectra and decompose the observed spectra into components associated with the estimated spectral templates.

The two approaches described above rely on different clues for making separation possible. Roughly speaking, the former approach focuses on the local time-frequency structure of each source, while the latter approach focuses on a relatively global structure of music spectrograms (such a property that a music signal typically consists of a limited number of recurring note events). Rather than discussing which clues are more useful, we believe that both of these clues can be useful for achieving a reliable monaural source separation algorithm. This belief has led us to develop a new model and method for monaural source separation that combine the features of both HTC and NMF. We call the present method “harmonic-temporal factor decomposition (HTFD).”

The present model is formulated as a probabilistic generative model in such a way that musically relevant information can be flexibly incorporated into the prior distributions of the model parameters. Given the recent progress of state-of-the-art methods for a variety of music information retrieval (MIR)-related tasks such as audio key detection, audio chord detection, and audio beat tracking, information such as key, chord and beat extracted from the given signal can potentially be utilized as reliable and useful prior information for source separation. The inclusion of auxiliary information in the separation scheme is referred to as informed source separation and is gaining increasing momentum in recent years (see e.g., among others, [5, 15, 18, 20]). This paper further describes some ideas how to design the prior distributions for the present model to incorporate musically relevant information.

We henceforth denote the normal, Dirichlet and Poisson



© Tomohiko Nakamura<sup>†</sup>, Kotaro Shikata<sup>†</sup>, Norihiro Takamune<sup>†</sup>, Hirokazu Kameoka<sup>†‡</sup>.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Tomohiko Nakamura<sup>†</sup>, Kotaro Shikata<sup>†</sup>, Norihiro Takamune<sup>†</sup>, Hirokazu Kameoka<sup>†‡</sup>. “Harmonic-temporal factor decomposition incorporating music prior information for informed monaural source separation”, 15th International Society for Music Information Retrieval Conference, 2014.

distributions by  $\mathcal{N}$ , Dir and Pois, respectively.

## 2. SPECTROGRAM MODEL OF MUSIC SIGNAL

### 2.1 Wavelet transform of source signal model

As in [8], this section derives the continuous wavelet transform of a source signal. Let us first consider as a signal model for the sound of the  $k$ th pitch the analytic signal representation of a pseudo-periodic signal given by

$$f_k(u) = \sum_{n=1}^N a_{k,n}(u) e^{j(n\theta_k(u) + \varphi_{k,n})}, \quad (1)$$

where  $u$  denotes the time,  $n\theta_k(u) + \varphi_{k,n}$  the instantaneous phase of the  $n$ -th harmonic and  $a_{k,n}(u)$  the instantaneous amplitude. This signal model implicitly ensures not to violate the ‘harmonicity’ and ‘coherent frequency modulation’ constraints of the auditory grouping cues. Now, let the wavelet basis function be defined by

$$\psi_{\alpha,t}(u) = \frac{1}{\sqrt{2\pi\alpha}} \psi\left(\frac{u-t}{\alpha}\right), \quad (2)$$

where  $\alpha$  is the scale parameter such that  $\alpha > 0$ ,  $t$  the shift parameter and  $\psi(u)$  the mother wavelet with the center frequency of 1 satisfying the admissibility condition.  $\psi_{\alpha,t}(u)$  can thus be used to measure the component of period  $\alpha$  at time  $t$ . The continuous wavelet transform of  $f_k(u)$  is then defined by

$$W_k(\log \frac{1}{\alpha}, t) = \int_{-\infty}^{\infty} \sum_{n=1}^N a_{k,n}(u) e^{j(n\theta_k(u) + \varphi_{k,n})} \psi_{\alpha,t}^*(u) du. \quad (3)$$

Since the dominant part of  $\psi_{\alpha,t}^*(u)$  is typically localized around time  $t$ , the result of the integral in Eq. (3) shall depend only on the values of  $\theta_k(u)$  and  $a_{k,n}(u)$  near  $t$ . By taking this into account, we replace  $\theta_k(t)$  and  $a_{k,n}(t)$  with zero- and first-order approximations around time  $t$ :

$$a_{k,n}(u) \simeq a_{k,n}(t), \quad \theta_k(u) \simeq \theta_k(t) + \dot{\theta}_k(t)(u-t). \quad (4)$$

Note that the variable  $\dot{\theta}_k(t)$  corresponds to the instantaneous fundamental frequency ( $F_0$ ). By undertaking the above approximations, applying the Parseval’s theorem, and putting  $x = \log(1/\alpha)$  and  $\Omega_k(t) = \log \dot{\theta}_k(t)$ , we can further write Eq. (3) as

$$W_k(x, t) = \sum_{n=1}^N a_{k,n}(t) \Psi^*(n e^{-x + \Omega_k(t)}) e^{j(n\theta_k(t) + \varphi_{k,n})}, \quad (5)$$

where  $x$  denotes log-frequency and  $\Psi$  the Fourier transform of  $\psi$ . Since the function  $\Psi$  can be chosen arbitrarily, as with [8], we employ the following unimodal real function whose maximum is taken at  $\omega = 1$ :

$$\Psi(\omega) = \begin{cases} e^{-\frac{(\log \omega)^2}{4\sigma^2}} & (\omega > 0) \\ 0 & (\omega \leq 0) \end{cases}. \quad (6)$$

Eq. (5) can then be written as

$$W_k(x, t) = \sum_{n=1}^N a_{k,n}(t) e^{-\frac{(x - \Omega_k(t) - \log n)^2}{4\sigma^2}} e^{j(n\theta_k(t) + \varphi_{k,n})}. \quad (7)$$

If we now assume that the time-frequency components are sparsely distributed so that the partials rarely overlap each other,  $|W_k(x, t)|^2$  is given approximately as

$$|W_k(x, t)|^2 \simeq \sum_{n=1}^N |a_{k,n}(t)|^2 e^{-\frac{(x - \Omega_k(t) - \log n)^2}{2\sigma^2}}. \quad (8)$$

This assumption means that the power spectra of the partials can approximately be considered additive. Note that a cutting plane of the spectrogram model given by Eq. (8)

at time  $t$  is expressed as a harmonically-spaced Gaussian mixture function. If we assume the additivity of power spectra, the power spectrogram of a superposition of  $K$  pitched sounds is given by the sum of Eq. (8) over  $k$ . It should be noted that this model is identical to the one employed in the HTC approach [8].

Although we have defined the spectrogram model above in continuous time and continuous log-frequency, we actually obtain observed spectrograms as a discrete time-frequency representation through computer implementations. Thus, we henceforth use  $Y_{l,m} := Y(x_l, t_m)$  to denote an observed spectrogram where  $x_l$  ( $l = 1, \dots, L$ ) and  $t_m$  ( $m = 1, \dots, M$ ) stand for the uniformly-quantized log-frequency points and time points, respectively. We will also use the notation  $\Omega_{k,m}$  and  $a_{k,n,m}$  to indicate  $\Omega_k(t_m)$  and  $a_{k,n}(t_m)$ .

### 2.2 Incorporating source-filter model

The generating processes of many sound sources in real world can be explained fairly well by the source-filter theory. In this section, we follow the idea described in [12] to incorporate the source-filter model into the above model. Let us assume that each signal  $f_k(u)$  within a short-time segment is an output of an all-pole system. That is, if we use  $f_{k,m}[i]$  to denote the discrete-time representation of  $f_k(u)$  within a short-time segment centered at time  $t_m$ ,  $f_{k,m}[i]$  can be described as

$$\beta_{k,m}[0] f_{k,m}[i] = \sum_{p=1}^P \beta_{k,m}[p] f_{k,m}[i-p] + \epsilon_{k,m}[i], \quad (9)$$

where  $i$ ,  $\epsilon_{k,m}[i]$ , and  $\beta_{k,m}[p]$  ( $p = 0, \dots, P$ ) denote the discrete-time index, an excitation signal, and the autoregressive (AR) coefficients, respectively. As we have already assumed in 2.1 that the  $F_0$  of  $f_{k,m}[i]$  is  $e^{\Omega_{k,m}}$ , to make the assumption consistent, the  $F_0$  of the excitation signal  $\epsilon_{k,m}[i]$  must also be  $e^{\Omega_{k,m}}$ . We thus define  $\epsilon_{k,m}[i]$  as

$$\epsilon_{k,m}[i] = \sum_{n=1}^N v_{k,n,m} e^{j n e^{\Omega_{k,m}} i u_0}, \quad (10)$$

where  $u_0$  denotes the sampling period of the discrete-time representation and  $v_{k,n,m}$  denotes the complex amplitude of the  $n$ th partial. By applying the discrete-time Fourier transform (DTFT) to Eq. (9) and putting  $B_{k,m}(z) := \beta_{k,m}[0] - \beta_{k,m}[1]z^{-1} - \dots - \beta_{k,m}[P]z^{-P}$ , we obtain

$$F_{k,m}(\omega) = \frac{\sqrt{2\pi}}{B_{k,m}(e^{j\omega})} \sum_{n=1}^N v_{k,n,m} \delta(\omega - n e^{\Omega_{k,m}} u_0), \quad (11)$$

where  $F_{k,m}$  denotes the DTFT of  $f_{k,m}$ ,  $\omega$  the normalized angular frequency, and  $\delta$  the Dirac delta function. The inverse DTFT of Eq. (11) gives us another expression of  $f_{k,m}[i]$ :

$$f_{k,m}[i] = \sum_{n=1}^N \frac{v_{k,n,m}}{B_{k,m}(e^{j n e^{\Omega_{k,m}} u_0})} e^{j n e^{\Omega_{k,m}} i u_0}. \quad (12)$$

By comparing Eq. (12) and the discrete-time representation of Eq. (1), we can associate the parameters of the source filter model defined above with the parameters introduced in 2.1 through the explicit relationship:

$$|a_{k,n,m}| = \left| \frac{v_{k,n,m}}{B_{k,m}(e^{j n e^{\Omega_{k,m}} u_0})} \right|. \quad (13)$$

### 2.3 Constraining model parameters

The key assumption behind the NMF model is that the spectra of the sound of a particular pitch is expressed as



a multiplication of time-independent and time-dependent factors. In order to extend the NMF model to a more reasonable one, we consider it important to clarify which factors involved in the spectra should be assumed to be time-dependent and which factors should not. For example, the  $F_0$  must be assumed to vary in time during vibrato or portamento. Of course, the scale of the spectrum should also be assumed to be time-varying (as with the NMF model). On the other hand, the timbre of an instrument can be considered relatively static throughout an entire piece of music.

We can reflect these assumptions in the present model in the following way. For convenience of the following analysis, we factorize  $|a_{k,n,m}|$  into the product of two variables,  $w_{k,n,m}$  and  $U_{k,m}$

$$|a_{k,n,m}| = w_{k,n,m} \sqrt{U_{k,m}}. \quad (14)$$

$w_{k,n,m}$  can be interpreted as the relative power of the  $n$ th harmonic and  $U_{k,m}$  as the time-varying normalized amplitude of the sound of the  $k$ th pitch such that  $\sum_{k,m} U_{k,m} = 1$ . In the same way, let us put  $v_{k,n,m}$  as

$$v_{k,n,m} = \tilde{w}_{k,n,m} \sqrt{U_{k,m}}. \quad (15)$$

Since the all-pole spectrum  $1/|B_{k,m}(e^{j\omega})|^2$  is related to the timbre of the sound of the  $k$ th pitch, we want to constrain it to be time-invariant. This can be done simply by eliminating the subscript  $m$ . Eq. (13) can thus be rewritten as

$$w_{k,n,m} = \left| \frac{\tilde{w}_{k,n,m}}{B_k(e^{j\omega_{k,m} u_0})} \right|. \quad (16)$$

We can use  $\Omega_{k,m}$  as is, since it is already dependent on  $m$ .

To sum up, we obtain a spectrogram model  $X_{l,m}$  as

$$X_{l,m} = \sum_{k=1}^K C_{k,l,m}, \quad C_{k,l,m} = \underbrace{\left( \sum_{n=1}^N w_{k,n,m}^2 e^{-\frac{(x_l - \Omega_{k,m} - \log n)^2}{2\sigma^2}} \right)}_{H_{k,l,m}} U_{k,m}, \quad (17)$$

where  $C_{k,l,m}$  stands for the spectrogram of the  $k$ th pitch. If we denote the term inside the parenthesis by  $H_{k,l,m}$ ,  $X_{l,m}$  can be rewritten as  $X_{l,m} = \sum_k H_{k,l,m} U_{k,m}$  and so the relation to the NMF model may become much clearer.

## 2.4 Formulating probabilistic model

Since the assumptions and approximations we made so far do not always hold exactly in reality, an observed spectrogram  $Y_{l,m}$  may diverge from  $X_{l,m}$  even though the parameters are optimally determined. One way to simplify the process by which this kind of deviation occurs would be to assume a probability distribution of  $Y_{l,m}$  with the expected value of  $X_{l,m}$ . Here, we assume that  $Y_{l,m}$  follows a Poisson distribution with mean  $X_{l,m}$

$$Y_{l,m} \sim \text{Pois}(Y_{l,m}; X_{l,m}), \quad (18)$$

where  $\text{Pois}(z; \xi) = \xi^z e^{-\xi} / \Gamma(z)$ . This defines our likelihood function

$$p(\mathbf{Y}|\boldsymbol{\theta}) = \prod_{l,m} \text{Pois}(Y_{l,m}; X_{l,m}), \quad (19)$$

where  $\mathbf{Y}$  denotes the set consisting of  $Y_{l,m}$  and  $\boldsymbol{\theta}$  the entire set consisting of the unknown model parameters. It should be noted that the maximization of the Poisson likelihood with respect to  $X_{l,m}$  amounts to optimally fitting  $X_{l,m}$  to  $Y_{l,m}$  by using the I-divergence as the fitting criterion.

Eq. (16) implicitly defines the conditional distribution

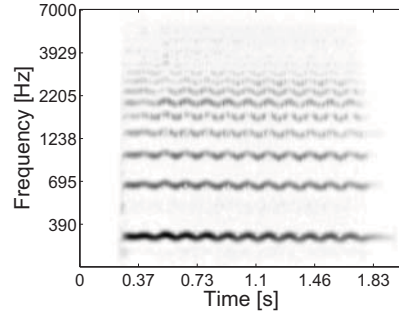


Figure 1. Power spectrogram of a violin vibrato sound.

$p(\mathbf{w}|\tilde{\mathbf{w}}, \boldsymbol{\beta}, \boldsymbol{\Omega})$  expressed by the Dirac delta function

$$p(\mathbf{w}|\tilde{\mathbf{w}}, \boldsymbol{\beta}, \boldsymbol{\Omega}) = \prod_{k,n,m} \delta \left( w_{k,n,m} - \left| \frac{\tilde{w}_{k,n,m}}{B_k(e^{j\omega_{k,m} u_0})} \right| \right). \quad (20)$$

The conditional distribution  $p(\mathbf{w}|\boldsymbol{\beta}, \boldsymbol{\Omega})$  can thus be obtained by defining the distribution  $p(\tilde{\mathbf{w}})$  and marginalizing over  $\tilde{\mathbf{w}}$ . If we now assume that the complex amplitude  $\tilde{w}_{k,n,m}$  follows a circular complex normal distribution

$$\tilde{w}_{k,n,m} \sim \mathcal{N}_{\mathbb{C}}(\tilde{w}_{k,n,m}; 0, \nu^2), \quad n = 1, \dots, N, \quad (21)$$

where  $\mathcal{N}_{\mathbb{C}}(z; 0, \xi^2) = e^{-|z|^2/\xi^2} / (\pi\xi^2)$ , we can show, as in [12], that  $w_{k,n,m}$  follows a Rayleigh distribution:

$$w_{k,n,m} \sim \text{Rayleigh}(w_{k,n,m}; \nu/|B_k(e^{j\omega_{k,m} u_0})|), \quad (22)$$

where  $\text{Rayleigh}(z; \xi) = (z/\xi^2) e^{-z^2/(2\xi^2)}$ . This defines the conditional distribution  $p(\mathbf{w}|\boldsymbol{\beta}, \boldsymbol{\Omega})$ .

The  $F_0$  of stringed and wind instruments often varies continuously over time with musical expressions such as vibrato. For example, the  $F_0$  of a violin sound varies periodically around the note frequency during vibrato, as depicted in Fig. 1. Let us denote the standard log- $F_0$  corresponding to the  $k$ th note by  $\mu_k$ . To appropriately describe the variability of an  $F_0$  contour in both the global and local time scales, we design a prior distribution for  $\boldsymbol{\Omega}_k := (\Omega_{k,1}, \Omega_{k,2}, \dots, \Omega_{k,M})^T$  by employing the product-of-experts (PoE) [6] concept using two probability distributions. First, we design a distribution  $q_g(\boldsymbol{\Omega}_k)$  describing how likely  $\Omega_{k,1}, \dots, \Omega_{k,L}$  stay near  $\mu_k$ . Second, we design another distribution  $q_l(\boldsymbol{\Omega}_k)$  describing how likely  $\Omega_{k,1}, \dots, \Omega_{k,L}$  are locally continuous along time. Here we define  $q_g(\boldsymbol{\Omega}_k)$  and  $q_l(\boldsymbol{\Omega}_k)$  as

$$q_g(\boldsymbol{\Omega}_k) = \mathcal{N}(\boldsymbol{\Omega}_k; \mu_k \mathbf{1}_M, \nu_k^2 \mathbf{I}_M), \quad (23)$$

$$q_l(\boldsymbol{\Omega}_k) = \mathcal{N}(\boldsymbol{\Omega}_k; \mathbf{0}_M, \tau_k^2 D^{-1}), \quad (24)$$

$$D = \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 2 & -1 \\ 0 & \cdots & 0 & 0 & -1 & 1 \end{bmatrix}, \quad (25)$$

where  $\mathbf{I}_M$  denotes an  $M \times M$  identity matrix,  $D$  an  $M \times M$  band matrix,  $\mathbf{1}_M$  an  $M$ -dimensional all-one vector, and  $\mathbf{0}_M$  an  $M$ -dimensional all-zero vector, respectively.  $\nu_k$  denotes the standard deviation from mean  $\mu_k$ , and  $\tau_k$  the standard deviation of the  $F_0$  jumps between adjacent frames. The prior distribution of  $\boldsymbol{\Omega}_k$  is then derived as

$$p(\boldsymbol{\Omega}_k) \propto q_g(\boldsymbol{\Omega}_k)^{\alpha_g} q_l(\boldsymbol{\Omega}_k)^{\alpha_l} \quad (26)$$

where  $\alpha_g$  and  $\alpha_l$  are the hyperparameters that weigh the

contributions of  $q_g(\mathbf{\Omega}_k)$  and  $q_l(\mathbf{\Omega}_k)$  to the prior distribution.

### 2.5 Relation to other models

It should be noted that the present model is related to other models proposed previously.

If we do not assume a parametric model for  $H_{k,l,m}$  and treat each  $H_{k,l,m}$  itself as the parameter, the spectrogram model  $X_{l,m}$  can be seen as an NMF model with time-varying basis spectra, as in [14]. In addition to this assumption, if we assume that  $H_{k,l,m}$  is time-invariant (i.e.,  $H_{k,l,m} = H_{k,l}$ ),  $X_{l,m}$  reduces to the regular NMF model [19]. Furthermore, if we assume each basis spectrum to have a harmonic structure,  $X_{l,m}$  becomes equivalent to the harmonic NMF model [16, 21].

If we assume that  $\Omega_{k,m}$  is equal over time  $m$ ,  $X_{l,m}$  reduces to a model similar to the ones described in [17, 22]. Furthermore, if we describe  $U_{k,m}$  using a parametric function of  $m$ ,  $X_{l,m}$  becomes equivalent to the HTC model [8, 10].

With a similar motivation, Hennequin *et al.* developed an extension to the NMF model defined in the short-time Fourier transform domain to allow the  $F_0$  of each basis spectrum to be time-varying [4].

## 3. INCORPORATION OF AUXILIARY INFORMATION

### 3.1 Use of musically relevant information

We consider using side-information obtained with the state-of-the-art methods for MIR-related tasks including key detection, chord detection and beat tracking to assist source separation.

When multiple types of side-information are obtained for a specific parameter, we can combine the use of the mixture-of-experts and PoE [6] concepts according to the ‘‘AND’’ and ‘‘OR’’ conditions we design. For example, pitch occurrences typically depend on both the chord and key of a piece of music. Thus, when the chord and key information are obtained, we may use the product-of-experts concept to define a prior distribution for the parameters governing the likeliness of the occurrences of the pitches. In the next subsection, we describe specifically how to design the prior distributions.

### 3.2 Designing prior distributions

The likeliness of the pitch occurrences in popular and classical western music usually depend on the key or the chord used in that piece. The likeliness of the pitch occurrences can be described as a probability distribution over the relative energies of the sounds of the individual pitches.

Since the number of times each note is activated is usually limited, inducing sparsity to the temporal activation of each note event would facilitate the source separation. The likeliness of the number of times each note is activated can be described as well as a probability distribution over the temporal activations of the sound of each pitch.

To allow for designing such prior distributions, we decompose  $U_{k,m}$  as the product of two variables: the pitch-wise relative energy  $R_k = \sum_m U_{k,m}$  (i.e.  $\sum_k R_k = 1$ ), and the pitch-wise normalized amplitude  $A_{k,m} = U_{k,m}/R_k$  (i.e.  $\sum_m A_{k,m} = 1$ ). Hence, we can write

$$U_{k,m} = R_k A_{k,m}. \quad (27)$$

This decomposition allows us to incorporate different kinds of prior information into our model by separately defining prior distributions over  $\mathbf{R} = (R_1, \dots, R_K)^\top$  and

$\mathbf{A}_k = (A_{k,1}, \dots, A_{k,M})^\top$ . Here we introduce Dirichlet distributions:

$$\mathbf{A}_k \sim \text{Dir}(\mathbf{A}_k; \boldsymbol{\gamma}_k^{(A)}), \quad \mathbf{R} \sim \text{Dir}(\mathbf{R}; \boldsymbol{\gamma}^{(R)}), \quad (28)$$

where  $\text{Dir}(\mathbf{z}; \boldsymbol{\xi}) \propto \prod_i z_i^{\xi_i}$ ,  $\boldsymbol{\gamma}_k^{(A)} := (\gamma_{k,1}^{(A)}, \dots, \gamma_{k,M}^{(A)})^\top$ , and  $\boldsymbol{\gamma}^{(R)} := (\gamma_1^{(R)}, \dots, \gamma_K^{(R)})^\top$ . For  $p(\mathbf{R})$ , we set  $\gamma_k^{(R)}$  at a reasonably high value if the  $k$ th pitch is contained in the scale and vice versa. For  $p(\mathbf{A}_k)$ , we set  $\gamma_{k,m}^{(A)} < 1$  so that the Dirichlet distribution becomes a sparsity inducing distribution.

## 4. PARAMETER ESTIMATION ALGORITHM

Given an observed power spectrogram  $\mathbf{Y} := \{Y_{l,m}\}_{l,m}$ , we would like to find the estimates of  $\Theta := \{\mathbf{\Omega}, \mathbf{w}, \boldsymbol{\beta}, \mathbf{V}, \mathbf{R}, \mathbf{A}\}$  that maximizes the posterior density  $p(\Theta|\mathbf{Y}) \propto p(\mathbf{Y}|\Theta)p(\Theta)$ . We therefore consider the problem of maximizing

$$\mathcal{L}(\Theta) := \ln p(\mathbf{Y}|\Theta) + \ln p(\Theta), \quad (29)$$

with respect to  $\Theta$  where

$$\ln p(\mathbf{Y}|\Theta) = \sum_{l,m} (Y_{l,m} \ln X_{l,m} - X_{l,m}) \quad (30)$$

$$\begin{aligned} \ln p(\Theta) = & \ln p(\mathbf{w}|\boldsymbol{\beta}, \mathbf{\Omega}) + \sum_k \ln p(\mathbf{\Omega}_k) \\ & + \ln p(\mathbf{R}) + \sum_k \ln p(\mathbf{A}_k). \end{aligned} \quad (31)$$

$=_c$  denotes equality up to constant terms. Since the first term of Eq. (30) involves summation over  $k$  and  $n$ , analytically solving the current maximization problem is intractable. However, we can develop a computationally efficient algorithm for finding a locally optimal solution based on the auxiliary function concept, by using a similar idea described in [8, 12].

When applying an auxiliary function approach to a certain maximization problem, the first step is to define a lower bound function for the objective function. As mentioned earlier, the difficulty with the current maximization problem lies in the first term in Eq. (30). By using the fact that the logarithm function is a concave function, we can invoke the Jensen’s inequality

$$Y_{l,m} \ln X_{l,m} \geq Y_{l,m} \sum_{k,n} \lambda_{k,n,l,m} \ln \frac{w_{k,n,m}^2 e^{-\frac{(x_l - \Omega_{k,m} - \log n)^2}{2\sigma^2}} U_{k,m}}{\lambda_{k,n,l,m}}, \quad (32)$$

to obtain a lower bound function, where  $\lambda_{k,n,l,m}$  is a positive variable that sums to unity:  $\sum_{k,n} \lambda_{k,n,l,m} = 1$ . Equality of (32) holds if and only if

$$\lambda_{k,n,l,m} = \frac{w_{k,n,m}^2 e^{-\frac{(x_l - \Omega_{k,m} - \log n)^2}{2\sigma^2}} U_{k,m}}{X_{l,m}}. \quad (33)$$

Although one may notice that the second term in Eq. (30) is nonlinear in  $\Omega_{k,m}$ , the summation of  $X_{l,m}$  over  $l$  can be approximated fairly well using the integral  $\int_{-\infty}^{\infty} X(x, t_m) dx$ , since  $\sum_l X_{l,m}$  is the sum of the values at the sampled points  $X(x_1, t_m), \dots, X(x_L, t_m)$  with an equal interval, say  $\Delta_x$ . Hence,

$$\begin{aligned} \sum_l X_{l,m} & \simeq \frac{1}{\Delta_x} \int_{-\infty}^{\infty} X(x, t_m) dx \\ & = \frac{1}{\Delta_x} \sum_{k,n} w_{k,n,m}^2 U_{k,m} \int_{-\infty}^{\infty} e^{-\frac{(x - \Omega_{k,m} - \log n)^2}{2\sigma^2}} dx \end{aligned}$$

$$= \frac{\sqrt{2\pi}\sigma}{\Delta_x} \sum_k U_{k,m} \sum_n w_{k,n,m}^2. \quad (34)$$

This approximation implies that the second term in Eq. (30) depends little on  $\Omega_{k,m}$ .

An auxiliary function can thus be written as

$$\begin{aligned} \mathcal{L}^+(\Theta, \lambda) = & \sum_{l,m} Y_{l,m} \sum_{k,n} \lambda_{k,n,l,m} \ln \frac{w_{k,n,m}^2 e^{-\frac{(x_l - \Omega_{k,m} - \ln n)^2}{2\sigma^2}} U_{k,m}}{\lambda_{k,n,l,m}} \\ & - \frac{\sqrt{2\pi}\sigma}{\Delta_x} \sum_m \sum_k U_{k,m} \sum_n w_{k,n,m}^2 + \ln p(\Theta). \end{aligned} \quad (35)$$

We can derive update equations for the model parameters, using the above auxiliary function. By setting at zero the partial derivative of  $\mathcal{L}^+(\Theta, \lambda)$  with respect to each of the model parameters, we obtain

$$w_{k,n,m}^2 \leftarrow \frac{\sum_l Y_{l,m} \lambda_{k,n,l,m} + 1/2}{\sqrt{2\pi} R_k A_{k,m} \sigma / \Delta_x + \nu^2 / (2|B_k(e^{j\Omega_{k,m}} u_0)|^2)}, \quad (36)$$

$$\begin{aligned} \Omega_k \leftarrow & \left( \frac{\alpha_1}{\tau^2} D + \frac{\alpha_g}{\nu_k^2} I_M + \sum_{n,l} \text{diag}(\mathbf{p}_{k,n,l}) \right)^{-1} \\ & \times \left( \mu_k \frac{\alpha_g}{\nu_k^2} \mathbf{1}_M + \sum_{n,l} (x_l - \ln n) \mathbf{p}_{k,n,l} \right), \end{aligned} \quad (37)$$

$$R_k \propto \frac{\sum_{l,m} Y_{l,m} \sum_n \lambda_{k,n,l,m} + \gamma_k^{(R)} - 1}{\sum_{m,n} A_{k,m} w_{k,m,n}^2}, \quad (38)$$

$$A_{k,m} \propto \frac{\sum_l Y_{l,m} \sum_n \lambda_{k,n,l,m} + \gamma_{k,m}^{(A)} - 1}{R_k \sum_n w_{k,m,n}^2}, \quad (39)$$

$$\mathbf{p}_{k,n,l} := \frac{1}{\sigma^2} [Y_{l,1} \lambda_{k,n,l,1}, Y_{l,2} \lambda_{k,n,l,2}, \dots, Y_{l,M} \lambda_{k,n,l,M}]^\top, \quad (40)$$

where  $\text{diag}(\mathbf{p})$  converts a vector  $\mathbf{p}$  into a diagonal matrix with the elements of  $\mathbf{p}$  on the main diagonal.

As for the update equations for the AR coefficients  $\beta$ , we can invoke the method described in [23] with a slight modification, since the terms in the auxiliary function that depend on  $\beta$  has the similar form as the objective function defined in [23]. It can be shown that  $\mathcal{L}^+$  can be increased by the following updates (the details are omitted owing to space limitations):

$$\mathbf{h}_k \leftarrow \hat{C}_k(\beta_k) \beta_k, \quad \beta_k \leftarrow C_k^{-1} \mathbf{h}_k, \quad (41)$$

where  $C_k$  and  $\hat{C}_k(\beta_k)$  are  $(P+1) \times (P+1)$  Toeplitz matrices, whose  $(p, q)$ -th elements are

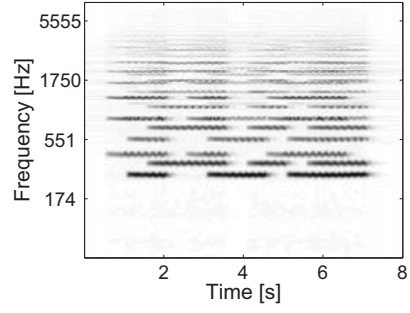
$$\begin{aligned} C_{k,p,q} &= \frac{1}{MN} \sum_{m,n} \frac{w_{k,m,n}^2}{2\nu} \cos[(p-q)n e^{\Omega_{k,m}} u_0], \\ \hat{C}_{k,p,q}(\beta_k) &= \frac{1}{MN} \sum_{m,n} \frac{1}{|B_k(e^{j\Omega_{k,m}} u_0)|^2} \cos[(p-q)n e^{\Omega_{k,m}} u_0]. \end{aligned} \quad (42)$$

## 5. EXPERIMENTS

In the following preliminary experiments, we simplified HTFD by omitting the source filter model and assuming the time-invariance of  $w_{k,m,n}$ .

### 5.1 $F_0$ tracking of violin sound

To confirm whether HTFD can track the  $F_0$  contour of a sound, we compared HTFD with NMF with the I-divergence, by using a 16 kHz-sampled audio signal which



**Figure 2.** Power spectrogram of a mixed audio signal of three violin vibrato sounds (Db4, F4 and Ab4).

were artificially made by mixing Db4, F4 and Ab4 violin vibrato sounds from the RWC instrument database [3]. In this paper, the  $F_0$  of the pitch name A4 was set at 440 Hz. The power spectrogram of the mixed signal is shown in Fig. 2. To convert the signal into a spectrogram, we employed the fast approximate continuous wavelet transform [9] with a 16 ms time-shift interval.  $\{x_l\}_l$  ranged 55 to 7040 Hz per 10 cent. The parameters of HTFD were set at  $\gamma_k^{(A)} = (1 - 3.96 \times 10^{-6}) \mathbf{1}_I$ ,  $(\tau_k, \nu_k) = (0.83, 1.25)$  for all  $k$ ,  $(N, K, \sigma, \alpha_g, \alpha_s) = (8, 73, 0.02, 1, 1)$ , and  $\gamma^{(R)} = (1 - 2.4 \times 10^{-3}) \mathbf{1}_K$ .  $\{\mu_k\}_k$  ranged A1 to A#7 with a chromatic interval, i.e.  $\mu_k = \ln(55) + \ln(2) \times (k-1)/12$ . The number of NMF bases were set at three. The parameter updates of both HTFD and NMF were stopped at 100 iterations.

While the estimates of spectrograms obtained with NMF were flat and the vibrato spectra seemed to be averaged (Fig. 3 (a)), those obtained with HTFD tracked the  $F_0$  contours of the vibrato sounds appropriately (Fig. 3 (b)), and clear vibrato sounds were contained in the separated audio signals by HTFD.

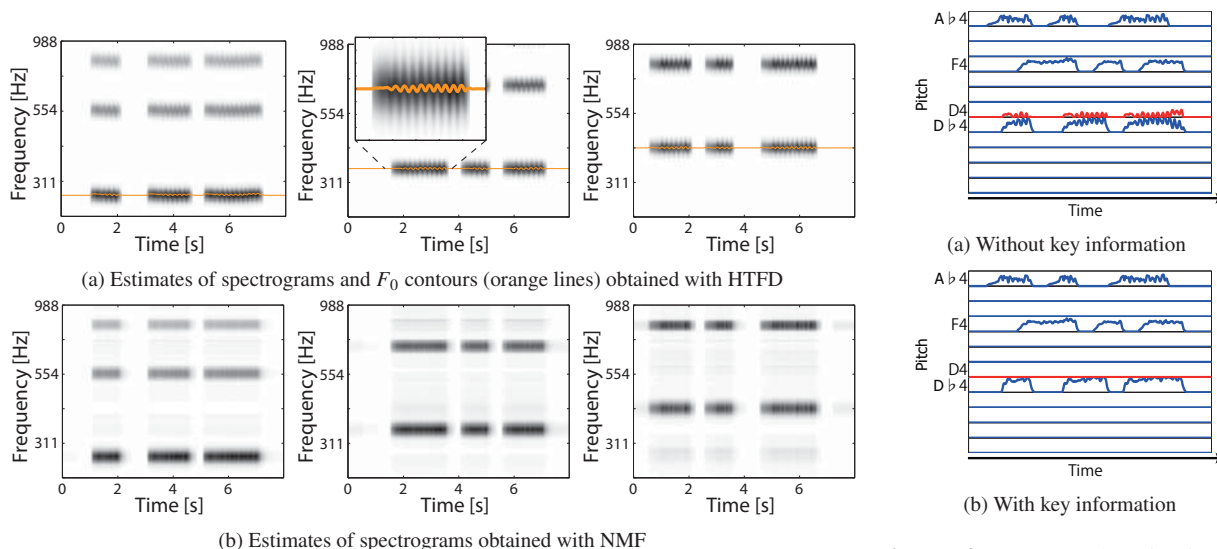
### 5.2 Separation using key information

We next examined whether the prior information of a sound improve source separation accuracy. The key of the sound used in 5.1, was assumed as Db major. The key information was incorporated in the estimation scheme by setting  $\gamma_k^{(R)} = 1 - 2.4 \times 10^{-3}$  for the pitch indices that are not contained in the Db major scale and  $\gamma_k^{(R)} = 1 - 3.0 \times 10^{-3}$  for the pitch indices contained in that scale. The other conditions were the same as 5.1.

With HTFD without using the key information, the estimated activations of the pitch indices that were not contained in the scale, in particular D4, were high as illustrated in Fig. 4 (a). In contrast, those estimated activations with HTFD using the key information were suppressed as shown in Fig. 4 (b). These results thus support strongly that incorporating prior information improve the source separation accuracy.

### 5.3 Transposing from one key to another

Here we show some results of an experiment on automatic key transposition [11] using HTFD. The aim of key transposition is to change the key of a musical piece to another key. We separated the spectrogram of a polyphonic sound into spectrograms of individual pitches using HTFD, transposed the pitches of the subset of the separated components, added all the spectrograms together to construct a pitch-modified polyphonic spectrogram, and constructed a



**Figure 3.** Estimated spectrogram models by harmonic-temporal factor decomposition (HTFD) and non-negative matrix factorization (NMF). In left-to-right fashion, the spectrogram models are for D♭4, F4 and A♭4.

time-domain signal from the modified spectrogram using the method described in [13]. For the key transposition, we adopted a simple way: To transpose, for example, from A major scale to A natural minor scale, we changed the pitches of the separated spectrograms corresponding to C♯, F♯ and G♯ to C, F and G, respectively.

Some results are demonstrated in <http://hil.t.u-tokyo.ac.jp/~nakamura/demo/HTFD.html>.

## 6. CONCLUSION

This paper proposed a new approach for monaural source separation called the “Harmonic-Temporal Factor Decomposition (HTFD)” by introducing a spectrogram model that combines the features of the models employed in the NMF and HTC approaches. We further described some ideas how to design the prior distributions for the present model to incorporate musically relevant information into the separation scheme.

## 7. ACKNOWLEDGEMENTS

This work was supported by JSPS Grant-in-Aid for Young Scientists B Grant Number 26730100.

## 8. REFERENCES

- [1] A. S. Bregman: *Auditory Scene Analysis*, MIT Press, Cambridge, 1990.
- [2] J. S. Downie, D. Byrd, and T. Crawford: “Ten years of ISMIR: Reflections on challenges and opportunities,” *Proc. ISMIR*, pp. 13–18, 2009.
- [3] M. Goto: “Development of the RWC Music Database,” *Proc. ICA*, pp. 1–553–556, 2004.
- [4] R. Hennequin, R. Badeau, and B. David: “Time-dependent parametric and harmonic templates in non-negative matrix factorization,” *Proc. DAFX*, pp. 246–253, 2010.
- [5] R. Hennequin, B. David, and R. Badeau: “Score informed audio source separation using a parametric model of non-negative spectrogram,” *Proc. ICASSP*, pp. 45–48, 2011.
- [6] G. E. Hinton: “Training products of experts by minimizing contrastive divergence,” *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [7] G. Hu, and D. L. Wang: “An auditory scene analysis approach to monaural speech segregation,” *Topics in Acoust. Echo and Noise Contr.*, pp. 485–515, 2006.
- [8] H. Kameoka: *Statistical Approach to Multipitch Analysis*, PhD thesis, The University of Tokyo, Mar. 2007.
- [9] H. Kameoka, T. Tabaru, T. Nishimoto, and S. Sagayama: (*Patent*) *Signal processing method and unit*, in Japanese, Nov. 2008.
- [10] H. Kameoka, T. Nishimoto, and S. Sagayama: “A multipitch analyzer based on harmonic temporal structured clustering,” *IEEE Trans. ASLP*, vol. 15, no. 3, pp. 982–994, 2007.
- [11] H. Kameoka, J. Le Roux, Y. Ohishi, and K. Kashino: “Music Factorizer: A note-by-note editing interface for music waveforms,” *IPSI SIG Tech. Rep.*, 2009-MUS-81-9, in Japanese, Jul. 2009.
- [12] H. Kameoka: “Statistical speech spectrum model incorporating all-pole vocal tract model and  $F_0$  contour generating process model,” *IEICE Tech. Rep.*, vol. 110, no. 297, SP2010-74, pp. 29–34, in Japanese, Nov. 2010.
- [13] T. Nakamura and H. Kameoka: “Fast signal reconstruction from magnitude spectrogram of continuous wavelet transform based on spectrogram consistency,” *Proc. DAFX*, 40, to appear, 2014.
- [14] M. Nakano, J. Le Roux, H. Kameoka, Y. Kitano, N. Ono, and S. Sagayama: “Nonnegative matrix factorization with Markov-chained bases for modeling time-varying patterns in music spectrograms,” *Proc. LVA/ICA*, pp. 149–156, 2010.
- [15] A. Ozerov, C. Févotte, R. Blouet, and J. L. Durrieu: “Multichannel nonnegative tensor factorization with structured constraints for user-guided audio source separation,” *Proc. ICASSP*, pp. 257–260, 2011.
- [16] S. A. Raczynski, N. Ono, and S. Sagayama: “Multipitch analysis with harmonic nonnegative matrix approximation,” *Proc. ISMIR*, pp. 381–386, 2007.
- [17] D. Sakaue, T. Otsuka, K. Itoyama, and H. G. Okuno: “Bayesian nonnegative harmonic-temporal factorization and its application to multipitch analysis,” *Proc. ISMIR*, pp. 91–96, 2012.
- [18] U. Simsekli and A. T. Cemgil: “Score guided musical source separation using generalized coupled tensor factorization,” *Proc. EUSIPCO*, pp. 2639–2643, 2012.
- [19] P. Smaragdis and J. C. Brown: “Non-negative matrix factorization for polyphonic music transcription,” *Proc. WASPAA*, pp. 177–180, 2003.
- [20] P. Smaragdis and G. J. Mysore: “Separation by “humming”: User-guided sound extraction from monophonic mixtures,” *Proc. WASPAA*, pp. 69–72, 2009.
- [21] E. Vincent, N. Bertin, and R. Badeau: “Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription,” *Proc. ICASSP*, pp. 109–112, 2008.
- [22] K. Yoshii and M. Goto: “Infinite latent harmonic allocation: A nonparametric Bayesian approach to multipitch analysis,” *Proc. ISMIR*, pp. 309–314, 2010.
- [23] A. El-Jaroudi, J. Makhoul: “Discrete all-pole modeling,” *IEEE Trans. SP*, vol. 39, no. 2, pp. 411–423, 1991.

**Figure 4.** Temporal activations of A3–A♭4 estimated with HTFD using and without using prior information of the key. The red curves represent the temporal activations of D4.



Oral Session 9  
**Rhythm & Beat**

This Page Intentionally Left Blank

# DESIGN AND EVALUATION OF ONSET DETECTORS USING DIFFERENT FUSION POLICIES

Mi Tian, György Fazekas, Dawn A. A. Black, Mark Sandler  
 Centre for Digital Music, Queen Mary University of London  
 {m.tian, g.fazekas, dawn.black, mark.sandler}@qmul.ac.uk

## ABSTRACT

Note onset detection is one of the most investigated tasks in Music Information Retrieval (MIR) and various detection methods have been proposed in previous research. The primary aim of this paper is to investigate different fusion policies to combine existing onset detectors, thus achieving better results. Existing algorithms are fused using three strategies, first by combining different algorithms, second, by using the linear combination of detection functions, and third, by using a late decision fusion approach. Large scale evaluation was carried out on two published datasets and a new percussion database composed of Chinese traditional instrument samples. An exhaustive search through the parameter space was used enabling a systematic analysis of the impact of each parameter, as well as reporting the most generally applicable parameter settings for the onset detectors and the fusion. We demonstrate improved results attributed to both fusion and the optimised parameter settings.

## 1. INTRODUCTION

The automatic detection of onset events is an essential part in many music signal analysis schemes and has various applications in content-based music processing. Different approaches have been investigated for onset detection in recent years [1,2]. As the main contribution of this paper, we present new onset detectors using different fusion policies, with improved detection rates relying on recent research in the MIR community. We also investigate different configurations of onset detection and fusion parameters, aiming to provide a reference for configuring onset detection systems.

The focus of ongoing onset detection work is typically targeting Western musical instruments. Apart from using two published datasets, a new database is incorporated into our evaluation, collecting percussion ensembles of Jingju, also denoted as Peking Opera or Beijing Opera, a major

genre of Chinese traditional music<sup>1</sup>. By including this dataset, we aim at increasing the diversity of instrument categories in the evaluation of onset detectors, as well as extending the research to include non-Western music types.

The goal of this paper can be summarised as follows: *i*) to evaluate fusion methods in comparison with the baseline algorithms, as well as a state-of-the-art method<sup>2</sup>; *ii*) to investigate which fusion policies and which pair-wise combinations of onset detectors yield the most improvement over standard techniques; *iii*) to find the best performing configurations by searching through the multi-dimensional parameter space, hence identifying emerging patterns in the performances of different parameter settings, showing good results across different datasets; *iv*) to investigate the performance difference in Western and non-Western percussive instrument datasets.

In the next section, we present a review of related work. Descriptions of the datasets used in this experiment are given in Section 3. In Section 4, we introduce different fusion strategies. Relevant post-processing and peak-picking procedures, as well as the parameter search process will be discussed in Section 5. Section 6 presents the results, with a detailed analysis and discussion of the performance of the fusion methods. Finally, the last section summarises our findings and provides directions for future work.

## 2. RELATED WORK

Many onset detection algorithms and systems have been proposed in recent years. Common approaches using energy or phase information derived from the input signal include the high frequency content (HFC) and complex domain (CD) methods. See [1,6] for detailed reviews and [9] for further improvements. Pitch contours and harmonic information can also be indicators for onset events [7]. These methods shows some superiority over energy based ones in case of soft onsets.

Onset detection systems using machine learning techniques have also been gaining popularity in recent years<sup>3</sup>. The winner of MIREX 2013 audio onset detection task utilises convolutional neural networks to classify and distinguish onsets from non-onset events in the spectrogram [13]. The data-driven nature of these methods makes the



© Mi Tian, György Fazekas, Dawn A. A. Black, Mark Sandler.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Mi Tian, György Fazekas, Dawn A. A. Black, Mark Sandler. "Design and Evaluation of Onset Detectors Using Different Fusion Policies", 15th International Society for Music Information Retrieval Conference, 2014.

<sup>1</sup> [http://en.wikipedia.org/wiki/Peking\\_opera](http://en.wikipedia.org/wiki/Peking_opera)

<sup>2</sup> Machine learning-based methods are excluded from this study to limit the scope of our work.

<sup>3</sup> [http://www.music-ir.org/mirex/wiki/2013:Audio\\_Onset\\_Detection](http://www.music-ir.org/mirex/wiki/2013:Audio_Onset_Detection)

detection less dependent on onset types, though a computationally expensive training process is required. A promising approach for onset detection lies in the fusion of multiple detection methods. Zhou et al. proposed a system integrating two detection methods selected according to properties of the target onsets [17]. In [10], pitch, energy and phase information are considered in parallel for the detection of pitched onsets. Another fusion strategy is to combine peak score information to form new estimations of the onset events [8]. Albeit fusion has been used in previous work, there is a lack of systematic evaluation of fusion strategies and applications in the current literature. This paper focusses on the assessment of different fusion policies, from feature-level and detection function-level fusion to higher level decision fusion.

The success of an onset detection algorithm largely depends on the signal processing methods used to extract salient features from the audio that emphasise the features characterising onset events as well as smoothing the noise in the detection function. Various signal processing techniques have been introduced in recent studies, such as vibrato suppression [3] and adaptive thresholding [1]. In [14], adaptive whitening is presented where each STFT bins magnitude is divided by the an average peak for that bin accumulated over time. This paper also investigates the performances of some commonly used signal processing modules within onset detection systems.

### 3. DATASETS

In this study, we use two previously released evaluation datasets and a newly created one. The first published dataset comes from [1], containing 23 audio tracks with a total duration of 190 seconds and having 1058 onsets. These are classified into four groups: pitched non-percussive (PNP), e.g. bowed strings, 93 onsets, pitched percussive (PP), e.g. piano, 482 onsets<sup>4</sup>, non-pitched percussive (NPP), e.g. drums, 212 onsets, and complex mixtures (CM), e.g. pop singing music, 271 onsets. The second set comes from [2] which is composed of 30 samples<sup>5</sup> of 10 second audio tracks, containing 1559 onsets in total, covering also four categories: PNP (233 onsets in total), PP (152 onsets), NPP (115 onsets), CM (1059 onsets). The use of these datasets enables us to test the algorithms on a range of different instruments and onset types, and provides for direct comparison with published work. The combined dataset used in the evaluation of our work is composed of these two sets.

The third dataset consists of recordings of the four major percussion instruments in Jingju: bangu (clapper-drum), daluo (gong-1), naobo (cymbals), and xiaoluo (gong-2). The samples are manually mixed using individual recordings of these instruments with possibly simultaneous onsets to closely reproduce real world conditions. See [15] for more details on the instrument types and the dataset. This dataset includes 10 samples of 30-second excerpts

<sup>4</sup> A 7-onset discrepancy (482 instead of 489) from the reference paper is reported by the original author due to revisions of annotations.

<sup>5</sup> Only a subset of this dataset presented in the original paper is received from the author for the evaluation in this paper.

with 732 onsets. We also use NPP onsets from the first two datasets to form the fourth one, providing a direct comparison with the Chinese NPP instruments. All stimuli are mono signals sampled at 44.1kHz<sup>6</sup> and 16 bits per sample, having 3349 onsets in total.

## 4. FUSION EXPERIMENT

The aim of information fusion is to merge information from heterogeneous sources to reduce uncertainty of inferences [11]. In our study, six spectral-based onset detection algorithms are considered as baselines for fusion: high frequency content (HFC), spectral difference (SD) complex domain (CD), broadband energy rise (BER), phase deviation (PD), outlined in [1], and SuperFlux (SF) from recent work [4]. We also developed and included in the fusion a method based on Linear Predictive Coding [12], where the LPC coefficients are computed using the Levinson-Durbin recursion, and the onset detection function is derived from the LPC error signal.

Three fusion policies are used in our experiments: *i*) feature-level fusion, *ii*) fusion using the linear combination of detection functions and *iii*) decision fusion by selecting and merging onset candidates. All pairwise combination of the baseline algorithms are amenable for the latter two fusion policies. However, not all algorithms can be meaningfully combined using feature-level fusion. For example CD can be considered as an existing combination of SD and PD, therefore combining CD with either of these two at a feature level is not sensible. In this study, 10 feature-level fusion, 13 linear combination based fusion and 15 decision fusion based methods are tested. These are compared to the 7 original methods, giving us 45 detectors in total. In the following, we describe specific fusion policies. We assume familiarity with onset detection principles and restrain from describing these details, please see [1] for a tutorial.

### 4.1 Feature-level Fusion

In feature-level fusion, multiple algorithms are combined to compute fused features. For conciseness, we provide only one example combining BER and SF, denoted BERSF, utilising the vibrato suppression capability of SF [4] for detecting soft onsets, as well as the good performance of BER for detecting percussive onsets with sharp energy bursts [1]. Here, we use the BER to mask the SF detection function as described by Equation (1). In essence, SF is used directly when there is evidence for a sharp energy rise, otherwise it is further smoothed using a median filter.

$$ODF(n) = \begin{cases} SF(n) & \text{if } BER(n) > \gamma \\ \lambda(SF(n)) & \text{otherwise,} \end{cases} \quad (1)$$

where  $\gamma$  is an experimentally defined threshold,  $\lambda$  is a weighting constant set to 0.9 and  $\overline{SF(n)}$  is the median filtered detection function with a window size of 3 frames.

<sup>6</sup> Some audio files were upsampled to obtain a uniform dataset.



## 4.2 Linear Combination of Detection Functions

In this method, two time aligned detection functions are used and their weighted linear combination is computed to form a new detection function as shown in Equation 2:

$$ODF(n) = wODF_1(n) + (1 - w)ODF_2(n), \quad (2)$$

where  $ODF_1$  and  $ODF_2$  are two normalised detection functions and  $w$  is a weighting coefficient ( $0 \leq w \leq 1$ ).

## 4.3 Decision Fusion

This fusion method operates at a later stage and combines prior decisions of two detectors. Post-processing and peak picking are applied separately yielding two lists of onset candidates. Onsets from the two lists occurring within a fixed temporal tolerance window will be merged and accepted. Let  $TS_1$  and  $TS_2$  be the lists of onset locations given by two different detectors,  $i$  and  $j$  be indexes of onsets in the candidate lists and  $\delta$  the tolerance time window. The final onset locations are generated using the fusion strategy described by Algorithm 1.

---

### Algorithm 1 Onset decision fusion

---

```

1: procedure DECISIONFUSION( $TS_1, TS_2$ )
2:    $I, J \leftarrow 0 : \text{len}(TS_1) - 1, 0 : \text{len}(TS_2) - 1$ 
3:    $TS \leftarrow \text{empty list}$ 
4:   for all  $i, j$  in  $\text{product}(I, J)$  do
5:     if  $\text{abs}(TS_1[i] - TS_2[j]) < \delta$  then
6:       insert sorted:  $TS \leftarrow \text{mean}(TS_1[i], TS_2[j])$ 
7:   return  $TS$ 

```

---

## 5. PEAK PICKING AND PARAMETER SEARCH

### 5.1 Smoothing and Thresholding

Post-processing is an optional stage to reduce noise that interferes with the selection of maxima in the detection function. In this study, three post-processing blocks are used: *i*) DC removal and normalisation, *ii*) zero-phase low-pass filtering and *iii*) adaptive thresholding. In conventional normalisation, data is scaled using a fixed constant. Here we use a normalisation coefficient computed by weighting the input exponentially. After removing constant offsets, the detection function is normalised using the coefficient *AlphaNorm* calculated by Equation (3):

$$\text{AlphaNorm} = \left( \frac{\sum_n |ODF(n)|^\alpha}{\text{len}(ODF)} \right)^{\frac{1}{\alpha}} \quad (3)$$

A low-pass filter is applied to the detection function to reduce noise. To avoid introducing delays, a zero phase filter is employed at this stage. Finally, adaptive thresholding using a moving median filter is applied following Bello [1], to avoid the common pitfalls of using a fixed threshold for peak picking.

### 5.2 Peak Picking

#### 5.2.1 Polynomial Fitting

The use of polynomial fitting allows for assessing the shape and magnitude of peaks separately. Here we fit a second-

degree polynomial on the detection function around local maxima using a least squares method, following the QM Vamp Plugins<sup>7</sup>. The coefficients  $a$  and  $c$  of the quadratic equation  $y = ax^2 + bx + c$  are used to detect both sharper peaks, under the condition  $a > th_a$ , and peaks with a higher magnitude, when  $c > th_c$ . The corresponding thresholds are computed from a single sensitivity parameter called *threshold* using  $th_a = (100 - \text{threshold})/1000$  for the quadratic term and  $th_c = (100 - \text{threshold})/1500$  for the constant term. The linear term  $b$  can be ignored.

#### 5.2.2 Backtracking

In case of many musical instruments, onsets have longer transients without a sharp burst of energy rise. This may cause energy based detection functions to exhibit peaks after the perceived onset locations. Vos and Rasch conclude that onsets are perceived when the envelope reaches a level of roughly 6-15 dB below the maximum level of the tones [16]. Using this rationale, we trace the onset locations from the detected peak position back to a hypothesised earlier “perceived” location. The backtracking procedure is based on measuring relative differences in the detection function, as illustrated by Algorithm 2, where  $\theta$  is the threshold used as a stopping condition. We use the implementation available in the QM Vamp Plugins.

---

### Algorithm 2 Backtracking

---

```

Require:  $idx$ : index of a peak location in the ODF
1: procedure BACKTRACKING( $idx, ODF, \theta$ )
2:    $\delta, \gamma \leftarrow 0$ 
3:   while  $idx > 1$  do
4:      $\delta \leftarrow ODF[idx] - ODF[idx - 1]$ 
5:     if  $\delta < \gamma * \theta$  then
6:       break
7:      $idx \leftarrow idx - 1$ 
8:      $\gamma \leftarrow \delta$ 
9:   return  $idx$ 

```

---

### 5.3 Parameter Search

An exhaustive search is carried out to find the configurations in the parameter space yielding the best detection rates. The following parameters and settings, related to the onset detection and fusion stages, are evaluated: *i*) adaptive whitening (*wht*) on/off; *ii*) detection sensitivity (threshold), ranging from 0.1 to 1.0 with an increment of 0.1; *iii*) backtracking threshold ( $\theta$ ), ranging from 0.4 to 2.4 with 8 equal subdivisions (the upper bound is set to an empirical value 2.4 in the experiment since the tracking will not go beyond the previous valley); *iv*) linear combination coefficient ( $w$ ), ranging from 0.0 to 1.0 with an increment of 0.1; *v*) tolerance window length ( $\delta$ ) for decision fusion, ranging from 0.01 to 0.05 (in second) having 8 subdivisions. This gives a 5-dimensional space and all combinations of all possible values described above are evaluated. This results in 180 configurations in case of standard detectors and feature-level fusion, 1980 in case of linear fusion and 1620 for decision fusion. The configurations are described

<sup>7</sup><http://www.vamp-plugins.org>

using the Vamp Plugin Ontology<sup>8</sup> and the resulting RDF files are used by Sonic Annotator [5] to configure the detectors. The test result will thus give us not only the overall performance of each onset detector, but also uncover their strengths and limitations across different datasets and parameter settings.

## 6. EVALUATION AND RESULTS

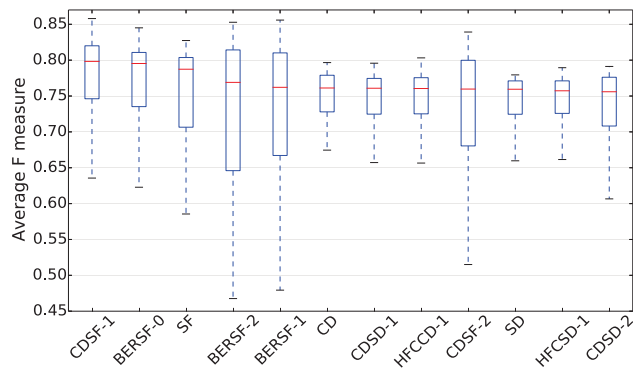
### 6.1 Analysis of Overall Performance

Figure 1 provides an overview of the results, showing the F-measure for the top 12 detectors in our study<sup>9</sup>. Detectors are ranked by the median showing the overall performance increase due to fusion across the entire range of parameter settings. Due to space limitations, only a subset of the results are reported in this paper. The complete result set for all tested detectors under all configurations on different datasets is available online<sup>10</sup>, together with Vamp plugins of all tested onset detectors. The names of the fusion algorithms come from the abbreviations of the constituent methods, while the numbers represent the fusion policy: *0*: feature-level fusion, *1*: linear combination of detection functions and *2*: decision fusion.

CDSF-1 yields improved F-measure for the combined dataset by 3.06% and 6.14% compared to the two original methods SF and CD respectively. Smaller interquartile ranges (IQRs) observed in case of CD, SD and HFC based methods show they have less dependency on the configuration. BERSF-2 and BERSF-1 vary the most in performance, also reflected from their IQRs. In case of BERSF-2, the best performance is obtained using the widest considered tolerance window (0.05s), with modest sensitivity (40%). However, decreasing the tolerance window size has an adverse effect on the performance, yielding one of the lowest detection rates caused by the significant drop of recall. In case of BERSF-1, a big discrepancy between the best and worst performing configurations can be observed. This is partly because the highest sensitivity setting has a negative effect on SF causing very low precision.

Table 1 shows the results ranked by F-measure, precision and recall with corresponding standard deviations for the ten best detectors as well as all baseline methods. Standard deviations are computed over the results for all configurations in each dataset. SF is ranked in the best performing ten, thus it is excluded from the baseline. Nine out of the top ten detectors are fusion methods. CDSF-1 performs the best for all datasets (including CHN-NPP and WES-NPP that are not listed in the table) while BERSF yields the second best performance in the combined, WES-NPP and JPB datasets. Corresponding parameter settings for the combined dataset are given in Table 2.

Fusion policies may perform differently in the evaluation. In case of feature-level fusion, we compared how combined methods score relative to their constituents. The



**Figure 1.** F-measure of all configurations for the top 12 detectors. (Min, first and third quartile and max value of the data are represented by the bottom bar of the whiskers, bottom and upper borders of the boxes and upper bar of the whiskers respectively. Median is shown by the red line)

method	threshold	$\theta$	wht	$w$	$\delta$ (s)
CDSF-1	10.0	2.15	off	0.20	n/a
BERSF-1	10.0	2.40	off	0.30	n/a
BERSF-2	40.0	2.15	off	n/a	0.05
BERSF-0	30.0	2.40	off	n/a	n/a
CDSF-2	50.0	2.40	off	n/a	0.05
SF	20.0	2.40	off	n/a	n/a
CDBER-1	10.0	2.40	off	0.50	n/a
BERSD-1	10.0	2.40	off	0.60	n/a
HFCCD-1	20.0	1.15	off	0.50	n/a
CDBER-2	50.0	1.15	off	n/a	0.05
mean	25.90	2.100	-	0.4200	0.05
std	15.01	0.4848	-	0.1470	0.00
median	20.0	2.15	-	0.50	0.05
mode	10.0	2.40	off	0.50	0.05

**Table 2.** Parameter settings for the ten best performing detectors, **threshold**: overall detection sensitivity;  $\theta$ : back-tracking threshold; **wht**: adaptive whitening;  $w$ : linear combination coefficient;  $\delta$ : tolerance window size.

performances vary between datasets, with only HFCBER-0 outperforming both HFC and BER on the combined and SB datasets in terms of mean F-measure. However, five perform better than their two constituents on JPB, two on CHN-NPP and five on WES-NPP dataset (these results are published online). A more detailed analysis of these performance differences constitutes future work.

When comparing linear fusion of detection functions with decision fusion, the former performs better across all datasets in all but one cases, the fusion of HFC and BER. Even in this case, linear fusion yields close performance in terms of mean F-measure. Interesting observations also emerge for particular methods on certain datasets. The linear fusion based detectors involving LPC and PD (SDPD-1 and LPCPD-1) show better performances in the case of the CHN-NPP dataset compared to their performances on other datasets as well those given by their constituent methods (please see table online). Further analysis, for instance, by looking at statistical significance of these observations is required to identify relevant instrument properties.

When comparing BERSF-2, CDSF-2 and CDBER-2 to the other detectors in Table 1, notably higher standard deviations in *recall* and *F-measure* are shown, indicating this

<sup>8</sup> <http://www.omras2.org/VampOntology>

<sup>9</sup> Due to different post-processing stages, the results reported here may diverge from previously published results.

<sup>10</sup> <http://isophonics.net/onset-fusion>

method	F (combined)	P (combined)	R (combined)	F (sb)	P (sb)	R (sb)	F (jpb)	P (jpb)	R (jpb)
CDSF-1	<b>0.8580</b> 0.0613	<b>0.9054</b> 0.1195	0.8153 <b>0.0609</b>	<b>0.8194</b> 0.0598	0.8455 0.1165	0.7949 <b>0.0681</b>	<b>0.9286</b> 0.0649	<b>0.9748</b> 0.1241	0.8865 <b>0.0525</b>
BERSF-1	0.8559 0.0941	0.8857 0.1363	<b>0.8280</b> 0.0866	0.8126 0.0961	0.8191 0.1306	0.8062 0.0988	0.9283 0.0925	0.9718 0.1463	<b>0.8885</b> 0.0710
BERSF-2	0.8528 0.1684	0.8901 0.1411	0.8186 0.2028	0.8088 0.1677	<b>0.8729</b> 0.1470	0.7536 0.2055	0.9230 0.1724	0.9637 0.1310	0.8856 0.2011
BERSF-0	0.8451 0.0722	0.8638 0.1200	0.8272 0.0701	0.8025 0.0723	0.8185 <b>0.1134</b>	0.7870 0.0744	0.9175 0.0747	0.9712 0.1322	0.8694 0.0658
CDSF-2	0.8392 0.1537	0.8970 <b>0.1129</b>	0.7884 0.1855	0.7892 0.1758	0.8336 0.1251	0.7493 0.2014	0.9165 0.1344	0.9642 <b>0.1001</b>	0.8732 0.1690
SF	0.8274 0.0719	0.8313 0.1209	0.8234 0.0657	0.8126 0.0744	0.8191 0.1241	<b>0.8063</b> 0.0737	0.8488 0.0704	0.8290 0.1177	0.8694 0.0558
CDBER-1	0.8145 0.0809	0.8210 0.1276	0.8080 0.0792	0.7877 0.0829	0.7972 0.1295	0.7785 0.0893	0.8560 0.0793	0.8678 0.1253	0.8446 0.0667
BERSD-1	0.8073 0.0792	0.8163 0.1311	0.7986 0.0812	0.7843 0.0828	0.7985 0.1358	0.7707 0.0915	0.8420 0.0756	0.8310 0.1252	0.8532 0.0685
HFCDD-1	0.8032 <b>0.0472</b>	0.8512 0.1179	0.7603 0.0734	0.7802 <b>0.0448</b>	0.8387 0.1239	0.7293 0.0765	0.8416 <b>0.0511</b>	0.8376 0.1101	0.8456 0.0705
CDBER-2	0.7967 0.2231	0.8423 0.1404	0.7558 0.2398	0.7605 0.2279	0.8140 0.1607	0.7138 0.2384	0.8498 0.2291	0.8853 0.1273	0.8170 0.2494
CD	0.7966 0.0492	0.8509 0.1164	0.7489 0.0672	0.7692 0.0467	0.8361 0.1191	0.7123 0.0709	0.8320 0.0535	0.8692 0.1128	0.7979 0.0636
BER	0.7883 0.0942	0.7776 0.1184	0.7994 0.1001	0.7626 0.0974	0.7521 0.1166	0.7138 0.1119	0.8254 0.0920	0.7968 0.1226	0.8561 0.0851
SD	0.7795 0.0466	0.8354 0.1269	0.7305 0.0733	0.7604 0.0450	0.8311 0.1326	0.7009 0.0785	0.8210 0.0491	0.8202 0.1190	0.8217 0.0676
HFC	0.7712 0.0412	0.8011 0.1225	0.7436 0.0898	0.7411 0.0375	0.7818 0.1291	0.7044 0.0844	0.8159 0.0496	0.8082 0.1138	0.8236 0.1002
LPC	0.7496 0.0658	0.7671 0.1103	0.7330 0.1061	0.7243 0.0657	0.7494 0.1069	0.7009 0.1019	0.7913 0.0662	0.8041 0.1164	0.7788 0.1118
PD	0.6537 0.1084	0.5775 0.1008	0.7530 0.2235	0.6143 0.1093	0.5230 0.0688	0.7308 0.2302	0.7114 0.1115	0.6513 0.1536	0.7836 0.2158

**Table 1.** F-measure (F), Precision (P) and Recall (R) for dataset **combined**, **SB**, **JPB** for detectors under best performing configurations from the parameter search, with corresponding standard deviations over different configurations.

statistic	Combined	SB	JPB	CHN-NPP	WES-NPP
mean	0.7731	0.7438	0.8183	0.8527	0.8358
std	0.0587	0.0579	0.0628	<b>0.1206</b>	0.0641
median	0.7818	0.7595	0.8226	0.8956	0.8580

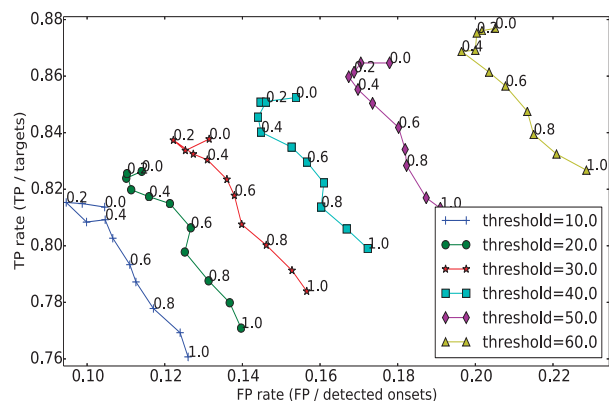
**Table 3.** Statistics for F-measure of the ten detectors with their best performances from Table 1 for different datasets

fusion policy is more sensitive to the choice of parameters. A possible improvement in this fusion policy would be to make the size of the tolerance window dependent on the magnitude of relevant peaks of the detection functions.

The results also vary across different datasets. Table 3 summarises F-measure statistics computed over the detectors listed in Table 1 at their best setting for each datasets used in this paper. In comparison with SB, the JPB dataset exhibits higher F-measure. This dataset has larger diversity in terms of the length of tracks and the level of complexity, while the SB dataset mainly consists of complex mixture (CM) onsets type. Both the Chinese and Western NPP onset class provides noticeably higher detection rate compared to the mix-typed datasets. Though the CHN-NPP set shows the largest standard deviation, suggesting a greater variation in performance between the different detectors for these instruments. Apart from aiming at optimal overall detection results, it is also useful to consider when and how a certain onset detector exhibits the best performance, which constitutes future work.

## 6.2 Parameter Specifications

For general datasets a low detection sensitivity value is favourable, which is supported by the fact that 30 out of the 45 tested methods yield the best performances with a sensitivity lower than 50% (see online). In 23 out of all cases, the value of the backtracking threshold was the highest considered in our study (2.4) when the detectors yield the best performances for the combined dataset, and it was unanimously at a high value for all other datasets including the percussive ones. This suggests that in many cases, the perceived onset will be better characterised by the valley of the detection function prior to the detected peak. Note that



**Figure 2.** Performances of CDSF-1 onset detector under different  $w$  (labelled in each curve) and **threshold** (annotated in the side box) settings

even at a higher threshold, the onset location would not be traced back further than the valley preceding the peak detected in our algorithm. An interesting direction for future work would thus be, given this observation, to take into account the properties of human perception.

Adaptive whitening had to be turned off for the majority of detectors to provide good performance for all datasets. This indicates that the method does not improve onset detection performance in general, although it is available in most onset detectors in the Vamp plugin library. The value of the tolerance window was always 0.05s for best performance in our study, suggesting that the temporal precision of the different detectors varies significantly, which requires a fairly wide decision horizon for successful combination.

Figure 2 shows how two parameters influence the performance of the onset detector CDSF-1. The figure illustrates the true positive rate (i.e., correct detections relative to the number of target onsets) and false positive rate (i.e., false detections relative to the number of detected onsets) and better performance is indicated by the curve shifting upwards and leftwards. All parameters except the *linear combination coefficient* ( $w$ ) and *detection sensitivity*

(*threshold*) are fixed at their optimal values. We can observe that the value of the linear combination coefficient is around 0.2 for best performance. This suggests that the detector works the best when taking the majority of the contribution from SF. With the *threshold* increasing from 10.0% to 60.0%, the true positive rate is increasing at the cost of picking more false onsets, thus a lower sensitivity is preferred in this case. Poorest performance in case of the linear fusion policy occurs in general when the linear combination coefficient overly favours one constituent detector, or the sensitivity (threshold) is too high and the backtracking threshold ( $\theta$ ) is at its lowest value.

## 7. CONCLUSION AND FUTURE WORK

In this work, we applied several fusion techniques to aid the music onset detection task. Different fusion policies were tested and compared to their constituent methods, including the state-of-the-art SuperFlux method. A large scale evaluation was performed on two published datasets showing improvements as a result of fusion, without extra computational cost, or the need for a large amount of training data as in the case of machine learning based methods. A parameter search was used to find the optimal settings for each detector to yield the best performance.

We found that some of the best performing configurations do not match the default settings of some previously published algorithms. This suggests that in some cases, better performance can be achieved just by finding better settings which work best overall for a given type of audio even without changing the algorithms.

In future work, a possible improvement in case of late decision fusion is to take the magnitude of the peaks into account when combining detected onsets, essentially treating the value as an estimation confidence. We will investigate the dependency of the selection of onset detectors on the type and the quality of the input music signal. We also intend to carry out more rigorous statistical analyses with significance tests for the reported results. More parameters could be included in the search to study their strengths as well as how they influence each other under different configurations. Another interesting direction is to incorporate more Non-Western music types as detection target and design algorithms using instrument specific priors.

## 8. REFERENCES

- [1] J.P. Bello, L. Daudet, S. Abdallan, C. Duxbury, and M. Davies. A tutorial on onset detection in music signals. In *IEEE Transactions on Audio, Speech, and Language Processing*, volume 13, 2005.
- [2] S. Böck, F. Krebs, and M. Schedl. Evaluating the online capabilities of onset detection methods. In *Proc. of the 13th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2012.
- [3] S. Böck and G. Widmer. Local group delay based vibrato and tremolo suppression for onset detection. In *Proc. of the 14th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2013.
- [4] S. Böck and G. Widmer. Maximum filter vibrato suppression for onset detection. In *Proc. of the 16th Int. Conf. on Digital Audio Effects (DAFx)*, 2013.
- [5] C. Cannam, M.O. Jewell, C. Rhodes, M. Sandler, and M. d'Inverno. Linked data and you: Bringing music research software into the semantic web. *Journal of New Music Research*, 2010.
- [6] N. Collins. A comparison of sound onset detection algorithms with emphasis on psychoacoustically motivated detection functions. In *Proc. of the 118th Convention of the Audio Engineering Society*, 2005.
- [7] N. Collins. Using a pitch detector for onset detection. In *Proc. of the 6th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2005.
- [8] N. Degara-Quintela, A. Pena, and S. Torres-Guijarro. A comparison of score-level fusion rules for onset detection in music signals. In *Proc. of the 10th Int. Soc. for Music Information Retrieval Conf. (ISMIR)*, 2009.
- [9] S. Dixon. Onset detection revisited. In *Proc. of the 9th Int. Conference on Digital Audio Effects (DAFx'06)*, 2006.
- [10] A. Holzapfel, Y. Stylianou, A.C. Gedik, and B. Bozkurt. Three dimensions of pitched instrument onset detection. *IEEE Transactions on Audio, Speech, and Language Processing*, 2010.
- [11] L.A. Klein. *Sensor and data fusion: a tool for information assessment and decision making*. SPIE, 2004.
- [12] J. Makhoul. Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4), 1975.
- [13] J. Schlüter and S. Böck. Musical onset detection with convolutional neural networks. In *6th Int. Workshop on Machine Learning and Music (MML)*, 2013.
- [14] D. Stowell and M. Plumbley. Adaptive whitening for improved real-time audio onset detection. In *Proceedings of the International Computer Music Conference (ICMC)*, 2007.
- [15] M. Tian, A. Srinivasamurthy, M. Sandler, and X. Serra. A study of instrument-wise onset detection in beijing opera percussion ensembles. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.
- [16] J. Vos and R. Rasch. The perceptual onset of musical tones. *Perception & Psychophysics*, 29(4), 1981.
- [17] R. Zhou, M. Mattavelli, and G. Zoia. Music onset detection based on resonator time frequency image. *IEEE Transactions on Audio, Speech, and Language Processing*, 2008.

# EVALUATING THE EVALUATION MEASURES FOR BEAT TRACKING

**Mathew E. P. Davies**

Sound and Music Computing Group  
INESC TEC, Porto, Portugal  
mdavies@inesctec.pt

**Sebastian Böck**

Department of Computational Perception  
Johannes Kepler University, Linz, Austria  
sebastian.boeck@jku.at

## ABSTRACT

The evaluation of audio beat tracking systems is normally addressed in one of two ways. One approach is for human listeners to judge performance by listening to beat times mixed as clicks with music signals. The more common alternative is to compare beat times against ground truth annotations via one or more of the many objective evaluation measures. However, despite a large body of work in audio beat tracking, there is currently no consensus over which evaluation measure(s) to use, meaning multiple accuracy scores are typically reported. In this paper, we seek to evaluate the evaluation measures by examining the relationship between objective accuracy scores and human judgements of beat tracking performance. First, we present the raw correlation between objective scores and subjective ratings, and show that evaluation measures which allow alternative metrical levels appear more correlated than those which do not. Second, we explore the effect of parameterisation of objective evaluation measures, and demonstrate that correlation is maximised for smaller tolerance windows than those currently used. Our analysis suggests that true beat tracking performance is currently being over-estimated via objective evaluation.

## 1. INTRODUCTION

Evaluation is a critical element of music information retrieval (MIR) [16]. Its primary use is a mechanism to determine the individual and comparative performance of algorithms for given MIR tasks towards improving them in light of identified strengths and weaknesses. Each year many different MIR systems are formally evaluated within the MIREX initiative [6].

In the context of beat tracking, the concept and purpose of evaluation can be addressed in several ways. For example, to measure reaction time across changing tempi [2], to identify challenging musical properties for beat trackers [9] or to drive the composition of new test datasets [10]. However, as with other MIR tasks, evaluation in beat tracking is most commonly used to estimate the performance of one or more algorithms on a test dataset.

This measurement of performance can happen via subjective listening test, where human judgements are used to determine beat tracking performance [3], to discover: *how perceptually accurate the beat estimates are when mixed with the input audio*. Alternatively, objective evaluation measures can be used to compare beat times with ground truth annotations [4], to determine: *how consistent the beat estimates are with the ground truth according to some mathematical relationship*. While undertaking listening tests and annotating beat locations are both extremely time-consuming tasks, the apparent advantage of the objective approach is that once ground truth annotations have been determined, they can easily be re-used without the need for repeated listening experiments. However, the usefulness of any given objective accuracy score (of which there are many [4]) is contingent on its ability to reflect human judgement of beat tracking performance. Furthermore, for the entire objective evaluation process to be meaningful, we must rely on the inherent accuracy of the ground truth annotations.

In this paper we work under the assumption that musically trained experts can provide meaningful ground truth annotations and rather focus on the properties of the objective evaluation measures. The main question we seek to address is: *to what extent do existing objective accuracy scores reflect subjective human judgement of beat tracking performance?* In order to answer this question, even in principle, we must first verify that human listeners can make reliable judgements of beat tracking performance. While very few studies exist, we can find supporting evidence suggesting human judgements of beat tracking accuracy are highly repeatable [3] and that human listeners can reliably disambiguate accurate from inaccurate beat click sequences mixed with music signals [11].

The analysis we present involves the use of a test database for which we have a set of estimated beat locations, annotated ground truth and human subjective judgements of beat tracking performance. Access to all of these components (via the results of existing research [12, 17]) allows us to examine the correlation between objective accuracy scores, obtained by comparing the beat estimates to the ground truth, with human listener judgements. To the best of our knowledge this is the first study of this type for musical beat tracking.

The remainder of this paper is structured as follows. In Section 2 we summarise the objective beat tracking evaluation measures used in this paper. In Section 3 we describe



© Mathew E. P. Davies, Sebastian Böck.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Mathew E. P. Davies, Sebastian Böck. "Evaluating the evaluation measures for beat tracking", 15th International Society for Music Information Retrieval Conference, 2014.

the comparison between subjective ratings and objective scores of beat tracking accuracy. Finally, in Section 4 we present discussion and areas for future work.

## 2. BEAT TRACKING EVALUATION MEASURES

In this section we present a brief summary each of the evaluation measures from [4]. While nine different approaches were presented in [4], we reduce them to seven by only presenting the underlying approaches for comparing a set of beats with a set of annotations (i.e. ignoring alternate metrical interpretations). We consider the inclusion of different metrical interpretations of the annotations to be a separate process which can be applied to any of these evaluation measures (as in [5, 8, 15]), rather than a specific property of one particular approach. To this end, we choose three evaluation conditions: *Annotated* – comparing beats to annotations, *Annotated+Offbeat* – including the “off-beat” of the annotations for comparison against beats and *Annotated+Offbeat+D/H* – including the off-beat and both double and half the tempo of the annotations. This doubling and halving has been commonly used in beat tracking evaluation to attempt to reflect the inherent ambiguity in music over which metrical level to tap the beat [13]. The set of seven basic evaluation measures are summarised below:

**F-measure** : accuracy is determined through the proportion of *hits*, *false positives* and *false negatives* for a given annotated musical excerpt, where *hits* count as beat estimates which fall within a pre-defined tolerance window around individual ground truth annotations, *false positives* are extra beat estimates, and *false negatives* are missed annotations. The default value for the tolerance window is  $\pm 0.07$ s.

**PScore** : accuracy is measured as the normalised sum of the cross-correlation between two impulse trains, one corresponding to estimated beat locations, and the other to ground truth annotations. The cross-correlation is limited to the range covering 20% of the median inter-annotation-interval (IAI).

**Cemgil** : a Gaussian error function is placed around each ground truth annotation and accuracy is measured as the sum of the “errors” of the closest beat to each annotation, normalised by whichever is greater, the number of beats or annotations. The standard deviation of this Gaussian is set at 0.04s.

**Goto** : the annotation interval-normalised timing error is measured between annotations and beat estimates, and a binary measure of accuracy is determined based on whether a region covering 25% of the annotations continuously meets three conditions – the maximum error is less than  $\pm 17.5\%$  of the IAI, and the mean and standard deviation of the error are within  $\pm 10\%$  of the IAI.

**Continuity-based** : a given beat is considered accurate if it falls within a tolerance window placed around an annotation and that the previous beat also falls within the pre-

ceding tolerance window. In addition, a separate threshold requires that the estimated inter-beat-interval should be close to the IAI. In practice both thresholds are set at  $\pm 17.5\%$  of the IAI. In [4], two basic conditions consider the ratio of the longest continuously correct region to the length of the excerpt (CMLc), and the total proportion of correct regions (CMLt). In addition, the AMLc and AMLt versions allow for additional interpretations of the annotations to be considered accurate. As specified above, we reduce these four to two principal accuracy scores. To prevent any ambiguity, we rename these accuracy scores **Continuity-C** (CMLc) and **Continuity-T** (CMLt).

**Information Gain** : this method performs a two-way comparison of estimated beat times to annotations and vice-versa. In each case, a histogram of timing errors is created and from this the **Information Gain** is calculated as the Kullback-Leibler divergence from a uniform histogram. The default number of bins used in the histogram is 40.

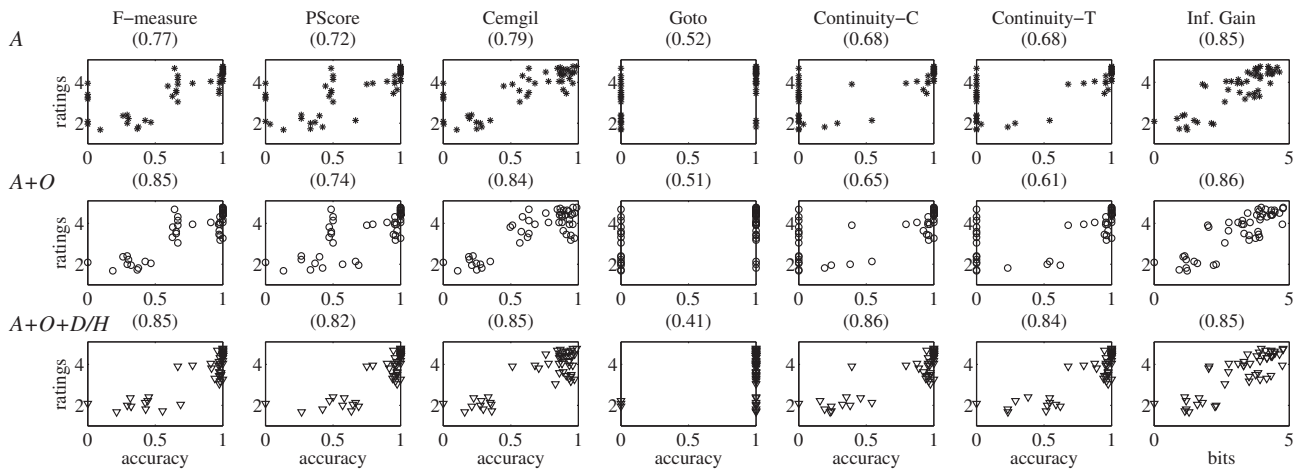
## 3. SUBJECTIVE VS. OBJECTIVE COMPARISON

### 3.1 Test Dataset

To facilitate the comparison of objective evaluation scores and subjective ratings we require a test dataset of audio examples for which we have both annotated ground truth beat locations and a set of human judgements of beat tracking performance for a beat tracking algorithm. For this purpose we use the test dataset from [17] which contains 48 audio excerpts (each 15s in duration). The excerpts were selected from the MillionSongSubset [1] according to a measurement of mutual agreement between a committee of five state of the art beat tracking algorithms. They cover a range from very low mutual agreement – shown to be indicative of beat tracking difficulty, up to very high mutual agreement – shown to be easier for beat tracking algorithms [10].

In [17] a listening experiment was conducted where a set of 22 participants listened to these audio examples mixed with clicks corresponding to automatic beat estimates and rated on a 1 to 5 scale how well they considered the clicks represented the beats present in the music. For each excerpt these beat times were the output of the beat tracker which most agreed with the remainder of the five committee members from [10]. Analysis of the subjective ratings and measurements of mutual agreement revealed low agreement to be indicative of poor subjective performance.

In a later study, these audio excerpts were used as one test set in a beat tapping experiment, where participants tapped the beat using a custom piece of software [12]. In order to compare the mutual agreement between tappers with their global performance against the ground truth, a musical expert annotated ground truth beat locations. The tempi range from 62 BPM (beats per minute) up to 181 BPM and, with the exception of two excerpts, all are in 4/4 time. Of the remaining two excerpts, one is in 3/4 time and



**Figure 1.** Subjective ratings vs. objective accuracy scores for different evaluation measures. The rows indicate different evaluation conditions. (top row) *Annotated*, (middle row) *Annotated+Offbeat*, and (bottom row) *Annotated+Offbeat+D/H*. For each scatter plot, the linear correlation coefficient is provided.

the other was deemed to have no beat at all, and therefore no beats were annotated.

In the context of this paper, this set of ground truth beat annotations provides the final element required to evaluate the evaluation measures, since we now have: i) automatically estimated beat locations, ii) subjective ratings corresponding to these beats and iii) ground truth annotations to which the estimated beat locations can be compared. We use each of the seven evaluation measures described in Section 2 to obtain the objective accuracy scores according to the three versions of the annotations: *Annotated*, *Annotated+Offbeat* and *Annotated+Offbeat+D/H*. Since all excerpts are short, and we are evaluating the output of an offline beat tracking algorithm, we remove the startup condition from [4] where beat times in the first five seconds are ignored.

## 3.2 Results

### 3.2.1 Correlation Analysis

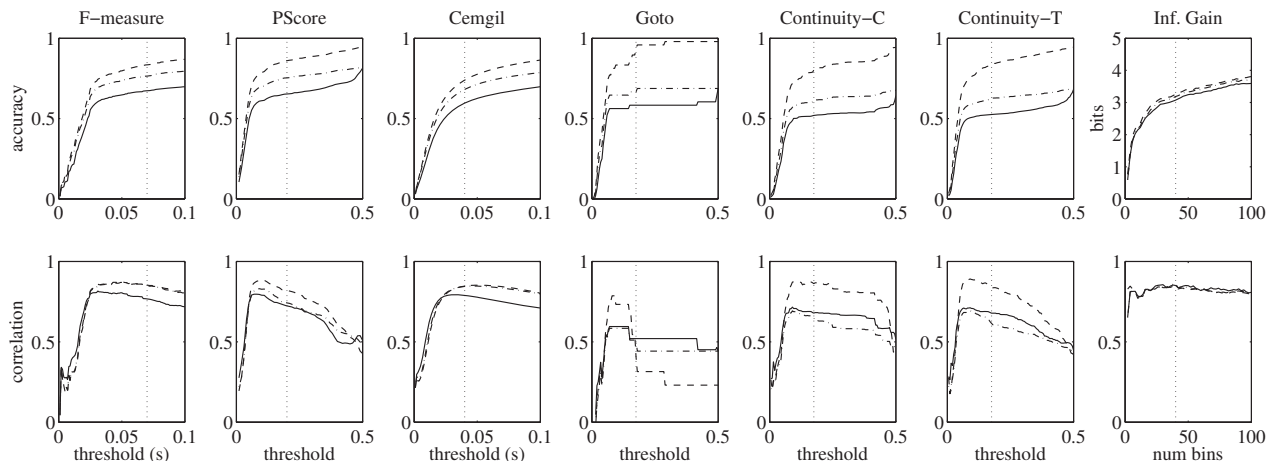
To investigate the relationship between the objective accuracy scores and subjective ratings, we present scatter plots in Figure 1. The title of each individual scatter plot includes the linear correlation coefficient which we interpret as an indicator of the validity of a given evaluation measure in the context of this dataset.

The highest overall correlation (0.86) occurs for **Continuity-C** when the offbeat and double/half conditions are included. However, for all but **Goto**, the correlation is greater than 0.80 once these additional evaluation criteria are included. It is important to note only **Continuity-C** and **Continuity-T** explicitly include these conditions in [4]. Since **Goto** provides a binary assessment of beat tracking performance, it is unlikely to be highly correlated with the subjective ratings from [17] where participants were explicitly required to use a five point scale rather than a good/bad response concerning beat tracking performance. Nevertheless, we retain it to maintain consistency with [4].

Comparing each individual measure across these evaluation conditions, reveals that **Information Gain** is least affected by the inclusion of additional interpretations of the annotations, and hence most robust to ambiguity over metrical level. Referring to the **F-measure** and **PScore** columns of Figure 1 we see that the “vertical” structure close to accuracies of 0.66 and 0.5 respectively is mapped across to 1 for the *Annotated+Offbeat+D/H* condition. This pattern is also reflected for **Goto**, **Continuity-C** and **Continuity-T** which also determine beat tracking accuracy according to fixed tolerance windows, i.e. a beat falling anywhere inside a tolerance window is perfectly accurate. However, the fact that a fairly uniform range of subjective ratings between 3 and 5 (i.e. “fair” to “excellent” [17]) exists for apparently perfect objective scores indicates a potential mismatch and over-estimation of beat tracking accuracy. While a better visual correlation appears to exist in the scatter plots of **Cemgil** and **Information Gain**, this is not reflected in the correlation values (at least not for the *Annotated+Offbeat+D/H* condition). The use of a Gaussian instead of a “top-hat” style tolerance window for **Cemgil** provides more information regarding the precise localisation of beats to annotations and hence does not have this clustering at the maximum performance. The **Information Gain** measure does not use tolerance windows at all, instead it measures beat tracking accuracy in terms of the temporal dependence between beats and annotations, and thus shows a similar behaviour.

### 3.2.2 The Effect of Parameterisation

For the initial correlation analysis, we only considered the default parameterisation of each evaluation measure as specified in [4]. However, to only interpret the validity of the evaluation measures in this way presupposes that they have already been optimally parameterised. We now explore whether this is indeed the case, by calculating the objective accuracy scores (under each evaluation condition) as a function of a threshold parameter for each measure.



**Figure 2.** (top row) Beat tracking accuracy as a function of threshold (or number of bins for Information Gain) per evaluation measure. (bottom row) Correlation between subjective ratings and accuracy scores as a function of threshold (or number of bins). In each plot the solid line indicates the *Annotated* condition, the dashed–dotted line shows *Annotated+Offbeat* and the dashed line shows *Annotated+Offbeat+D/H*. For each evaluation measure, the default parameterisation from [4] is shown by a dotted vertical line.

We then re-compute the subjective vs. objective correlation. We adopt the following parameter ranges as follows:

**F-measure** : the size of the tolerance window increases from  $\pm 0.001$ s to  $\pm 0.1$ s.

**PScore** : the width of the cross-correlation increases from 0.01 to 0.5 times the median IAI.

**Cemgil** : the standard deviation of the Gaussian error function grows from 0.001s to 0.1s.

**Goto** : to allow a similar one-dimensional representation, we make all three parameters identical and vary them from  $\pm 0.005$  to  $\pm 0.5$  times the IAI.

**Continuity-based** : the size of the tolerance window increases from  $\pm 0.005$  to  $\pm 0.5$  times the IAI.

**Information Gain** : we vary the number of bins in multiples of 2 from 2 up to 100.

In the top row of Figure 2 the objective accuracy scores as a function of different parameterisations are shown. The plots in the bottom row show the corresponding correlations with subjective ratings. In each plot the dotted vertical line indicates the default parameters. From the top row plots we can observe the expected trend that, as the size of the tolerance window increases so the objective accuracy scores increase. For the case of **Information Gain** the beat error histograms become increasingly sparse due to having more histogram bins than observations, hence the entropy reduces and the information gain increases. In addition, **Information Gain** does not have a maximum value of 1, but instead,  $\log_2$  of the number of histogram bins [4].

Looking at the effect of correlation with subjective ratings in the bottom row of Figure 2, we see that for most evaluation measures there is rapid increase in the correlation as the tolerance windows grow from very small sizes

	Default Parameters	Max. Correlation Parameters
<b>F-measure</b>	0.070s	0.049s
<b>PScore</b>	0.200	0.110
<b>Cemgil</b>	0.040s	0.051s
<b>Goto</b>	0.175	0.100
<b>Continuity-C</b>	0.175	0.095
<b>Continuity-T</b>	0.175	0.090
<b>Information Gain</b>	40	38

**Table 1.** Comparison of default parameters per evaluation measure with those which provide the maximum correlation with subjective ratings in the *Annotated+Offbeat+D/H* condition.

after which the correlation soon reaches its maximum and then reduces. Comparing these change points with the dotted vertical lines (which show the default parameters) we see that correlation is maximised for smaller (i.e. more restrictive) parameters than those currently used. By finding the point of maximum correlation in each of the plots in the bottom row of Figure 2 we can identify the parameters which yield the highest correlation between objective accuracy and subjective ratings. These are shown for the *Annotated+Offbeat+D/H* evaluation condition in Table 1 for which the correlation is typically highest. Returning to the plots in the top row of Figure 2 we can then read off the corresponding objective accuracy with the default and then maximum correlation parameters. These accuracy scores are shown in Table 2.

From these Tables we see that it is only **Cemgil** whose default parameterisation is lower than that which maximises the correlation. However this does not apply for the *Annotated* only condition which is implemented in [4]. While there is a small difference for **Information Gain**, in-



	<i>Annotated</i>		<i>Annotated+Offbeat</i>		<i>Annotated+Offbeat+D/H</i>	
	Default Params	Max Corr. Params	Default Params	Max Corr. Params	Default Params	Max Corr. Params
<b>F-measure</b>	0.673	0.607	0.764	0.738	0.834	0.797
<b>PScore</b>	0.653	0.580	0.753	0.694	0.860	0.792
<b>Cemgil</b>	0.596	0.559	0.681	0.702	0.739	0.779
<b>Goto</b>	0.583	0.563	0.667	0.646	0.938	0.813
<b>Continuity-C</b>	0.518	0.488	0.605	0.570	0.802	0.732
<b>Continuity-T</b>	0.526	0.505	0.624	0.587	0.837	0.754
<b>Information Gain</b>	3.078	2.961	3.187	3.187	3.259	3.216

**Table 2.** Summary of objective beat tracking accuracy under the three evaluation conditions: *Annotated*, *Annotated+Offbeat* and *Annotated+Offbeat+D/H* per evaluation measure. Accuracy is reported using the default parameterisation from [4] and also using the parameterisation which provides maximal correlation to the subjective ratings. For **Information Gain** only performance is measured in bits.

spection of Figure 2 shows that it is unaffected by varying the number of histogram bins in terms of the correlation. In addition, the inclusion of the extra evaluation criteria also leads to a negligible difference in reported accuracy. Therefore **Information Gain** is most robust to parameter sensitivity and metrical ambiguity. For the other evaluation measures the inclusion of the *Annotated+Offbeat* and the *Annotated+Offbeat+D/H* (in particular) leads to more pronounced differences. The highest overall correlation between objective accuracy scores and subjective ratings (0.89) occurs for **Continuity-T** for a tolerance window of  $\pm 9\%$  of the IAI rather than the default value of  $\pm 17.5\%$ . Referring again to Table 2 we see that this smaller tolerance window causes a drop in reported accuracy from 0.837 to 0.754. Indeed a similar drop in performance can be observed for most evaluation measures.

#### 4. DISCUSSION

Based on the analysis of objective accuracy scores and subjective ratings on this dataset of 48 excerpts, we can infer that: i) a higher correlation typically exists when the *Annotated+Offbeat* and/or *Annotated+Offbeat+D/H* conditions are included, and ii) for the majority of existing evaluation measures, this correlation is maximised for a more restrictive parameterisation than the default parameters which are currently used [4]. A strict following of the results presented here would promote either the use of **Continuity-T** for the *Annotated+Offbeat+D/H* condition with a smaller tolerance window, or **Information Gain** since it is most resilient to these variable evaluation conditions while maintaining a high subjective vs. objective correlation.

If we are to extrapolate these results to all existing work in the beat tracking literature this would imply that any papers reporting only performance for the *Annotated* condition using **F-measure** and **PScore** may not be as representative of subjective ratings (and hence true performance) as they could be by incorporating additional evaluation conditions. In addition, we could infer that most presented accuracy scores (irrespective of evaluation measure or evaluation condition) are somewhat inflated due to the use of artificially generous parameterisations. On this basis, we

might argue that the apparent glass ceiling of around 80% for beat tracking [10] (using **Continuity-T** for the *Annotated+Offbeat+D/H* condition) may in fact be closer to 75%, or perhaps lower still. In terms of external evidence to support our findings, a perceptual study evaluating human tapping ability [7] used a tolerance window of  $\pm 10\%$  of the IAI, which is much closer to our “maximum correlation” **Continuity-T** parameter of  $\pm 9\%$  than the default value of  $\pm 17.5\%$  of the IAI.

Before making recommendations to the MIR community with regard to how beat tracking evaluation should be conducted in the future, we should first revisit the makeup of the dataset to assess the scope from which we can draw conclusions. All excerpts are just 15s in duration, and therefore not only much shorter than complete songs, but also significantly shorter than most annotated excerpts in existing datasets (e.g. 40s in [10]). Therefore, based on our results, we cannot yet claim that our subjective vs. objective correlations will hold for evaluating longer excerpts. We can reasonably speculate that an evaluation across overlapping 15s windows could provide some local information about beat tracking performance for longer pieces, however this is currently not how beat tracking evaluation is addressed. Instead, a single score of accuracy is normally reported regardless of excerpt length. With the exception of [3] we are unaware of any other research where subjective beat tracking performance has been measured across full songs.

Regarding the composition of our dataset, we should also be aware that the excerpts were chosen in an unsupervised data-driven manner. Since they were sampled from a much larger collection of excerpts [1] we do not believe there is any intrinsic bias in their distribution other than any which might exist across the composition of the Million-SongSubset itself. The downside of this unsupervised sampling is that we do not have full control over exploring specific interesting beat tracking conditions such as off-beat tapping, expressive timing, the effect of related metrical levels and non-4/4 time-signatures. We can say that for the few test examples where the evaluated beat tracker tapped the off-beat (shown as zero accuracy points in the *Anno-*

tated condition but non-zero for the *Annotated+Offbeat* condition in Figure 1), were not rated as “bad”. Likewise, there did not appear to be a strong preference over a single metrical level. Interestingly, the ratings for the *unannotatable* excerpt were among the lowest across the dataset.

Overall, we consider this to be a useful pilot study which we intend to follow up in future work with a more targeted experiment across a much larger musical collection. In addition, we will also explore the potential for using bootstrapping measures from Text-IR [14] which have also been used for the evaluation of evaluation measures. Based on these outcomes, we hope to be in a position to make stronger recommendations concerning how best to conduct beat tracking evaluation, ideally towards a single unambiguous measurement of beat tracking accuracy. However, we should remain open to the possibility that different evaluation measures may be more appropriate than others and that this could depend on several factors, including: the goal of the evaluation; the types of beat tracking systems evaluated; how the ground truth was annotated; and the make up of the test dataset.

To summarise, we believe the main contribution of this paper is to further raise the profile and importance of evaluation in MIR, and to encourage researchers to more strongly consider the properties of evaluation measures, rather than merely reporting accuracy scores and assuming them to be valid and correct. If we are to improve underlying analysis methods through iterative evaluation and refinement of algorithms, it is critical to optimise performance according to meaningful evaluation methodologies targeted towards specific scientific questions.

While the analysis presented here has only been applied in the context of beat tracking, we believe there is scope for similar subjective vs. objective comparisons in other MIR topics such as chord recognition or structural segmentation, where subjective assessments should be obtainable via similar listening experiments to those used here.

## 5. ACKNOWLEDGMENTS

This research was partially funded by the Media Arts and Technologies project (MAT), NORTE-07-0124-FEDER-000061, financed by the North Portugal Regional Operational Programme (ON.2-O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT) as well as FCT post-doctoral grant SFRH/BPD/88722/2012. It was also supported by the European Union Seventh Framework Programme FP7 / 2007-2013 through the GiantSteps project (grant agreement no. 610591).

## 6. REFERENCES

- [1] T. Bertin-Mahieux, D. P. W. Ellis, B. Whitman, and P. Lamere. The million song dataset. In *Proceedings of 12th International Society for Music Information Retrieval Conference*, pages 591–596, 2011.
- [2] N. Collins. *Towards Autonomous Agents for Live Computer Music: Realtime Machine Listening and Interactive Music Systems*. PhD thesis, Centre for Music and Science, Faculty of Music, Cambridge University, 2006.
- [3] R. B. Dannenberg. Toward automated holistic beat tracking, music analysis, and understanding. In *Proceedings of 6th International Conference on Music Information Retrieval*, pages 366–373, 2005.
- [4] M. E. P. Davies, N. Degara, and M. D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. Technical Report C4DM-TR-09-06, Queen Mary University of London, Centre for Digital Music, 2009.
- [5] S. Dixon. Evaluation of audio beat tracking system beatroot. *Journal of New Music Research*, 36(1):39–51, 2007.
- [6] J. S. Downie. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008.
- [7] C. Drake, A. Penel, and E. Bigand. Tapping in time with mechanically and expressively performed music. *Music Perception*, 18(1):1–23, 2000.
- [8] M. Goto and Y. Muraoka. Issues in evaluating beat tracking systems. In *Working Notes of the IJCAI-97 Workshop on Issues in AI and Music - Evaluation and Assessment*, pages 9–16, 1997.
- [9] P. Grosche, M. Müller, and C. S. Sapp. What Makes Beat Tracking Difficult? A Case Study on Chopin Mazurkas. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 649–654, 2010.
- [10] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech and Language Processing*, 20(9):2539–2460, 2012.
- [11] J. R. Iversen and A. D. Patel. The beat alignment test (BAT): Surveying beat processing abilities in the general population. In *Proceedings of the 10th International Conference on Music Perception and Cognition*, pages 465–468, 2008.
- [12] M. Miron, F. Gouyon, M. E. P. Davies, and A. Holzapfel. Beat-Station: A real-time rhythm annotation software. In *Proceedings of the Sound and Music Computing Conference*, pages 729–734, 2013.
- [13] D. Moelants and M. McKinney. Tempo perception and musical content: what makes a piece fast, slow or temporally ambiguous? In *Proceedings of the 8th International Conference on Music Perception and Cognition*, pages 558–562, 2004.
- [14] T. Sakai. Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the International ACM SIGIR conference on research and development in information retrieval*, pages 525–532, 2006.
- [15] A. M. Stark. *Musicians and Machines: Bridging the Semantic Gap in Live Performance*. PhD thesis, Centre for Digital Music, Queen Mary University of London, 2011.
- [16] J. Urbano, M. Schedl, and X. Serra. Evaluation in Music Information Retrieval. *Journal of Intelligent Information Systems*, 41(3):345–369, 2013.
- [17] J. R. Zapata, A. Holzapfel, M. E. P. Davies, J. L. Oliveira, and F. Gouyon. Assigning a confidence threshold on automatic beat annotation in large datasets. In *Proceedings of 13th International Society for Music Information Retrieval Conference*, pages 157–162, 2012.

# IMPROVING RHYTHMIC TRANSCRIPTIONS VIA PROBABILITY MODELS APPLIED POST-OMR

**Maura Church**  
Applied Math, Harvard University  
and Google Inc.  
maura.church@gmail.com

**Michael Scott Cuthbert**  
Music and Theater Arts  
M.I.T.  
cuthbert@mit.edu

## ABSTRACT

Despite many improvements in the recognition of graphical elements, even the best implementations of Optical Music Recognition (OMR) introduce inaccuracies in the resultant score. These errors, particularly rhythmic errors, are time consuming to fix. Most musical compositions repeat rhythms between parts and at various places throughout the score. Information about rhythmic self-similarity, however, has not previously been used in OMR systems.

This paper describes and implements methods for using the prior probabilities for rhythmic similarities in scores produced by a commercial OMR system to correct rhythmic errors which cause a contradiction between the notes of a measure and the underlying time signature. Comparing the OMR output and post-correction results to hand-encoded scores of 37 polyphonic pieces and movements (mostly drawn from the classical repertory), the system reduces incorrect rhythms by an average of 19% (min: 2%, max: 36%).

The paper includes a public release of an implementation of the model in `music21` and also suggests future refinements and applications to pitch correction that could further improve the accuracy of OMR systems.

## 1. INTRODUCTION

Millions of paper copies of musical scores are found in libraries and archival collections and hundreds of thousands of scores have already been scanned as PDFs in repositories such as IMSLP [5]. A scan of a score cannot, however, be searched or manipulated musically, so Optical Music Recognition (OMR) software is necessary to transform an image of a score into symbolic formats (see [7] for a recent synthesis of relevant work and extensive bibliography; only the most relevant citations from this work are included here). Projects such as Peachnote [10] show both the feasibility of recognizing large bodies of scores and also the limitations that errors introduce, par-

ticularly in searches such as chord progressions that rely on accurate recognition of multiple musical staves.

Understandably, the bulk of OMR research has focused on improving the algorithms for recognizing graphical primitives and converting them to musical objects based on their relationships on the staves. Improving score accuracy using musical knowledge (models of tonality, meter, form) has largely been relegated to “future work” sections and when discussed has focused on localized structures such as beams and measures and requires access to the “guts” of a recognition engine (see Section 6.2.2 in [9]). Improvements to score accuracy based on the output of OMR systems using multiple OMR engines have been suggested [2] and when implemented yielded results that were more accurate than individual OMR engines, though the results were not statistically significant compared to the best commercial systems [1]. Improving the accuracy of an OMR score using musical knowledge and a single engine’s output alone remains an open field.

This paper proposes using rhythmic repetition and similarity within a score to create a model where measure-level metrical errors can be fixed using correctly recognized (or at least metrically consistent) measures found in other places in the same score, creating a self-healing method for post-OMR processing conditioned on probabilities based on rhythmic similarity and statistics of symbolic misidentification.

## 2. PRIOR PROBABILITIES OF DISTANCE

Most Western musical scores, excepting those in certain post-common practice styles (e.g., Boulez, Cage), use and gain cohesion through a limited rhythmic vocabulary across measures. Rhythms are often repeated immediately or after a fixed distance (e.g., after a 2, 4, or 8 measure distance). In a multipart score, different instruments often employ the same rhythms in a measure or throughout a passage. From a parsed musical score, it is not difficult to construct a hash of the sequence of durations in each measure of each part (hereafter simply called “measure”; “measure stack” will refer to measures sounding together across all parts); if grace notes are handled separately, and interior voices are flattened (e.g., using the `music21 chordify` method) then hash-key collisions will only occur in the rare cases where two graphically distinct symbols equate to the same length in quarter notes (such as a dotted-triplet eighth note and a normal eighth).

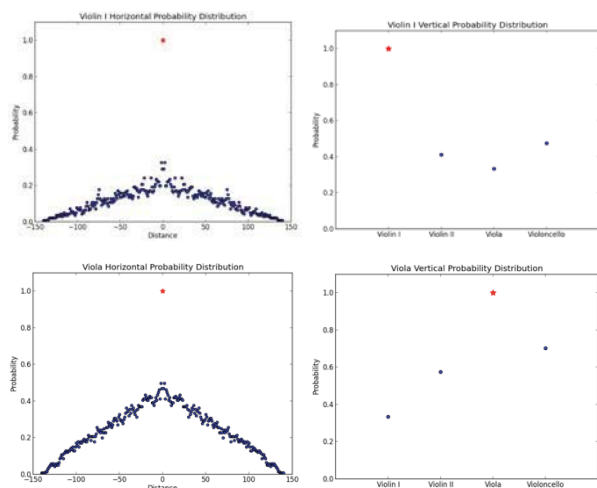


© Maura Church, Michael Scott Cuthbert.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Maura Church and Michael Scott Cuthbert. “Improving Rhythmic Transcriptions via Probability Models Applied Post-OMR”, 15th International Society for Music Information Retrieval Conference, 2014.

Within each part, the prior probability that a measure  $m_0$  will have the same rhythm as the measure  $n$  bars later (or earlier) can be computed (the prior-based-on-distance, or PrD). Similarly, the prior probability that, within a measure stack, part  $p$  will have the same rhythm as part  $q$  can also be computed (the prior-based-on-part, or PrP).

Figure 1 shows these two priors for the violin I and viola parts of the first movement of Mozart K525 (*Eine kleine Nachtmusik*). Individual parts have their own characteristic shapes; for instance, the melodic violin I (*top left*), shows less rhythmic similarity overall than the viola (*bot. left*). This difference results from the greater rhythmic variety of the violin I part compared to the viola part. Moments of large-scale repetition such as between the exposition and recapitulation, however, are easily visible as spikes in the PrD graph for violin I. (Possible refinements to the model taking into account localized similarities are given at the end of this paper.) The PrP graphs (*right*) show that both parts are more similar to the violoncello part than to any other part. However, the viola is more similar to the cello (and to violin II) than violin I is to any other part.



**Figure 1.** Priors based on distance ( $l$  in measure separation) and part ( $r$ ) for the violin I (*top*) and viola (*bot.*) parts in Mozart, K525.

### 3. PRIOR PROBABILITIES OF CHANGE

#### 3.1 Individual Change Probabilities

The probability that any given musical glyph will be read correctly or incorrectly is dependent on the quality of scan, the quality of original print, the OMR engine used, and the type of repertory. One possible generalization used in the literature [8] is to classify errors as class confusion (e.g., rest for note, with probability of occurring  $c$ ), omissions (e.g., of whole symbols or of dots, tuplet marks: probability  $o$ ), additions ( $a$ ), and general value confusion (e.g., quarter for eighth:  $v$ ). Other errors, such as sharp for natural or tie for slur, do not affect rhythmic accuracy. Although accuracy would be improved by

computing these values independently for each OMR system and quality of scan, such work is beyond the scope of the current paper. Therefore, we use Rossant and Bloch's recognition rates, adjusting them for the differences between working with individual symbols (such as dots and note stems) and symbolic objects (such as dotted-eighth and quarter notes). The values used in this model are thus:  $c = .003$ ,  $o = .009$ ,  $a = .004$ ,  $v = .016$ .<sup>1</sup> As will become clear, more accurate measures would only improve the results given below. Subtracting these probabilities from 1.0, the rate of equality,  $e$ , is .968.

#### 3.2 Aggregate Change Distances

The similarity of two measures can be calculated in a number of different ways, including the earth mover distance, the Hamming distance, and the minimum Levenshtein or edit distance. The nature of the change probabilities obtained from Rossant and Bloch along with the inherent difficulties of finding the one-to-one correspondence of input and output objects required for other methods, made Levenshtein distance the most feasible method. The probability that certain changes would occur in a given originally scanned measure (source,  $S$ ) to transform it into the OMR output measure (destination,  $D$ ) is determined by finding, through an implementation of edit distance, values for  $i$ ,  $j$ ,  $k$ ,  $l$ , and  $m$  (for number of class changes, omissions, additions, value changes, and unchanged elements) that maximize:

$$p_{S,D} = c^i \cdot o^j \cdot a^k \cdot v^l \cdot e^m \quad (1)$$

Equation (1), the prior-based-on-changes or PrC, can be used to derive a probability of rhythmic change due to OMR errors between any two arbitrary measures, but the model employed here concerns itself with measures with incorrect rhythms, or flagged measures.

#### 3.3 Flagged Measures

Let  $F_{P_i}$  be the set of flagged measures for part  $P_i$ , that is, measures whose total durations do not correspond to the total duration implied by the currently active time signature, and  $F = \{F_{P_1}, \dots, F_{P_j}\}$  for a score with  $j$  parts. (Measure stacks where each measure number is in  $F$  can be removed as probable pickup or otherwise intended incomplete measures, and long stretches of measures in  $F$  in all parts can be attributed to incorrectly identified time signatures and reevaluated, though neither of these refinements is used in this model). It is possible for rhythms within a measure to be incorrectly recognized without the entire measure being in  $F$ ; though this problem only arises in the rare case where two rhythmic errors cancel out each other (as in a dotted quarter read as a quarter with an eighth read as a quarter in the same measure).

<sup>1</sup> Rossant and Bloch give probabilities of change given that an error has occurred. The numbers given here are renormalizations of those error rates after removing the prior probability that an error has taken place.

#### 4. INTEGRATING THE PRIORS

For each  $m \in F_{P_i}$ , the measure  $n$  in part  $P_i$  with the highest likelihood of representing the prototype source rhythm before OMR errors were introduced is the source measure  $S_D$  that maximizes the product of the prior-based-on-distance, that is, the horizontal model, and the prior-based-on-changes:

$$S_D = \operatorname{argmax}(\operatorname{PrD}_n \cdot \operatorname{PrC}_n) \forall n \notin F. \quad (2)$$

(In the highly unlikely case of equal probabilities, a single measure is chosen arbitrarily) Similarly, for each  $m$  in  $F_P$  the measure  $t$  in the measure stack corresponding to  $m$ , with the highest likelihood of being the source rhythm for  $m$ , is the source measure  $S_P$  that maximizes the product of the prior-based-on-part, that is, the vertical model, and the prior-based-on-changes:

$$S_P = \operatorname{argmax}(\operatorname{PrP}_t \cdot \operatorname{PrC}_t) \forall t \notin F. \quad (3)$$

Since the two priors  $\operatorname{PrD}$  and  $\operatorname{PrP}$  have not been normalized in any way, the best match from  $S_D$  and  $S_P$  can be obtained by simply taking the maximum of the two:

$$S = \operatorname{argmax}(P(m)) \forall m \text{ in } [S_D, S_P] \quad (4)$$

Given the assumption that the time signature and barlines have accurately been obtained and that each measure originally contained notes and rests whose total durations matched the underlying meter, we do not need to be concerned with whether  $S$  is a “better” solution for correcting  $m$  than the rhythms currently in  $m$ , since the probability of a flagged measure being correct is zero. Thus any solution has a higher likelihood of being correct than what was already there. (Real-world implementations, however, may wish to place a lower bound on  $P(S)$  to avoid substitutions that are below a minimum threshold to prevent errors being added that would be harder to fix than the original.)

#### 5. EXAMPLE

In this example from Mozart K525, mvmt. 1, measure stack 17, measures in both Violin I and Violin II have been flagged as containing rhythmic errors (marked in purple in Figure 2).

Both the OMR software and our implementation of the method, described below, can identify the violin lines as containing rhythmic errors, but neither can know that an added dot in each part has caused the error. The vertical model ( $\operatorname{PrP} * \operatorname{PrC}$ ) will look to the viola and cello parts for corrections to the violin parts. Violin II and viola share five rhythms ( $e^5$ ) and only one omission of a dot is required to transform the viola rhythm into violin II ( $o^1$ ), for a  $\operatorname{PrC}$  of 0.0076. The prior on similarities between violin II and viola ( $\operatorname{PrP}$ ) is 0.57, so the complete probability of this transformation is 0.0043. The prior on similarities between violin II and cello is slightly higher, 0.64, but the



Figure 2. Mozart, K525 I, in OMR ( $l$ ) and scanned ( $r$ ) versions.

prior based on changes is much smaller ( $4 \cdot 10^{-9}$ ). Violin I is not considered as a source since its measure has also been flagged as incorrect. Therefore the viola’s measure is used for  $S_P$ .

A similar search is done for the other (unflagged) measures in the rest of the violin II part in order to find  $S_D$ . In this case, the probability of  $S_P$  exceeds that of  $S_D$ , so the viola measure’s rhythm is, correctly, used for violin II.

#### 6. IMPLEMENTATION

The model developed above was implemented using conversion and score manipulation routines from the open-source Python-based toolkit, `music21` [4] and has been contributed back to the toolkit as the `omr.correctors` module in v.1.9 and above. Example 1 demonstrates a round-trip in MusicXML of a raw OMR score to a post-processed score.

```
from music21 import *
s = converter.parse('/tmp/k525omrIn.xml')
sc = omr.correctors.ScoreCorrector(s)
s2 = sc.run()
s2.write('xml', fp='/tmp/k525post.xml')
```

Example 1. Python/`music21` code for correcting OMR errors in Mozart K525, I.

Figure 3, below, shows the types of errors that the model is able, and in some cases unable, to correct.

#### 7. RESULTS

Nine scores of four-movement quartets by Mozart (5),<sup>1</sup> Haydn (1), and Beethoven (4) were used for the primary evaluation. (Mozart K525, mvmt. 1 was used as a test score for development and testing but not for evaluation.) Scanned scores came from out-of-copyright editions (mainly Breitkopf & Härtel) via IMSLP and were converted to MusicXML using SmartScore X2 Pro (v.10.5.5). Ground truth encodings in MuseData and MusicXML formats came via the `music21` corpus originally from the Stanford’s CCARH repertoires [6] and Project Gutenberg.

<sup>1</sup> Mozart K156 is a three-movement quartet, however, both the ground truth and the OMR versions include the abandoned first version of the Adagio as a fourth movement.

The figure consists of three vertically stacked musical staves for measures 35-39 of Mozart's K525 I. The top staff is the original scan, showing some noise and irregularities. The middle staff is the SmartScore OMR output, with some notes circled in blue to indicate errors. The bottom staff is the post-OMR processed score, with five flags (1-5) indicating specific corrections or emendations. Flag 1 is a green bar over a measure, flag 2 is a green bar over a measure with a 'p' dynamic marking, flag 3 is a green bar over a measure, flag 4 is a red bar over a measure with a 'p' dynamic marking, and flag 5 is a red bar over a measure.

**Figure 3:** Comparison of Mozart K525 I, mm. 35–39 in the original scan (*top*), SmartScore OMR output (*middle*), and after post-OMR processing (*bot.*). Flags 1–3 were corrected successfully; Flags 4 and 5 result in metrically plausible but incorrect emendations. The model was able to preserve the correct pitches for Flags 2 (added quarter rest) and Flag 3 (added augmentation dot). Flag 1 (omitted eighth note) is considered correct in this evaluation, based solely on rhythm, even though the pitch of the reconstructed eighth note is not correct.

The pre-processed OMR movement was aligned with the ground truth by finding the minimum edit distance between measure hashes. This step was necessary for the many cases where the OMR version contained a different number of measures than the ground truth. The number of differences between the two versions of the same movement was recorded. A total of 29,728 measures with 7,196 flagged measures were examined. Flag rates ranged from 0.6% to 79.2% with a weighed mean of 24.2% and median of 21.7%.

The model was then run on each OMR movement and the number of differences with the ground truth was recorded again. (In order to make the outputted score useful for performers and researchers, we added a simple algorithm to preserve as much pitch information as possible from the original measure.) From 2.1% to 36.1% of flagged measures were successfully corrected, with a weighed mean of 18.8% and median of 18.0%: a substantial improvement over the original OMR output.

Manually checking the pre- and post-processed OMR scores against the ground truth showed that the highest rates of differences came from scores where single-pitch repetitions (tremolos) were spelled out in one source and written in abbreviated form in another; such differences could be corrected for in future versions. There was no significant correlation between the percentage of measures originally flagged and the correction rate ( $r = .17, p > .31$ ).

The model was also run on two scores outside the classical string quartet repertory to test its further relevance. On a fourteenth-century vocal work (transcribed into modern notation), *Gloria: Clemens Deus artifex* and the first movement of Schubert’s “Unfinished” symphony, the results were similar to the previous findings (16.8% and 18.7% error reduction, respectively).

The proportion of suggestions taken from the horizontal (PrD) and vertical models (PrP) depended significantly on the number of parts in the piece. In Mozart K525 quartet, 72% of the suggestions came from the horizontal model while for the Schubert symphony (fourteen parts), only 39% came from the horizontal model.

## 8. APPLICATIONS

The model has broad applications for improving the accuracy of scores already converted via OMR, but it would have greater impact as an element of an improved user experience within existing software. Used to its full potential, the model could help systems provide suggestions as users examine flagged measures. Even a small scale implementation could greatly improve the lengthy error-correcting process that currently must take place before a score is useable. See Figure 4 for an example interface.

The figure shows a screenshot of a music score interface. A yellow callout box with the text "Should this be:" is overlaid on a measure of the score. Below the callout box are two buttons: a green "Yes" button and a red "No" button. The score is written on multiple staves, with some notes highlighted in purple.

**Figure 4.** A sample interface improvement using the model described.

A similar model to the one proposed here could also be integrated into OMR software to offer suggestions for pitch corrections if the user selects a measure that was not flagged for rhythmic errors. Integration within OMR software would also potentially give the model access to

rejected interpretations for measures that may become more plausible when rhythmic similarity within a piece is taken into account.

The model could be expanded to take into account spatial separation between glyphs as part of the probabilities. Simple extensions such as ignoring measures that are likely pickups or correcting wrong time signatures and missed barlines (resulting in double-length measures) have already been mentioned. Autocorrelation matrices, which would identify repeating sections such as recapitulations and rondo returns, would improve the prior-based-on-distance metric. Although the model runs quickly on small scores (in far less than the time to run OMR despite the implementation being written in an interpreted language), on larger scores the  $O(\text{len}(F) \cdot \text{len}(\text{Part}))$  complexity of the horizontal model could become a problem (though correction of the lengthy Schubert score took less than ten minutes on an i7 MacBook Air). Because the prior-based-on-distance tends to fall off quickly, examining only a fixed-sized window worth of measures around each flagged measure would offer substantial speed-ups.

Longer scores and scores with more parts offered more possibilities for high-probability correcting measures. Thus we encourage the creators of OMR competitions and standard OMR test examples [3] to include entire scores taken from standard repertoires in their evaluation sets.

The potential of post-OMR processing based on musical knowledge is still largely untapped. Models of tonal behavior could identify transposing instruments and thus create better linkages between staves across systems that vary in the number of parts displayed. Misidentifications of time signatures, clefs, ties, and dynamics could also be reduced through comparison across parts and with similar sections in scores. While more powerful algorithms for graphical recognition will always be necessary, substantial improvements can be made quickly with the selective deployment of musical knowledge.

## 9. ACKNOWLEDGEMENTS

The authors thank the Radcliffe Institute of Harvard University, the National Endowment for the Humanities/Digging into Data Challenge, the Thomas Temple Hoopes Prize at Harvard, and the School of Humanities, Arts, and Social Sciences, MIT, for research support, four anonymous readers for suggestions, and Margo Levine, Beth Chen, and Suzie Clark of Harvard's Applied Math and Music departments for advice and encouragement.

## 10. REFERENCES

- [1] E. P. Bugge, et al.: "Using sequence alignment and voting to improve optical music recognition from multiple recognizers," *Proc. ISMIR*, Vol. 12, pp. 405–410, 2011.
- [2] D. Byrd, M. Schindele: "Prospects for improving OMR with multiple recognizers," *Proc. ISMIR*, Vol. 7, pp. 41–47, 2006.
- [3] D. Byrd, J. G. Simonsen, "Towards a Standard Testbed for Optical Music Recognition: Definitions, Metrics, and Page Images," [http://www.informatics.indiana.edu/donbyrd/Papers/OMRStandardTestbed\\_Final.pdf](http://www.informatics.indiana.edu/donbyrd/Papers/OMRStandardTestbed_Final.pdf), in progress.
- [4] M. Cuthbert and C. Ariza: "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," *Proc. ISMIR*, Vol. 11, pp. 637–42, 2010.
- [5] E. Guo et al.: *Petrucchi Music Library*, imslp.org, 2006–.
- [6] W. Hewlett, et al.: *MuseData: an Electronic Library of Classical Music Scores*, musedata.org, 1994, 2000.
- [7] A. Rebelo, et al.: "Optical music recognition: State-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, Vol. 1, No. 3, pp. 173–190, 2012.
- [8] F. Rossant and I. Bloch, "A fuzzy model for optical recognition of musical scores," *Fuzzy sets and systems*, Vol. 141, No. 2, pp. 165–201, 2004.
- [9] F. Rossant, I. Bloch: "Robust and adaptive OMR system including fuzzy modeling, fusion of musical rules, and possible error detection," *EURASIP Journal on Advances in Signal Processing*, 2007.
- [10] V. Viro: "Peachnote: Music score search and analysis platform," *Proc. ISMIR*, Vol. 12, pp. 359–362, 2011.

This Page Intentionally Left Blank



# CLASSIFYING EEG RECORDINGS OF RHYTHM PERCEPTION

Sebastian Stober, Daniel J. Cameron and Jessica A. Grahn

Brain and Mind Institute, Department of Psychology, Western University, London, ON, Canada  
 {sstober, dcamer25, jgrahn}@uwo.ca

## ABSTRACT

Electroencephalography (EEG) recordings of rhythm perception might contain enough information to distinguish different rhythm types/genres or even identify the rhythms themselves. In this paper, we present first classification results using deep learning techniques on EEG data recorded within a rhythm perception study in Kigali, Rwanda. We tested 13 adults, mean age 21, who performed three behavioral tasks using rhythmic tone sequences derived from either East African or Western music. For the EEG testing, 24 rhythms – half East African and half Western with identical tempo and based on a 2-bar 12/8 scheme – were each repeated for 32 seconds. During presentation, the participants' brain waves were recorded via 14 EEG channels. We applied stacked denoising autoencoders and convolutional neural networks on the collected data to distinguish African and Western rhythms on a group and individual participant level. Furthermore, we investigated how far these techniques can be used to recognize the individual rhythms.

## 1. INTRODUCTION

Musical rhythm occurs in all human societies and is related to many phenomena, such as the perception of a regular emphasis (i.e., beat), and the impulse to move one's body. However, the brain mechanisms underlying musical rhythm are not fully understood. Moreover, musical rhythm is a universal human phenomenon, but differs between human cultures, and the influence of culture on the processing of rhythm in the brain is uncharacterized.

In order to study the influence of culture on rhythm processing, we recruited participants in East Africa and Canada to test their ability to perceive and produce rhythms derived from East African and Western music. Besides behavioral tasks, which have already been discussed in [4], the East African participants also underwent electroencephalography (EEG) recording while listening to East African and Western musical rhythms thus enabling us to study the neural mechanisms underlying rhythm perception. We were interested in differences between neuronal entrainment to the periodicities in East African versus Western rhythms for participants from those respective cultures. Entrainment was defined as

the magnitudes of steady state evoked potentials (SSEPs) at frequencies related to the metrical structure of rhythms. A similar approach has been used previously to study entrainment to rhythms [17, 18].

But it is also possible to look at the collected EEG data from an information retrieval perspective by asking questions like *How well can we tell from the EEG whether a participant listened to an East African or Western rhythm?* or *Can we even say from a few seconds of EEG data which rhythm somebody listened to?* Note that answering such question does not necessarily require an understanding of the underlying processes. Hence, we have attempted to let a machine figure out how best to represent and classify the EEG recordings employing recently developed deep learning techniques. In the following, we will review related work in [Section 2](#), describe the data acquisition and pre-processing in [Section 3](#) present our experimental findings in [Section 4](#), and discuss further steps in [Section 5](#).

## 2. RELATED WORK

Previous research demonstrates that culture influences perception of the metrical structure (the temporal structure of strong and weak positions in rhythms) of musical rhythms in infants [20] and in adults [16]. However, few studies have investigated differences in brain responses underlying the cultural influence on rhythm perception. One study found that participants performed better on a recall task for culturally familiar compared to unfamiliar music, yet found no influence of cultural familiarity on neural activations while listening to the music while undergoing functional magnetic resonance imaging (fMRI) [15].

Many studies have used EEG and magnetoencephalography (MEG) to investigate brain responses to auditory rhythms. Oscillatory neural activity in the gamma (20-60 Hz) frequency band is sensitive to accented tones in a rhythmic sequence and anticipates isochronous tones [19]. Oscillations in the beta (20-30 Hz) band increase in anticipation of strong tones in a non-isochronous sequence [5, 6, 10]. Another approach has measured the magnitude of SSEPs (reflecting neural oscillations entrained to the stimulus) while listening to rhythmic sequences [17, 18]. Here, enhancement of SSEPs was found for frequencies related to the metrical structure of the rhythm (e.g., the frequency of the beat).

In contrast to these studies investigating the oscillatory activity in the brain, other studies have used EEG to investigate event-related potentials (ERPs) in responses to tones occurring in rhythmic sequences. This approach has been used to show distinct sensitivity to perturbations of the rhythmic pat-



tern vs. the metrical structure in rhythmic sequences [7], and to suggest that similar responses persist even when attention is diverted away from the rhythmic stimulus [12].

In the field of music information retrieval (MIR), retrieval based on brain wave recordings is still a very young and unexplored domain. So far, research has mainly focused on emotion recognition from EEG recordings (e.g., [3, 14]). For rhythms, however, Vlek et al. [23] already showed that imagined auditory accents can be recognized from EEG. They asked ten subjects to listen to and later imagine three simple metric patterns of two, three and four beats on top of a steady metronome click. Using logistic regression to classify accented versus unaccented beats, they obtained an average single-trial accuracy of 70% for perception and 61% for imagery. These results are very encouraging to further investigate the possibilities for retrieving information about the perceived rhythm from EEG recordings.

In the field of deep learning, there has been a recent increase of works involving music data. However, MIR is still largely under-represented here. To our knowledge, no prior work has been published yet on using deep learning to analyze EEG recordings related to music perception and cognition. However, there are some first attempts to process EEG recordings with deep learning techniques.

Wulsin et al. [24] used deep belief nets (DBNs) to detect anomalies related to epilepsy in EEG recordings of 11 subjects by classifying individual “channel-seconds”, i.e., one-second chunks from a single EEG channel without further information from other channels or about prior values. Their classifier was first pre-trained layer by layer as an autoencoder on unlabelled data, followed by a supervised fine-tuning with backpropagation on a much smaller labeled data set. They found that working on raw, unprocessed data (sampled at 256Hz) led to a classification accuracy comparable to hand-crafted features.

Langkvist et al. [13] similarly employed DBNs combined with a hidden Markov model (HMM) to classify different sleep stages. Their data for 25 subjects comprises EEG as well as recordings of eye movements and skeletal muscle activity. Again, the data was segmented into one-second chunks. Here, a DBN on raw data showed a classification accuracy close to one using 28 hand-selected features.

### 3. DATA ACQUISITION & PRE-PROCESSING

#### 3.1 Stimuli

African rhythm stimuli were derived from recordings of traditional East African music [1]. The author (DC) composed the Western rhythmic stimuli. Rhythms were presented as sequences of sine tones that were 100ms in duration with intensity ramped up/down over the first/final 50ms and a pitch of either 375 or 500 Hz. All rhythms had a temporal structure of 12 equal units, in which each unit could contain a sound or not. For each rhythmic stimulus, two individual rhythmic sequences were overlaid – each at a different pitch. For each cultural type of rhythm, there were 2 groups of 3 individual rhythms for which rhythms could be overlaid with the others in their group. Because an individual rhythm could be one

**Table 1.** Rhythmic sequences in groups of three that pairings were based on. All ‘x’s denote onsets. Larger, bold ‘X’s denote the beginning of a 12 unit cycle (downbeat).

Western Rhythms												
1	<b>X</b>	x	x	x	x	x	x	x	<b>X</b>	x	x	x
2	<b>X</b>		x	x	x	x	x	<b>X</b>		x	x	x
3	<b>X</b>	x	x	x	x	x	x	<b>X</b>	x	x	x	x
4	<b>X</b>	x	x	x	x	x	x	<b>X</b>	x	x	x	x
5	<b>X</b>	x	x	x	x	x	x	<b>X</b>	x	x	x	x
6	<b>X</b>	x	x	x	x	x	x	<b>X</b>	x	x	x	x
East African Rhythms												
1	<b>X</b>	x	x	x	x	x	x	<b>X</b>	x	x	x	x
2	<b>X</b>	x	x	x	x	x	x	<b>X</b>	x	x	x	x
3	<b>X</b>	x	x	x	x	x	x	<b>X</b>	x	x	x	x
4	<b>X</b>	x	x	x	x	x	x	<b>X</b>	x	x	x	x
5	<b>X</b>	x	x	x	x	x	x	<b>X</b>	x	x	x	x
6	<b>X</b>	x	x	x	x	x	x	<b>X</b>	x	x	x	x

of two pitches/sounds, this made for a total of 12 rhythmic stimuli from each culture, each used for all tasks. Furthermore, rhythmic stimuli could be one of two tempi: having a minimum inter-onset interval of 180 or 240ms.

#### 3.2 Study Description

Sixteen East African participants were recruited in Kigali, Rwanda (3 female, mean age: 23 years, mean musical training: 3.4 years, mean dance training: 2.5 years). Thirteen of these participated in the EEG portion of the study as well as the behavioral portion. All participants were over the age of 18, had normal hearing, and had spent the majority of their lives in East Africa. They all gave informed consent prior to participating and were compensated for their participation, as per approval by the ethics boards at the Centre Hospitalier Universitaire de Kigali and the University of Western Ontario. After completion of the behavioral tasks, electrodes were placed on the participant’s scalp. They were instructed to sit with eyes closed and without moving for the duration of the recording, and to maintain their attention on the auditory stimuli. All rhythms were repeated for 32 seconds, presented in counterbalanced blocks (all East African rhythms then all Western rhythms, or vice versa), and with randomized order within blocks. All 12 rhythms of each type were presented – all at the same tempo (fast tempo for subjects 1–3 and 7–9, and slow tempo for the others). Each rhythm was preceded by 4 seconds of silence. EEG was recorded via a portable Grass EEG system using 14 channels at a sampling rate of 400Hz and impedances were kept below 10kΩ.

#### 3.3 Data Pre-Processing

EEG recordings are usually very noisy. They contain artifacts caused by muscle activity such as eye blinking as well as possible drifts in the impedance of the individual electrodes over the course of a recording. Furthermore, the recording equipment is very sensitive and easily picks up interferences from the surroundings. For instance, in this experiment, the power supply dominated the frequency band around 50Hz. All these issues have led to the common practice to invest a lot of effort

into pre-processing EEG data, often even manually rejecting single frames or channels. In contrast to this, we decided to put only little manual work into cleaning the data and just removed obviously bad channels, thus leaving the main work to the deep learning techniques. After bad channel removal, 12 channels remained for subjects 1–5 and 13 for subjects 6–13.

We followed the common practice in machine learning to partition the data into *training*, *validation* (or model selection) and *test* sets. To this end, we split each 32s-long trial recording into three non-overlapping pieces. The first four seconds were used for the validation dataset. The rationale behind this was that we expected that the participants would need a few seconds in the beginning of each trial to get used to the new rhythm. Thus, the data would be less suited for training but might still be good enough to estimate the model accuracy on unseen data. The next 24 seconds were used for training and the remaining four seconds for testing.

The data was finally converted into the input format required by the neural networks to be learned.<sup>1</sup> If the network just took the raw EEG data, each waveform was normalized to a maximum amplitude of 1 and then split into equally sized frames matching the size of the network’s input layer. No windowing function was applied and the frames overlapped by 75% of their length. If the network was designed to process the frequency spectrum, the processing involved:

1. computing the short-time Fourier transform (STFT) with given window length of 64 samples and 75% overlap,
2. computing the log amplitude,
3. scaling linearly to a maximum of 1 (per sequence),
4. (optionally) cutting of all frequency bins above the number requested by the network,
5. splitting the data into frames matching the network’s input dimensionality with a given hop size of 5 to control the overlap.

Here, the number of retained frequency bins and the input length were considered as hyper-parameters.

## 4. EXPERIMENTS & FINDINGS

All experiments were implemented using Theano [2] and pylearn2 [8].<sup>2</sup> The computations were run on a dedicated 12-core workstation with two Nvidia graphics cards – a Tesla C2075 and a Quadro 2000.

As the first retrieval task, we focused on recognizing whether a participant had listened to an East African or Western rhythm (Section 4.1). This binary classification task is most likely much easier than the second task – trying to predict one out of 24 rhythms (Section 4.2). Unfortunately, due to the block design of the study, it was not possible to train a classifier for the tempo. Trying to do so would yield a classifier that “cheated” by just recognizing the inter-individual differences because every participant only listened to stimuli of the same tempo.

<sup>1</sup> Most of the processing was implemented through the *librosa* library available at <https://github.com/bmcfee/librosa/>.

<sup>2</sup> The code to run the experiments is publicly available as supplementary material of this paper at <http://dx.doi.org/10.6084/m9.figshare.1108287>

As the classes were perfectly balanced for both tasks, we chose the *accuracy*, i.e., the percentage of correctly classified instances, as evaluation measure. Accuracy can be measured on several levels. The network predicts a class label for each input frame. Each frame is a segment from the time sequence of a single EEG channel. Finally, for each trial, several channels were recorded. Hence, it is natural to also measure accuracy also at the sequence (i.e. channel) and trial level. There are many ways to aggregate frame label predictions into a prediction for a channel or a trial. We tested the following three ways to compute a score for each class:

- **plain:** sum of all 0-or-1 outputs per class
- **fuzzy:** sum of all raw output activations per class
- **probabilistic:** sum of log output activations per class

While the latter approach which gathers the log likelihoods from all frames worked best for a softmax output layer, it usually performed worse than the fuzzy approach for the DLSVM output layer with its hinge loss (see below). The plain approach worked best when the frame accuracy was close to the chance level for the binary classification task. Hence, we chose the plain aggregation scheme whenever the frame accuracy was below 52% on the validation set and otherwise the fuzzy approach.

We expected significant inter-individual differences and therefore made learning good individual models for the participants our priority. We then tested configuration that worked well for individuals on three groups – all participants as well as one group for each tempo, containing 6 and 7 subjects respectively.

### 4.1 Classification into African and Western Rhythms

#### 4.1.1 Multi-Layer Perceptron with Pre-Trained Layers

Motivated by the existing deep learning approaches for EEG data (cf. Section 2), we choose to pre-train a MLP as an autoencoder for individual channel-seconds – or similar fixed-length chunks – drawn from all subjects. In particular, we trained a stacked denoising autoencoder (SDA) as proposed in [22] where each individual input was set to 0 with a *corruption probability* of 0.2.

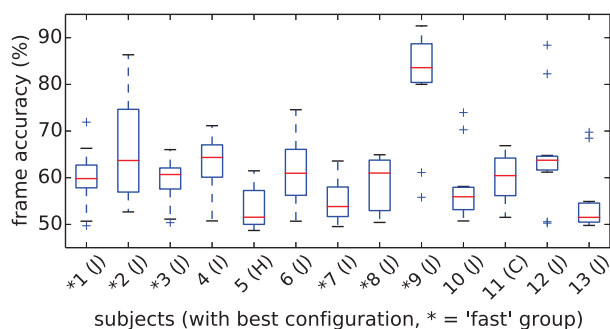
We tested several structural configurations, varying the input sample rate (400Hz or down-sampled to 100Hz), the number of layers, and the number of neurons in each layer. The quality of the different models was measured as the mean squared reconstruction error (MSRE). Table 2 gives an overview of the reconstruction quality for selected configurations. All SDAs were trained with tied weights, i.e., the weight matrix of each decoder layer equals the transpose of the respective encoder layer’s weight matrix. Each layer was trained with stochastic gradient descent (SGD) on mini-batches of 100 examples for a maximum of 100 epochs with an initial learning rate of 0.05 and exponential decay.

In order to turn a pre-trained SDA into a multilayer perceptron (MLP) for classification, we replaced the decoder part of the SDA with a DLSVM layer as proposed in [21].<sup>3</sup> This special kind of output layer for classification uses the hinge

<sup>3</sup> We used the experimental implementation for pylearn2 provided by Kyle Kastner at [https://github.com/kastnerkyle/pylearn2/blob/svm\\_layer/pylearn2/models/mlp.py](https://github.com/kastnerkyle/pylearn2/blob/svm_layer/pylearn2/models/mlp.py)

**Table 2.** MSRE and classification accuracy for selected SDA (top, A-F) and CNN (bottom, G-I) configurations.

id	neural network configuration (sample rate, input format, hidden layer sizes)	MSRE		MLP Classification Accuracy (for frames, channels and trials) in %											
		train	test	individ. subjects			fast (1–3, 7–9)			slow (4–6, 10–13)			all (1–13)		
A	100Hz, 100 samples, 50-25-10 (SDA for subject 2)	4.35	4.17	61.1	65.5	72.4	58.7	60.6	61.1	53.7	56.0	59.5	53.5	56.6	60.3
B	100Hz, 100 samples, 50-25-10	3.19	3.07	58.1	62.0	66.7	58.1	60.7	61.1	53.5	57.7	57.1	52.1	53.5	54.5
C	100Hz, 100 samples, 50-25	1.00	0.96	61.7	65.9	71.2	58.6	62.3	63.2	54.4	56.4	57.1	53.4	54.8	56.4
D	400Hz, 100 samples, 50-25-10	0.54	0.53	51.7	58.9	62.2	50.3	50.6	50.0	50.0	51.8	51.2	50.1	50.2	50.0
E	400Hz, 100 samples, 50-25	0.36	0.34	60.8	65.9	71.8	56.3	58.6	66.0	52.0	55.0	56.0	49.9	50.1	56.1
F	400Hz, 80 samples, 50-25-10	0.33	0.32	52.0	59.9	62.5	52.3	53.9	54.9	50.5	53.5	55.4	50.2	51.0	50.3
G	100Hz, 100 samples, 2 conv. layers			62.0	63.9	67.6	57.1	57.9	59.7	49.9	50.2	50.0	51.7	52.8	52.9
H	100Hz, 200 samples, 2 conv. layers			64.0	64.8	67.9	58.2	58.5	61.1	49.5	49.6	50.6	50.9	50.2	50.6
I	400Hz, 1s freq. spectrum (33 bins), 2 conv. layers			69.5	70.8	74.7	58.1	58.0	59.0	53.8	54.5	53.0	53.7	53.9	52.6
J	400Hz, 2s freq. spectrum (33 bins), 2 conv. layers			72.2	72.6	77.6	57.6	57.5	60.4	52.9	52.9	54.8	53.1	53.5	52.3

**Figure 1.** Boxplot of the frame-level accuracy for each individual subject aggregated over all configurations.<sup>5</sup>

loss as cost function and replaces the commonly applied softmax. We observed much smoother learning curves and a slightly increased accuracy when using this cost function for optimization together with rectification as non-linearity in the hidden layers. For training, we used SGD with dropout regularization [9] and momentum, a high initial learning rate of 0.1 and exponential decay over each epoch. After training for 100 epochs on minibatches of size 100, we selected the network that maximized the accuracy on the validation dataset. We found that the dropout regularization worked really well and largely avoided over-fitting to the training data. In some cases, even a better performance on the test data could be observed. The obtained mean accuracies for the selected SDA configurations are also shown in Table 2 for MLPs trained for individual subjects as well as for the three groups. As Figure 1 illustrates, there were significant individual differences between the subjects. Whilst learning good classifiers appeared to be easy for subject 9, it was much harder for subjects 5 and 13. As expected, the performance for the groups was inferior. Best results were obtained for the “fast” group, which comprised only 6 subjects including 2 and 9 who were amongst the easiest to classify.

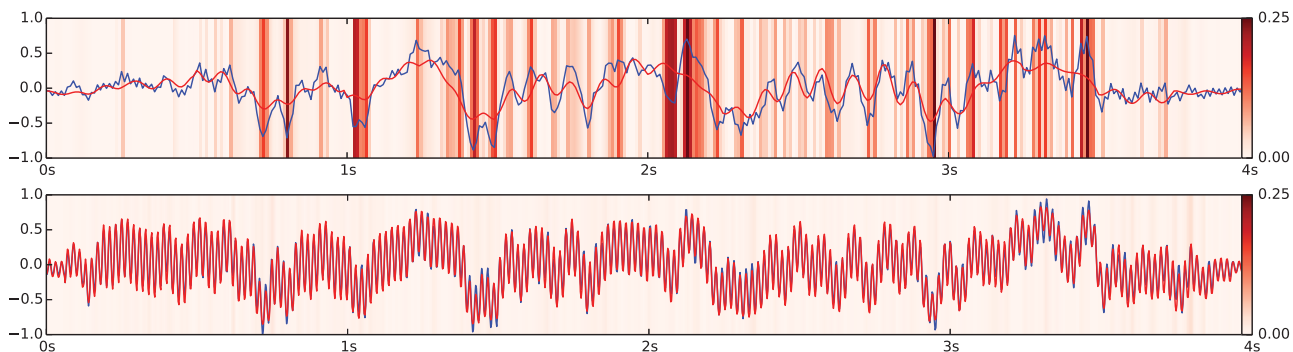
We found that two factors had a strong impact on the MSRE: the amount of (lossy) *compression* through the autoencoder’s bottleneck and the amount of *information* the

network processes at a time. Configurations A, B and D had the highest compression ratio (10:1). C and E lacked the third autoencoder layer and thus only compressed at 4:1 and with a lower resulting MSRE. F had exactly twice the compression ratio as C and E. While the difference in the MSRE was remarkable between F and C, it was much less so between F and E – and even compared to D. This could be explained by the four times higher sample rate of D–F. Whilst A–E processed the same amount of *samples* at a time, the input for A–C contained much more information as they were looking at 1s of the signal in contrast to only 250ms. Judging from the MSRE, the longer time span appears to be harder to compress. This makes sense as EEG usually contains most information in the lower frequencies and higher sampling rates do not necessarily mean more content. Furthermore, with growing size of the input frames, the variety of observable signal patterns increases and they become harder to approximate. Figure 2 illustrates the difference between two reconstructions of the same 4s raw EEG input segment using configurations B and D. In this specific example, the MSRE for B is ten times as high compared to D and the loss of detail in the reconstruction is clearly visible. However, D can only see 250ms of the signal at a time whereas B processes one channel-second.

Configuration A had the highest MSRE as it was only trained on data from subject 2 but needed to process all other subjects as well. Very surprisingly, the respective MLP produced much better predictions than B, which had identical structure. It is not clear what caused this effect. One explanation could be that the data from subject 2 was cleaner than for other participants as it also led to one amongst the best individual classification accuracies.<sup>6</sup> This could have led to more suitable features learned by the SDA. In general, the two-hidden-layer models worked better than the three-hidden-layer ones. Possibly, the compression caused by the third hidden layer was just too much. Apart from this, it was hard to make out a clear “winner” between A, C and E. There seemed to be a trade-off between the accuracy of the reconstruction (by choosing a smaller window size and/or higher sampling rate) and learning more suitable features

<sup>5</sup> Boxes show the 25th to 75th percentiles with a mark for the median within, whiskers span to furthest values within the 1.5 interquartile range, remaining outliers are shown as crossbars.

<sup>6</sup> Most of the model/learning parameters were selected by training just on subject 2.



**Figure 2.** Input (blue) and its reconstruction (red) for the same 4s sequence from the test data. The background color indicates the squared sample error. Top: Configuration B (100Hz) with MSRE 6.43. Bottom: Configuration D (400Hz) with MSRE 0.64. (The bottom signals shows more higher-frequency information due to the four-times higher sampling rate.)

**Table 3.** Structural parameters of the CNN configurations.

id	input		convolutional layer 1				convolutional layer 2			
	dim.	shape	patterns	pool	stride	shape	patterns	pool	stride	
G	100x1	15x1	10	7	1	70x1	10	7	1	
H	200x1	25x1	10	7	1	151x1	10	7	1	
I	22x33	1x33	20	5	1	9x1	10	5	1	
J	47x33	1x33	20	5	1	9x1	10	5	1	

for recognizing the rhythm type at a larger time scale. This led us to try a different approach using convolutional neural networks (CNNs) as, e.g., described in [11].

#### 4.1.2 Convolutional Neural Network

We decided on a general layout consisting of two convolutional layers where the first layer was supposed to pick up beat-related patterns while the second would learn to recognize higher-level structures. Again, a DLSVM layer was used for the output and the rectifier non-linearity in the hidden layers. The structural parameters are listed in Table 3. As pooling operation, the maximum was applied. Configurations G and H processed the same raw input as A–F whereas I and J took the frequency spectrum as input (using all 33 bins). All networks were trained for 20 epochs using SGD with a momentum of 0.5 and an exponential decaying learning rate initialized at 0.1.

The obtained accuracy values are listed in Table 2 (bottom). Whilst G and H produced results comparable to A–F, the spectrum-based CNNs, I and J, clearly outperformed all other configurations for the individual subjects. For all but subjects 5 and 11, they showed the highest frame-level accuracy (c.f. Figure 1). For subjects 2, 9 and 12, the trial classification accuracy was even higher than 90% (not shown).

#### 4.1.3 Cross-Trial Classification

In order to rule out the possibility that the classifiers just recognized the individual trials – and not the rhythms – by coincidental idiosyncrasies and artifacts unrelated to rhythm perception, we additionally conducted a cross-trial classification experiment. Here, we only considered all subjects with frame-level accuracies above 80% in the earlier experiments – i.e., subjects 2, 9 and 12. We formed 144 rhythm pairs by combining each East African with each Western rhythm from

the fast stimuli (for subjects 2 and 9) and the slow ones (for subject 12) respectively. For each pair, we trained a classifier with configuration J using all but the two rhythms of the pair.<sup>7</sup> Due to the amount of computation required, we trained only for 3 epochs each. With the learned classifiers, the mean frame-level accuracy over all 144 rhythm pairs was 82.6%, 84.5% and 79.3% for subject 2, 9 and 12 respectively. These values were only slightly below those shown in Figure 1, which we considered very remarkable after only 3 training epochs.

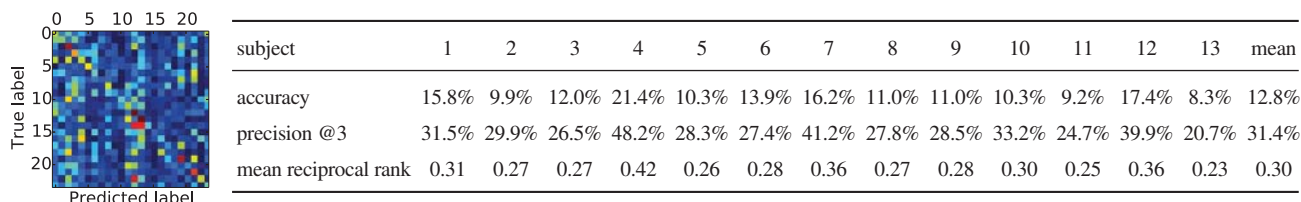
## 4.2 Identifying Individual Rhythms

Recognizing the correct rhythm amongst 24 candidates was a much harder task than the previous one – especially as all candidates had the same meter and tempo. The chance level for 24 evenly balanced classes was only 4.17%. We used again configuration J as our best known solution so far and trained an individual classifier for each subject. As Figure 3 shows, the accuracy on the 2s input frames was at least twice the chance level. Considering that these results were obtained without any parameter tuning, there is probably still much room for improvements. Especially, similarities amongst the stimuli should be considered as well.

## 5. CONCLUSIONS AND OUTLOOK

We obtained encouraging first results for classifying chunks of 1-2s recorded from a single EEG channel into East African or Western rhythms using convolutional neural networks (CNNs) and multilayer perceptrons (MLPs) pre-trained as stacked denoising autoencoders (SDAs). As it turned out, some configurations of the SDA (D and F) were especially suited to recognize unwanted artifacts like spikes in the waveforms through the reconstruction error. This could be elaborated in the future to automatically discard bad segments during pre-processing. Further, the classification accuracy for individual rhythms was significantly above chance level and encourages more research in this direction. As this has been an initial and by no means exhaustive exploration of the model- and learning parameter space, there seems to be a lot more potential – especially in CNNs processing the frequency spectrum – and

<sup>7</sup> Deviating from the description given in Section 3.3, we used the first 4s of each recording for validation and the remaining 28s for training as the test set consisted of full 32s from separate recordings in this special case.



**Figure 3.** Confusion matrix for all subjects (left) and per-subject performance (right) for predicting the rhythm (24 classes).

we will continue to look for better designs than those considered here. We are also planning to create publicly available data sets and benchmarks to attract more attention to these challenging tasks from the machine learning and information retrieval communities.

As expected, individual differences were very high. For some participants, we were able to obtain accuracies above 90%, but for others, it was already hard to reach even 60%. We hope that by studying the models learned by the classifiers, we may shed some light on the underlying processes and gain more understanding on why these differences occur and where they originate. Also, our results still come with a grain of salt: We were able to rule out side effects on a trial level by successfully replicating accuracies across trials. But due to the study's block design, there remains still the chance that unwanted external factors interfered with one of the two blocks while being absent during the other one. Here, the analysis of the learned models could help to strengthen our confidence in the results.

The study is currently being repeated with North America participants and we are curious to see whether we can replicate our findings. Furthermore, we want to extend our focus by also considering more complex and richer stimuli such as audio recordings of rhythms with realistic instrumentation instead of artificial sine tones.

**Acknowledgments:** This work was supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD), by the Natural Sciences and Engineering Research Council of Canada (NSERC), through the Western International Research Award R4911A07, and by an AUCC Students for Development Award.

## 6. REFERENCES

- [1] G.F. Barz. *Music in East Africa: experiencing music, expressing culture*. Oxford University Press, 2004.
- [2] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proc. of the Python for Scientific Computing Conference (SciPy)*, 2010.
- [3] R. Cabredo, R.S. Legaspi, P.S. Inventado, and M. Numao. An emotion model for music using brain waves. In *ISMIR*, pages 265–270, 2012.
- [4] D.J. Cameron, J. Bentley, and J.A. Grahn. Cross-cultural influences on rhythm processing: Reproduction, discrimination, and beat tapping. *Frontiers in Human Neuroscience*, to appear.
- [5] T. Fujioka, L.J. Trainor, E.W. Large, and B. Ross. Beta and gamma rhythms in human auditory cortex during musical beat processing. *Annals of the New York Academy of Sciences*, 1169(1):89–92, 2009.
- [6] T. Fujioka, L.J. Trainor, E.W. Large, and B. Ross. Internalized timing of isochronous sounds is represented in neuromagnetic beta oscillations. *The Journal of Neuroscience*, 32(5):1791–1802, 2012.
- [7] E. Geiser, E. Ziegler, L. Jancke, and M. Meyer. Early electrophysiological correlates of meter and rhythm processing in music perception. *Cortex*, 45(1):93–102, 2009.
- [8] I.J. Goodfellow, D. Warde-Farley, P. Lamblin, V. Dumoulin, M. Mirza, R. Pascanu, J. Bergstra, F. Bastien, and Y. Bengio. Pylearn2: a machine learning research library. *arXiv preprint arXiv:1308.4214*, 2013.
- [9] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- [10] J.R. Iversen, B.H. Repp, and A.D. Patel. Top-down control of rhythm perception modulates early auditory responses. *Annals of the New York Academy of Sciences*, 1169(1):58–73, 2009.
- [11] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.
- [12] O. Ladinig, H. Honing, G. Háden, and I. Winkler. Probing attentive and preattentive emergent meter in adult listeners without extensive music training. *Music Perception*, 26(4):377–386, 2009.
- [13] M. Långkvist, L. Karlsson, and M. Loutfi. Sleep stage classification using unsupervised feature learning. *Advances in Artificial Neural Systems*, 2012:5–5:5, Jan 2012.
- [14] Y.-P. Lin, T.-P. Jung, and J.-H. Chen. EEG dynamics during music appreciation. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual Int. Conf. of the IEEE*, pages 5316–5319, 2009.
- [15] S.J. Morrison, S.M. Demorest, E.H. Aylward, S.C. Cramer, and K.R. Maravilla. Fmri investigation of cross-cultural music comprehension. *Neuroimage*, 20(1):378–384, 2003.
- [16] S.J. Morrison, S.M. Demorest, and L.A. Stambaugh. Enculturation effects in music cognition the role of age and music complexity. *Journal of Research in Music Education*, 56(2):118–129, 2008.
- [17] S. Nozaradan, I. Peretz, M. Missal, and A. Mouraux. Tagging the neuronal entrainment to beat and meter. *The Journal of Neuroscience*, 31(28):10234–10240, 2011.
- [18] S. Nozaradan, I. Peretz, and A. Mouraux. Selective neuronal entrainment to the beat and meter embedded in a musical rhythm. *The Journal of Neuroscience*, 32(49):17572–17581, 2012.
- [19] J.S. Snyder and E.W. Large. Gamma-band activity reflects the metric structure of rhythmic tone sequences. *Cognitive brain research*, 24(1):117–126, 2005.
- [20] G. Soley and E.E. Hannon. Infants prefer the musical meter of their own culture: a cross-cultural comparison. *Developmental psychology*, 46(1):286, 2010.
- [21] Y. Tang. Deep Learning using Linear Support Vector Machines. *arXiv preprint arXiv:1306.0239*, 2013.
- [22] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *The Journal of Machine Learning Research*, 11:3371–3408, Dec 2010.
- [23] R.J. Vlek, R.S. Schaefer, C.C.A.M. Gielen, J.D.R. Farquhar, and P. Desain. Shared mechanisms in perception and imagery of auditory accents. *Clinical Neurophysiology*, 122(8):1526–1532, Aug 2011.
- [24] D.F. Wulsin, J.R. Gupta, R. Mani, J.A. Blanco, and B. Litt. Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement. *Journal of Neural Engineering*, 8(3):036015, Jun 2011.



## **MIREX Oral Session**

This Page Intentionally Left Blank



## TEN YEARS OF MIREX: REFLECTIONS, CHALLENGES AND OPPORTUNITIES

**J. Stephen Downie**

University of Illinois  
jdownie@illinois.edu

**Kahyun Choi**

University of Illinois  
ckahyu2@illinois.edu

**Xiao Hu**

University of Hong Kong  
xiaoxhu@hku.hk

**Sally Jo Cunningham**

University of Waikato  
sallyjo@waikato.ac.nz

**Jin Ha Lee**

University of Washington  
jinhalee@uw.edu

**Yun Hao**

University of Illinois  
yunhao2@illinois.edu

### ABSTRACT

The Music Information Retrieval Evaluation eXchange (MIREX) has been run annually since 2005, with the October 2014 plenary marking its tenth iteration. By 2013, MIREX has evaluated approximately 2000 individual music information retrieval (MIR) algorithms for a wide range of tasks over 37 different test collections. MIREX has involved researchers from over 29 different countries with a median of 109 individual participants per year. This paper summarizes the history of MIREX from its earliest planning meeting in 2001 to the present. It reflects upon the administrative, financial, and technological challenges MIREX has faced and describes how those challenges have been surmounted. We propose new funding models, a distributed evaluation framework, and more holistic user experience evaluation tasks—some evolutionary, some revolutionary—for the continued success of MIREX. We hope that this paper will inspire MIR community members to contribute their ideas so MIREX can have many more successful years to come.

### 1. INTRODUCTION

Music Information Retrieval Evaluation eXchange (MIREX) is an annual evaluation campaign managed by the International Music Information Retrieval Systems Evaluation Laboratory (IMIRSEL) at the University of Illinois at Urbana-Champaign (UIUC). MIREX provides a framework and venue for music information retrieval (MIR) researchers to compare, contrast, and discuss the results of their MIR algorithms and systems, similar to what Text Retrieval Conference (TREC) has provided to the text information retrieval community [7]. MIREX has significantly contributed to the growth and maturity of the research community, and also affected the shaping of research priorities.

As this year marks the tenth running of MIREX, we take this opportunity to reflect upon its history, impact, and challenges over the past decade. We are at the critical point where a new funding model and distributed evaluation framework must be developed to ensure the sustainability of MIREX. We propose one of many possible solutions, and hope that this will spark the discussion in the

MIR community. In addition, based on the feedback and criticisms provided by the members of MIR community, we make recommendations on future directions of MIREX, emphasizing more holistic user experience evaluation tasks.

### 2. HISTORY, STRUCTURE, AND IMPACT

The history of MIREX can be traced back to 1999 when the Exploratory Workshop on Music Information Retrieval was held as part of the ACM Special Interest Group Information Retrieval (SIGIR) Conference. Two years later, the attendees of ISMIR2001 passed the “Bloomington Manifesto” which called for a formal evaluation platform for MIR research. Afterwards, two additional workshops on MIR evaluation were held in ISMIR2002 and SIGIR2003, which led to funding for MIREX from both the Andrew W. Mellon Foundation and the National Science Foundation (NSF).

In 2004, the local committee of ISMIR, the Music Technology Group (MTG) of the University Pompeu Fabra organized an evaluation session, the Audio Description Contest (ADC) [1], which provided valuable insights for MIREX. After these preludes, MIREX officially began in 2005 and had its first plenary and poster sessions in ISMIR 2005, held at Queen Mary College, University of London. Building on successful runs of MIREX, three additional projects were funded to explore new technologies and models to uplift MIREX: Networked Environment for Music Analysis (NEMA), Structural Analysis of Large Amounts of Music Information (SALAMI) and MIREX: Next Generation (MIREX: NG). From 2000 to date, MIREX and related projects have received approximately \$3,100,000 in funding from the NSF, The Andrew W. Mellon Foundation, University of Illinois, and the Korean Electronics Technology Institute (KETI). Table 1 summarizes these and other important events in the development of MIREX.

By 2013, MIREX has evaluated 1997 individual MIR algorithms over 37 different test collections, and has involved researchers from over 20 different countries with a median of 108 individual participants per year (Table 2). The tasks and subtasks evaluated in MIREX represent a wide spectrum of research interests among MIR researchers in the last decade (Table 3), including classical machine-learning train-test tasks (e.g., Audio Tag Classification), “low-level” tasks on which many MIR systems depend (e.g., Audio Beat Tracking), and tasks involving some types of user-issued music queries (e.g., Query-by-Singing/Humming). The evidence that MIREX has sig-



© J. Stephen Downie, Xiao Hu, Jin Ha Lee, Kahyun Choi, Sally Jo Cunningham, Yun Hao. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** J. Stephen Downie, Xiao Hu, Jin Ha Lee, Kahyun Choi, Sally Jo Cunningham, Yun Hao. “Ten Years of MIREX: Reflections, Challenges and Opportunities”, 15th International Society for Music Information Retrieval Conference, 2014.

nificantly grown over the years is clear. The addition and retirement of the tasks reflect the shift in interests in the field.

1999	Music retrieval workshop at SIGIR proposed a range of evaluation scenarios
2000	First ISMIR held at Plymouth with participants holding brainstorming sessions
2001	ISMIR at Indiana University; “Bloomington Manifesto” on evaluation published
2002	Planning grant from the Andrew W. Mellon Foundation awarded
2002	ISMIR at Paris hosted special evaluation workshop
2003	SIGIR at Toronto held Workshop on the Evaluation of Music Information Retrieval Systems
2003	Andrew W. Mellon Foundation and NSF funding awarded
2004	Audio Description Contest run at ISMIR Barcelona
2005	First MIREX plenary session held at ISMIR London
2008	NEMA project funded by the Andrew W. Mellon Foundation
2009	SALAMI funded by the NSF, SSHRC and JISC
2012	MIREX:NG project funded by the Andrew W. Mellon Foundation

**Table 1.** Important Events in MIREX History

	Datasets	Individuals	Countries	Runs
2005	10	82	19	86
2006	13	50	14	92
2007	12	73	15	122
2008	18	84	19	169
2009	26	138	15	289
2010	31	152	21	331
2011	32	156	16	296
2012	35	109	20	302
2013	37	116	29	310

**Table 2.** Descriptive Statistics for MIREX 2005-2013

Due to restrictive intellectual property issues surrounding music materials, the test data used in MIREX cannot be distributed to participants. This distinguishes the structure of MIREX from those of other major evaluation frameworks such as TREC. MIREX has been operated under an “algorithm-to-data” or “non-consumptive computation” model: researchers submit their MIR algorithms to IMIRSEL which are then evaluated by IMIRSEL personnel and volunteers against the ground truth data hosted in IMIRSEL.

Beyond the technical infrastructure, the communications infrastructure is also critical for MIREX as it is a community-driven endeavor. The MIREX wikis were set up for the community to collaboratively define the evaluation tasks, metrics, and general rules in every spring, and to publish and archive results data for each task and associated algorithms in every autumn. Besides being used by participants for preparing their mandatory presentations in the annual MIREX poster session in ISMIR, the MIREX results data also provide unique and valuable materials for publications in the field. In addition, the MIREX “EvalFest” mailing list is used for discussions about evaluation issues. To date, 531 people have subscribed to EvalFest. IMIRSEL also creates task-specific

mailing lists where researchers can have detailed discussions about metrics, collections, and input/output formats.

From its inception, MIREX has had a clear (and growing) impact on MIR research. Updating an earlier analysis of MIREX-related publications in [3], as of April 2014, 314 MIREX extended abstracts and 1,070 publications based on MIREX trials and results can be found through Google Scholar (Table 4). These publications have received a total of 18,239 citations (Table 5). We limited the analysis period to the seven years ending in 2011, as there is a considerable lag between the publication of a document and its appearance in Google Scholar (and then a similar lag before the paper can be cited). The growing number of Master’s and PhD dissertations building on MIREX results—and in many cases, participating in MIREX trials—is particularly significant; MIREX has clearly become a fundamental aspect of MIR research infrastructure. In addition to this impact on academic research, 13 patents have explicitly referenced MIREX extended abstracts [4].

	‘05	‘06	‘07	‘08	‘09	‘10	‘11
<b>Tech. report</b>	0	4	4	3	10	5	11
<b>Book chapter</b>	0	2	1	2	8	9	20
<b>Dissertation</b>	1	17	13	25	22	35	48
<b>Conference</b>	12	46	68	88	127	144	137
<b>Journal article</b>	1	15	27	21	29	50	65
<b>Total</b>	14	84	113	139	196	243	281

**Table 4.** Publication Types for MIREX-derived Papers

To elicit further, less easily measured contributions of MIREX to the research community, interviews of 18 influential MIR researchers were conducted in the MIREX Next Generation project [10]. From these, four key contributions were identified: 1) **Benchmarking and evaluation**: MIREX was born from the recognition that the field could not progress unless MIR researchers could benchmark their work against each other’s; 2) **Training and induction into MIR**: Emerging researchers and graduate students gain hands-on experience with MIR research and development, and build a reputation with potential employers within both the music industry and academia; 3) **Dissemination of new research**: The annual MIREX trials and subsequent MIREX session at ISMIR provide a natural focus for the research community, and allow researchers to showcase their work to the MIR community at large; 4) **Dissemination of data**: MIREX has been an important venue for the community to access previous high-quality evaluation datasets created by MIREX team or donated by researchers.

Year	MIREX extended abstracts				MIREX-derived publications			
	No.	citations	mean	med.	No.	citations	mean	med.
2005	55	418	7.60	5	14	879	62.79	17.5
2006	35	217	6.20	2	51	2656	31.62	13
2007	32	403	12.60	4	113	1449	21.27	8
2008	39	136	2.61	3	139	3560	26.61	8
2009	48	144	3.00	0	196	2790	14.23	5
2010	61	135	2.21	0	243	3093	12.73	6
2011	44	63	1.43	1	281	2296	8.17	2

**Table 5.** Overview of MIREX Citation Data, 2005-2011

TASK NAME	2005	2006	2007	2008	2009	2010	2011	2012	2013
Audio Artist Identification	7		7	11					
Audio Beat Tracking		5		15 <sup>(2)</sup>	22 <sup>(2)</sup>	26 <sup>(2)</sup>	24 <sup>(2)</sup>	60 <sup>(3)</sup>	54 <sup>(3)</sup>
Audio Chord Detection				11	18 <sup>(2)</sup>	15	18	22 <sup>(2)</sup>	36 <sup>(3)</sup>
Audio Classical Composer ID			7	8	30	27	16	15	14
Audio Cover Song Identification		8	8		6 <sup>(2)</sup>	6 <sup>(2)</sup>	4 <sup>(2)</sup>		2
Audio Drum Detection	8								
Audio Genre Classification	15		7	26 <sup>(2)</sup>	65 <sup>(2)</sup>	48 <sup>(2)</sup>	31 <sup>(2)</sup>	31 <sup>(2)</sup>	26 <sup>(2)</sup>
Audio Key Detection	7					5	8	6	3
Audio Melody Extraction	10	10 <sup>(2)</sup>		21 <sup>(3)</sup>	12 <sup>(6)</sup>	30 <sup>(6)</sup>	60 <sup>(6)</sup>	24 <sup>(6)</sup>	24 <sup>(6)</sup>
Audio Mood Classification			9	13	33	36	17	20	23
Audio Music Similarity		6	12		15	8	18	10	8
Audio Onset Detection	9	13	17		12	18	8	10	11
Audio Tag Classification				11	24 <sup>(3)</sup>	26 <sup>(2)</sup>	30 <sup>(2)</sup>	18 <sup>(2)</sup>	8 <sup>(2)</sup>
Audio Tempo Extraction	13	7				7	6	4	11
Discovery of Repeated Themes & Sections									16
Multiple Fundamental Frequency Estimation & Tracking			27 <sup>(2)</sup>	28 <sup>(2)</sup>	26 <sup>(3)</sup>	23 <sup>(3)</sup>	16 <sup>(2)</sup>	16 <sup>(2)</sup>	6 <sup>(2)</sup>
Query-by-Singing/Humming		23 <sup>(2)</sup>	20 <sup>(2)</sup>	16 <sup>(2)</sup>	12 <sup>(4)</sup>	20 <sup>(4)</sup>	12 <sup>(4)</sup>	24 <sup>(4)</sup>	28 <sup>(5)</sup>
Query-by-Tapping				5	9 <sup>(3)</sup>	6 <sup>(3)</sup>	3 <sup>(3)</sup>	6 <sup>(3)</sup>	6 <sup>(3)</sup>
Real-time Audio to Score Alignment (a.k.a Score Following)		2		4		5	2	3	2
Structural Segmentation					5	12 <sup>(2)</sup>	12 <sup>(2)</sup>	27 <sup>(3)</sup>	26 <sup>(3)</sup>
Symbolic Genre Classification	5								
Symbolic Key Finding	5								
Symbolic Melodic Similarity	7	18 <sup>(3)</sup>	8			13	11	6	6
<b>Total Number of Runs per Year</b>	<b>86</b>	<b>92</b>	<b>122</b>	<b>169</b>	<b>289</b>	<b>331</b>	<b>296</b>	<b>302</b>	<b>310</b>
<b>Total Number of Runs (2005-2013)</b>	<b>1997</b>								

Notes: 1) Superscript numbers represent the number of subtasks included. 2) Since 2009, the Audio Classical Composer ID task, Audio Genre Classification task, and Audio Mood Classification task have become subtasks of Train-Test Task.

**Table 3.** MIREX Tasks and the Number of Runs

### 3. CHALLENGES

#### 3.1 Sustainability of Current Administration Model

The current model for administering the evaluations is costly and unsustainable. Since its inception, all MIREX tasks have required manual execution of submitted algorithms. As algorithms are written in different languages and require a range of executing environments, running one algorithm takes about 5 hours of focused attention on average, including but not limited to the time spent on communicating with participants, debugging algorithms, reconfiguring input/output interfaces and execution environment, etc. More often than not, algorithms may have to be updated by participants and tested by IMIRSEL for multiple rounds before they can be executed correctly. Besides the algorithms, some tasks require ground truth data in every iteration of MIREX (e.g., similarity tasks, further discussed in Section 3.4), which takes a significant amount of time to build. To meet all these demands, IMIRSEL has been relying on a small number of graduate students fully devoted to running MIREX in each fall. Nonetheless, participants sometimes still have to wait for a long time to receive evaluation results.

To mitigate the problem, the Networked Environment for Music Analysis (NEMA) project was established to “construct a web-service framework that would make MIREX evaluation tasks, test collections, and automated evaluation scripts available to the community on a yearly basis” (p.113, the so-called “Do-It-Yourself” model) [6]. However, due to the large variety of execution environments of algorithms, the built framework has not been widely adopted in the MIR community, except for the

automated evaluation package in the NEMA framework which has been used in recent iterations of MIREX to automate the evaluation of tasks such as Train-test and Audio Tag Classification. This has greatly improved the efficiency of MIREX and productivity of IMIRSEL personnel, but such procedures still require manual input of raw results produced by the algorithms. The sustainability of MIREX calls for new technology and structures that can streamline the entire process of data/algorithm ingest, evaluation code generation/modification, and results posting, so that the evaluations can not only be effective, but also efficient, robust, and scalable.

#### 3.2 Financial Sustainability Challenges

The fact that MIREX has been providing significant value to the MIR community is clearly evident. However, IMIRSEL has effectively offered MIREX as a free service to the community. This model is unsustainable; in January 2015, the current Mellon funding concludes, leaving MIREX with no financial support for the first time in its history. A back-of-the-envelope calculation using the amount of grant funding (\$3,100,000) divided by number of runs (1997) gives an estimate of the cost per run of \$1,552. Cost estimates per participant (960 total) come in at \$3,229. These rough numbers illustrate the general magnitude of the funding challenge MIREX is facing.

#### 3.3 Knowledge Management and Transfer

Over the past decade, the leading task organizers of MIREX have left IMIRSEL, including Dr. Andreas Ehmann (now at Pandora.com) and Dr. Mert Bay—both in-

strumental in creating MIREX processes and techniques. Considerable time and energy are being expended in reconstructing past practices to help new IMIRSEL members and new task organizers complete their assigned duties. MIREX needs more effective mechanisms to manage corporate memory so as to successfully transfer knowledge to new lab members and external volunteers. Notwithstanding recent efforts to more thoroughly document MIREX technologies and procedures, more work needs to be done to support hands-on training sessions for all who manage and run MIREX tasks.

### 3.4 Ground Truth Data Shortage

The lack of ground truth data is one of the primary obstacles facing the field of MIR. There is a strong demand for large, high-quality ground truth datasets for various evaluation tasks. However, generating any kind of user data is expensive. Crowdsourcing has been suggested as a possible solution by a number of MIR researchers (e.g., [12][16]). Although previous studies have shown that the user evaluation results collected by crowdsourcing and from music experts in the conventional MIREX framework are comparable, the issues of representativeness and noise in data still exist.

In order to generate the ground truth data, human evaluators must listen to sample music pieces and manually input their responses. The task must be carried out by individuals who have had a baseline level of training, making the data even more expensive to collect. Currently, most ground truth data is generated within academic institutions through the use of graduate and undergraduate student labor. Funding opportunities for generating ground truth data are limited, and the fact that audio data is often not transferrable between multiple researchers or labs due to copyright restrictions further complicates dataset creation.

There are a variety of sources for ground truth data, some released by MIREX, and also by other researchers in an ad hoc fashion. However, academic scholars as well as researchers in industry have difficulty identifying and obtaining relevant datasets. Currently, there is no organization or lab that is taking the role of creating, maintaining, and sharing ground truth data. In other IR domains, there are central organizations that fulfill at least part of this responsibility to support evaluations [10]. For example, ground truth data in TREC is created and/or managed by National Institute of Standards and Technology (NIST) and is released after each evaluation [7]. In the field of speech recognition, the Linguistic Data Consortium (LDC) creates ground truth datasets that can be purchased for use by individual labs [10]. This cycle of refreshed data allows the research community to conduct high-quality evaluation. As this has not been the case for MIREX, the same ground truth data must sometimes be used for multiple years.

### 3.5 Intellectual Property Issues

Another major problem facing the MIREX community is the lack of usable music data upon which to build realistic test collections, due to intellectual property issues surrounding music materials. The datasets used in MIREX

are very limited in terms of size, variety, recency, and novelty. Moreover, the fact that datasets cannot be distributed after being used in MIREX effectively prevents researchers from replicating the evaluation and benchmarking their newly developed algorithms on their own. To tackle this issue that has plagued MIR research since day one, the MIR community needs to work together to explore possible solutions such as negotiating with copyright holders collectively, using creative audio and/or music in the public domain, and running algorithms against multiple datasets hosted in different labs. The latter approach has been attempted by projects such as NEMA. However, none of the possibilities is straightforward and this battle is likely to exist for many years to come.

### 3.6 System vs. User-centered Evaluations

MIREX has followed the conventional, Cranfield IR system-centered evaluation paradigm [2]. Recently, this evaluation approach has been criticized by multiple researchers for excluding users from the evaluation process. To name a few, Hu and Liu [9], Hu and Kando [8], Lee [11], Schedl and Flexer [15], and Lee and Cunningham [13] all argued that the goal of MIR systems is to help users meet their music information needs, and thus MIR evaluation must take users into account. For instance, a number of MIR researchers have questioned the validity of system-centered evaluation on tasks that involve human judgments such as the similarity tasks [12], [15], [16]. Music similarity may be interpreted differently for different people, yet the variance across users is simply ignored in the current evaluation protocol. As noted by Lee and Cunningham [13], a result of system-centered evaluation “may not be effectively translated to something meaningful or practical for real users (p. 517).” They suggested introducing tasks that “seems closer to what would be useful for real users” such as playlist generation, known-item search, or personal music collection management.

Notwithstanding the importance of traditional system-centered tasks, some suggestions have been made to MIREX to bridge the gap between system-centered and user-centered evaluation (e.g., incorporating user context in test queries, use terms familiar to users, combine multiple tasks in [11][9]), although they are yet to be reflected in the MIREX tasks. As the field matures, in order to move forward, it is vital to explore user-centered and realistic evaluation tasks.

## 4. FUTURE DIRECTIONS

### 4.1 Developing a User Experience Task

In keeping with our desire to expand MIREX beyond its current system-centered paradigm, we are conducting the first user-centered grand challenge evaluation task. The “Grand Challenge ‘14 User Experience” (GC14UX)<sup>1</sup> task is unlike any previous MIREX task. The GC14UX is directly inspired by the grand challenge idea proposed in Downie, Crawford and Byrd [5], which noted the persis-

<sup>1</sup> <http://www.music-ir.org/mirex/wiki/2014:GC14UX>

tent absence of complete MIR systems presented at ISMIR that could be released to the public for music searching and discovery. Thus, the GC14UX has two underpinning goals: 1) to inspire the development of complete MIR systems to be shared at ISMIR; and 2) to promote the notion of user experience as a first-class research objective in the MIR community.

The choice of “Grand Challenge” to describe our first UX task was made, in part, to signify that MIREX will be entering into uncharted evaluation territory. By finally undertaking a user-centered evaluation task, the GC14UX will require the MIREX team (and the MIR community) to come up with new evaluation methods and criteria that will be made manifest in ways significantly different from our now standard MIREX operation procedures. We argue that the current state of the art in conventional MIREX tasks is sufficient to support an acceptable degree of efficiency and effectiveness for most of the now classic MIREX system-centered tasks. It is now time to look towards the more holistic user experience: subjective explorations of hedonic aspects of use such as satisfaction, enjoyment, and stimulation. To that end, the MIREX team is proposing several radical departures from MIREX tradition that promise to better support the focus on the user experience. The most radical changes include: 1) no submission of algorithms to IMIRSEL; and 2) distribution of audio data to participants.

To ensure that the GC14UX does not become a system-centered evaluation in disguise, the process is designed to remain as agnostic as possible concerning the technological means by which participating systems create and deliver their experiences to the users. This deliberate indifference suggests that the GC14UX has no need to run or evaluate the underlying system code that delivers the content to the users. Since the GC14UX will not be evaluating the system-code per se, it makes sense that the GC14UX does not follow MIREX’s usual practice of running code on behalf of the submitters. There are obvious benefits to this non-submission approach, including greatly reduced system requirements and significantly reduced MIREX staff time requirements for debugging and administration.

Dropping the usual algorithm-to-data procedures does, obviously, beg the question about data sources for the systems to use. All the usual copyright reasons why music distribution is problematic for MIREX still apply and therefore we need data sources that are amenable to distribution. For the first running of GC14UX, the test collection will be drawn from Creative Commons music. We believe that a set in the magnitude range of 10,000 songs would strike a nice balance between being non-trivial in size and breadth while not posing too great of a data management burden for participants. A common dataset helps mitigate against the possible user experience bias induced by the differential presence (or absence) of popular or known music within the participating systems.

The GC14UX task is all about how users perceive their experiences with the systems. We intend to capture the user perceptions in a minimally intrusive manner under as-realistic-as-possible use scenarios. To this end, all

participating systems are required to be constructed as websites accessible to users through normal web browsers. For user evaluation, we also do not want to burden the users/evaluators with too many questions or required data inputs. Our main goal is to determine whether each system was able to provide a satisfying user experience ([14], [17]). Thus, a question asking about the level of overall satisfaction is posed to each user for each system. An option for open-ended responses is provided so as to capture the expressions of the users in their own words.

There are many potential challenges that could prevent GC14UX from being the progenitor of future MIREX UX evaluations. For example, the utility and possible side-effects of using Creative Commons music as the common dataset have yet to be ascertained. Also, the effectiveness of the current GC14UX user inputs will most likely spark lively debate among MIR researchers after our first round of data is collected. Notwithstanding these known problems, as well as the challenges currently unknown, we are eager to see GC14UX proceed and inspire new evaluations. It is well past time that MIREX act to create a real user-centered evaluation stream. If we allow perfection to be the enemy of the good, MIREX might never be able to launch a vibrant UX evaluation thread.

#### 4.2 Funding Models

In order to continue providing benefits to the MIR community, MIREX must explore a range of funding options. In order to reduce the dependencies and burdens placed upon any one funding source, it is necessary to seek multiple sources of income. Some of the current possibilities include:

- **Lab Memberships:** MIREX is exploring the possibility of setting up a lab membership system for labs that are active in MIR. Member labs would be represented on MIREX’s governing committee, and would have access to the new datasets that MIREX creates.
- **Sponsorship:** MIREX would also like set up a sponsorship program for leaders in industry. A sponsorship program would give companies a chance to support and/or discover interesting new MIR work by emerging researchers. Identification of recruiting opportunities is a valuable benefit that industry currently derives from MIREX (Section 2).
- **Institutional Support:** The University of Illinois has provided significant in-kind support for MIREX in the past. MIREX seeks to extend this partnership into the future. However, budget shortfalls at the State level are diminishing the prospects of ongoing University support.
- **Data Creation and Curation:** The MIREX team completed a collaborative project developing ground truth genre and mood data for, and funded by, Korea Electronics Technology Institute (KETI) in 2013. The data created is being folded into the MIREX task pool. The success of the KETI project, combined with the precedent set by the LDC (Section 3.4), inspires future data creation actions. In a similar line, we are exploring the possibility of providing fee-based data

curation and management services to those who have data sets that require long-term preservation.

While it will need to seek more support from its participants, MIREX recognizes the need to balance this with openness and accessibility. MIREX aims to remain open to any researcher who wants to participate, with a healthy funding mix making this goal more likely to be achieved.

#### 4.3 Distributed Management Model: Task Captains

MIREX is pursuing a more decentralized model in order to reduce the strain on IMIRSEL and to more actively involve the entire MIR community in task creation, organization and delivery. Under this model, multiple labs can run particular tasks while IMIRSEL functions as a central organizer and algorithm submission point. This model was piloted in 2012 with Query-by-Singing/Humming (QBSH) and Audio Melody Extraction (AME) run by KETI. In MIREX 2013, Audio Beat Tracking (ABT), Audio Chord Estimation (ACE), Audio Key Detection (AKD), Audio Onset Detection (AOD), Audio Tempo Estimation (ATE), and Discovery of Repeated Themes & Sections (DRTS) were led by non-IMIRSEL volunteer “Task Captains” who managed the tasks from start to finish. While shortcomings in MIREX documentation were evident, the Task Captain initiative was successful and will be developed further.

### 5. CONCLUSIONS

In this paper, we reflect on ten years of experience of MIREX. As the major community-based evaluation framework, MIREX has made unprecedented contributions to the MIR research field. However, MIREX also faces a number of significant challenges including financial sustainability, restrictions on data and intellectual property, and governance. Future directions of MIREX are proposed to meet these challenges. By moving towards the evaluation of entire systems and emphasizing holistic user experience, MIREX will allow us to compare and evaluate startups and experimental systems, as well as commercial MIR systems. We hope this paper will serve as a catalyst for the community to come together and seek answers to the question: what is the future of MIREX? More importantly, we hope this paper will inspire MIR community members to actively engage in and contribute to the continuation of MIREX. MIREX has always been a community-driven endeavor; without the active leadership and involvement of MIR researchers, MIREX simply cannot exist.

### 6. REFERENCES

- [1] P. Cano, E. Gomez, F. Gouyon, P. Herrera, M. Koppenberger, B. Ong, X. Serra, S. Streich, N. Wack: “ISMIR 2004 audio description contest,” MTG Technical Report, MTG-TR-2006-02 (Music Technology Group, Barcelona, Spain), 2004.
- [2] C. W. Cleverdon and E. M. Keen: “Factors determining the performance of indexing systems. Vol. 1: Design, Vol. 2: Results,” Cranfield, UK: Aslib Cranfield Research Project, 1966.
- [3] S. J. Cunningham, D. Bainbridge, and J. S. Downie: “The impact of MIREX on scholarly research,” *Proceedings of the ISMIR*, pp. 259-264, 2012.
- [4] S. J. Cunningham and J. H. Lee: “Influences of ISMIR and MIREX Research on Technology Patents,” *Proceedings of the ISMIR*, pp.137-142, 2013.
- [5] J. S. Downie, D. Byrd, and T. Crawford: “Ten Years of ISMIR: Reflections on Challenges and Opportunities.” *Proceedings of the ISMIR*, pp. 13-18. 2009.
- [6] J. S. Downie, A. F. Ehmann, M. Bay, and M. C. Jones: “The music information retrieval evaluation exchange: Some observations and insights.” In *Advances in music information retrieval*, pp. 93-115, Springer Berlin Heidelberg, 2010.
- [7] D. Harman: “Overview of the Second Text Retrieval Conference (TREC-2).” *Information Processing & Management*, 31(3), 271–289, 1995.
- [8] X. Hu and N. Kando: “User-centered Measures vs. System Effectiveness in Finding Similar Songs,” *Proceedings of the ISMIR*, pp.331-336, 2012.
- [9] X. Hu and J. Liu: “Evaluation of Music Information Retrieval: Towards a User-Centered Approach”. *Proceedings of the 4th Workshop on Human-Computer Interaction and Information Retrieval*, 2010.
- [10] Ithaca S+R: “MIREX Consulting Report and Proposed Business Plan,” 2013.
- [11] J. H. Lee: “Analysis of user needs and information features in natural language queries for music information retrieval,” *Journal of the American Society for Information Science & Technology*, 61(5), pp. 1025-1045, 2010.
- [12] J. H. Lee: “Crowdsourcing Music Similarity Judgments using Mechanical Turk,” *Proceedings of the ISMIR*, pp. 183 - 188, 2010.
- [13] J. H. Lee and S. J. Cunningham: “Toward an understanding of the history and impact of user studies in music information retrieval”, *Journal of Intelligent Information Systems*, 41, pp. 499-521, 2013.
- [14] H. Petrie and N. Bevan: “The evaluation of accessibility, usability and user experience,” *The Universal Access Handbook*, pp. 10-20, 2009.
- [15] M. Schedl and A. Flexer: “Putting the User in the Center of Music Information Retrieval.” *Proceedings of the ISMIR*, pp. 385-390, 2012.
- [16] J. Urbano: “Information Retrieval Meta-Evaluation: Challenges and opportunities in the Music Domain,” *Proceedings of the ISMIR*, pp. 609 - 611, 2011.
- [17] A. Vermeeren, E. L-C Law, V. Roto, M. Obrist, J. Hoonhout, and K. Väänänen-Vainio-Mattila: “User experience evaluation methods: current state and development needs.” In *Proceedings of the 6th Nordic Conference on HCI*, pp. 521-530, 2010.

## Author Index

### A

Ali, Hasan, 457  
 Ali-MacLachlan, Islah, 15  
 Aljanaki, Anna, 373  
 Antila, Christopher, 71  
 Arjannikov, Tom, 95  
 Arzt, Andreas, 549  
 Athwal, Cham, 21

### B

Badeau, Roland, 175  
 Bainbridge, David, 457  
 Barbancho, Ana M., 47, 277, 567  
 Barbancho, Isabel, 41, 277, 567  
 Bardeli, Rolf, 89  
 Batista, Gustavo E. A. P. A., 113  
 Bazzica, Alessio, 201  
 Bello, Juan P., 155, 265, 591  
 Benetos, Emmanouil, 53, 175  
 Bisot, Victor, 337  
 Bittner, Rachel, 155  
 Bittner, Rachel M., 591  
 Black, Dawn A. A., 161, 631  
 Böck, Sebastian, 513, 603, 637  
 Bogaards, Niels, 537  
 Bogdanov, Dmitry, 573  
 Boland, Daniel, 561  
 Bonada, Jordi, 65  
 Bountouridis, Dimitrios, 207, 379  
 Brown, Daniel G., 3, 471  
 Burgoyne, John Ashley, 525

### C

Caetano, Marcelo, 331  
 Cameron, Daniel J., 649  
 Cannam, Chris, 155  
 Carabias-Orti, Julio José, 125  
 Chacón, Carlos Eduardo Cancino, 195  
 Chen, Chun-Ta, 289  
 Cherla, Srikanth, 53, 101  
 Chew, Elaine, 489  
 Chi, Tai-Shih, 617  
 Chien, Jen-Tzung, 507  
 Choi, Kahyun, 385, 657  
 Church, Maura, 643

Clausen, Michael, 35  
 Collins, Nick, 21  
 Cumming, Julie, 71  
 Cunningham, Sally Jo, 457, 657  
 Cuthbert Michael Scott, 643

### D

Danger, Marc, 597  
 Davies, Mathew E. P., 637  
 d'Avila Garcez, Artur S., 53, 101  
 de Haas, W. Bas, 47, 525  
 Dieleman, Sander, 29  
 Disch, Sascha, 611  
 Dixon, Simon, 53, 83, 187  
 Downie, J. Stephen, 385, 657  
 Driedger, Jonathan, 611  
 Driessen, Peter F., 41  
 Duan, Zhiyao, 181  
 Dufour, Richard, 465  
 Dupont, Stéphane, 349  
 Dutoit, Thierry, 349  
 Dutta, Shrey, 397

### E

Ellis, Daniel P.W., 167, 367, 405  
 Emiya, Valentin, 89  
 Ewert, Sebastian, 83

### F

Farbood, Morwaread M., 265, 411  
 Farrahi, Katayoun, 483  
 Fazekas, György, 631  
 Flexer, Arthur, 245  
 Forsyth, Jon, 591  
 Fourer, Dominique, 295  
 Frisson, Christian, 349

### G

Gauß, Sven, 543  
 Giraud, Mathieu, 59  
 Glazyrin, Nikolay, 149  
 Godec, Primož, 355

Gómez, Emilia, 65, 219, 573  
 Gómez-Martín, Francisco, 65  
 Goto, Masataka, 227, 585  
 Grachten, Maarten, 47, 195  
 Grahn, Jessica A., 649  
 Grill, Thomas, 417  
 Grohgan, Harald, 35  
 Guastavino, Catherine, 65  
 Guna, Jože, 355

## H

Hamanaka, Masatoshi, 325  
 Han, Yoonchang, 77  
 Hanjalic, Alan, 143, 201  
 Hanna, Pierre, 295  
 Hao, Yun, 657  
 Hasegawa-Johnson, Mark, 477  
 Hauger, David, 483  
 Herrera, Perfecto, 573  
 Hirata, Keiji, 325  
 Holzapfel, Andre, 425  
 Honingh, Aline, 537  
 Horner, Andrew, 253  
 Hornung, Thomas, 543  
 Hsu, Chung-Chien, 507  
 Hsu, Ling-Chi, 495  
 Hu, Xiao, 385, 579, 657  
 Huang, Po-Sen, 477  
 Humphrey, Eric J., 367, 591  
 Hwang, Eenjun, 519

## I

Itoyama, Katsutoshi, 233

## J

Jančovič, Peter, 15  
 Janer, Jordi, 125  
 Jang, Jyh-Shing Roger, 289, 555  
 Jehan, Tristan, 265  
 Jensen, Bjørn Sand, 319  
 Joachims, Thorsten, 439  
 Jones, Andrew, 301  
 Joren, Six, 259  
 Jun, Sanghoon, 519

## K

Kameoka, Hirokazu, 623  
 Karsdorp, Folgert, 391  
 Kim, Minje, 477  
 King, Richard, 137  
 Kirlin, Phillip B., 213  
 Köküer, Münevver, 15  
 Kong, LamWang, 501  
 Korzeniowski, Filip, 513  
 Krebs, Florian, 425, 603  
 Kroher, Nadine, 65  
 Kruspe, Anna M., 271

## L

Laaksonen, Antti, 119  
 Langlois, Thibault, 89  
 Laplante, Audrey, 451  
 Larsen, Jan, 319  
 Larson, Martha, 143  
 Lartillot, Olivier, 361  
 Lattner, Stefan, 195  
 Lausen, Georg, 543  
 Lee, Chung, 253  
 Lee, Jin Ha, 385, 579, 657  
 Lee, Kyogu, 77  
 Lee, Tan, 501  
 Leman, Marc, 259  
 Leonard, Brett, 137  
 Leonardis, Aleš, 131  
 Levé, Florence, 59  
 Liang, Dawen, 167, 367  
 Liem, Cynthia C. S., 143, 201  
 Lin, Yi-Ju, 495  
 Linarès, Georges, 465  
 Lu, Chun-Hung, 289, 555  
 Luo, Yin-Jyun, 617

## M

Madsen, Jens, 319  
 Maezawa, Akira, 233  
 Man, Brecht De, 137  
 Marolt, Matija, 131, 355  
 Marshall, David, 301  
 Martorell, Agustín, 219  
 Masuda, Taro, 227  
 Mauch, Matthias, 155, 187  
 McFee, Brian, 367, 405  
 Mercier, Florent, 59  
 Metcalf, Cheryl D., 495



Miron, Marius, 125  
 Molina, Emilio, 277, 567  
 Moore, Joshua L., 439  
 Morchid, Mohamed, 465  
 Morishima, Shigeo, 227, 585  
 Müller, Meinard, 35, 307, 611  
 Murray-Smith, Roderick, 561  
 Murthy, Hema A.397

## N

Nakamura, Eita, 531  
 Nakamura, Tomohiko, 623  
 Nakano, Tomoyasu, 585  
 Nayak, Nagesh, 283  
 Nichols, David M., 457  
 Nieto, Oriol, 265, 367, 411, 591

## O

Okuno, Hiroshi G., 233  
 Ono, Nobutaka, 531

## P

Paisley, John, 167  
 Panteli, Maria, 537  
 Pauwels, Johan, 525  
 Peeters, Geoffroy, 337  
 Pesek, Matevž, 131, 355  
 Pogačnik, Matevž, 355  
 Poredoš, Mojca, 355  
 Prätzlich, Thomas, 307  
 Przyjaciół-Zablocki, Martin, 543  
 Pugin, Laurent, 107

## R

Raffel, Colin, 367  
 Rao, Preeti, 283  
 Reiss, Joshua D., 137  
 Repetto, Rafael Caro, 313, 343, 431  
 Rezende, Solange O., 113  
 Rho, Seungmin, 519  
 Richard, Gaël, 175  
 Riche, Nicolas, 349  
 Rigaudière, Marc, 59  
 Robine, Matthias, 295  
 Rodríguez López, Marcelo, 207  
 Roland, Perry, 107  
 Rossi, Rafael G.113  
 Rouas, Jean-Luc, 295  
 Rousseaux, Francis, 597

## S

Sagayama, Shigeki, 531  
 Salamon, Justin, 155, 367, 591  
 Sandler, Mark, 631  
 Sasaki, Shoto, 585  
 Saurel, Pierre, 597  
 Schätzle, Alexander, 543  
 Schedl, Markus, 483  
 Schlüter, Jan, 417  
 Schrauwen, Benjamin, 29  
 Serra, Xavier, 313, 343, 431, 573  
 Shikata, Kotaro, 623  
 Sidorov, Kirill, 301  
 Siebert, Xavier, 349  
 Sigtia, Siddharth, 53  
 Silva, Diego F., 113  
 Singhi, Abhishek, 3, 471  
 Smaragdis, Paris, 477  
 Sonnleitner, Reinhard, 549  
 Srinivasamurthy, Ajay, 425, 431  
 Stober, Sebastian, 649  
 Stojmenova, Emilija, 355  
 Strle, Gregor, 355  
 Sturm, Bob L., 21, 89  
 Su, Alvin W.Y., 495  
 Su, Li, 9  
 Sundar, Harshavardhan, 431

## T

Takamune, Norihiro, 623  
 Tang, Zheng, 161  
 Tardón, Lorenzo J., 41, 277, 567  
 Taxidou, Io, 543  
 Temperley, David, 181  
 Thompson, Lucas, 187  
 Thorez, Donatien, 59  
 Tian, Mi, 631  
 Tierney, Mike, 155  
 Tkalčić, Marko, 483  
 Tojo, Satoshi, 325  
 Turnbull, Douglas, 439  
 Tzanetakis, George, 41

## U

Ullrich, Karen, 417  
 Urbano, Julián, 573

## V

Vall, Andreu, 483  
 van den Oord, Aäron, 29  
 van Herwaarden, Sam, 47  
 van Kranenburg, Peter, 391  
 Van Balen, Jan, 379  
 Velmurugan, Rajbabu, 283  
 Veltkamp, Remco C., 373, 379  
 Venkataramani, Shrikant, 283  
 Vera, Bogdan, 489  
 Volk, Anja, 207

## W

Wang, Hsin-Min, 239  
 Wang, Ju-Chiang, 239  
 Wang, Siying, 83  
 Wang, Xinxi, 445  
 Wang, Ye, 445  
 Wang, Yu-Lin, 495  
 Weyde, Tillman, 53, 101, 175  
 Widmer, Gerhard, 513, 549, 603  
 Wiering, Frans, 331, 373, 379  
 Wong, Leanne Ka Yan, 579  
 Wu, Bin, 253  
 Wu, Ming-Ju, 555

## X

Xing, Zhe, 445

## Y

Yadati, Karthik, 143  
 Yang, Po-Kai, 507  
 Yang, Yi-Hsuan, 9, 239  
 Yen, Frederick, 617  
 Yen, Ming-Chi, 239  
 Yoshii, Kazuyoshi, 227, 233, 585  
 Yu, Li-Fan, 9  
 Yvart, Willy, 349

## Z

Zhang, John Z., 95  
 Zhang, Shuo, 343  
 Zitellini, Rodolfo, 107

*We wish to thank the following sponsors for their contribution to the success of this conference:*

### Gold Sponsors



### Silver Sponsors



### Bronze Sponsors



## Co-Hosts



中華民國電腦音樂學會  
*Taiwan Computer Music Association*



經濟部  
國際貿易局 經貿資訊網  
Bureau of Foreign Trade



觀光傳播局  
*Department of Information and Tourism*

## Corporate Partner



