SONG2GUITAR: A DIFFICULTY-AWARE ARRANGEMENT SYSTEM FOR GENERATING GUITAR SOLO COVERS FROM POLYPHONIC AUDIO OF POPULAR MUSIC

Shunya Ariga
The University of Tokyo
ariga@iis-lab.org

Satoru Fukayama
AIST
s.fukayama@aist.go.jp

Masataka Goto
AIST
m.goto@aist.go.jp

ABSTRACT

This paper describes Song2Guitar which automatically generates difficulty-aware guitar solo cover of popular music from its acoustic signals. Previous research has utilized hidden Markov models (HMMs) to generate playable guitar piece from music scores. Our Song2Guitar extends the framework by leveraging MIR technologies so that it can handle beats, chords and melodies extracted from polyphonic audio. Furthermore, since it is important to generate a guitar piece to meet the skill of a player, Song2Guitar generates guitar solo covers in consideration of playing difficulty. We conducted a data-driven investigation to find what factor makes a guitar piece difficult to play, and restricted Song2Guitar to use certain hand forms adaptively so that the player can play the piece without experiencing too much difficulty. The user interface of Song2Guitar is also implemented and is used to conduct user tests. The results indicated that Song2Guitar succeeded in generating guitar solo covers from polyphonic audio with various playing difficulties.

1. INTRODUCTION

A guitar solo cover version of an original song adds new pleasure to the music experience of the song. Various musical elements such as beats, melodies, and harmonies in an original song are represented in a uniform but expressive timbre of a guitar. However, a guitar solo cover of one's favorite song is not always available, and creating guitar arrangements requires advanced skills and knowledge and takes a lot of time. If such a guitar solo cover of any song can be generated from music audio signals, music listeners can enjoy their favorite songs in a different way, and guitarists who do not have skills for playing by ear can also enjoy performing any songs on their guitars.

The goal of this research is to develop a system that can automatically generate a guitar solo cover version from audio signals. By leveraging Music Information Retrieval (MIR) technologies, we propose a guitar arrangement system, *Song2Guitar*, that generates guitar solo covers from polyphonic audio signals of popular music, which contain sounds of various instruments. We also aim at creating difficult-aware guitar arrangements — i.e., generating guitar tablatures having different levels of playing difficulty for guitarists. There are three issues that should be considered:

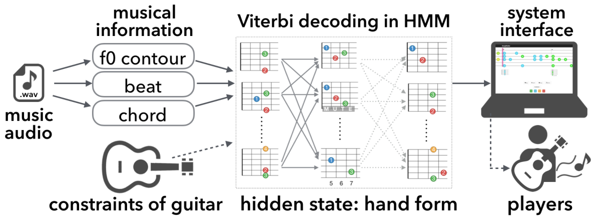

Figure 1: Overview of the Song2Guitar system.

(1) Generate from polyphonic audio of popular music

(2) Difficulty-aware arrangement

(3) Interface to perform the arrangement result

An overview of our solutions to address these issues is shown in Fig. 1. As for issue (1), even if we use the state-of-the-art MIR technologies, we cannot obtain completely-transcribed musical scores from such complex audio signals. We therefore directly extract important musical elements, such as melody lines represented as F0 (fundamental frequency) contour, beats, and chords, from polyphonic audio. We then reflect the extracted elements in generating guitar solo covers by using a novel extension of a hidden Markov model. As for issue (2), we conducted a data-driven survey to find what factors make a guitar tablature difficult to play. Based on the survey, Song2Guitar controls the movement of an index finger and the number of fingers to press the strings. Finally, as for issue (3), we designed and implemented an interface that enables a guitarist to change the degree of difficulty to perform the result. In this paper, we will also discuss a desirable interface for generating various arrangement results and providing training materials for guitarist. The design of the interface and the results generated by our system are available on the web [1].

[1] https://youtu.be/fN4-ibh7ZDI

2. RELATED WORK

2.1 Creative MIR

Our research is addressed in a Creative MIR approach. MIR researchers have recently explored creativity-oriented music technologies by applying technologies developed in the MIR community. This emerging field is named *Creative MIR* [11] where music analysis and transformation technologies are used in various creative applications. For example, AutoMashUpper [4] is an interactive system that creates music mashups by automatically selecting and mixing chosen songs. They achieved automatic mashup by estimating *mashability*, which is calculated by using MIR technologies to estimate various musical elements such as beats, downbeats, and chromagram. Song2Quartet [18] generates a cover song in the style of string quartet by combining probabilistic models estimated from a corpus of symbolic classical music with the target audio file of a song.

2.2 Generating playable guitar Solo

In order to generate a playable guitar covers, Hori *et al.* [7–9] used a hidden Markov model (HMM) to generate guitar arrangements from a symbolic musical score while considering natural fingerings. Audio signals, however, were not used as the input. By taking audio signals of an individual separated guitar part as the input, Yazawa *et al.* [25] developed an automatic transcription system specialized for a guitar performance and generated a guitar tablature by using multi-pitch analysis and playability constraints. Yazawa *et al.* [24] then extended their previous work to transcribe a guitar tablature while considering acoustical reproducibility and fingering easiness. Even though guitarist's proficiency was considered, creating guitar arrangements from polyphonic music including multiple instruments was not tackled so far.

Research of automatic fingering decision can also be regarded as related work of ours. This is because fingering decision is a sub-problem to generate playable guitar solo, and the existence of a fingering for a song is a necessary condition for the song to be playable. Radicioni discussed in his thesis how to computationally model the fingering in music performance [19]. The fingering is often determined by searching the fingering sequence as an optimal path search problem [20, 21].

Fingerings are represented in a tablature score or tabs, and they are often utilized to analyze and generate playable scores. A method to analyze and search valuable information in the tablature database has been proposed [14]. AutoGuitarTab [15–17] generates guitar music according to different styles of various guitarists by training individual probabilistic model using a tablature database. Genetic algorithms have been used to search the fingering sequence efficiently to generate an guitar solo arrangement [22].

Tablature transcription from music audio are also the related work. MIR technologies such as multi-pitch analysis and chord recognition have been used to capture notes and chords in the audio signal. Given those music elements, dynamic programming or Viterbi decoding with HMM has been leveraged to output reasonable fingerings [1, 10, 12].

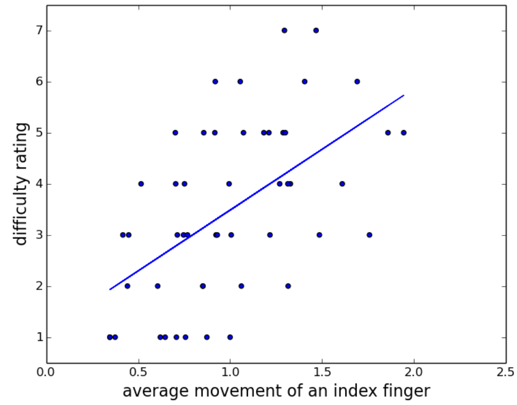

Figure 2: Average movement of an index finger and difficulty rating of 50 tablatures. The correlation coefficient was 0.55. The line indicates the linear regression result and the R-squared value was 0.30.

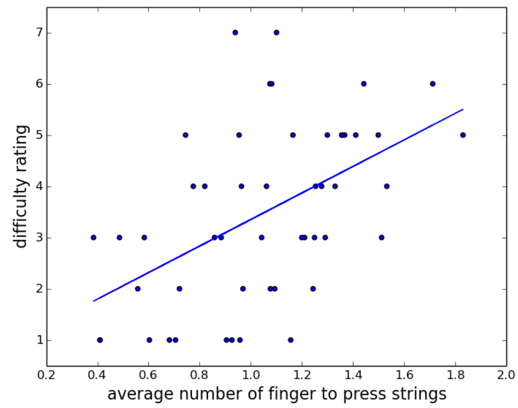

Figure 3: Average number of finger pressuring strings and difficulty rating of 50 tablatures. The correlation coefficient was 0.51. The line indicates the linear regression result and the R-squared value was 0.26.

3. PLAYING DIFFICULTY OF A GUITAR SOLO

3.1 Analysis of guitar tablatures

We first investigated what are the factors that affect the playing difficulty. We collected tablatures from a web site distributing classical guitar music [2]. These tabs were written in a plain text format and did not have uniformity in data structure. We therefore implemented a parser to retrieve structured tab information. Since Song2Guitar assumes only the standard tuning (E-A-D-G-B-E), we excluded tabs that were instructed to play in another tuning. Furthermore, we also excluded scores for guitar duo as we focus on guitar solo covers.

[2] http://www.classtab.org

For each tablature, we considered and calculated two factors: the average movements of the index finger of a hand to hold the guitar, and the average number of fingers to press the strings. The position of index finger is determined as follows: if the fingering contains a barre, we use the fret position of the barre, otherwise the position of the index finger is set to be the minimum fret number among the frets being pressed. We hypothesized that these two factors affect the playing difficulty of a guitar solo.

3.2 Subjective test to evaluate the playing difficulty

We verified our hypothesis by asking proficient guitarists to rate the difficulty of the tabs. Five independent raters subjectively evaluated the difficulty with a 7-point Likert scale (1: easiest – 7: most difficult). The raters were instructed to consider only the complexity of the fingerings of the left hand. As they respectively rated randomly-selected 10 tabs, we consequently obtained 50 ratings. Fig. 2 and Fig. 3 show the plots of our hypothesized features (average movement of an index finger, average number of fingers to press strings) and the result of difficulty ratings. The correlation coefficients calculated with these two features and the ratings were 0.55 and 0.51, respectively. We also conducted a linear regression on data. The regression results are also shown in Fig. 2 and Fig. 3. R-squared values for these two regressions were 0.30 and 0.26, respectively. These results indicated that the tabs were evaluated to be more difficult when the values of both features get larger.

4. CREATING DIFFICULTY-AWARE GUITAR ARRANGEMENTS FROM MUSIC SIGNALS

4.1 Our problem setting

In solving the problem of generating guitar solo covers from audio signals of popular music, we want to maintain and reproduce major characteristics of an original song in a generated guitar solo cover. The followings are the major characteristics that most songs in popular music have in common.

- It contains a clear melody line that is performed by a vocal part.
- It contains a bass line corresponding to a chord sequence.
- It gives a rhythmic groove emerged from sounds of rhythmic instruments.

Although previous work of generating guitar arrangements from symbolic musical scores [7] formulates the problem by using HMM, it have not tested with audio input. To generate from polyphonic music audios of popular music, we formulate the problem by a novel extension of HMM. We also propose how we can generate various results with different levels of playing difficulty for guitarists.

4.2 Guitar Arrangement by using HMM

We start from reviewing how HMM can be applied to the fingering decision problem. Suppose we have a collection of guitar music scores, and we want to model this collection statistically. This means that we need to obtain a function that returns high probability if the music seems to be included in the guitar music collection, and low probability when the music is obviously not a guitar music. Designing a generative model is one method to achieve this.

The generative process of a guitar music is apparently the process of performing a guitar instrument. When the guitar is played, one hand holds the neck and its fingers press strings on the frets. Fingers of the other hand pluck the strings, and eventually a sound is generated. We can see that the output sound is determined when the states of both hands are determined.

In terms of the hand to press the strings, it is less likely to observe a drastic change of the hand form in a very short duration because of physical constraints of the human body. It is also unlikely to observe a long distance move of position of a hand to hold the neck of a guitar. Since these two aspects are relationships between the current and the previous state of holding the neck, we can model them by the first-order Markov chain. Let X_t be the fingering at time t. We can define a probability for observing fingering X_{t+1} as $P\left(X_{t+1}|X_t\right)$. X_t contains four components each of which corresponds the state of each finger of the left hand. Each component has two values: one indicates the string index of a guitar to put pressure on, and the other indicates the fret number to put the finger on. Fret number 0 indicates that the finger does not touch any string.

The output sound is audible when strings are plucked. The sounding notes are biased by the fingering. Let Y_t be the set of notes played at time t, such as set consisting of C3, E4 and G4. Y_t follows a probability distribution $P\left(Y_t|X_t\right)$ which models the playing notes biased from the fingering.

A guitar performance can be realized as a time sequence of both the fingering ($X_1^T = X_1 \cdots X_T$) and the plucking of strings at each fingering ($Y_1^T = Y_1 \cdots Y_T$). Note that T indicates the length of a sequence, not indicating transposition. The probability of generating notes from the given fingering is calculated by the product of these probabilities as:

$$P\left(Y_1^T|X_1^T\right) = \prod_{t=1}^{T} P\left(Y_t|X_t\right) P\left(X_t|X_{t-1}\right). \quad (1)$$

Since the fingering cannot be observed from the guitar music afterwards, X_1^T is hidden and therefore this probabilistic model is called as *hidden* Markov model. $P\left(Y_t|X_t\right)$ is called as emission probability, and $P\left(X_t|X_{t-1}\right)$ is called as transition probability. By using Viterbi decoding, we can efficiently estimate the most likely fingerings which maximize the likelihood in terms of X_1^T [23].

Now we can extend the generative model discussed above to let the model generate music that is not necessary to be guitar music, but *could be arranged* into guitar music. In particular, the emission probability is revised so that the model can output notes which are octave higher or lower than the notes of the actually played pitches by plucking the string. As Hori *et al.* formulated in their works [7–9], the emission probability is set to allow the number of notes more than the guitar can perform simultaneously.

By executing the Viterbi algorithm with this extension, we can obtain a sequence of fingering from not only guitar music but also from any music which is not originally composed for a guitar. Since the existence of proper fingering is the necessary condition for a guitar arrangement, we can generate a guitar solo cover by the above extension of fingering decision formulation.

4.3 Creative MIR Approach to Guitar Arrangement

4.3.1 Leveraging MIR technologies

One of the novelties of this research is the further extension of the generative model so that it can generate guitar solo covers from polyphonic music audios by leveraging MIR technologies. Since the melody, beats, and chords are main elements that can be reflected in a guitar solo arrangement, it is not necessary to try to obtain all notes by using multi-pitch analysis. We therefore use methods that can estimate the melody (F0 contour), beats, and chords in polyphonic audio including drums. We show that these methods developed in the MIR community largely contribute to generating a guitar solo cover.

The melody estimation here assumes that the melody is sung by a singer. We first extract a singing voice track by using an existing singing voice extraction method. We then applied a melody estimation method proposed by Goto [5] to obtain the F0 contour. We also smooth the F0 contour by using an FIR low-pass filter with 5 Hz cutoff frequency in order to remove the vibrato. To discretize the F0 contour into musical notes, we used beat estimation results to approximately obtain what musical note is played as the melody line at every 16th note.

Chord estimation provides chord labels (chord names). Since the label contains "on-chords" such as "C/E" or "C on E", we literally use the bass note described in the chord label. We used a chord estimation method developed by Korezeniowski *et al.* [13], which is available in Madmom or an audio signal processing library written in python [2].

Finally, the beat estimation plays an important role in generating a guitar solo cover. The beat estimation results give us a set of segments corresponding to quarter notes. By dividing every quarter note into four parts, we obtain a finer set of segments with the resolution of 16th notes. These 16th-note segments can be used to quantize the F0 contour of the melody line as explained above, and also quantize chord estimation results. In other words, all note lengths extracted from the audio are quantized into integer multiples of the 16th-note duration. We used a beat estimation method proposed by Böck *et al.* [3] which is also available in Madmom [2].

4.3.2 Emission probability

Based on the results of these estimation methods, we set the emission probability as follows so that the HMM can handle music audio to generate a guitar solo cover:

$$\begin{aligned} P\left(Y_t|X_t\right) &\propto P_{\text{chord}}\left(C_t|X_t\right) + P_{\text{melody}}\left(M_t|X_t\right) \\ &+ P_{\text{bass}}\left(B_t|X_t\right) \end{aligned} \tag{2}$$

The subscript t denotes the index of the onset. Since the onsets are not apparent in audio signals, we regard the timing of a sudden increase of power in singing voice and the timing of every chord change as onset timings. The onset timings are discretized by using the beat estimation result. Y denotes the audio segment with 16th-note duration. C, M, and B are the chord label, melody pitch, and bass pitch, respectively. X denotes the fingering to press the fret, which is a set of the pressing position of each finger including open strings. Open strings are represented as pressing the imaginary 0^{th} position of the fret.

Probability P_{chord} is set based on how the current fingering achieves the chord observed at the time. For example, when the fingering is given to play "C, E, G", the probability for observing "C maj7" would be high, but the probability for observing "F# maj" would be low. This can be measured by the number of elements in intersection between the set of notes derived from the fingering and the chord label. In this example, the set of notes for "C maj7" is "C, E, G, B", and the set of notes for "F# maj" is "F#, A#, C#". The probability for observing "C maj7" is higher since the intersection has "C, E, G" (3 elements) whereas "F# maj" has no elements as intersection. We implemented this as follows:

$$P_{\text{chord}}\left(C_t|X_t\right) \propto \exp\left(-\alpha \cdot \#\left(\mathbf{N}\left(C_t\right) \cap \mathbf{N}\left(X_t\right)\right)\right) \tag{3}$$

where $\mathbf{N}\left(\cdot\right)$ denotes the set of consisting notes of chord label or guitar fingering. We adjusted the parameter to be $\alpha = 3.0$ in our experiments.

Probability P_{melody} is designed by considering how the highest note of the playing notes with the fingering is relevant to the melody pitch observed in the acoustic signal. Let $M\left(X_t\right)$ be the highest note could be played from the current fingering. The probability can be designed as:

$$P_{\text{melody}}\left(M_t|X_t\right) \propto \begin{cases} 1.0 & \left(M_t = M\left(X_t\right)\right) \\ \varepsilon_1 & \left(M_t = M\left(X_t\right) + 12n\left(n \neq 0\right)\right) \\ \varepsilon_2 & \text{otherwise} \end{cases} \tag{4}$$

where the parameters are set as $\varepsilon_1 = 0.3$ and $\varepsilon_2 = 10^{-5}$. These parameters were set heuristically by iteratively generating and subjectively evaluating the results.

Finally, probability P_{bass} is set similarly to P_{melody}. Let $B\left(X_t\right)$ be the lowest note that could be played from the current fingering X_t. The probability is designed as:

$$P_{\text{bass}}\left(B_t|X_t\right) \propto \begin{cases} 1.0 & \left(B_t = B\left(X_t\right)\right) \\ \varepsilon_1 & \left(B_t = B\left(X_t\right) + 12n\left(n \neq 0\right)\right) \\ \varepsilon_3 & \text{otherwise} \end{cases} \tag{5}$$

where the parameter ε_1 shared the same value as in P_{melody}, and ε_3 was set as $\varepsilon_3 = 0.0027$ in our experiment.

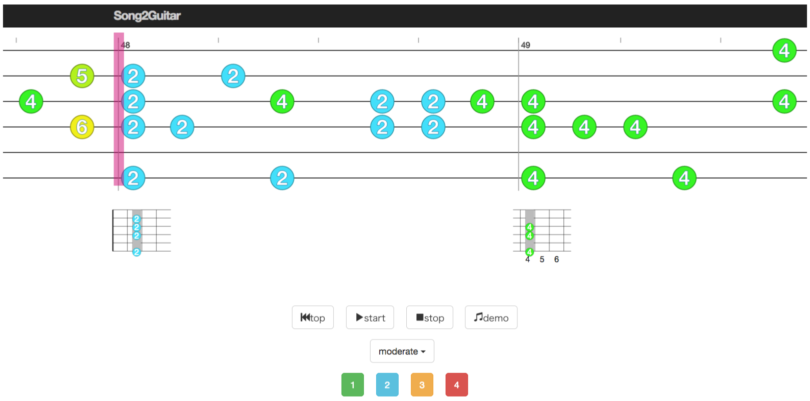

Figure 4: User interface of the Song2Guitar system.

4.3.3 Transition probability

For setting transition probability $P(X_{t+1}|X_t)$, we basically followed the formulation by Hori *et al.* [7]. We defined the transition probability given the time interval d_t between onsets as:

$$P(X_{t+1}|X_t) \propto \frac{1}{2d_t} \exp\left(-\lambda_m \frac{|I(X_{t+1}) - I(X_t)|}{d_t}\right) \times \frac{1}{1+I(X_{t+1})} \times \frac{1}{1+W(X_{t+1})} \times \frac{1}{1+N(X_{t+1})} \tag{6}$$

where $I(X)$ denotes the position of an index finger when holding the fret of a guitar with a fingering X. $W(X)$ denotes the length between the leftmost fret used and the rightmost fret used under the fingering X. $N(X)$ denotes the number of fingers used to achieve the fingering X.

4.4 Controlling the Degree of Difficulty

Based on the survey described in section 3, we determined the following two parameters to control the playing difficulty for generating a guitar solo cover: the average movement of the index finger of a hand to hold the guitar, which is denoted as a_{move}, and the average number of fingers to press the strings, which is denoted as a_{string}.

Song2Guitar supports three different levels of playing difficulty: EASY, NORMAL, and HARD. To create these levels by changing a_{move} and a_{string}, we adaptively restricted the use of fingering X according to the following constraints:

$$\begin{aligned} \text{EASY}: a_{\text{move}} \leq 2.0 \quad \&\& \quad a_{\text{string}} \leq 2.0 \\ \text{NORMAL}: a_{\text{move}} \leq 4.0 \quad \&\& \quad a_{\text{string}} \leq 3.0 \\ \text{HARD}: \text{use all available fingerings.} \end{aligned} \tag{7}$$

5. INTERFACE DESIGN OF SONG2GUITAR

Song2Guitar aims at not only generating a guitar solo cover automatically but also enabling a guitarist to easily practice and perform the generated result. Fig. 4 shows the main interface of Song2Guitar. The design of the interface and the results generated by our system are available on the web [3].

Because the tablature form is more intuitive than the music score, Song2Guitar visualizes the tablature score of the generated cover song. This tablature score scrolls automatically while playing since a guitarist uses both hands to perform the guitar and no hands are left to control the system.

We also implemented an interface to control the playing difficulty of the results. When we aim at creating playable arrangements for human guitarists, it is important to control how difficult the generated score is. Guitarists would be discouraged if the score is too difficult or too easy for them.

The tablature shown in the interface contains additional notations to make the practice and performance easier. Numbers in colored circles on the strings indicate the fret that the guitarist should press on the string. The indicator with a purple vertical line (in the left of Fig. 4) shows the timing to pluck the string.

The interface of Song2Guitar also supports non-proficient guitarists to find the position to press the indicated frets. Usually a guitarist needs to prepare the hand form to press the fret in advance of plucking the strings. Even though the tablature score is shown, a novice guitarist often gets stuck in keeping finding where to put their left hand to hold the neck of a guitar. This is because the tablature usually indicates only the fingerings on the frets, but does not indicate the position of the left hand to hold the neck of the guitar. Therefore we implemented to show small diagrams (shown below of the tablature in Fig. 4) representing how to place the fingers in a similar fashion to a guitar fingerboard. The diagram is shown when there is a position change in a hand to hold the frets.

The Song2Guitar interface also supports a demonstration mode which playbacks the generated tablature by using synthesized guitar sounds so that music listeners can simply enjoy the system output.

[3] https://youtu.be/fN4-ibh7ZDI

6. EVALUATION

To evaluate how the performing difficulty varied among the generated guitar solo covers, we conducted an experiment in a qualitative evaluation approach.

6.1 Experimental setting

We asked a guitarist who is proficient in playing the classical guitar to participate in the evaluation. The guitarist was male and 24 years old, and had an experience in playing the acoustic guitar (both steel and nylon strings) for around six years.

We used RWC-MDB-P-2001 No.7 from the RWC Music Database [6] to generate a guitar solo cover. We generated three different covers with different levels of difficulty (EASY, NORMAL, and HARD) by using Song2Guitar. To focus on evaluating varying difficulty, we manually corrected estimated beats and chords before generating them.

The guitarist was first asked to practice each score for 15 minutes. Since the duration of generated pieces were about five minutes long and it was too long to practice the entire song, we asked the guitarist to practice only the intro, the first verse, and the chorus section. After the practice, we asked the guitarist to play all designated sections of each cover. Finally, we conducted a short interview to obtain comments on Song2Guitar. The obtained comments were originally in Japanese, and they were translated into English as shown in this paper.

6.2 Evaluation results

We obtained a comment indicating that the participant enjoyed using the system:

I think this is a really great app.. I can play a song endlessly, and it was like some kind of a game.

We also found a comment to indicate that our system generated covers in three different levels of playing difficulty (EASY, NORMAL, and HARD):

Well, playing difficulties were appropriate, difference between NORMAL and HARD makes sense.

Although we intended to make three covers as getting gradually difficult, the participant commented that the playing difficulty of EASY and NORMAL were reversed:

EASY score was not easy, it was more difficult than NORMAL one, for me. The HARD score was like in the middle of NORMAL and EASY.

The participant reported why “EASY score was not easy” as follows:

I guess that it’s easier when it consists of chords (multiple notes) moderately than full of simple notes. Chords are the basic form, and I can figure out how to do fingering in my mind. When only two notes appear in the tab, of course, I can figure out the fingerings, however, it didn’t go well [...]

This comment indicated that smaller number of notes are not always easy to play. The fingering of chords provides a basic form, and a guitarist is more familiar with it than the other irregular fingerings for fewer notes.

The participant also pointed out the playing difficulty comes from the note value of the generated results.

The difficulty is, I think it’s easy if all notes were eighth note. Sixteenth note is difficult to figure out the timing.

He also indicated the issue in the interface design:

It’s hard to understand beats and timings of notes with the interface. I appreciate if every half beat were highlighted, somehow.

7. DISCUSSION

We confirmed that Song2Guitar was able to generate guitar solo covers from polyphonic audio of popular music by leveraging MIR technologies. We found that the HMM formulation to generate guitar solo combined with estimation of melody (F0), beats, and chords was effective even from music audio which multi-pitch analysis cannot be sufficiently applied to.

We also found that Song2Guitar was able to generate output with different playing difficulties. We introduced two parameters: the average movement of an index finger and the average number of fingers to press the strings, to control the playing difficulty. The evaluation results, however, suggested that there would be more factors that affect the playing difficulty. One possible factor for determining the difficulty is the familiarity of particular fingerings such as chords.

The interface of Song2Guitar enabled the player to practice and perform the generated result. The comments obtained in the experiment revealed that the rhythms of the generated covers were sometimes hard to recognize. The interface did not visualize the timing except for showing the indicator bar. Highlighting half beats would help the player recognize the rhythm much easier.

The future work of this research is to enable Song2Guitar to generate cover songs in real time considering the player’s proficiency. Conducting an objective evaluation is also included in future work. Since the generative model is designed as a probabilistic model, we can verify the fingering model by calculating cross-entropy.

8. CONCLUSION

We proposed Song2Guitar that generates guitar solo covers from polyphonic music signals of popular songs. The formulation using HMM was combined with MIR technologies so that it can generate covers considering the melody, bass and rhythm of the songs. Furthermore, Song2Guitar generated covers with different levels of playing difficulty. The interface was implemented and a guitarist succeeded in playing different guitar solo covers. In the future, cover song generation from music audio signals will be further improved by leveraging other MIR technologies.

9. ACKNOWLEDGEMENT

This work was supported in part by JST ACCEL Grant Number JPMJAC1602, Japan.

10. REFERENCES

[1] Ana M. Barbancho, Anssi Klapuri, Lorenzo J. Tardon, and Isabel Barbancho. Automatic transcription of guitar chords and fingering from audio. *IEEE/ACM TASLP*, 20(3):915–921, 2011.

[2] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. madmom: a new Python Audio and Music Signal Processing Library. In *Proceedings of the 24th ACM International Conference on Multimedia*, pages 1174–1178, Amsterdam, The Netherlands, 10 2016.

[3] Sebastian Böck, Florian Krebs, and Gerhard Widmer. Joint beat and downbeat tracking with recurrent neural networks. In *Proc. ISMIR*, ISMIR '16, 2016.

[4] Matthew E. P. Davies, Philippe Hamel, Kazuyoshi Yoshii, and Masataka Goto. Automashupper: Automatic creation of multi-song music mashups. *IEEE/ACM TASLP*, 22(12):1726–1737, Dec 2014.

[5] Masataka Goto. A real-time music-scene-description system: predominant-f0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication*, 43(4):311 – 329, 2004. Special Issue on the Recognition and Organization of Real-World Sound.

[6] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC music database: Popular, classical and jazz music databases. In *Proc. ISMIR*, volume 2 of *ISMIR '02*, pages 287–288, 2002.

[7] Gen Hori, Hirokazu Kameoka, and Shigeki Sagayama. Input-output HMM applied to automatic arrangement for guitars. *Journal of Information Processing*, 21(2):264–271, 2013.

[8] Gen Hori and Shigeki Sagayama. HMM-based automatic arrangement for guitars with transposition and its implementation. In *Proc. ICMC/SMC*, 2014.

[9] Gen Hori and Shigeki Sagayama. Minimax viterbi algorithm for HMM-based guitar fingering decision. In *Proc. ISMIR*, 2016.

[10] Eric J. Humphrey and Juan P. Bello. From music audio to chord tablature: Teaching deep convolutional networkds to play guitar. In *Proc. IEEE ICASSP*, pages 7024–7028, 2014.

[11] Eric J. Humphrey, Douglas Turnbull, and Tom Collins. A brief review of Creative MIR. *Proc. ISMIR (Late Breaking/Demo Session)*, 2013.

[12] Christian Kehling, Jakob Abesser, Chirstian Dittmar, and Gerald Schuller. Automatic tablature transcription of electric guitar recordings by estimation of score- and instrument-related parameters. In *Proc. DAFx*, 2014.

[13] Filip Korzeniowski and Gerhard Widmer. Feature learning for chord recognition: the deep chroma extractor. In *Proc. ISMIR*, ISMIR '16, pages 37–43, 2016.

[14] Robert Macrae and Simon Dixon. Guitar tab mining, analysis and ranking. In *Proc. ISMIR*, pages 453–458, 2011.

[15] Matt McVicar, Satoru Fukayama, and Masataka Goto. Autoleadguitar: Automatic generation of guitar solo phrases in the tablature space. In *Proc. IEEE ICSP*, pages 599–604, Oct 2014.

[16] Matt McVicar, Satoru Fukayama, and Masataka Goto. Autorhythmguitar: Computer-aided composition for rhythm guitar in the tab space. In *Proc. ICMC/SMC*, 2014.

[17] Matt McVicar, Satoru Fukayama, and Masataka Goto. Autoguitartab: Computer-aided composition of rhythm and lead guitar parts in the tablature space. *IEEE/ACM TASLP*, 23(7):1105–1117, 2015.

[18] Graham Percival, Satoru Fukayama, and Masataka Goto. Song2quartet: A system for generating string quartet cover songs from polyphonic audio of popular music. In *Proc. ISMIR*, pages 114–120. Citeseer, 2015.

[19] D.P. Radicioni. *Computational Modeling of Fingering in Music Performance*. PhD thesis, Università di Torino, Centro di Scienza Cognitiva, 2005.

[20] Aleksander Radisavljevic and Peter Driessen. Path difference learning for guitar fingering problem. In *Proc. ICMC*, volume 28, 2004.

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

[22] Daniel R. Tuohy and W. D. Potter. GA-based music arranging for guitar. In *Proc. IEEE Congress on Evolutional Computation*, pages 1065–1070, 2006.

[23] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theor.*, 13(2):260–269, September 2006.

[24] Kazuaki Yazawa, Katsutoshi Itoyama, and Hiroshi G. Okuno. Automatic transcription of guitar tablature from audio signals in accordance with player's proficiency. In *Proc. IEEE ICASSP*, pages 3146–3150, 2014.

[25] Kazuaki Yazawa, Daichi Sakaue, Kohei Nagira, Katsutoshi Itoyama, and Hiroshi G. Okuno. Audio-based guitar tablature transcription using multipitch analysis and playability constraints. In *Proc. IEEE ICASSP*, pages 196–200, 2013.