JAZZ ENSEMBLE EXPRESSIVE PERFORMANCE MODELING

Bantula, Helena **Giraldo, Sergio** **Ramirez, Rafael**
Music Technology Group, Universitat Pompeu Fabra, Barcelona (Spain)
helenabantula@gmail.com, {sergio.giraldo, rafael.ramirez}@upf.edu

ABSTRACT

Computational expressive music performance studies the analysis and characterisation of the deviations that a musician introduces when performing a musical piece. It has been studied in a classical context where timing and dynamic deviations are modeled using machine learning techniques. In jazz music, work has been done previously on the study of ornament prediction in guitar performance, as well as in saxophone expressive modeling. However, little work has been done on expressive ensemble performance. In this work, we analysed the musical expressivity of jazz guitar and piano from two different perspectives: solo and ensemble performance. The aim of this paper is to study the influence of piano accompaniment into the performance of a guitar melody and vice versa. Based on a set of recordings made by professional musicians, we extracted descriptors from the score, we transcribed the guitar and the piano performances and calculated performance actions for both instruments. We applied machine learning techniques to train models for each performance action, taking into account both solo and ensemble descriptors. Finally, we compared the accuracy of the induced models. The accuracy of most models increased when ensemble information was considered, which can be explained by the interaction between musicians.

1. INTRODUCTION

Music is a very important part in the life of milions of people, whether they are musicians, they enjoy attending live music concerts or simply like listening to musical recordings at home. The engaging part of music is the human component added to the performance: instead of a ”dead” score, musicians shape the music by changing parameters such as intensity, velocity, volume and articulation. The study of music expressive performance from a computational point of view consists of characterising the deviations that a musician introduces in a score, often in order to render human-like performances from inexpressive music scores.

There are numerous works which study expressive performance in classical music, and most of these studies have been done on piano performances (for an overview, see Goebl [24]). Other works analyse expressivity in a jazz context. For instance, Giraldo and Ramírez [8] study and model the ornamentation introduced to a jazz melody by using machine learning techniques. In an ensemble context, the musicians' performance is influenced by what is being played by the other musicians. Although most works are focused on soloist performances, some works take into account ensemble performances in classical music ([26], [12], [15]). However, to our knowledge little work addresses ensemble expressive performance in a jazz context. In this work, we present a method to study the interaction between jazz musicians from a computational perspective. Our data set consisted of 7 jazz pieces recorded by a jazz quartet (guitar, piano, bass and drums), in which each instrument was recorded on a separate track. In this study we considered the interaction between guitar melodies and the accompaniment of piano. We extracted individual (soloist) score descriptors as well as ensemble descriptors. We calculated performance actions for both guitar (embellishments) and piano (chord density, range and weight). We applied machine learning techniques to predict these performance actions using Artificial Neural Networks, Support Vector Machine and Decision Trees. We generated *individual models* for each instrument and measured the level of interaction between musicians by introducing ensemble descriptors into each individual model to create *mixed models*, and compared the *individual models* with the *mixed models*. Finally, we evaluated the performance of the algorithms by computing statistical significance tests (Paired T-Test).

The rest of the paper is organised as follows. In Section 2, we present related work in expressive music performance. In section 3, we describe the materials we have used. In Section 4, the proposed method is described and the evaluation process is explained. In Section 5, the results of the evaluation are presented. Finally, in Section 6 we put forward conclusions and future improvements.

2. RELATED WORK

Many works study expressive performance actions in music, defined as variations in timing (duration and onsets), energy, articulation, and vibrato from different perspectives, including psychology ([6], [19]), neurology ([13]), musicology ([22]) and at a computational level. Previous work has been done in a **classical context** by Friberg [5], who develops a set of rules using analysis by synthesis to generate the deviations to be applied to a score, obtaining human-like performances. Widmer [25] analyses recordings of 13 complete Mozart piano sonatas and uses a machine learning approach to create predictive rules for note-level timing, dynamics and articulation. In a **jazz context**, previous work has focused on the saxophone: Lopez de Mántaras et al. [1] use case-based reasoning to develop a system capable of modeling expressive performances (onset, duration and energy) of jazz saxophone. Ramírez and Hazan [20] apply classification and regression methods to a set of acoustic features extracted from saxophone recordings and a set of descriptors which characterised the context of the performed notes. In jazz guitar, Giraldo and Ramírez ([9], [10]) use machine learning techniques to model and synthesise embellishments by training models to classify score notes as embellished or not, according to the characteristics of the notes' context.

2.1 Ensemble Performance

In ensemble performance, the expressivity of a soloist might be influenced by what the other musicians are playing. Most of the literature refers to **classical context**, studying timing asynchrony among performers. Repp [21] studies the synchronisation of the task of tapping by taking into account phase and frequency correction mechanisms. Wing et al. [26] develop a model for studying synchronisation in string quartets in different contexts (democratic or dictatorial). Goebl and Palmer [12] investigate the effect of the auditory and visual feedback so to study the synchronisation among musicians. More recently, Marchini [15] studies the interaction between musicians by generating independent machine learning models of expressive performance for each musician and taking into account the influence of the other musicians.

3. MATERIALS

We recorded 7 jazz standards performed by a jazz quartet both in wav and MIDI format, using the digital audio workstation Logic Pro X [14]. The scores were written in Music-XML format using Muse Score [18] to extract descriptors. We developed code by using the computing environment Matlab [17], concretely, the MidiToolBox Library developed by Toiviainen and Eerola [4] to process the data in MIDI format. We used the fundamental frequency estimator YIN [3] to create an automatic guitar melody transcriber. We performed beat tracking of the recordings using the beat tracker developed by Zapata [27]. Finally, we used the Weka Data Mining Software [16] for machine learning modeling.

4. METHODOLOGY

The methodology is divided into three stages, which are depicted in Figure 1. Firstly, we acquired the data from recordings and its respective scores (Section 4.1). Secondly, the data was analysed to extract the chords played by the pianist, which were aligned with the score afterwards so as to obtain piano performance actions. For guitar, we transcribed the audio into MIDI, and aligned the played notes with the score to obtain guitar performance actions. From the score, descriptors for notes and chords were extracted. We manually transcribed the audio of the piano and guitar into a new score in order to also extract descriptors from the *performed* score. The audio mix was used for beat tracking, and to compute a mean tempo. Thirdly, machine learning techniques were applied using the different data sets created from the extracted data to predict the calculated performance actions (Section 4.3).

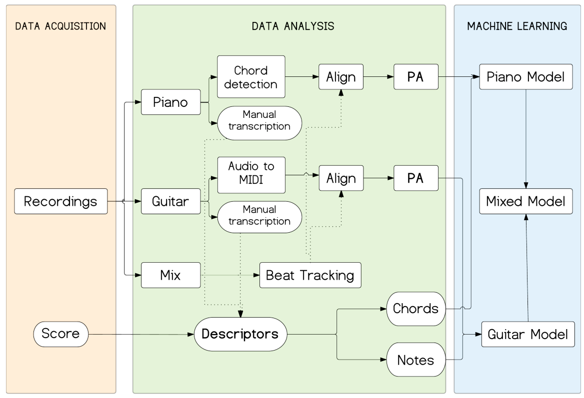

Figure 1. Overall framework: the data related to the score is shown inside ellipses while the data related to the recordings is placed inside rectangles.

4.1 Data acquisition and pre-processing

We recorded a jazz ensemble consisting of keyboard, electric guitar, electric bass and drums. We only used the guitar data in wav format, the piano data in MIDI format and an audio mix of the band in wav format for further tempo computation. Improvisers usually play the main melody at the beginning and end of a performance with improvisations in the central part and so both the recordings and the scores were segmented in order to contain only the melody part (no introductions or solos).

4.2 Data analysis

The aim of this part was to obtain a machine readable representation from the input data (recordings and scores) in the form of **descriptors** (data extracted from the score which characterised both notes and chords by taking into account their properties and the properties of their contexts) and **performance actions** (deviations from the score introduced by the musician to add expressivity, extracted from the recordings). In this stage, there were 4 types of input data: piano recordings, guitar recordings, scores and audio mix recordings. The following Sections explain the processing of this data.

4.2.1 Piano Data

We detected chords in the piano data by grouping together individual notes. The process consisted of identifying groups of notes played at the same time and so we created heuristic rules based on the work done by Traube and Bernays [23], who identify groups of notes which have near-synchronous onsets by analysing onset differences between two consecutive notes. Our approach consisted of three rules: the first one searched for and grouped notes which were played at the same time. The second one, was in charge of merging chords with an inter onset difference $< 100ms$. Finally, the third rule took into consideration pedal notes (notes that remain while two or more chords are played consequently).

Alignment was performed to link the detected chords with the score chords. It was done at a beat level: since the position of the beats was computed by the beat tracker (see Section 4.2.4), we converted the onsets/offsets of each performed chord from seconds to beats. Based on beat information, we aligned a chord written in the score with the chord (or chords) that had been played.

Based on the alignment of the played chords to the score, the performance actions for every chord in the score were calculated according to what had been played. We computed three performance actions: **density** (Equation 1), defined as *low* or *high* depending on the number of chords used to perform a score chord (i.e. chords played with a duration of half-note or more were labelled as low while a duration of less than a half note corresponded to a "high" label); **weight** (Equation 2), defined as *low* or *high* according to the total number of notes which were utilised to perform a score chord; and **range** (Equation 3), defined as *low* or *high* if the distance in semitones from the highest to the lowest performed note per chord in the score was larger than 18 (an octave and a half) .

$$den(chord_S) = \begin{cases} low & \text{if } \dfrac{\sum chords_P}{dur(chord_S)} < 1/2 \\ high & \text{if } \dfrac{\sum chords_P}{dur(chord_S)} \geq 1/2 \end{cases} \tag{1}$$

Where:

$chord_S$: is the corresponding chord on the score

$\sum chords_P$: is the amount of performed chords for a chord on the score

$dur(chord_S)$: is the duration of the corresponding chord on the score

$$wei(chord_S) = \begin{cases} low & \text{if } \dfrac{\sum notes_P}{\sum chords_P} < 4 \\ high & \text{if } \dfrac{\sum notes_P}{\sum chords_P} \geq 4 \end{cases} \tag{2}$$

Where:

$chord_S$: is the corresponding chord on the score

$\sum notes_P$: is the total number of performed notes for a chord on the score

$\sum chords_P$: is the amount of performed chords for a chord on the score

$$ran(chord_S) = \begin{cases} low & \text{if } max(pitch_{PN}) - min(pitch_{PN}) < 18 \\ high & \text{if } max(pitch_{PN}) - min(pitch_{PN}) \geq 18 \end{cases} \tag{3}$$

Where:

$chord_S$: is the corresponding chord on the score

$pitch_{PN}$: is the vector of pitch of the performed notes (PN) for a chord on the score

4.2.2 Guitar Data

We automatically converted the guitar recording in wav format into a MIDI format in order to obtain a note representation based on pitch, onset (in seconds) and offset (in seconds) by following the framework presented in Bantula et al. work [2].

Alignment was then performed at two levels. Firstly, the onsets and offsets of the MIDI notes were converted from seconds to beats using the beats' information computed by the beat tracker (see Section 4.2.4). Secondly, we performed manual alignment between the performed notes and the score notes by using a graphical interface that allowed to link the performed notes and the score notes in two pianoroll representations [11]. Embellishments were computed by following the same approach by Giraldo and Ramírez [11]: a note was considered to be embellished if two or more notes were played in its place. Then, each score note was labelled as embellished or not (y/n) according to the previous alignment.

4.2.3 Score Data

In this stage, we extracted horizontal and vertical descriptors from the score to characterise both chords and notes.

- **Chord Descriptors (Figure 2)** For chords, the horizontal context concerned harmonic information and the vertical context considered melodic, ensemble information. In Table 1, the intrinsic descriptors of the reference chords are listed. In Table 3, the harmonic horizontal descriptors, computed according to the neighbours of the reference chord are shown. Table 2 includes the vertical descriptors computed by averaging or weighting the single note descriptors of the notes below the region defined by the reference chord.

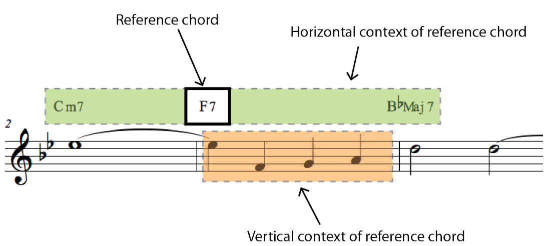

Figure 2. Excerpt of *Autumn Leaves*: horizontal and vertical contexts for the reference chord F7

descriptor	units	computation	range
id	num	$root \rightarrow number$	[0,11]
type	label	$type$	{M, m, +,7, dim, half_dim}
tens	label	tension based on musical criteria	{++, +, -, - -}
chord_dur	beats	$chord\_dur$	[1,∞)
on_b	beats	on_b	[1,∞)

Table 1. Individual descriptors for a reference chord (no context).

descriptor	units	computation	range
onset_b	beats	$min(onset_{notes})$	[1,∞)
dur_b	beats	$max(onset_{notes}) + max(dur_{notes}) - min(onset_{notes})$	[1,∞)
meanPitch (mP)	MIDI note	$mean(pitch_{notes})$	[36,96]
onset_s	seconds	$60 * \frac{onset\_b}{tempo}$	[1,∞)
dur_s	seconds	$60 * \frac{dur\_b}{tempo}$	[1,∞)
chroma	half tones	$mod_{12}(mP)$	[0,11]
measure	num	$measure$	[1,∞)
pre_dur_b	beats	pre_dur_b	[1,∞)
pre_dur_s	seconds	$60 * \frac{pre\_dur\_b}{tempo}$	[1,∞)
nxt_dur_b	beats	nxt_dur_b	[1,∞)
nxt_dur_s	seconds	$60 * \frac{nxt\_dur\_b}{tempo}$	[1,∞)
prev_int	half tones	$prev_{mP} - mP$	[1,∞)
next_int	half tones	$mP - next_{mP}$	[1,∞)
note2key	half tones	$chroma - key$	[0,11]
note2chord	half tones	$chroma - id$	[0,11]
isChordN*	label	-	{y,n}
mtr*	label	$mean(met_{pos}(notes))$	{strong, weak}
intHop*	num	$mean(intervals)$	[0,96]
melody*	num	$\frac{\#notes}{chord\_dur}$	-

Table 2. Chord melodic descriptors (vertical)

descriptor	units	computation	range
tempo	bpm	$tempo$	[1,300]
keyMode	label	$keyMode$	{major, minor}
numKey	num	key position in the Fifths Circle	[0,11]
keyDistance	half tones	$id - numKey$	[0,11]
metP*	label	metrical position	{strongest, strong, weak, weakest}
function	label	harmonic analysis from $keyDistance$	{tonic, subdom, dom, no_func}
next_root_int	half tones	$id - next_{id}$	[0,11]
prev_root_int	half tones	$prev_{id} - id$	[0,11]

Table 3. Chord harmonic descriptors (horizontal)

- **Note descriptors (Figure 3)** For note descriptors, the horizontal context included melodic information while the vertical context included harmonic, ensemble information. Following the approach made by Giraldo [7], we computed horizontal note descriptors using the information of the reference notes' neighbours whereas we computed vertical note descriptors by using the chords' information. Since every note belonged to a chord, the features of the note were merged with the descriptors of the corresponding chord by concatenating both lists and eliminating repeated items.

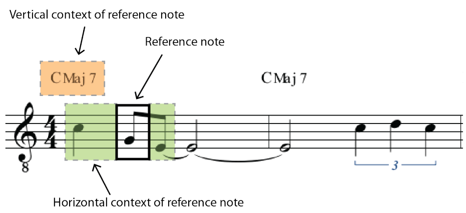

Figure 3. Excerpt of *All Of Me*: Horizontal and Vertical contexts for a reference note

4.2.4 Audio mix Data

For every recording, we performed a semi-automatic alignment between the performance and the score.The tempo varied during the performance because no metronome was used. Hence, beat positions were not equidistant and beat-tracking was performed to create a beat grid which allowed to link the performed information to the score information. We used the algorithm developed by Zapata et al. [27] to track the beats, followed by manual correction. Afterwards, the mean tempo of each song was computed using Equation 4, where $beats$ was the vector of beats computed in the previous step.

$$tempo = round\left(\frac{60}{mean(diff(beats))}\right) \quad (4)$$

4.3 Machine learning

4.3.1 Datasets

As it can be seen in Figure 1, the inputs of the Machine Learning stage were the performance actions for both piano and guitar as well as the score descriptors (for chords and notes). Hence, we constructed three types of datasets, shown in Figure 4.

- **Simple Datasets (D1)**: Horizontal score context. It only contained individual descriptors of the chords or notes.
- **Score Mixed Datasets (D2)**: D1 plus vertical **score** context, which contained merged descriptors of chords and notes.
- **Performance Mixed Datasets (D3)**: D1 plus vertical **performance** context (extracted from the manual

transcriptions of the performances), which contained merged features of chords and notes, taking into account the real interaction between musicians.

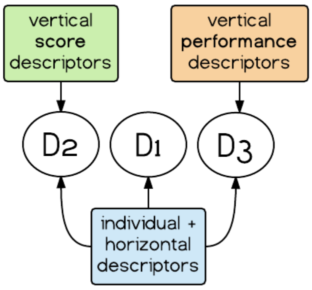

Figure 4. Three different datasets depending on the included descriptors

Therefore, for piano we trained models to learn the function shown in Equation 5 while for guitar, the function to learn is presented in Equation 6.

$$f(Chord) \rightarrow (Den, Wei, Ran) \tag{5}$$

Where:

$Chord$: is a chord only characterised by harmonic descriptors plus melodic descriptors (depending on the dataset)

Den, Wei, Ran: are the predicted density, weight and range labels (high/low), respectively.

$$f(Note) \rightarrow (Emb) \tag{6}$$

Where:

$Note$: is a note characterised by the set of melodic descriptors plus harmonic descriptors (depending on the dataset)

Emb: corresponds to the predicted embellishment label (yes/no).

4.3.2 Feature Selection

The aim of this part was to identify specific score descriptors that best described the previously defined performance actions, so as to train models with the most representative ones. Therefore, for every dataset we evaluated the descriptors by their information gain. Tables 4, 5, 6 and 7 show the best ranked descriptors for density, weight, range and embellishments, respectively.

D1	D2	D3
metP	mtr	metP
chord_dur	metP	chord_dur
function	chord_dur	intHop
type	isChordN	isChordN
tens	type	function
metP	tens	type
		tens

Table 4. Selected features for density

D1	D2	D3
tens	tens	tens
function	function	function
chord_dur	type	type
metP	metP	metP
	isChordN	keyMode
	keyMode	isChordN
	mtr	tens

Table 5. Selected features for weight

D1	D2	D3
numKey	numKey	numKey
function	dur_s	pre_dur_s
type	duration_b	prev_int
tens	pre_dur_b	function
keyMode	nxt_dur_b	type
metP	isChordN	mtr
	function	isChordN
	mtr	tens
	type	keyMode
	tens	metP
	keyMode	
	metP	

Table 6. Selected features for range

D1	D2	D3
phrase	phrase	phrase
dur_b	dur_b	dur_b
dur_s	dur_s	dur_s
pre_dur_b	pre_dur_b	pre_dur_b
pre_dur_s	pre_dur_s	pre_dur_s
onset	onset	onset
	tens	tens
	type	type
	function	function
	isChordN	isChordN
	keyMode	keyMode
		metP

Table 7. Selected features for embellishments

4.3.3 Algorithms

The aim of this stage was to compare the results of the widely used algorithms *Decision Trees*, *Support Vector Machine (SVM)* (with a linear kernel) and *Neural Networks (NN)* (with one hidden layer). We used the implementation of these algorithms in the Weka Data Mining Software [16], utilising the default parameters.

5. RESULTS

Since every performance action contained 3 datasets, we generated a model for each of them. Thus, the results we present include a comparison between the datasets as well as the algorithms.

5.1 Piano data: density, weight and range

We evaluated the accuracy (percentage of correct classifications) using 10-cross fold validation with 10 iterations. We performed statistical testing by using the t-test with a significance value of 0.05 to compare the methods with the baseline (Zero Rule Classifier) and decide if one produced measurably better results than the other.

Table 8 shows the results for **density**. It can be seen that the accuracy increased when ensemble information was considered (datasets D2 and D3). The significant improvements were achieved by the algorithms NN and SVM, being 65.13 the highest accuracy reached with the dataset D2 which consisted in both harmonic and melodic score descriptors. For **weight** (Table 9), none of the results was statistically significant and the performance of the three models can be interpreted as random. The highest results were achieved when only piano information was considered (D1), showing no interaction between this performance action and the guitar melody. Table 10 presents the results for **range**. In that case, the three algorithms reached their maximum accuracy when information of the ensemble performance (D3) was considered, which can be explained as a presence of correlation between the range of the chords performed and the melody the piano player was hearing. Moreover, the results for the algorithms Decision Trees and SVM were statistically significant.

Dataset	Baseline	NN	SVM	Decision Tree
D1	51.82	61.19 ∘	62.13 ∘	53.75
D2	51.82	61.72	65.13 ∘	55.34
D3	51.82	55.75	61.75	57.65

∘, • statistically significant improvement or degradation

Table 8. Accuracy for the models of density in comparison to the baseline using NN, SVM and Decision Trees

Dataset	Baseline	NN	SVM	Decision Tree
D1	53.73	63.52	52.96	54.48
D2	53.73	50.62	49.64	51.85
D3	53.73	57.70	50.90	51.36

∘, • statistically significant improvement or degradation

Table 9. Accuracy for the models of weight in comparison to the baseline using NN, SVM and Decision Trees

Dataset	Baseline	NN	SVM	Decision Tree
D1	56.73	54.51	62.06	63.72
D2	56.73	57.11	60.90	60.93
D3	56.73	58.83	67.85 ∘	67.98 ∘

∘, • statistically significant improvement or degradation

Table 10. Accuracy for the models of range in comparison to the baseline using NN, SVM and Decision Trees

5.2 Guitar data: embellishments

In that case, there was a skewed classes distribution, which led us to evaluate the sensitivity (true positive rate) rather than the accuracy of the model. Table 11 presents the results obtained. It can be seen that, despite the low percentage of sensitivity, the results for the three algorithms increased when considering ensemble information (D2, D3).

Dataset	NN	SVM	Decision Tree
D1	26	20	12
D2	30	38	26
D3	30	32	24

Table 11. Sensitivity percentage for embellishments

6. CONCLUSIONS

In this work we have developed a system which studies the interaction between musicians by using techniques related to computational analysis of expressive music performance and machine learning. We have created a database consisting of recordings of 7 jazz standards played by a quartet (piano, guitar, bass and drums) and their corresponding scores. For processing both the recordings and the scores, we have developed code libraries consisting of specific functions for every stage of the process: select chords, extract vertical and horizontal descriptors for both notes and chords, align and compare the recordings with the score and extract performance actions. Finally, we have generated models for different datasets consisting of information from individual performances and ensemble performances. Based on the accuracy and sensitivity of the models, we have obtained numerical results which have allowed us to estimate the level of interaction between musicians. The data analysis indicated that, in general terms, the performance actions of the accompaniment are influenced by the soloist and vice versa, since both written and performed descriptors contributed to a better performance of the models.

In a future work, it would be interesting to extract other performance actions such as energy or duration for both chords and notes and to study the extent to which the measures are sensitive to the incorporation of other instruments. Moreover, since we have at our disposal a database which contains the recordings of bass and drums, it would be interesting to incorporate both instruments into the analysis. We have observed that the majority of the models got better results with ensemble information but the accuracies of the models could still improve by collecting more data (making new recordings) or extracting more descriptors. Finally, the parameters of the used algorithms could be further investigated so as to improve the results.

7. ACKNOWLEDGEMENTS

This work has been partly sponsored by the Spanish TIN project TIMUL (TIN2013-48152-C2-2-R) and the H2020-ICT-688268 TELMI project.

8. REFERENCES

[1] Josep Lluís Arcos, Ramon Lopez De Mantaras, and Xavier Serra. Saxex: A case-based reasoning system for generating expressive musical performances*. *Journal of New Music Research*, 27(3):194–210, 1998.

[2] Helena Bantulà, Sergio Giraldo, and Rafael Ramírez. A Rule-based System to Transcribe Guitar Melodies. In *Inernational Symposium on Computer Music Multidisciplinary Research (CMMR)*, pages 1–8, 2015.

[3] Alain de Cheveigne and Hideki Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917, 2002.

[4] Tuomas Eerola and Petri Toiviainen. MIDI toolbox: MATLAB tools for music research. *University of Jyväskylä, Jyväskylä, Finland*, 2004.

[5] Anders Friberg. *A quantitative rule system for musical performance*. PhD thesis, PhD Thesis, KTH, Sweden, 1995.

[6] Alf Gabrielsson. The performance of music. *The psychology of music*, 2:501–602, 1999.

[7] Sergio Giraldo. Modelling embellishment, duration and energy expressive transformations in jazz guitar, 2012.

[8] Sergio Giraldo and Rafael Ramírez. Computational generation and synthesis of jazz guitar ornaments using machine learning modeling. In *International Workshop on Machine Learning and Music (MML)*, 2015.

[9] Sergio Giraldo and Rafael Ramírez. Computational Modeling and Synthesis of Timing, Dynamics and Ornamentation in Jazz Guitar Music. In *Inernational Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2015.

[10] Sergio Giraldo and Rafael Ramírez. Computational modeling of ornamentation in jazz guitar music. In *International Symposium on Performance Science*, 2015.

[11] Sergio Giraldo and Rafael Ramírez. Score Sequence Matching for Automatic Ornament Detection in Jazz Music. In *Inernational Conference on New Music Concepts (ICNMC)*, 2015.

[12] Werner Goebl and Caroline Palmer. Synchronization of timing and motion among performing musicians. 2009.

[13] Talar Hopyan, Maureen Dennis, Rosanna Weksberg, and Cheryl Cytrynbaum. Music skills and the expressive interpretation of music in children with williams-beuren syndrome: pitch, rhythm, melodic imagery, phrasing, and musical affect. *Child Neuropsychology*, 7(1):42–53, 2001.

[14] Apple Inc. Logic pro x. http://www.apple.com/logic-pro/.

[15] Marco Marchini. *Analysis of Ensemble Expressive Performance in String Quartets: A Statistical and Machine Learning Approach*. PhD thesis, PhD Thesis, UPF, Barcelona, 2014.

[16] Geoffrey Holmes Bernhard Pfahringer Peter Reutemann Ian H. Witten Mark Hall, Eibe Frank. *"The WEKA Data Mining Software: An Update"*, volume 11. 2009.

[17] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[18] MuseScore. *version 1.3*. 2013.

[19] Caroline Palmer. Music performance. *Annual review of psychology*, 48(1):115–138, 1997.

[20] Rafael Ramirez and Amaury Hazan. Modeling expressive music performance in jazz. In *FLAIRS Conference*, pages 86–91, 2005.

[21] Bruno H Repp. Sensorimotor synchronization: a review of the tapping literature. *Psychonomic bulletin & review*, 12(6):969–992, 2005.

[22] John Rink. *Musical performance: a guide to understanding*. Cambridge University Press, 2002.

[23] Caroline Traube and Michel Bernays. Piano Touch Analysis: a Matlab Toolbox for Extracting Performance Descriptors from High Resolution Keyboard and Pedalling Data. *Journées d'Informatique Musicale (JIM)*, 2012.

[24] G. De Poli A. Friberg R. Bresin W. Goebl, S. Dixon and G. Widmer. *Sound to sense-sense to sound: a state of the art in sound and music computing*, chapter Sense in expressive music performance: Data acquisition, computational studies, and models, pages 195–242. Berln, 2008.

[25] Gerhard Widmer and Asmir Tobudic. Playing mozart by analogy: Learning multi-level timing and dynamics strategies. *Journal of New Music Research*, 32(3):259–268, 2003.

[26] Alan M Wing, Satoshi Endo, Adrian Bradbury, and Dirk Vorberg. Optimal feedback correction in string quartet synchronization. *Journal of The Royal Society Interface*, 11(93):20131125, 2014.

[27] José R. Zapata, André Holzapfel, Matthew E.P. Davies, Joao Lobato Oliveira, and Fabien Gouyon. Assigning a Confidence Threshold on Automatic Beat Annotation in Large Datasets. In *ISMIR 2012, Proceedings of the 13th International Society for Music Information Retrieval Conference.*, number ISMIR, pages 157–162, 2012.