TABLATURE GENERATION FROM LEAD SHEETS FOR FINGER-STYLE SOLO GUITAR

Shunsuke SAKAI[1], **Hinata SEGAWA**[1], and **Tetsuro KITAHARA** (kitahara@kthrlab.jp)[1]

[1]**Nihon University**, Tokyo, Japan

ABSTRACT

In the finger-style guitar, players often play both melody notes and chords simultaneously with a single guitar. This playing style is difficult because the player has to find a playable set of fretboard positions that include both melody notes and chords. In this paper, we propose a method for generating a tablature that enables users to play both melody notes and chords simultaneously from a given lead sheet, which consists of melody and chord symbol notations. Our method achieves the tablature generation as an optimaization problem of searching for the minimal-cost sequence of states. A combination of the fingering positions for the five fingers on the fretboard is regarded as a state. And three types of performing costs (an initial cost, a transition cost, and an emission cost) are formulated for each state. Then, a sequence of states having the minimal cost is searched with a Viterbi algorithm. Experimental results showed that tablatures generated with our method were moderately evaluated by an expert of the classical guitar.

1. INTRODUCTION

The finger-style guitar is one of the common playing styles for acoustic guitars. Unlike plectrum playing, the finger-style guitar enables guitarists to play complicated performances because they can independently pluck different strings with different fingers. Indeed, it is common to play both melody notes and chords simultaneously with a single guitar without any accompaning players. This playing style is called *solo guitar* in this paper.

The solo guitar is considered a difficult playing style. One of the major difficulties resides in the fact that it is complicated to find a playable set of fingering positions on the fretbaord, which play both melody notes and chords. Because melody notes, chord notes, and bass notes are played with a single guitar, the fingering positions of all fingers for the melody, chord, and bass notes have to fall within a physically playable range. Searching for such fingering positions is not easy for many non-professional players.

The purpose of this study is to achieve automatic generation of a tablature for the finger-style solo guitar for a given lead sheet. Because a lead sheet consists of the notations of melody notes and chord symbols, there are multiple possibilities of the voicing for each chord. To generate a playable tablature, the system has to carefully choose an appropriate voicing that can be played together with the melody notes.

Automatic tablature generation has been attempted by a number of researchers, but no one has attempted tablature generation from a lead sheet under the assumption that both the melody and the chord progression are played with a single guitar. In 2004, Miura et al. developed a system that selects the optimal fingering positions for monophonic melodies by minimizing finger motions on the fretboard [1]. In 2005, Tuohy et al. applied a genetic algorithm to find playable tablatures by using a fitness function that assesses the playability of a given set of fretboard positions. They used simple polyphonic music in their experiment but the inputs were not lead sheets [2]. In 2016, Hori et al. proposed a minimax Viterbi algorithm to minimize the maximal difficulty in fingering motion on the fretboard. They tested their method with simple monophonic melodies [3]. In addition, there have been some attempts to introduce a tablature representation in automatic music transcription to consider the playability in the guitar (e.g., [4]). Their inputs were audio signals, not lead sheets.

In our previous paper [5], we developed a tablature generation system for polyphony that consists of only melodies and bass lines. Because the number of simultaneously played notes are one or two in such polyphony, we defined a set of states that represents fingering positions for one or two simultaneous notes. The system then generates a tablature by searching for the state sequence that has the minimal performing cost.

To extend this method to a combination of melody notes and chords, we add states that represent more than two notes. However, if this addition is simply done, we will have two problems: (1) states that represent unsuitable (very difficult-to-play) note combinations may be added, and (2) the computational cost may become higher due to the increase of the number of states. To avoid these problems, we introduce *typical forms*, a set of very common fingering forms for commonly used chords. By introducing them, we limit the set of states to common fingering forms, so we can avoid that unsuitable fingering forms are contained in generated tablatures.

2. TABLATURE GENERATION ALGORITHM

Our system generates a tablature from a lead sheet consisting of a melody and a chord progression. Because our target is finger-style solo guitar, in which the guitarist plays both melody notes and chords simultaneously, fingering (string-pressing) positions on the fretboard for melody notes and chords must fall within a physically playable range. We achieve this by optimizing state transitions in which physically unplayable states had been removed.

2.1 Data representation

We denote a gjven melody and chord progression as $X = \{(x_1, c_1), \cdots, (x_N, c_N)\}$, where x_n is the MIDI note number (or rest) of the n-th note and c_n is the chord symbol assigned to the n-th note. In the current implementation, each chord is played only at its beginning time; for other timings (no chord symbol or a chord continues from the previous note), a special symbol ϵ is set to c_n.

Next, we define a 6-dimensional vector, $\boldsymbol{v} = (v^{(1)}, v^{(2)}, v^{(3)}, v^{(4)}, v^{(5)}, v^{(6)})$, which represents a fingering form on the fretboard. $v^{(m)}$ represents the fret position of the finger pressing the m-th string ($v^{(m)} = 1, \cdots, 14$). $v^{(m)} = 0$ represents an open string (the string is not pressed although plucked) and $v^{(m)} = -1$ represents no plucking.

Let V be a playable set of fingering form vectors $\boldsymbol{v}$. When we denote the optimal fingering form for (x_n, c_n) as $\boldsymbol{q}_n$ $(\in V)$, our optimization problem can be formulated as:

$$\text{minimize } C(Q) \text{ s.t. } Q = \{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_N\}$$

where $C(Q)$ is a performing cost function (representing the degree of the non-optimality).

The cost function consists of the following three types of costs:

- **Initial cost** $C(\boldsymbol{q}_1)$
  It is said that neck-side positions are more common than body-side positions in classical guitar, so lower costs are set to neck-side positions.
- **Transition cost** $C(\boldsymbol{q}_{n+1}|\boldsymbol{q}_n)$
  As the motion of the fingering positions from $\boldsymbol{q}_n$ to $\boldsymbol{q}_{n+1}$ is smaller, playing it becomes easier. Therefore, we give lower costs to smaller motions of the fingering positions.
- **Emission cost** $C((x_n, c_n)|\boldsymbol{q}_n)$
  The fingering form $\boldsymbol{q}_n$ must emit the melody note x_n and chord tones of the chord c_n. In addition, it is more desirable to emit more chord tones to make a rich harmony. The emission cost is defined from these points of view.

The overall cost $C(Q)$ is defined by the following equation:

$$\begin{aligned} C(Q) &= C(\boldsymbol{q}_1) \\ &\quad + \sum_{n=1}^{N-1} \{C((x_n, c_n)|\boldsymbol{q}_n) + C(\boldsymbol{q}_{n+1}|\boldsymbol{q}_n)\} \\ &\quad + C((x_N, c_N)|\boldsymbol{q}_N) \end{aligned}$$

This formulation is inspired by a hidden Markov model.

2.2 Defining a playable set of fingering form vectors

As mentioned above, V is a playable set of fingering form vectors. Each element $\boldsymbol{v} \in V$ is a 6-dimensional vector $\boldsymbol{v} = (v^{(1)}, v^{(2)}, v^{(3)}, v^{(4)}, v^{(5)}, v^{(6)})$, where $v^{(m)}$ represents the fret position of the finger pressing the m-th string.

To define V, we adopt the following policies for chord voicing:

- The highest note shall always be a melody note.
- The lowest note (bass note) shall always be the root note.
- The fretboard positions for different strings should be close enough to each other (within three frets).
- Fingering forms familiar for guitarists are preferred.

We add fingering position vectors to V based on the following methods.

2.2.1 The case of two or less simultaneous note(s)

When the number of simultaneous notes of a fingering position vector is two or less, this fingering is used to play only a melody note or a combination of a melody note and a bass note. We therefore add fingering form vectors $\boldsymbol{v}$ satisfying the following requirements to V:

1. $\max_+(\boldsymbol{v}) - \min_+(\boldsymbol{v}) \leq 3$
2. $n(\{m \mid v^{(m)} > 0\}) < 2$.
3. Multiple strings do not correspond to the same note

where $\max_+(\boldsymbol{v})$ and $\min_+(\boldsymbol{v})$ denote the maximum and minimum values of the positive-value elements of $\boldsymbol{v}$, respectively. $n(\cdot)$ is the number of the elements in a given set.

2.2.2 The case of three or more simultaneous notes

When the number of simultaneous notes is three or more, the fingering is used to play both a melody note and chords simultaneously (or chord only if the melody note is a rest). The important here is to give a higher priority to more familiar fingering forms for players. To achieve this, we define *typical forms*, which represent fingering forms for basic chords for familiar for guitarists. For example, the typical forms for the chords C and F are $(0, 1, 0, 2, 3, -1)$ and $(1, 1, 2, 3, 3, 1)$, respectively. We manually made a dataset that include 3658 typical forms.

In addition to all of the 3658 typical forms, *modified forms* are added to V. The modified forms are fingering forms in which a few elements are changed from a typical form. They were made by changing each typical form $\boldsymbol{v}$ as follows:

- Reduce string-pressing:
  Replace one or two element(s) of $\boldsymbol{v}$ with 0 (an open string) or -1 (no plucking).
- Add string-pressing:
  Replace one of the elements having 0 or -1 with the maximal value of $\boldsymbol{v}$, if the little finger is not

used. For example, the chord C, the fingering form of which is $(0, 1, 0, 2, 3, -1)$, is played without the little finger. Then, modified forms $(3, 1, 0, 2, 3, -1)$ and $(0, 1, 3, 2, 3, -1)$ are generated.

2.3 Narrowing down the set of fingering form vectors

The set of fingering form vectors, V, have a large number of vectors, so it requires a high computational cost to directly search the optimal fingering form from V. Therefore, we narrow down V. For n-th note (x_n, c_n), the elements that do not emit the melody note x_n and the chord c_n are removed from V. The narrowed-down set for (x_n, c_n) is denoted as V_n. Specifically, V_n has only vectors $\boldsymbol{v} = (v^{(1)}, v^{(2)}, v^{(3)}, v^{(4)}, v^{(5)}, v^{(6)})$ that satisfy the following:

- The highest note matches the melody note. $\exists m_0 : \mathrm{note}(v^{(m_0)}) = x_n$ and $\forall m < m_0 : v^{(m)} = -1$.
- The lowest note matches the root note of the chord. $\exists m_1 : \mathrm{note}(v^{(m_1)}) = \mathrm{root}(c_n)$ and $\forall m > m_1 : v^{(m)} = -1$.
- The notes between the highest and lowest notes are chord notes. $\forall m \in (m_0, m_1) : \mathrm{note}(v^{(m)}) \in \mathrm{notes}(c_n)$.

$\mathrm{note}(v^{(m)})$ denotes the note corresponding to the fret position $v^{(m)}$ for string m, $\mathrm{root}(c_n)$ denotes the root note of the chord c_n, and $\mathrm{notes}(c_n)$ denotes the set of the chord notes of c_n.

When $c_n = \epsilon$, only the melody note is played. Therefore, V_n only includes vectors that have a value satisfying $\mathrm{notes}(v^{(m)}) = x_n$ for one element m and -1 for the others.

2.4 Defining initial costs

On acoustic guitars, positions closer to the neck is more common than positions closer to the body, so positions closer to the body should have a slightly higher cost.

$$C(\boldsymbol{q}_1) = \begin{cases} 2.5 & (\max_+(\boldsymbol{q}_1) \leq 4) \\ 5.0 & (\text{otherwise}) \end{cases}$$

2.5 Defining transition costs

We define the transition cost to reduce the difficulty of the performance by increasing the cost of state transitions that involve large motions. In addition, for the same reason as above, a higher priority is given to positions closer to the neck.

$$C(\boldsymbol{q}_{n+1}|\boldsymbol{q}_n) = \begin{cases} 0.0 & (\max(\boldsymbol{q}_{n+1}) = 0 \text{ or } \max(\boldsymbol{q}_n) = 0) \\ |\min_+(\boldsymbol{q}_n) - \min_+(\boldsymbol{q}_{n+1})| + \alpha & (\text{otherwise}) \end{cases}$$

The α is introduced to give a higher preference to fingering positions closer to the neck (5.0 when $\max(\boldsymbol{q}_{n+1}) \geq 5$; 0.0 otherwise).

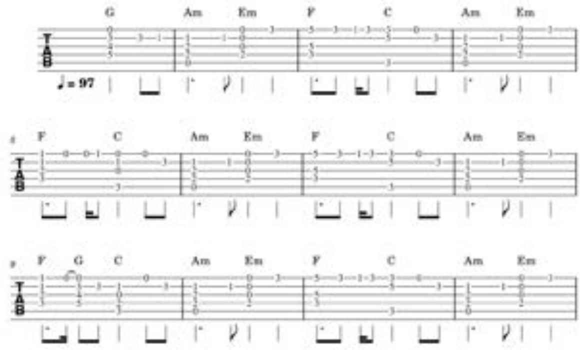

Figure 1. An example of the tabulatures generated by our system

2.6 Definng emission costs

Because the vectors that do not emit (x_n, c_n) have been removed from V_n, we do not have to consider if the fingering vector emits the melody note x_n and the chord c_n when we define the emission cost. We therefore define the emission cost based on the following policy:

- When the fingering vector is not modified from a typical form, a lower emission cost is given, because typical forms are expected to be easier to play than modified forms.
- For fingering vectors that emits more tones, a lower emission cost is given, because such vectors are expected to make a richer harmony.

Specifically, we define the emission cost as follows:

(i) When $c_n = \epsilon$ We simply set 0 to $C((x_n, c_n)|\boldsymbol{q}_n)$.

(ii) When $c_n \neq \epsilon$ We define the emission cost as:

$$C((x_n, c_n)|\boldsymbol{q}_n) = \mathrm{voices}(\boldsymbol{q}_n) + \mathrm{typical}(\boldsymbol{q}_n),$$

where $\mathrm{voices}(\boldsymbol{q}_n)$ is a cost calculated from the number of simultaneous notes and $\mathrm{typical}(\boldsymbol{q}_n)$ is a cost determined from whether the fingering form is a typical form. In defining $\mathrm{voices}(\boldsymbol{q}_n)$, we consider the number of simultanously plucked strings $\mathrm{pluck}(\boldsymbol{q}_\mathrm{n})$:

$$\mathrm{voices}(\boldsymbol{q}_n) = \begin{cases} \infty & (\mathrm{pluck}(\boldsymbol{q}_n) = 6) \\ \infty & (\mathrm{pluck}(\boldsymbol{q}_n) = 5 \\ & \text{and } \mathrm{dur}(x_n) < 2) \\ -\beta\, \mathrm{pluck}(\boldsymbol{q}_n) & (\text{otherwise}) \end{cases}$$

where $\mathrm{dur}(x_n)$ is the duration of the melody note x_n ($\mathrm{dur}(x_n) < 2$ means that x_n is shorter than a half note) and β is a positive constant. $\mathrm{typical}(\boldsymbol{q}_n)$ is defined as:

$$\mathrm{typical}(\boldsymbol{q}_n) = \begin{cases} 10 & (\boldsymbol{q}_n \text{ is a typical form}) \\ 12 & (\text{otherwise}) \end{cases}$$

3. EXPERIMENT

3.1 Method

The tablatures of 8 pieces of music output by the proposed method (an example is shown in Fig. 1) were evaluated by a classical guitar expert. For each piece, she answered the questions in table 1 on a 7-point scale.

Table 1. Questions asked to the evaluator

Q1	Was there anything in the voicing of each chord that made it difficult to play?
Q2	Did you find the transition from chord to chord difficult to play?
Q3	Did you find any part of the melody difficult to play?
Q4	Do you think the voicing is musically appropriate?
Q5	Was the difficulty level appropriate for intermediate level players?
Q6	Do you think the arrangement is useful as a starting point for intermediate players to make their own arrangements?

Table 2. Evaluation of each piece

	Q1	Q2	Q3	Q4	Q5	Q6
Let it Snow	6	4	4	3	5	4
The Wellerman	5	5	5	6	5	5
Canary	7	6	7	5	7	5
Cherry	7	7	6	6	7	7
Sora mo Toberu hazu	6	4	6	5	6	6
Tsuki wo Miteita	6	5	6	5	7	6
Uchiage Hanabi	5	6	7	5	7	6
Yoake to Hotaru	4	3	4	5	5	5

3.2 Results

The results are listed in Table 2. From the table, we can see that the overall evaluation was good. However, there were some large differences in evaluation depending on the music piece, such as "Cherry" and "Yoake to Hotaru".

For "Let it Snow", the ratings for Q2, Q3, Q4, and Q6 were low. Regarding Q4, we received the opnion that the voicings (especially the number of simultaneous notes) should be changed based on the metrical position; the harmony should be richer for the begining of each measure and less richer for the rest). For "Sora mo Toberu hazu", the rating for Q2 was low. This is because the number of simultaneous notes were too large, caused by the definition of the emission cost.

4. CONCLUSION

In this paper, we proposed a tablature generation system for solo guitar, in which the melody and chords are played simultaneously. To obtain the optimal fingering form for a given melody and chord notes, the Viterbi algorithm is applied with three types of costs: initial costs, transition costs, and emission costs. Through the experiment, we confirmed the the generated tablatures are moderaltely good, but they have parts that are difficult to play, especially for chord voicings. Finding a tradeoff between ease-to-play and harmonic richness is one of the important remaining issues.

Acknowledgments

This work was supported by KAKENHI Grants-in-Aid for Scientific Research 22H03711 and 21H03572. We thank Dr. Nami Iino for her help in the experiment.

5. REFERENCES

[1] 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.

[2] D.R.Tuohy and W.D.Potter: A Genetic Algolithm for the Automatic Generation of Playable Guitar Tablature", Proceedings of ICMC 2005, pp. 499–502, 2005.

[3] G. Hori and S. Sagayama: Minimax Viterbi algorithm for HMM-based guitar fingering decision, Proceedings of ISMIR 2016, pp. 448–453, 2016.

[4] K. Yazawa, D. Sakaue, K. Nagira, K. Itoyama, and H. G. Okuno: Audio-based guitar tablature transcription using multipitch analysis and playability constraints, Proceedings of ICASSP 2013, pp. 196–200, 2013.

[5] S. Sakai, H. Segawa and T. Kitahara: Generating Tablature of Polyphony Consisting of Melody and Bass Line, Proceedings of CMMR 2023, pp. 724–727, 2023.