HMM-BASED AUTOMATIC ARRANGEMENT FOR GUITARS WITH TRANSPOSITION AND ITS IMPLEMENTATION

Gen Hori† **and Shigeki Sagayama**‡
†Faculty of Business Administration, Asia University
5-24-10 Sakai, Musashino-shi, Tokyo 180-8629, Japan
‡School of Interdisciplinary Mathematical Sciences, Meiji University
4-21-1 Nakano, Nakano-ku, Tokyo 164-8525, Japan
hori@brain.riken.jp

ABSTRACT

Automatic generation of guitar tablatures for given songs has been one of research interests in music information processing. Furthermore, some of recent studies have attempted automatic arrangement of given songs before tablature generation for producing guitar scores for songs composed for other instruments. In our previous work, we have formulated "fingering decision" and "arrangement" in a unified framework based on hidden Markov model (HMM) whose hidden states are the left hand forms on the fingerboard and an observed sequence is a note sequence of a given song. The purpose of the present paper is to extend the HMM-based automatic arrangement to "arrangement with transposition" and introduce a web application that implements the arrangement with transposition. The optimal transposition for arrangement of a given song is obtained through a full search for all the possible keys where the resulting arrangements are evaluated based on the probabilities of the sequences of the left hand forms.

1. INTRODUCTION

Tablature generation for the guitar is not a straightforward task because there are several left hand forms to play a single chord or even a single note and finding the optimal sequence of left hand forms for a given song takes some experience. This is why automatic generation of guitar tablatures has been among research interests in music information processing. Furthermore, some of recent studies in this direction have attempted "automatic arrangement" of given songs before tablature generation aiming at automatic generation of guitar scores for songs that can not be played by the guitar in their original forms. While "fingering decision" is a task of determining which finger should be placed on which string and fret for each note given a guitar score without tablature, "arrangement" is a task of finding a reasonable fingering for a given score which is not playable by the guitar due to the limitations of the pitch range or the number of voices (simultaneous notes). It makes as few modifications as possible to the given score to make it playable by the guitar and then determines a fingering for the modified version of the score. In our previous work [1, 2], we have developed a unified framework for solving "fingering decision" and "arrangement" based on HMM (hidden Markov model).

The purpose of the paper is to extend the HMM-based "automatic arrangement" to "arrangement with transposition" and introduce a web application that implements "fingering decision" as well as "arrangement with transposition." Even when all the notes of the given piece are within the pitch range of the guitar, it is still meaningful to transpose the given piece up or down to find better arrangements and easier fingerings. As for the fingering decision, that is a subproblem of arrangement, several works have been made in the last two decades. Sayegh [3] introduced "optimum path paradigm" to fingering decision of generic string instruments. Miura et al. [4] developed software that generates guitar fingerings for given melodies (sequences of single notes). Radicioni et al. [5] extended Sayegh's approach paying attention to cognitive aspects underlying the fingering decision. Radisavljevic and Driessen [6] proposed a method for designing cost functions required in dynamic programming (DP) for fingering decision. Tuohy and Potter [7] introduced a genetic algorithm (GA) for fingering decision and Tuohy [8] extended their approach to arrangement for guitars. Baccherini et al. [9] introduced finite state automaton to fingering decision of generic string instruments. Comparing to those previous works, the novelty of our work mainly lies in its stochastic approach and intentional use of transposition.

The rest of the paper is organized as follows. Section 2 introduces HMM-based fingering decision and sets the HMM parameters such as the state transition and output probabilities so that the HMM performs fingering decision. Section 3 extends the HMM-based fingering decision to HMM-based automatic arrangement by adding output symbols to HMM. Section 4 introduces the transposition before arrangement and discusses how to find the optimal transposition. Section 5 introduces a web application that implements HMM-based automatic arrangement with transposition. Section 6 concludes the paper and discusses related future works.

Throughout the paper, we suppose a guitar with six strings and 19 frets in the standard tuning [1] .

[1] E2-A2-D3-G3-B3-E4

2. HMM-BASED FINGERING DECISION

For fingering decision, we employ HMM whose hidden states are left hand forms and output symbols are chords played by the left hand forms. The HMM outputs only pieces that can be played by the guitar.

2.1 HMM for fingering decision

In a guitar performance, a piece of music (a sequence of notes) is played by a fingering (a sequence of left hand forms). Conversely, "fingering decision" is a task of finding a fingering that plays the given piece of music. Generally, there are several fingerings for a single piece of music and various factors influence the choice of fingerings.

Fig.1 presents the hidden Markov model (HMM) we employ for fingering decision whose hidden states are left hand forms and output symbols are chords that are played by the forms. Here we restrict our attention to the pieces that can be considered as a "sequence of chords" where a chord is a set of notes that start and stop together. We note that, in the HMM for fingering decision, each hidden state (left hand form) outputs a unique output symbol (chord) while several hidden states can output the same output symbol. In this framework, the given sequence of chords (a piece of music) is considered to be generated from a hidden sequence of forms (a fingering). Although there are several fingerings for a single piece of music, the most probable fingering can be determined based on the HMM parameters such as the state transition and output probabilities discussed in the following subsection. Those HMM parameters model various factors that influence the choice of fingerings. The problem of finding the most probable sequence of hidden states is called the "decoding problem" and can be solved efficiently using the Viterbi algorithm [10].

2.2 HMM parameters

In standard applications of HMM, parameters such as the state transition and output probabilities are usually estimated using training data but HMM for fingering decision has a huge number of hidden states for which it is difficult to prepare enough training data. We choose to set those parameters manually as explained in the following.

We set the probability of the state transition from the form q_i to the form q_j given the time interval d_t between those two forms as

$$\begin{aligned} a_{ij}(d_t) &= p(\, z_t = q_j \mid z_{t-1} = q_i,\ d_t) \\ &\sim \frac{1}{2d_t} \exp(-\frac{|I_i - I_j|}{d_t}) \\ &\quad \times \frac{1}{1+I_j} \times \frac{1}{1+W_j} \times \frac{1}{1+N_j} \end{aligned}$$

where the first term of the right hand side is the density function of the Laplace distribution with the variance d_t. We define three difficulty levels of the form q_i, that is, the index finger position I_i, the width W_i and the number of working fingers N_i, and reflect them independently to the state transition probability. The number of working fingers N_i is obviously one of the difficulty levels of the form q_i. The index finger position I_i is also a difficulty level of the form q_i because a parallel translation to a higher position makes playing a chord more difficult. Also $|I_i - I_j|$ represents the movement of the left hand along the neck. Usually, guitarist's index finger position is not changing all the time but is staying at a position for several notes and leaps a few frets to a new position. Consequently, the distribution of the difference of the index finger position is sparse and concentrates on the center. To approximate such a sparse distribution concentrated on the center, we employ the Laplace distribution. The time interval d_t between two forms inhibits the dependency of the next form choice on the previous form. For example, when the time interval is very long, the choice of next form is almost independent of the previous form. To take such inhibition into account, we let the variance of the Laplace distribution be proportional to the time interval d_t. This is an extension of standard HMM introduced by Bengio and Frasconi [11] and is called "input-output HMM."

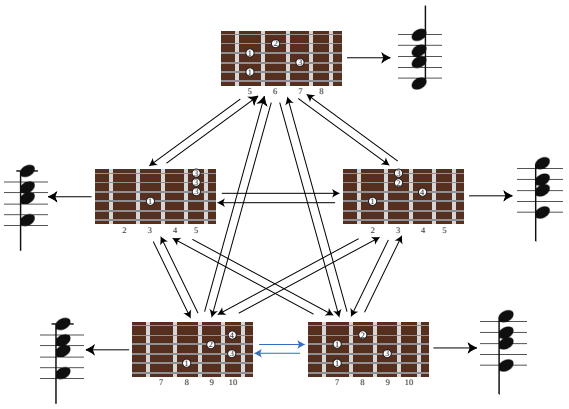

Figure 1. HMM for fingering decision: The left hand forms are hidden states and the chords played by the forms are the output symbols. Each hidden state (left hand form) outputs a unique output symbol (chord) while several hidden states can output the same output symbol.

We set the output probability of the chord x_t from the form q_i as

$$\begin{aligned} b_{it} &= p(\, x_t \mid z_t = q_i \,) \\ &\sim \begin{cases} 1 & (\text{if } x_t = chord(q_i)) \\ 0 & (\text{if } x_t \neq chord(q_i)) \end{cases} \end{aligned}$$

where $chord(q_i)$ denotes the chord played by the form q_i. This guarantees that the most probable sequence of forms is a fingering that plays the given piece.

3. HMM-BASED AUTOMATIC ARRANGEMENT

We employ the same input-output HMM for fingering decision with additional output symbols. HMM with additional output symbols can output pieces that can not be played by the guitar and relate such pieces to form sequences.

3.1 HMM for automatic arrangement

Basically the same HMM we employ for fingering decision can perform automatic arrangement by adding output symbols. In the HMM for fingering decision in Fig.1, each form outputs the chord played by the form. This guarantees that the most probable sequence of forms is a fingering that plays the given piece. On the other hand, in the HMM for automatic arrangement in Fig.2, each form also outputs some additional chords that can not be played by the guitar but can be modified to the chord played by the form. This qualifies the most probable sequence of forms as a fingering that plays a sequence of chords similar to the given piece.

Such chords that can not be played by the guitar can be modified to playable ones by omitting notes. When we omit notes of unplayable chords, we have to pay attention to the top and bottom notes that play important roles to create the impression of the piece. First, the top notes of chords basically form the melody line of the piece and cannot be omitted. If any chord includes a note above the pitch range of the guitar, we have no other choice than to transpose the piece down. We consider such "arrangement with transposition" in Section 4. Second, the bottom notes of chords are the "roots" and should not be omitted if possible. If a bottom note is below the pitch range of the guitar, it is better to move it up an octave. In our formulation of arrangement for guitars, we modify unplayable chords of a given piece using the following two operations: (1) to omit a note, and (2) to change octaves of a note. We use the operation (2) only when the changed note does not exceed the top note of the chord. If the changed note overlaps with an existing note, then the changed note is omitted.

3.2 Output probability for arrangement

For automatic arrangement, we set the transition probabilities exactly the same as the HMM for fingering decision and change the output probability according to the additional output symbols. We set the output probability b_{it} to zero if the t-th chord x_t of a given piece cannot be modified to the chord played by the form q_i using the operations (1) and (2) explained in previous subsection. Otherwise we set the output probability b_{it} to a positive value. This setting of the output probability implements the HMM in Fig.2 and makes the most probable sequence of forms qualified as a fingering that plays a sequence of chords similar to the given piece. Furthermore, to choose chords with the minimum modifications, the output probability b_{it} needs to be a monotone decreasing function of the number of the operations required to modify the t-th chord x_t of a given piece to the chord played by the form q_i. For this purpose, we set the output probability of the chord x_t from the form q_i as

$$b_{it} = p(\, x_t \mid z_t = q_i \,) \sim \begin{cases} \dfrac{1}{1+M_{it}} & (\text{if } x_t \Rightarrow chord(q_i)) \\ 0 & (\text{if } x_t \not\Rightarrow chord(q_i)) \end{cases}$$

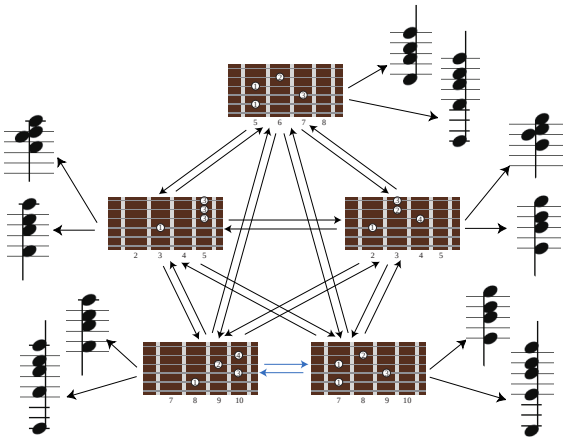

Figure 2. HMM for automatic arrangement: Basically the same HMM for fingering decision can perform automatic arrangement by adding output symbols. Each left hand form outputs the chord played by the form as well as those with a few modifications so that the HMM can output pieces that can not be played by the guitar.

where M_{it} denotes the number of operations (1) and (2) required to modify the chord x_t to the chord played by the form q_i and we write $x_t \Rightarrow chord(q_i)$ when the chord x_t can be modified to the chord played by the form q_i using the operations.

4. ARRANGEMENT WITH TRANSPOSITION

As we have pointed out in the previous section, when the highest note of the given piece exceeds the highest pitch of the guitar, we have no choice but to transpose the piece down before arrangement. Such pieces need to be transposed down at least such number of semitones that the highest note of the piece coincides with the highest pitch of the guitar. In addition, transposing down more semitones can help to find better arrangements and easier fingerings. Generally, even when all the notes of the given piece are within the pitch range of the guitar, it is still meaningful to transpose the given piece up or down to find better arrangements and easier fingerings.

However, for implementing the transposition before arrangement, the main problem is how to find the optimal number of semitones for transposition that gives the best results in arrangement and fingering. All the transpositions are not simply parallel displacements on the fingerboard due to the pitch ranges of the strings, that is, all the keys are not equivalent for the guitar. Each key has a possibility of clever arrangement or fingering exploiting open strings that can not be applied to any other keys and we can not see it until we perform arrangement or fingering decision.

In our formulation, the evaluation of arrangement or fingering is well-defined as the probability of the sequence of forms that gives the arrangement or the fingering. When one key is selected for transposition, the Viterbi algorithm quickly finds the most probable sequence of forms that outputs the given piece transposed to the key. Then we can compare two keys or more for transposition using the

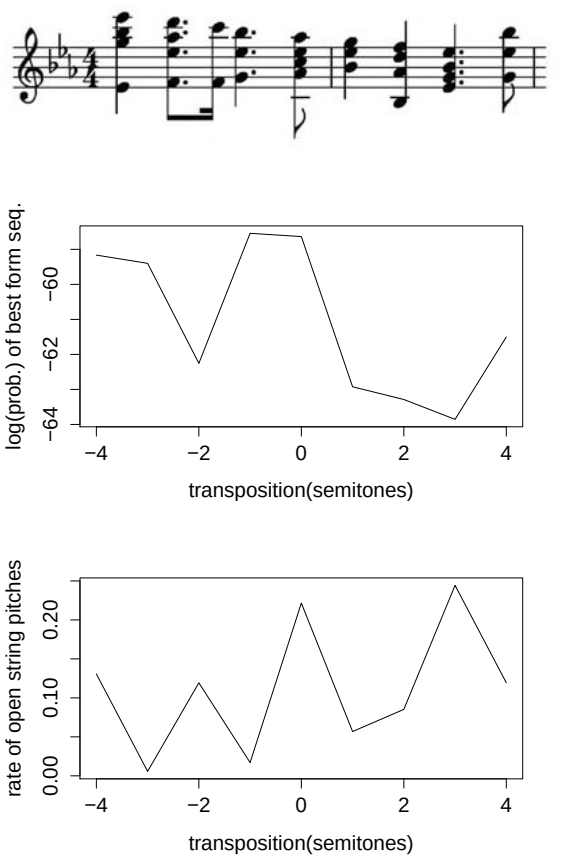

Figure 3. The opening section of "Joy to the world" (top), the logarithm of the probability of the best form sequence for the above score for each transposition (middle) and the rate of notes with the open string pitches for each transposition (bottom): The probability of the best form sequence is influenced by more factors and shows more complex response to transpositions than the rate of notes with the open string pitches.

probabilities of the most probable sequences for each keys. Furthermore, we can run the Viterbi algorithm for all the possible keys and compare the probabilities to find the optimal key among the possible keys. Comparing to typical discrete optimization problems, the search space (the set of possible keys) is very small and the full search is a realistic solution. Finding the optimal key for arrangement with transposition by a full search is an effective use of computation for music.

Fig.3 exemplifies the probability of the most probable sequence of forms as a function of the number of semitones for transposition for the opening section of "Joy to the world" given in its top. The logarithm of the probability of form sequence is given in the middle for transpositions from 4 semitones down to 4 semitones up and the rate of notes with the open string pitches of the guitar is given in the bottom for the sake of comparison. We take the rate of open string pitches for comparison because the number of open strings in left hand forms is related to the easiness of fingerings as well as the probability of form sequences and, in addition, the rate can be easily calculated from only

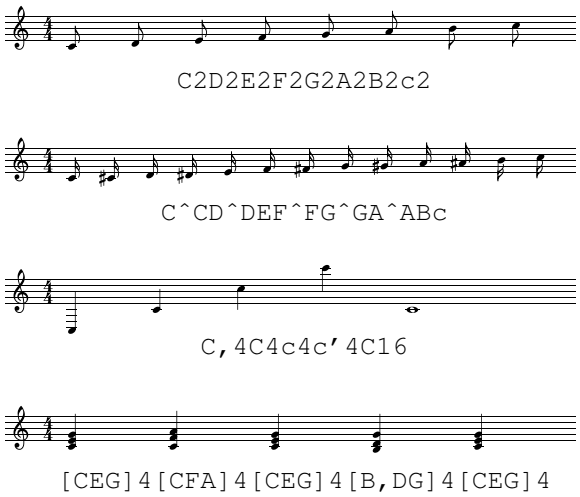

Figure 4. Examples of the simplified ABC notation used for loading song data in the web application. The unit note length is fixed to sixteenth note. The last example shows how to notate a "sequence of chords" which is an assumption on a piece to arrange in our formulation.

the distribution (or histogram) of the notes of a given piece of music. The rate of open string pitches exhibits alternate rises and falls due to the relation between the scale structure and the open string pitches. On the other hand, the probability of the best sequence of forms exhibits more complex response to transpositions mainly because it depends not only on the histogram of notes but also on the aspect of time series of the notes in a given piece of music. The probability heavily depends on how notes are ordered while the rate does not. From this observation, it seems that there is no easy way to find the optimal number of semitones for transposition but to run the Viterbi algorithm for all the possible keys.

We implement "arrangement with transposition" based on the full search in the web application introduced in the following section.

5. WEB APPLICATION

We have implemented automatic arrangement for guitars with transposition described in the previous sections in a web application and made it open to public at

`http://genhori.jp/guitar/`.

This section describes how to load song data and arrange it for guitars using the web application.

5.1 Song data format

The web application reads in song data using a simplified version of the standard ABC music notation [2]. The ABC music notation is a format designed to notate music using plain text, thus song data can be typed with any text editor. Fig.4 gives a few examples of the simplified ABC

[2] `http://abcnotation.com`

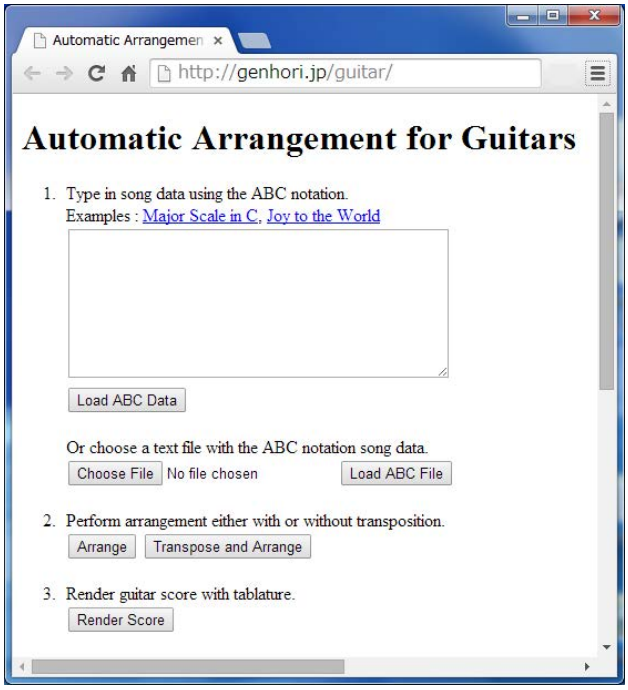

Figure 5. The user interface of the web application. The users are able to load song data using the simplified ABC notation, perform arrangement either with or without transposition and render guitar scores with tablature for printing using standard web browsers.

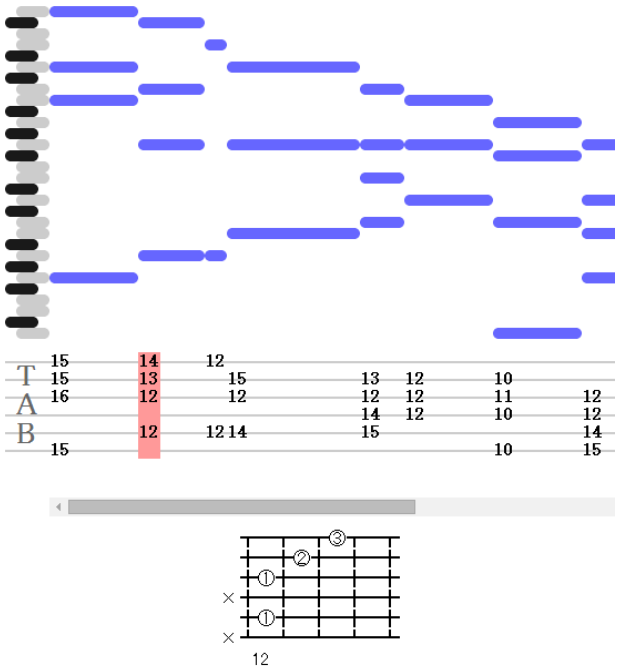

Figure 6. The pianoroll and the tablature displayed on the graphical user interface of the web application. Locating the cursor over any chord on the tablature displays a diagram of the chord to indicate the placements of each fingers on the fingerboard.

notation used for loading song data in the web application. First, a one-octave major scale from C3 to C4 with eighth notes is typed as `C2D2E2F2G2A2B2c2`, where the letters and the numbers show the pitches and the lengths of the notes respectively. The capital letters show the notes on the third octave (C3-B3) while the small letters the notes on the forth octave (C4-B4). For simplicity, we fix the unit note length to sixteenth note in the simplified ABC notation so that the number `2` means an eighth note. Secondly, a chromatic scale from C3 to C4 with sixteenth notes is typed as `C^CD^DEF^FG^GA^ABc`, where `^` moves the following note up a semitone and the missing numbers mean the unit note lengths, that is, sixteenth notes. The third example shows that `'` and `,` moves the preceding note up and down an octave respectively. The last example shows that notes in square brackets form a chord whose length is given by the number following the brackets. The ABC notation is suitable for notating a "sequence of chords" which is an assumption on a given piece to arrange in our formulation. A drag-and-drop user interface for loading song data using SMF (standard MIDI file) is presently being implemented.

5.2 User interface

Fig.5 presents the user interface of the web application. The users type in song data into the textarea and push the "Load ABC Data" button or choose a text file with song data and push the "Load ABC File" button then the loaded song data is displayed in the pianoroll like shown in the upper half of Fig.6. Next, the users push the "Arrange" button or the "Transpose and Arrange" button to perform arrangement without or with transposition then the resulting arrangement is displayed in the tablature below the pianoroll like shown in Fig.6. In the tablature, locating the cursor over any chord displays a highlight box on the chord as well as a diagram of the chord below the tablature to indicate the placements of each fingers on the fingerboard. Finally, the users push the "Render Score" button to render a guitar score with tablature for printing.

6. CONCLUDING REMARKS

The HMM-based automatic arrangement has been extended to "arrangement with transposition." The optimal transposition for arrangement of a given piece has been obtained through a full search for all the possible keys. Each arrangements have been evaluated based on the probabilities of the sequences of forms. The probability of the best sequence of forms has shown complex response to transpositions mainly because it depends on the histogram of notes as well as the aspect of time series of the notes. From this observation, we have seen that running the Viterbi algorithm for all the possible keys is a realistic solution.

Next, we have developed a web application that implements automatic arrangement for guitars with transposition and made it open to public. The web application reads in song data in ABC notation and carry out arrangement with or without transposition. In near future, we will use the web application for subjective assessment with guitarists of the robustness and the limitations of the proposed method.

7. REFERENCES

[1] G. Hori, H. Kameoka, S. Sagayama, "Input-Output HMM Applied to Automatic Arrangement for Guitars," *Journal of Information Processing*, Vol. 21, No. 2, pp. 264–271, 2013.

[2] G. Hori, Y. Yoshinaga, S. Fukayama, H. Kameoka, S. Sagayama, "Automatic arrangement for guitars using hidden Markov model," *Proc. 9th Sound and Music Computing Conference (SMC2012)*, pp. 450–456, 2012.

[3] S. I. Sayegh, "Fingering for string instruments with the optimum path paradigm," *Computer Music Journal*, Vol. 13, No. 3, pp. 76–83, 1989.

[4] M. Miura, I. Hirota, N. Hama and M. Yanagida, "Constructing a system for finger-position determination and tablature generation for playing melodies on guitars," *Systems and Computers in Japan*, Vol. 35, No. 6, pp. 10–19, 2004.

[5] D. Radicioni, L. Anselma and V. Lombardo, "A segmentation-based prototype to compute string instruments fingering," *Proc. Conference on Interdisciplinary Musicology*, Graz, Austria, 2004.

[6] A. Radisavljevic and P. Driessen, "Path difference learning for guitar fingering problem," *Proc. International Computer Music Conference*, Miami, U.S.A., 2004.

[7] D. R. Tuohy and W. D. Potter, "A genetic algorithm for the automatic generation of playable guitar tablature," *Proc. International Computer Music Conference*, pp. 499–502, 2005.

[8] D. R. Tuohy, "Creating tablature and arranging music for guitar with genetic algorithms and artificial neural networks," M.Sc. Theses, The University of Georgia, Georgia, U.S.A., 2006.

[9] D. Baccherini, D. Merlini and R. Sprugnoli, "Tablatures for stringed instruments and generating functions," In P. Crescenzi, G. Prencipe and G. Pucci Eds., *Fun with Algorithms*, Lecture Notes in Computer Science, Vol. 4475, Springer-Verlag, pp. 40–52, 2007.

[10] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Information Theory*, Vol. 13, No. 2, pp. 260–269, 1967.

[11] Y. Bengio and P. Frasconi, "An input output HMM architecture," In G. Tesauro, D. S. Touretzky, and T. K. Leen Eds., *Advances in Neural Information Processing Systems*, Vol. 7, pp. 427–434, 1995.