Melody Reduction for Beginners' Guitar Practice

Hinata Segawa, Shunsuke Sakai, and Tetsuro Kitahara*

College of Humanities and Sciences, Nihon Univesity
{segawa,sakai,kitahara}@kthrlab.jp

Abstract. This paper describes a system that reduces a melody for novice guitar players to practice guitar solo phrases without lowering their motivation. Although there have been systems already that generate the guitar tablature for given melodies, they did not deal with melody reduction for novice guitar players. In this paper, we propose a system that generates melodies in which the difficulty of the play is reduced by using a Viterbi-like dynamic programming search. A preliminary results shows that our system can reduce melodies based on the difficulty of the play.

1 Introduction

Our goal is to develop a system for beginners to practice their favorite solo phrases on the electric guitar. However, it is not easy because their favorite solo phrases may be too difficult for them to play. If a system can generate melodies that are reduced but similar enough to their favorite melodies, it will be effective in maintaining their motivation.

There have been studies on generating tablatures from given melodies. For example, Hori et al. used a hidden Markov model to generate a tablature by minimizing the moving distance of fingering positions[1]. Tuohy et al. used a genetic algorithm to generate a tablature [2]. However, they did not deal with reducing melodies of the guitar solos. On the other hand, there have been many attempts of melody reduction, e.g., Ryan Groves's method using a probabilistic context-free grammar[3]. However, they do not consider the difficulty of playing melodies on the guitar.

Our system reduces melodies by introducing a playing cost and a melody modification penalty, and by finding a tradeoff between them. In particular, the playing cost becomes high when the fingering moves at upbeat, so the system generates melodies that do not move in pitch at upbeat.

2 Proposed System

The system inputs a monophonic tablature and generates a reduced melody tablature using a Viterbi-like search algorithm. It calculates the playing cost and melody mod-

* This work was supported by JSPS Kakenhi Nos. JP22H03711 and JP21H03572.

ification penalty, and minimizes the sum of them to obtain an optimal sequence of fingering from the viewpoint of the ease of the play.

2.1 Loading MusicXML data

Given a tablature of a monophonic guitar solo melody in the MusicXML format, it is converted to a sequence of fingering positions, $\{x_1, x_2, \cdots, x_N\}$. Here, the fingering positions for the n-th note, x_n, is defined as $x_n = s_n + 6f_n$ where the string number $s_n (= 0, 1, 2, \cdots, 5)$ and the fret number $f_n (= 0, 1, 2, \cdots, 20)$ are combined. We alsp identify whether that each note is on a downbeat or upbeat.

2.2 Modeling Melody Reduction

The reduced melody is assumed to have the same number of notes and rhythm as the original melody, and a sequence of its fingering positions is represented by $\{x'_1, x'_2, \cdots, x'_N\}$. The fingering position for the n-th note, x_n is defined as $x'_n = s'_n + 6f'_n$ using the string number s'_n and fret number f'_n.

Our system outputs a melody that is close to the original melody, but with reduced movements shift of the fingering position. For example, we set a high cost for movements of the fingering position on upbeats. This allows us to obtain the melodies such that the pitch only changes on downbeats.

2.2.1 Playing cost The playing cost represents the difficulty in moving the fingering position from note to note. In order to reduce pitch motions on the upbeat, we set different costs for the downbeat and upbeat.Because the use of open strings is not accepted in the current implementation, we assign a sufficiently higher cost to the movement to an open string.

(i) When x'_n is on a downbeat
We define the playing cost from x'_n to x'_{n+1} as follows.

$$C(x'_{n+1}|x'_n) = \begin{cases} \alpha & (|f'_n - f'_{n+1}| > 3) \\ \alpha & (f'_{n+1} = 0) \\ 10000 & (|s'_n - s'_{n+1}| = 3), \\ 7500 & (|s'_n - s'_{n+1}| = 2, |f'_n - f'_{n+1}| = 2, 3) \\ 5000 & (|s'_n - s'_{n+1}| = 2, |f'_n - f'_{n+1}| = 1) \\ 1000 & (|s'_n - s'_{n+1}| = 2, f'_n = f'_{n+1}) \\ 400 & (|s'_n - s'_{n+1}| = 1, |f'_n - f'_{n+1}| = 2, 3) \\ 200 & (|s'_n - s'_{n+1}| = 1, |f'_n - f'_{n+1}| = 1, 0) \\ 50 & (s'_n = s'_{n+1}, |f'_n - f'_{n+1}| = 2, 3) \\ 0 & (s'_n = s'_{n+1}, |f'_n - f'_{n+1}| = 1, 0) \end{cases}$$

Let α be a sufficiently large positive value (1000000000 in the current implementation).

(ii) When x'_n is on an upbeat
We define the cost of playing from x'_n to x'_{n+1} as follows.

$$C(x'_{n+1}||x'_n) = \begin{cases} \alpha & (x'_n \neq x_n) \\ 0 & (x'_n = x_n) \end{cases}$$

As a criterion for identifying whether each note is on the downbeat or upbeat, the user can select quarter-note-level one and eighth-note-level one.

2.2.2 Melody Modification Penalty The melody modification penalty represents how much the output melody differs from the original melody. The melody modification penalty $P(x'_n|x_n)$ for x'_n is defined as follows.

$$P(x'_n|x_n) = \begin{cases} 0 \ (x'_n = x_n) \\ \alpha \ (x'_n \neq x_n) \end{cases}$$

2.3 Viterbi-like dynamic programming search

The system searches $\{x'_1, x'_2, \cdots, x'_N\}$ that minimizes the following values

$$S = \left\{ \sum_{n=1}^{N-1} (P(x'_n|x_n) + C(x'_{n+1}|x'_n)) \right\} + P(x'_N|x_N)$$

This minimization can be performed using a Viterbi-like dynamic programming algorithm.

2.4 Tablature Output

The system finally outputs $x'_1, x'_2, \cdots, x'_N$ as a tablature in the MusicXML format.

3 Preliminary Results

We attempted to generate a reduced melody's tablature. Figure 1 shows the tablature used as an input (representing the original melody). This melody has a series of sixteenth notes, so requires a fast fingering movement to play. On the other hand, Figure 2 shows the generated tablature (representing the reduced melody). The fingering movements on the upbeats have been removed, and the fingering movements on the downbeats have been made smaller.

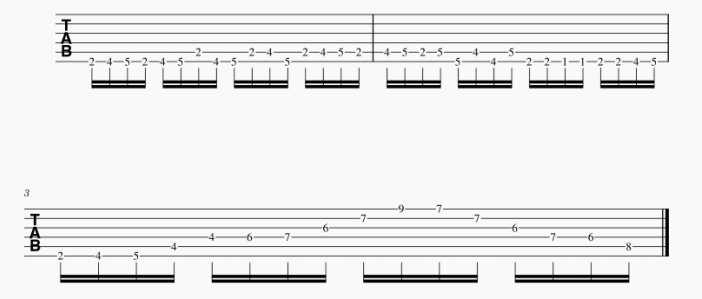

Fig. 1. A tablature used in the preliminary experiment

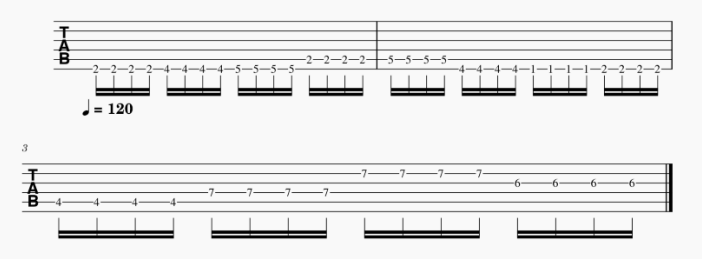

Fig. 2. A tablature of the reduced melody

4 Conclusion

In this paper, we developed a prototype system for generating a tablature that can be played more easily than the original melody. This system is intended to allow novice guitar players to practice solo melodies without lowering their motivations. After learning to play reduced melodies, they can try to practice the original melody.

However, there are still various issues to be addressed. First, it is necessary to evaluate the similarity of the reduced melody and the original melody to assess our melody reduction method. Because there have been many methods for melody reduction, such as GTTM-based ones, we need to compare our method with those ones.

References

1. Gen Hori, Shigeki Sagayama: Minimax Viterbi algorithm for HMM-based guitar fingering decision, 17th International Society for Music Information Retrieval Conference,(2016).
2. D.R Tuohy and W.D. Potter: A genetic algorithm for the automatic generation of playable guitar tablature(2005).
3. Ryan Groves: Automatic Melodic Reduction Using a Supervised Probabilistic Context-free Grammar, Proceedings of the 17th ISMIR Conference (2016).