ROBOTABA GUITAR TABLATURE TRANSCRIPTION FRAMEWORK

Gregory Burlet and Ichiro Fujinaga
Centre for Interdisciplinary Research in Music Media and Technology
McGill University, Montréal, Québec, Canada
gregory.burlet@mail.mcgill.ca, ich@music.mcgill.ca

ABSTRACT

This paper presents *Robotaba*, a web-based guitar tablature transcription framework. The framework facilitates the creation of web applications in which polyphonic transcription and guitar tablature arrangement algorithms can be embedded. Such a web application is implemented, and consists of an existing polyphonic transcription algorithm and a new guitar tablature arrangement algorithm. The result is a unified system that is capable of transcribing guitar tablature from a digital audio recording and displaying the resulting tablature in the web browser. Additionally, two ground-truth datasets for polyphonic transcription and guitar tablature arrangement are compiled from manual transcriptions gathered from the tablature website `ultimate-guitar.com`. The implemented transcription web application is evaluated on the compiled ground-truth datasets using several metrics.

1. INTRODUCTION

Tablature has become the primary form of communication between guitarists on the Internet. Guitar tablature is a music notation system with a six-line staff that represents the strings on a guitar. A numeric entry on a line represents the fret to depress on a particular string (Figure 1).

Manually transcribing guitar tablature from an audio recording is a difficult and laborious task, even for experienced guitarists. In response to the time-consuming process of manual transcription, automatic music transcription systems aim to extract a symbolic music score from an acoustical signal [5]. Specifically, automatic guitar tablature transcription systems transform a digital guitar signal into tablature notation. The task of automatic guitar tablature transcription can be decomposed into two subproblems: polyphonic transcription and guitar tablature arrangement. Polyphonic transcription algorithms extract the pitch, onset time, and duration of notes occurring in an audio recording. Guitar tablature arrangement algorithms assign a string and fret combination to each note occurring in an input music score. Adding more ambiguity to the transcription process, the guitar can produce the same note in several ways. For example, there exists six string and fret combinations that can produce the note E4 on a 24-fret guitar in standard tuning (Figure 1).

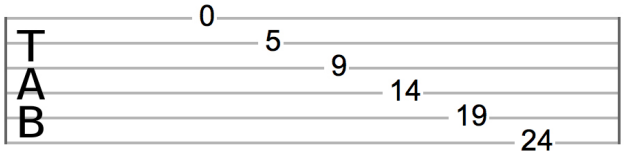

Figure 1. Tablature notation depicting six different string and fret combinations to perform the note E4 on a 24-fret guitar in standard tuning.

While several polyphonic transcription and guitar tablature arrangement algorithms have been proposed in the literature, no frameworks have been developed to facilitate the combination of these algorithms to produce an automatic guitar tablature transcription system. Moreover, after a new polyphonic transcription or guitar tablature arrangement algorithm is developed, the code has no immediately available vessel to be used by music researchers and the large community of guitarists on the Internet.

A web-based guitar tablature transcription framework, entitled *Robotaba*, has been designed and implemented to facilitate the creation of guitar tablature transcription web applications in which polyphonic transcription and guitar tablature arrangement algorithms can be embedded. An existing polyphonic transcription algorithm and a new guitar tablature arrangement algorithm is implemented; these algorithms are embedded in a transcription web application using the Robotaba framework. Two ground-truth datasets have been compiled to evaluate the performance of the implemented guitar tablature transcription system.

The structure of this paper is as follows: The next section provides an overview of polyphonic transcription and guitar tablature arrangement algorithms. Section 3 presents the design of Robotaba, followed by a description of the implemented transcription web application in Section 4. Section 5 presents the compiled ground-truth datasets and their use in the evaluation of the implemented polyphonic transcription and guitar tablature arrangement algorithms.

2. LITERATURE REVIEW

Though the transcription of monophonic musical passages is considered a solved problem [5], the transcription of

polyphonic music is still an open problem [1]. Several techniques have been proposed to accomplish the task of polyphonic transcription, including human audition modelling [12]; salience methods, which apply transformations to the input audio signal in order to emphasize the underlying fundamental frequencies [17]; iterative and joint fundamental frequency estimation algorithms followed by note tracking algorithms [2]; and a variety of machine learning methods such as non-negative matrix factorization [14], support vector machines [8], neural networks [6], and hidden Markov models [10].

Several algorithms have also been proposed to generate tablature from a symbolic music score according to criteria that minimizes the performance difficulty of the tablature arrangement. Heijink and Meulenbroek [4] established that guitarists have a disposition toward instrument fingering positions that are biomechanically easy to perform. Specifically, guitarists favour hand positions near the head of the guitar and avoid composing arrangements that require extensive hand repositioning and large finger spans. Shortest-path graph search algorithms [13], constraint satisfaction algorithms [9], neural networks [15], and genetic algorithms [16] have been used to search for tablature arrangements with minimal performance difficulty.

3. FRAMEWORK DESIGN

The Robotaba transcription framework is composed of three modules: a polyphonic transcription module, a guitar tablature arrangement module, and a guitar tablature engraving module (Figure 2).

Three benefits arise from this modular design: First, each module can be used independently or together. Used independently, an input file is sent directly to a module for processing, which returns a result instead of passing the output to the next module in the workflow. Using each module in sequence, guitar tablature can be generated from an input audio file and displayed in the web browser. Second, the modular design facilitates algorithm interchangeability. Assuming an algorithm produces valid output, it can be inserted into a module without disturbing the functionality of surrounding modules. As a result, the transcription framework can accommodate new state-of-the-art polyphonic transcription or guitar tablature arrangement algorithms without substantial changes to the web application. Third, the use of a single symbolic music file format for data interchange between modules promotes polyphonic transcription and tablature arrangement algorithms to adhere to a common interface. Robotaba uses the 2012 release of the music encoding initiative (MEI): an extensible markup language (XML) file format that encodes symbolic music notation in a hierarchical fashion [11].

3.1 Polyphonic Transcription Module

The polyphonic transcription module accepts an audio file as input, which is passed to the polyphonic transcription algorithm embedded in the module. The polyphonic transcription algorithm is responsible for generating an MEI file containing the estimates of note events occurring in the input audio file. The polyphonic transcription module optionally postprocesses the output symbolic music file by limiting the number of simultaneous notes to six, and discarding or transposing estimated notes that are outside of the range of a specific guitar. Properties of the guitar (number of frets, tuning, and capo position) [1] and postprocessing options are specified by the user of the web application.

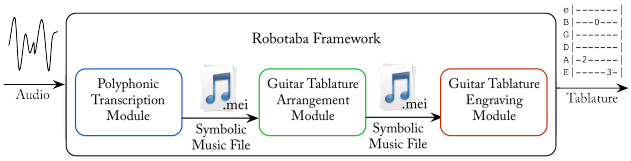

Figure 2. Modular architecture of the Robotaba guitar tablature transcription framework.

3.2 Guitar Tablature Arrangement Module

The guitar tablature arrangement module accepts an MEI file as input, which is mandatorily preprocessed in an identical manner as the postprocessing step of the polyphonic transcription module described in the previous section. The preprocessed MEI file is subsequently passed to the guitar tablature arrangement algorithm embedded in the module, which is responsible for assigning a guitar string and fret combination to each note occurring in the MEI file.

3.3 Guitar Tablature Engraving Module

The guitar tablature engraving module is responsible for parsing an MEI file containing a sequence of note events that have each been assigned a guitar string and fret combination and displaying the encoded tablature in the web browser. Robotaba uses the digital guitar tablature engraving library *AlphaTab* [2] to render tablature symbols on the hypertext markup language (HTML) canvas element. AlphaTab parses drawing scripts called AlphaTex, in which structured keywords inform the rendering engine about the contents of the tablature and how it should be displayed. When an MEI file is passed to the tablature engraving module, the contents of the file are converted to an AlphaTex drawing script to be rendered by AlphaTab. A rendered tablature score can be seen in Figure 1.

3.4 Framework Implementation

Robotaba is implemented using Django, [3] a Python web framework that facilitates rapid development of database-driven web applications by automatically translating Python classes called *models* into relational database tables. Models are created for audio and symbolic music files, as well as their associated metadata. Additionally, the database

[1] A capo is a device that is clipped onto the fretboard of a guitar and raises the pitches of the open strings.

[2] `www.alphatab.net`

[3] `www.djangoproject.com`

stores the user-specified parameters used to generate a transcription, such as the guitar properties and file pre and post-processing options, to allow reproducibility of results.

4. TRANSCRIPTION WEB APPLICATION

Using the Robotaba framework, a web application for guitar tablature transcription is developed that incorporates the polyphonic transcription and guitar tablature arrangement algorithms described in this section. [4]

4.1 Polyphonic Transcription Algorithm

A polyphonic transcription application, which uses the state-of-the-art algorithm proposed by Zhou and Reiss [17], is implemented. This algorithm was selected for several reasons: First, this algorithm ranked highest out of the polyphonic transcription algorithms evaluated in the Music Information Retrieval Evaluation eXchange (MIREX) on the piano dataset from 2007–2012 when considering the accuracy of pitch and note onset times only. Second, the authors tuned underlying parameters of the algorithm according to a dataset composed of both piano and guitar recordings [17]. Third, this algorithm is capable of performing polyphonic transcriptions in realtime. Finally, the source code of this algorithm is open source.

The aforementioned polyphonic transcription algorithm is distributed as a Vamp plugin written in the C++ programming language. A Vamp plugin is an audio feature extraction module that must be "plugged into" a host application. [5] In order to integrate the polyphonic transcription Vamp plugin into Robotaba, the plugin is first divorced from the host to produce a standalone application. A Python interface is created using the Boost.Python library [6] to access the standalone application from Robotaba. A Python application is implemented, which sets parameters of the polyphonic transcription algorithm, imports an audio file, sends the audio data to the Python bindings of the polyphonic transcription Vamp plugin, and generates an MEI document containing the resulting note event estimates.

4.2 Guitar Tablature Arrangement Algorithm

A new guitar tablature arrangement algorithm entitled *A-star-guitar* is developed, written in the Python programming language, and embedded in the Robotaba guitar tablature arrangement module. Extending the approach proposed by Sayegh [13], which uses the Viterbi algorithm to search for an optimal path through a weighted graph of candidate fretboard locations for notes in a *monophonic* musical passage, A-star-guitar uses the popular A* pathfinding algorithm [3] to search for an optimal tablature arrangement of a sequence of notes and chords in a *polyphonic* musical passage.

The A* pathfinding algorithm searches for an optimal path through a directed graph, in which vertices represent candidate string and fret combinations for a note or chord in a symbolic music score. Candidate string and fret combinations are calculated by considering the number of frets on the user's guitar, the tuning, and optional fret position of a capo. Vertices that correspond to adjacent notes or chords in the music score are connected by an edge.

The weight of an edge $w_{ij} \in \mathbb{N}$ between vertices i and j represents the biomechanical difficulty associated with the transition between the two hand positions on the fretboard. Following the study of left-hand movements of professional guitar players by Heijink and Meulenbroek [4], the edge weight between two vertices is the cumulation of three biomechanical complexity factors: the fretwise distance that the fretting hand must move to accommodate note transitions, the fretwise finger span required to perform chords, and a penalty of one if the fretting hand surpasses the seventh fret. The values of this penalty and fret threshold number were chosen on the basis of preliminary tests and encourage tablature arrangements near the beginning of the fretboard. In the event of a note played by depressing fret number f, followed by a chord comprised of multiple notes with the set of fret numbers g, the fretwise distance is calculated by the expression

$$\operatorname{abs}\left(f - \left(\frac{\max(g) - \min(g)}{2}\right)\right). \qquad (1)$$

The A* algorithm uses a heuristic function that returns an estimate of the summation of edge weights from a given vertex to the goal vertex. This must be an admissible heuristic: the cost of traversing the graph from a vertex to the goal vertex must not be overestimated. Using the proposed biomechanical complexity factors to assign weights to edges, the heuristic function is zero for all vertices and the resulting algorithm is effectively the Dijkstra pathfinding algorithm [3]. To illustrate why the heuristic function is zero, consider a music score with a sequence of identical notes that may be performed by repeatedly plucking an open string on the guitar. [7] In this case, each edge connecting the open-string vertices has a weight of zero: there is no fretwise distance between notes, there are no chords in the music score, and no frets are depressed.

Figure 3 displays a directed acyclic weighted graph of candidate string and fret combinations for an example music score consisting of four notes above the tablature arrangement produced by the A* search algorithm.

5. TRANSCRIPTION EVALUATION

This section focuses on the procedure for evaluating the implemented transcription web application, specifically the polyphonic transcription and guitar tablature arrangement algorithms embedded within.

5.1 Ground-truth Datasets

For the purposes of evaluating polyphonic transcription and guitar tablature arrangement algorithms, two new ground-truth datasets are presented. The datasets were compiled

[4] ddmal.music.mcgill.ca/robotaba
[5] www.vamp-plugins.org
[6] www.boost.org/libs/python/doc
[7] The term *open string* refers to a pluck whereby no fret is depressed.

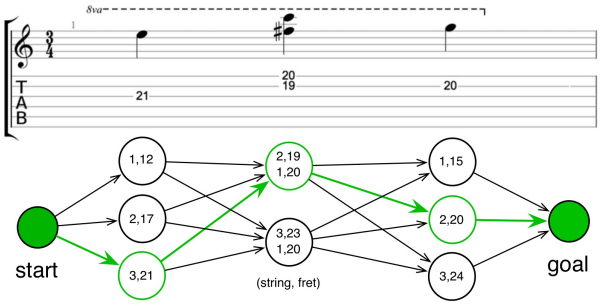

Figure 3. A directed acyclic weighted graph of candidate string and fret combinations for a music score consisting of four notes. The bold edges in the graph identify the optimal path found by the A* search algorithm. Edge weights are omitted for space considerations.

by harvesting the wealth of community-moderated and publicly available manual transcriptions of popular musical works uploaded to the *Ultimate Guitar* tablature website. [8]

Manual transcriptions were downloaded as *Guitar Pro* symbolic music files: a proprietary file format that is read and manipulated by the Guitar Pro desktop application. [9] 75 Guitar Pro files were selected from the Ultimate Guitar Top 100 list and the Ultimate Guitar Fresh Tabs list. The Ultimate Guitar Top 100 list sorts every Guitar Pro file uploaded to the website by its rating—from one to five stars—and displays the top 100 files. The Ultimate Guitar Fresh Tabs list is a catalogue of Guitar Pro files that have recently been uploaded to the website and is sorted by number of views. Only uploaded tablature with a five-star rating agreed upon by at least ten unique users was considered for selection. Tablature was selected on the basis of musical genre, average polyphony, and tempo, in order to compile a set of pieces with a variety of different attributes.

5.1.1 Polyphonic Guitar Transcription Dataset

To form the polyphonic guitar transcription dataset, several preprocessing steps were first applied to each Guitar Pro file. These symbolic music files often contain multiple instrument tracks, many of which are not guitar tracks. Extraneous tracks containing instruments such as the bass guitar, drums, and vocals were removed. Finally, all tempo, volume, and pan automations were removed from the remaining guitar track.

Each guitar track was then synthesized using Guitar Pro's *clean* and *distortion* guitar model to create two audio files. [10] Subsequently, a file listing the pitch, onset time, and duration of notes occurring in the synthesized audio files was generated. This ground-truth file was created by first exporting the Guitar Pro file as a MusicXML file and retrieving pertinent information about the music score from a MusicXML parser program, which was originally developed to gather information from symbolic music scores for the purpose of finding patterns in the relationships between melody, lyrics, and instrumentation [7].

The polyphonic guitar transcription dataset consists of 75 isolated guitar tracks; 125,192 note events; 30,914 chords; and an average tempo of 112 beats per minute. There are approximately five and a half hours of clean guitar recordings and five and a half hours of distortion guitar recordings, yielding approximately eleven hours of synthesized audio in total.

5.1.2 Guitar Tablature Arrangement Dataset

The ground-truth dataset for guitar tablature arrangement was compiled using the same 75 Guitar Pro files that were collected and preprocessed for the polyphonic guitar transcription dataset. Excerpts were selected from the symbolic music files on the basis of overall length (no one excerpt exceeds eight measures), the number of times the excerpt occurred throughout the entire piece, and the average polyphony of the excerpt. Each excerpt in Guitar Pro was exported as a MusicXML file, which was subsequently converted to an MEI file using an open-source MusicXML to MEI conversion application written in Python. [11]

The ground-truth dataset for guitar tablature arrangement consists of 75 tablature arrangements—with 4,845 notes and 1,143 chords—encoded in both the MEI and MusicXML symbolic music file formats.

5.2 Polyphonic Transcription Evaluation

The implemented polyphonic transcription algorithm is evaluated using the compiled ground-truth dataset for polyphonic guitar transcription. Using an evaluation procedure similar to the MIREX multiple fundamental frequency estimation and tracking task, the output of the polyphonic transcription algorithm on each audio recording in the dataset is compared to the corresponding ground-truth note events using the metrics of precision, recall, and f-measure. Precision describes the ratio of correctly transcribed note events to the total number of estimated note events. Recall describes the ratio of correctly transcribed note events to the total number of ground-truth note events. The f-measure statistic combines precision and recall into a single metric. An estimated note event is deemed to be correctly transcribed if the fundamental frequency is within 50 cents of the ground-truth note event and the onset time is within a 50-millisecond range of the ground-truth note event.

Four experiments were conducted: Experiment 1 reports the precision, recall, and f-measure of the polyphonic transcription algorithm on the clean synthesized guitar recordings, considering the accuracy of note pitch and onset time only. Experiment 2 is identical to the first experiment, except that octave errors are ignored. Experiment 3 and 4 are identical to Experiment 1 and 2, respectively, except that the polyphonic transcription algorithm is evaluated on the distortion synthesized guitar recordings in the ground-truth dataset. The results of these experiments are presented in Table 1, alongside the results of the polyphonic transcription algorithm in the 2008 MIREX experiments. In MIREX 2008, the polyphonic transcription

[8] `www.ultimate-guitar.com`
[9] `www.guitar-pro.com`
[10] *Clean* guitar refers to a guitar signal with no audio effects applied.
[11] `github.com/gburlet/musicxml-mei-conversion`

	PRECISION	RECALL	f-MEASURE
Experiment 1	0.71	0.42	0.50
Experiment 2	0.75	0.45	0.53
Experiment 3	0.48	0.36	0.39
Experiment 4	0.56	0.42	0.46
MIREX 2008	0.74	0.78	0.76

Table 1. Results of four experiments evaluating the implemented polyphonic transcription algorithm, alongside the results of the same algorithm in MIREX 2008.

algorithm was evaluated on a relatively small dataset of piano recordings and considered the accuracy of note pitch and onset time.

Analyzing the results, the performance of the polyphonic transcription algorithm on clean guitar recordings is significantly better than on distortion guitar recordings. A possible contributing factor to the reduced transcription performance on distortion guitar recordings could be the modification of the relative amplitudes of harmonics in the frequency domain of the guitar signal caused by the distortion audio effect. In line with past research, the performance of the polyphonic transcription algorithm improves when ignoring octave errors. The polyphonic transcription algorithm receives inferior results on the compiled dataset in comparison to the piano dataset used in the 2008 MIREX. Apart from containing pieces of a different genre, one possible explanation for the degraded performance is that the synthesized guitar recordings in the compiled dataset contain ornamentation such as slides, bends, hammer-ons, hammer-offs, palm-muting, and dead notes. The piano is incapable of replicating many of these ornaments.

5.3 Guitar Tablature Arrangement Evaluation

The implemented guitar tablature arrangement algorithm is evaluated using the compiled ground-truth dataset for guitar tablature arrangement. Since there is no standardized method for evaluating guitar tablature arrangement algorithms, a new evaluation method is used. The evaluation is performed by calculating a fitness value

$$f = \frac{1}{f_{GT}\left(1 + \sum_{i=1}^{N} w_i\right)} \tag{2}$$

for each generated tablature arrangement that is normalized with respect to the fitness of the corresponding ground-truth tablature arrangement f_{GT}. In Equation 2, N is the number of vertices in the optimal path returned by the A* search algorithm and w_i is the weight of the i^{th} edge along the optimal path. The normalized fitness value $f \in \mathbb{R}^+$ is interpreted as the biomechanical ease of performing a generated tablature arrangement relative to the ground-truth tablature arrangement.

By the central limit theorem, it can be assumed that the distribution of normalized fitness values for guitar tablature arranged using the A* search algorithm approximately

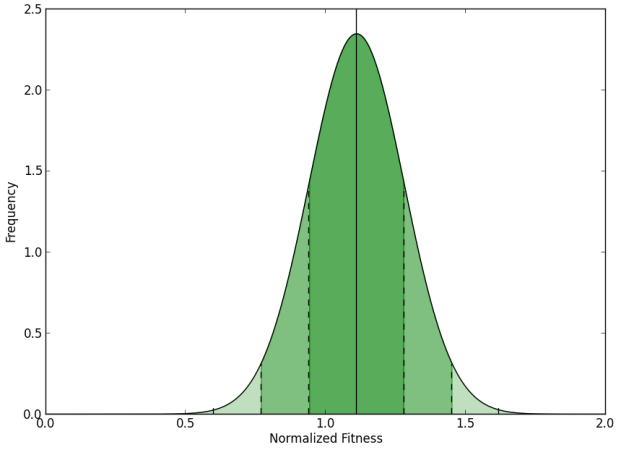

Figure 4. Gaussian distribution $N\left(\hat{\mu} = 1.11, \hat{\sigma} = 0.17\right)$ of normalized fitness values for guitar tablature arranged using the A* search algorithm.

follows a normal distribution. The median $\hat{\mu}$ and standardized median absolute deviation $\hat{\sigma}$ are robust descriptive statistics that are used to describe the central tendency and dispersion, respectively, of this distribution. Robust descriptive statistics are used because they are less sensitive to outliers present in small datasets in comparison to the sample mean and sample standard deviation. The resulting Gaussian distribution $N\left(\hat{\mu} = 1.11, \hat{\sigma} = 0.17\right)$ of normalized fitness values is displayed in Figure 4.

Having a median normalized fitness value of $\hat{\mu} = 1.11$, the A* search algorithm generates guitar tablature arrangements that, on average, are of similar performance difficulty as tablature arranged by humans. By the empirical rule, 68.2% of tablature arrangements generated by the A* search algorithm are estimated to have normalized fitness values that lie within one standard deviation of the mean (0.94–1.28).

6. CONCLUSION

An open-source and web-based guitar tablature transcription framework, entitled *Robotaba*, is designed and implemented. [12] The framework facilitates the rapid development of guitar tablature transcription web applications, providing a vessel for music researchers to publicize their polyphonic transcription and guitar tablature arrangement algorithms, while allowing researchers to focus on algorithm development instead of application development.

As a proof of concept, a guitar tablature transcription web application is developed using the Robotaba framework. An open-source polyphonic transcription application is implemented, which uses the state-of-the-art algorithm proposed by Zhou and Reiss [17]. [13] A new guitar tablature arrangement algorithm, which uses the popular A* pathfinding algorithm, is proposed and implemented as an open-source application. [14]

[12] `github.com/gburlet/robotaba`
[13] `github.com/gburlet/zhoutranscription`
[14] `github.com/gburlet/astar-guitar`

Two ground-truth datasets are compiled from manual tablature transcriptions downloaded from the *Ultimate Guitar* tablature website. The polyphonic guitar transcription dataset consists of 150 synthesized guitar recordings totalling approximately 11 hours of audio, which have been semi-automatically annotated. The guitar tablature arrangement dataset consists of 75 hand-arranged tablature scores encoded in the MEI file format. It is hoped that these datasets will stimulate future research in the area of automatic guitar tablature transcription.

Using the compiled ground-truth datasets, the implemented polyphonic transcription and guitar tablature arrangement algorithms are evaluated. Several experiments illustrate that the polyphonic transcription algorithm exhibits reduced precision and recall on guitar recordings with a distortion audio effect applied. Moreover, in comparison to the quality of transcriptions produced by the polyphonic transcription algorithm on piano recordings in the 2008 MIREX evaluation, the algorithm receives inferior results on the compiled dataset of synthesized guitar recordings. An evaluation of the proposed guitar tablature arrangement algorithm indicates that the algorithm arranges tablature that, on average, is of similar performance difficulty as human-arranged tablature.

Now that a framework exists to combine and evaluate polyphonic transcription and guitar tablature arrangement algorithms, more work can be done to improve the algorithms themselves.

7. ACKNOWLEDGEMENTS

This research was supported by the Social Sciences and Humanities Research Council of Canada. Special thanks are owed to the individuals who have uploaded manual tablature transcriptions to the Ultimate Guitar website, as well as Ruohua Zhou and Joshua Reiss for the open-source polyphonic transcription algorithm used in this research.

8. REFERENCES

[1] Benetos, E., S. Dixon, D. Giannoulis, H. Kirchoff, and A. Klapuri. 2012. Automatic music transcription: Breaking the glass ceiling. In *Proceedings of the International Society for Music Information Retrieval Conference*, Porto, Portugal, 1002–7.

[2] Chang, W., A. W. Su, C. Yeh, A. Roebel, and X. Rodet. 2008. Multiple-F0 tracking based on a high-order HMM model. In *Proceedings of the International Conference on Digital Audio Effects*, Espoo, Finland.

[3] Hart, P., N. Nilsson, and B. Raphael. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics* 4 (2): 100–7.

[4] Heijink, H., and R. Meulenbroek. 2002. On the complexity of classical guitar playing: Functional adaptations to task constraints. *Journal of Motor Behavior* 34 (4): 339–51.

[5] Klapuri, A. 2004. Automatic music transcription as we know it today. *Journal of New Music Research* 33 (3): 269–82.

[6] Marolt, M. 2004. A connectionist approach to automatic transcription of polyphonic piano music. *IEEE Transactions on Multimedia* 6 (3): 439–49.

[7] Nichols, E., D. Morris, S. Basu, and C. Raphael. 2009. Relationships between lyrics and melody in popular music. In *Proceedings of the International Society for Music Information Retrieval Conference*, Kobe, Japan, 471–6.

[8] Poliner, G., and D. Ellis. 2007. Improving generalization for polyphonic piano transcription. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 86–9.

[9] Radicioni, D., and V. Lombardo. 2005. Guitar fingering for music performance. In *Proceedings of the International Computer Music Conference*, Barcelona, Spain, 527–30.

[10] Raphael, C. 2002. Automatic transcription of piano music. In *Proceedings of the International Society for Music Information Retrieval Conference*, Paris, France, 1–5.

[11] Roland, P. 2002.The music encoding initiative (MEI). In *Proceedings of the International Conference on Musical Applications using XML*, Milan, Italy, 55–9.

[12] Ryynänen, M., and A. Klapuri. 2005. Polyphonic music transcription using note event modeling. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 319–22.

[13] Sayegh, S. 1989. Fingering for string instruments with the optimum path paradigm. *Computer Music Journal* 13 (3): 76–84.

[14] Smaragdis, P., and J. Brown. 2003. Non-negative matrix factorization for polyphonic music transcription. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 177–80.

[15] Tuohy, D., and W. Potter. 2006. An evolved neural network/HC hybrid for tablature creation in GA-based guitar arranging. In *Proceedings of the Computer Music Conference*, New Orleans, LA, 576–9.

[16] Tuohy, D., and W. Potter. 2006. GA-based music arranging for guitar. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Vancouver, BC, 1065–70.

[17] Zhou, R., and J. Reiss. 2008. A real-time polyphonic music transcription system. In the *Music Information Retrieval Evaluation eXchange*, `www.music-ir.org/mirex/abstracts/2008/F0_zhou.pdf`.