STRUMMER: AN INTERACTIVE GUITAR CHORD PRACTICE SYSTEM

Shunya Ariga[1], *Masataka Goto*[2], *Koji Yatani*[1]

[1]Interactive Intelligent Systems Laboratory, The University of Tokyo
{ariga, koji}@iis-lab.org
[2]National Institute of Advanced Industrial Science and Technology (AIST)
m.goto@aist.go.jp

ABSTRACT

Musical instrument playing is a skill many people desire to acquire, and learners now have a wide variety of learning materials. However, their volume is enormous, and novice learners may easily get lost in which songs they should practice first. We develop Strummer: an interactive multimedial system for guitar practice. Strummer provide data-driven and personalized practice for learners in order to identify important and easy-to-learn chords and songs. This practice design is intended to encourage smooth skill transfers to songs that learners even have not seen. Our user study confirms the benefits and possible improvements of the Strummer system. In particular, participants expressed their positive impressions on lessons provided by the system.

***Index Terms*—** Musical instrument training; musical chords; data-driven approaches.

1. INTRODUCTION

Playing musical instruments requires motor skills of fingers and hands. Training for these instruments can be demanding and time-consuming though many people have strong desire to acquire such skills. This problem is more prominent in polyphonic instruments, such as guitars and pianos. These instruments involve playing chords (i.e., harmonic sets of more than two notes). Learners thus have to develop sufficient dexterity to play chords, which can be discouraging.

There are many possible approaches for self-training. Learning materials for self-training include chord books (book illustrating finger placements for chords); music theory books discussing relationships between chords; and textbooks providing comprehensive tutorials and exercises. Many online resources for music training are also available. Examples are chordify[1], GuitarTricks[2] and Yousician[3]. However, the volume of available materials is enormous, and learners without experience and knowledge cannot easily identify songs to choose for their training. For instance, novice learners may not be aware of which songs are easy or difficult to practice. Some songs may have important characteristics for effective training (e.g., songs including chords frequently used in other songs). This motivates us to explore another computer-supported music training approach mainly targeted to novice learners.

We present Strummer, an interactive guitar chord practice system. Strummer considers three characteristics of chords in a song for prioritization within the given set of music annotations. These characteristics include: chord difficulty, chord frequency, and chord transition difficulty. It then prioritizes songs including chords and their transitions that are frequently-used and easy-to-play. Learners can go through step-by-step, interactive tutorials that are designed to practice from a chord to the entire song. Tutorial contents also change depending on song sets that learners choose in order to provide personalized tutorials for accommodating their preferences. Our contributions are two-folded:

- **A chord and song prioritization algorithm**: Our algorithm identifies which songs users should practice first based on three elements of chords.
- **An interactive system that provides guitar chord training**: Strummer provides step-by-step tutorials to practice chords and songs in a guitar. It can also analyze audio recordings of guitar performances by learners to judge whether each chord is correctly performed.

2. RELATED WORK

It is pedagogically desirable that learners are directly taught by an instructor from the beginning, and start with basic principles [1]. However, human instructors are expensive and not always available. Furthermore, supporting self-motivation and independence is important for successful skill acquisition [2].

Playing musical instruments requires a considerable amount of training and effort. Prior work has investigated how computers can support learners' practice of musical instruments. Dannenberg et al.'s Piano Tutor [3] is one of the early work in this space. This system analyzes learners' errors by taking matches with the given score and using training heuristics. Takegawa et al. [4] developed a system

[1]https://chordify.net/
[2]https://www.guitartricks.com/
[3]https://get.yousician.com/

that recognizes learners' finger and hand positions on the piano, and then offers visual feedback onto the piano keys by a projector on the ceiling. Wearable devices are often used for supporting musical instrument practice. Johnson et al. [5] created a system to support a practice of playing the violin with wearable sensors. Huang et al.'s Piano touch [6] informs learners of finger movements in piano playing through vibrotactile feedback with glove-shaped devices.

There are several projects targeted to guitar practice. Motokawa et al. [7] created a system to teach finger placement on the guitar strings by displaying a hand position image of the given chord. Barthet et al.'s Hotttabs [8] offers online practice materials, containing multimedia contents, including video tutorials and guitar TABs.

There are commercially available games and systems for encouraging learners' practice, and they greatly inspired some of our interface design. Guitar Hero[4] uses a simplified guitar device, and users press buttons as the display instructs. Rocksmith[5] provides a similar game environment to Guitar Hero, but users can connect a real guitar to the system. Yousician is an online training system for pianos, guitars, and ukuleles. It offers step-by-step tutorials and immediate feedback on the user's performance. Yousician is one of the closest systems to Strummer, but we incorporate a data-driven approach for prioritizing chords and songs for guitar practices.

3. MUSIC AND CHORD DATASET

3.1. Preprocessings

We used Billboard dataset [9] for the source of chord practice materials. The dataset includes chord notations of popular 890 songs (151 songs are duplicated thus it contains 739 unique songs) which ranked in the Billboard "Hot 100" chart in U.S. at least once. The notation is described in a context-free chord transition representation developed by Harte et al. [10].

We decided to remove twelve songs that only contain fewer than three chords because they are considered to have little training effect. We built a parser and analyzer for this chord dataset. In pre-processing, the parser first extracts the basic information of a song, including the title and artist. It then parses the chord information that contains timing and chord symbols. Most songs in the music dataset have repeated *section* patterns (e.g., verses and choruses). The parser performs simple matching of descriptions available in the dataset to mark such patterns. During the parsing, we broke down each song to smaller components for our tutorials. We fixed the time signature as 4/4 (i.e., each bar consists four quarter notes). A bar containing more than three different chords can be difficult to practice, thus we divide it into two half notes. We define a *segment* as a part of a section containing four bars (with an exception that the last segment can contain fewer bars).

We performed two kinds of simplification before developing Strummer. First, we performed representation simplification (e.g., $C : maj$ to C). We then executed chord simplification on tension chords except add 9th chords. For example, the chord of minor 9th is converted to minor 7th. We also converted inversions and fraction chords into simple forms. The types of guitar chords we use in Strummer are: *major*, *minor*, *7th*, *major 7th*, *minor 7th*, *minor major 7th*, *augumented*, *diminished*, *half-diminished 7th*, *major 6th*, *minor 6th*, *add 9th*, *suspended 4th*, and *5th(power chord)*.

After finishing all processing, our parser then converts all information into a JSON file in the format compatible with Songle Widget [11]. We use this converted data in our Strummer system.

3.2. Chord and Transition Difficulty Rating

As our application is guitar practice, we also need to consider how difficult each chord transition is to play. We first determined chord difficulty through manual rating. Two raters, including one of the authors, independently evaluated chord difficulty with a 5-Likert scale (1: easiest – 5: most difficult). The raters were instructed to consider only the complexity of the finger placement for each chord. They rated 180 chords including 163 appeared in the dataset. As a result, 52.2% of the chords had the exact same ratings. The square-weighted Cohen's Kappa was $\kappa = .81$ (95% CI: [.75, .85], indicating a substantial agreement. We use the average rating as a score of chord difficulty in our system.

We also developed difficulty ratings for chord transitions. However, the number of possible chord combinations in the dataset is theoretically 26,569, and it is infeasible to manually rate all of them. We thus decided to use a statistical approach to determining chord transition difficulty. First, we obtained 5-Likert scale ratings for 100 most frequently-used transitions in a similar manner to the chord difficulty through manual labeling by two raters. In addition, we randomly chose 30 of chord transitions, and rated in a similar manner to broaden the coverage of our model. Through our manual ratings, 60.0% of the 130 transitions had the exact same difficulty scores. The square-weighted Cohen's Kappa was $\kappa = .76$ (95% CI: [.66, .84], indicating a substantial agreement.

Using these sampled transitions with ratings, we ran a linear regression to create a model to estimate the difficulty of all chord transitions. We initially started with two potential independent variables, and used AIC for determining the optimal model. As a result, we had the following independent variables and estimated coefficient values ($p < .001$ for all) besides the intercept of 0.790:

- *diff_t* (0.310): The difficulty of the target chord.
- fig_num (0.211): The increase or decrease of the number of middle, ring, and little fingers (ranged from -2 to 2). We

[4]https://www.guitarhero.com/

[5]http://rocksmith.ubi.com/rocksmith/

excluded the index finger for this count as it is involved in more than 80% of the chords.

- $\log(move\_base+1)$ (0.336): The logarithm of the absolute movement of the hand, defined as the difference in the fret positions.
- *move_figR* (0.233): The absolute movement of the ring finger. It is defined as the Euclidean distance of the ring finger's position the between the origin and target chord. For example, if the ring finger is moved from the second fret in the fourth string to the third fret in the fifth string, this value is $\sqrt{2}$. It is zero when the origin or target chord does not involve the ring finger.
- *move_figL* (0.242): The absolute movement of the little finger. The definition is similar to *move_figR*.
- *barre_t* (0.481): The target chord is a barre (1 if true, otherwise 0).
- *slide* (-1.401): Both origin and target chords have the same barre form (1 if true, otherwise 0). Their positions can be different.

This model achieved high R-squared value, $R^2 = .786$. We use this model to estimate the difficulty of the given chord transition, expressed in the value ranging [1.0, 5.0].

4. CHORD AND SONG PRIORITIZATION

Strummer needs to identify which chords and songs learners should practice first. This information determines which songs they should prioritize for practice.

We first map all chords in the coordinate of difficulty and inverse frequency of occurrences. The frequency of occurrences of the chord c_i is calculated as $f_{c_i} = N_{c_i} / \sum_C N_{c_i}$. The denominator represents the total number of occurrences of all chords in the given song set. We define chord primariness, CP_{c_i}, for the chord c_i as an Euclidean distance from the origin in this coordinate: $CP_{c_i} = -\sqrt{w_f\,(1/f_{c_i})^2 + w_d {d_{c_i}}^2}$, where d represents the difficulty of a chord, and w_f and w_d are the associated coefficients ($w_f = 0.1$ and $w_d = 10$ in our current prototype system). We add the negative coefficient of -1 so that higher CP represents a more important and easy-to-learn chord.

With CP, we can calculate the primariness of each song, indicating how likely it includes chords that are important and easy to learn. In addition, this calculation should consider the difficulty of chord transitions. Therefore, we formulate the song primariness, or SP, as $SP = \sum_C CP_{c_i} + w_{T_h} \sum_{T_h} TD + w_{T_f} \sum_{T_f} TD$, where C, T_h, and T_f represent all chords, its transitions in the half note, and transitions in the whole note. We set w_{T_h} and w_{T_f} to -1. In this manner, we can only penalize highly difficult transitions in the whole note. The maximum computational complexity of SP is $O(n(n+m))$ where n is the average number of chords (10.2) in the given songs and m is its total number (727). It is possible to run this calculation for all songs in real time.

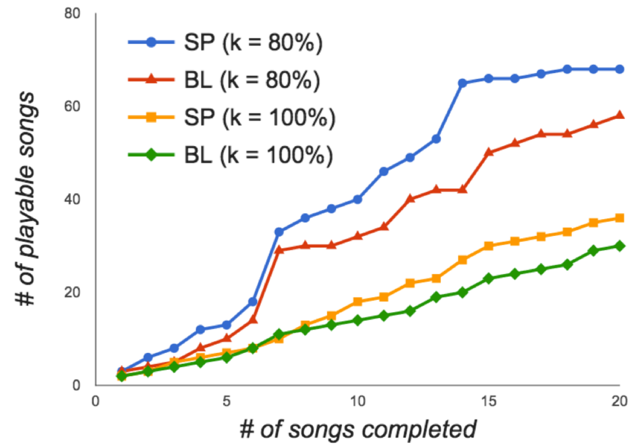

Fig. 1: Plots of k%-playable songs (genre: pops) across the number of songs completed ($k = 80$ and 100) in two ranking algorithms (*SP* and *BL* (the baseline method based on the total chord difficulty scores)).

To validate benefits of *SP*, we define an unseen song to be "k%-playable" if learners would be able to perform k% of the chords in that song. 100%-playable means that learners have already practiced all chords in the given song. We regard song prioritization as effective if it achieves a higher number of k%-playable with practice in a small number of song.

We compared *SP* against the total chord difficulty (TCD) as a baseline approach (*BL*). TCD is defined as the sum of difficulty scores of all chords in a song. Sorting with the *BL* method means that the system prioritizes songs that have fewer and easy-to-play chords. Fig. 1 shows the number of 80% and 100%-playable songs under an assumption that learners have completed practicing the first up to 20 songs in the two rankings. In both 80% and 100%-playable, *SP* constantly exhibits a higher number than *BL*. This means that the chords learners would practice can be more useful to play unseen songs in *SP*. Strummer thus uses *SP* to rank songs in the given set, and sorts them in a descending order of *SP*.

5. STRUMMER SYSTEM

The Strummer interface provides step-by-step tutorials to offer efficient chord practices. When learners log into the system, they first can choose genres, artists, and songs they desire to practice. Strummer then displays the songs in the chosen set in the order of *SP* by default (Fig. 2a). This list also provides information about the number of chords and sections in each song, and learners can sort them if they like. Learners can select a song to start tutorials. Our accompanying video provides a demonstration of the entire system.

5.1. Strummer Tutorials

Tutorials in the Strummer system are designed to provide step-by-step practice for the given song. As we described

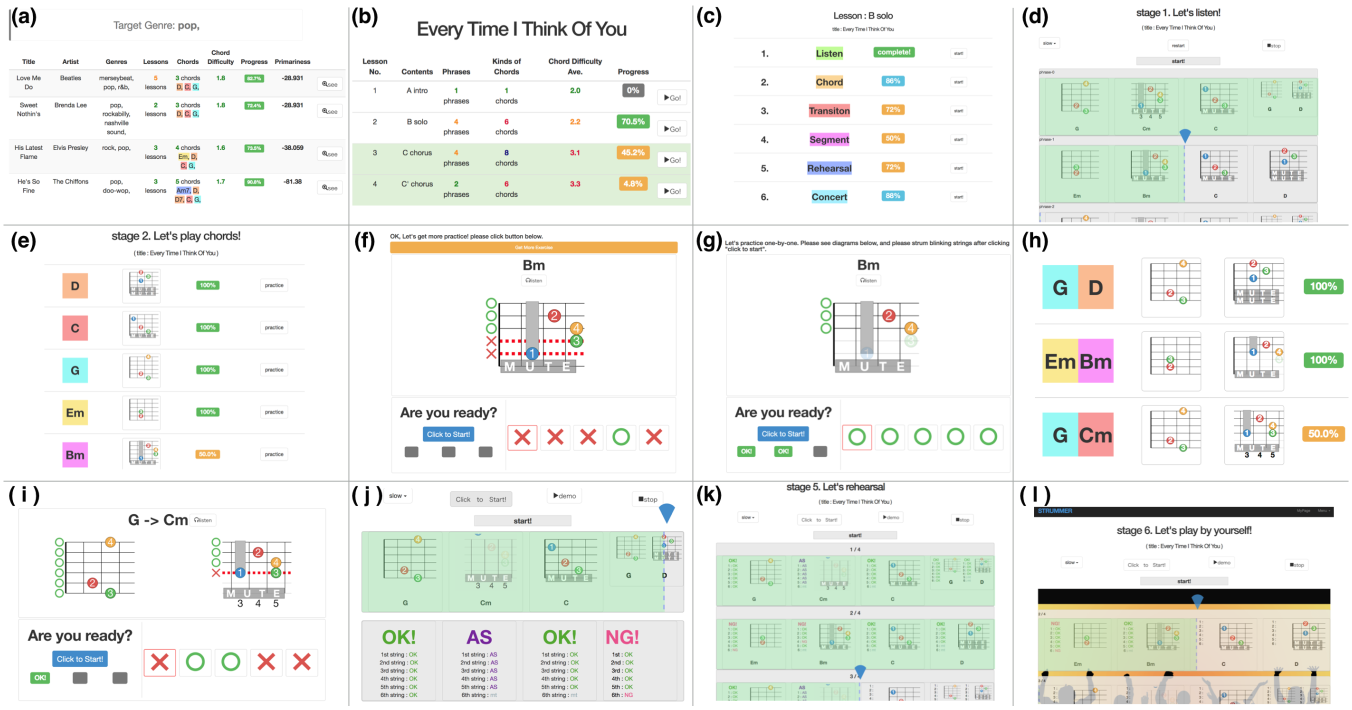

Fig. 2: The walkthrough of the Strummer system. Refer to the accompanying video for the full demonstration. (a) Learners choose songs to practice. They are sorted by *SP*. (b) Each song is divided into sections to make practices manageable. (c) Strummer has six practice stages. (d) Learners can listen to chords along with finger placements in the Listen stage. (e) In the Chord Practice stage, The chords are sorted in a way that learners start with frequently-used and easy-to-play chords. (f) Learners practice chords one-by-one in the Chord Practice stage. (g) Strummer offers additional, lower-level tutorials for unsuccessful chords (*Bm* in this example). (h) The system records a partial finger combition of a chord learners can play, and only enables performance assessment in later tutorials. Note that the finger placement of *Bm* is different from the original (shown in (f)). In this manner, Strummer encourages learners to practice the entire song even if they cannot play some chords well. (i) Learners practice chord transitions in a similar way to individual chords. (j) A screen for practicing a *segment* of the given song. (k) Learners practice a portion of the song in the Rehearsal stage. (l) Learners can also practice the entire song with a similar interface to music games.

above, a song generally has some repetitions (i.e., sections). The system first removes such repetitions and generates lessons each of which contains a unique set of chords. When learners choose a song, the system presents a set of tutorials each of which focuses on practicing one of the sections (Fig. 2b).

Learners can start a tutorial by simply clicking it. During a tutorial, Strummer provides six practice stages explained later in this section (Fig. 2c).

- **Listen Stage:** At the beginning of a practice, learners are encouraged to listen to the entire section before starting practice. They can listen to the song along with chord information (Fig 2d).
- **Chord Practice Stage:** In this stage, learners practice all chords used in the chosen section one-by-one. The system sorts the chords by *CP* in the descending order (Fig 2e). When learners click a chord, the Strummer system displays the finger placement in a similar fashion to guitar TAB (Fig 2f). The system plays the chord in a strumming style. Learners can then confirm what the chord sounds like. When learners feel ready, they press the start button to check their performance. The system then activates an audio-based chord recognition module (we explain its details later). The Strummer interface then informs them whether they have successfully played the given chord. Upon unsuccessful performance, it offers feedback about which strings the learners failed to play (Fig. 2f). If they are able to play the chord three times continuously, the system records the completion.
- **Supplemental Chord Practice:** When they miss the chord more than five times, or missed three times in sequence, the system enables a button for additional tutorials. The learners can continue to practice the chord separately in those tutorials (Fig. 2g). This separate tutorials offers lower-level training. It starts with practice for each finger, and progressively increases the number of fingers to be used. However, learners still may struggle playing a chord even after this additional training. The system records with which finger combinations the learner have been successful, and enables performance assessment only for these fingers in later tutorials (Fig. 2f, g). For example, suppose that a learner can play the chord C with the index and middle

finger, but not with all three fingers. The system then disables performance assessment for the ring finger.

- **Chord Transition Practice Stage:** Learners can practice chord transitions in a similar manner (Fig. 2h and i). The chord transition practice stage offers the same tutorial except that learners practice transitions.
- **Segment Practice Stage:** After completing the chord transition practice stage, learners move to practice the chosen segment. Similarly to the previous stage, the system displays the chord name and finger placement (Fig. 2j). When the learners press the start button, an animation starts over the chord sequence, indicating the timing to play. The system also activates the chord recognition module, and provides feedback about whether the learners have successfully played each chord at the right timing. The learners can complete this stage by correctly playing the segment at least once. For the chords learners have not completed even after additional training, the system disables the recognition module, and synthesizes the sound on behalf of the learners as scaffolding. In this manner, the learners can continue their practice without being constrained to chords they cannot play well.
- **Rehearsal Stage:** This stage (Fig. 2k) is intended to provide additional repetitions in practice, helping skill fixation [12]. The system marks the stage as completed when learners are able to play all chords successfully. But they can move to the next stage even if they have not completed.
- **Live Concert Stage:** At the end of the tutorial, learners are invited to play the song they have been practicing as if they would do in a concert. We take a similar user interface to popular music games and training systems, such as Guitar Hero and Yousician (Fig. 2l). In this stage, learners are asked to play all sections they have completed as well as what they are currently practicing. After the performance, the screen displays the score based on the number of successfully-played chords. Learners can complete the lesson by being able to perform 60% of the chords.

5.2. Chord Recognition

Fig. 3 shows the overview of our chord recogition workflow. We implement the Strummer system in Ruby on Rails and JavaScript. Fast Fourier Transform (FFT) in Web API does not take sufficiently long samples for chord recognition. We thus run the chord recognition module locally on the learners' machine. It is written in Python with the numpy and pyaudio libraries. The sampling rate is 44.1 kHz, and the window size of FFT is 8192 samples. We used the Hanning window in sampling. The module then matches the fundamental frequencies of a chord with the observed frequency peaks to determine whether it is correctly played. The module sends the recognition results to the main system via HTTP.

6. USER EVALUATION

We conducted a preliminary user study to examine the effect of Strummer. We mainly investigated the user experience of the features and tutorials to validate the design of the Strummer interface. We recruited 5 people (P1 – P5, all male, the average age of 21.6). All of them had nearly no experience of playing the guitar.

We designed our experiment to be five days long. We asked our participants to come to our lab every day during the experiment. They were instructed to practice with the given condition for at least one hour. Two of our participants were not able to accommodate this schedule, and they practiced for two hours at the fourth day. We lent the participants a classical guitar during the experiment.

At the beginning of the first day, we explained the interface and experimental procedure. We also manually provided basic tutorials about how to play the guitar. After this pre-experimental briefing, the participants were asked to practice starting from the first song in the given list.

When participants completed practicing one song, we conducted a short interview to understand their experience. In addition, we conducted another short interview to elicit any opinion about the system at the end of each day.

On the last day, we asked participants to perform a new song in the dataset that they had not practiced. The intention of this sight reading task was to emulate experience of playing at sight, requiring sufficient skill fixation on chords. We also wanted to ensure that the participants learned how to play chords. We chose "*(Sittin' On) The Dock Of The Bay*" by Otis Redding. This song requires quick transitions in frequently-used chords (C, D, G), and also includes bar chords (F, B) and unseen chords (A, B). After 15-minute practice, they played the song in front of the experimenter. We audio-taped their performance for later analysis.

Our interviews were not in English because none of our participants was native or proficient in the language. We transcribed and translated the interviews to English for the report in this paper. We offered the participants approximately 50 USD in local currency as compensation at the end of the study.

7. RESULTS

All participants successfully completed the study. One of the authors evaluated the performance of the sight reading task. Table 1 presents the number of successful chord playing in the sight reading tasks, showing that all participants performed at least half of all chords correctly.

Our interviews also uncovered perceived benefits of the features in the Strummer interface. Participants agreed that step-by-step tutorials were beneficial. P3 expressed his appreciation of the tutorials as follows:

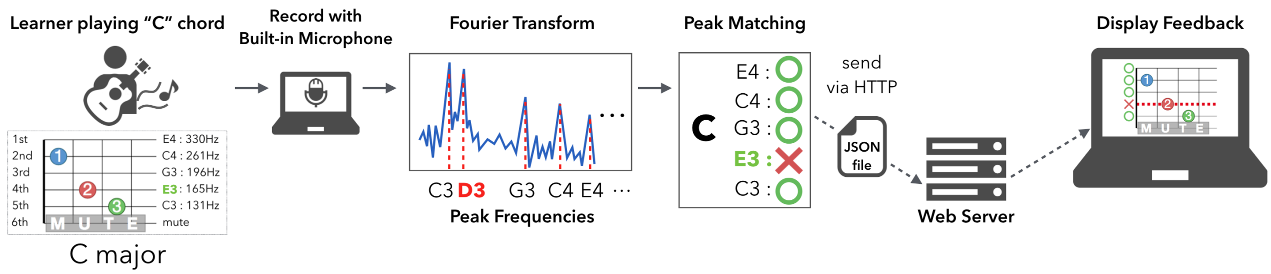

Fig. 3: An overview of our chord recognition workflow.

	Seen						Unseen		
Chord	Em	D	C	E	G	F	A	B	Total
Occurrences	2	4	5	1	14	1	3	2	32
Difficulty	1	1.5	2	2	2	3	2	3.5	2.13(ave.)
P1	2	4	3	1	14	0	3	1	28
P2	1	4	3	0	12	1	0	0	21
P3	1	0	3	0	12	0	0	0	16
P4	2	4	5	1	14	0	3	0	29
P5	2	4	5	1	13	1	3	2	31
Success Rate (%)	85	82	76	50	93	40	57	25	78

Table 1: The results of the sight reading task.

It's like I complete one-by-one, step-by-step. That gave me a good motivation. [P3]

In the step-by-step tutorials, the system included the chord recognition module to assess how well the participants played. This feature was also useful to the participants.

I really like that the system checks my performance... When the system showed unsuccessful placement, I found that I did not put my fingers strongly enough. [P2]

I think the sound feedback was helpful when I was busy with the guitar and was not able to pay attention to the display. [P4]

8. CONCLUSION

Many people have a desire to acquire musical instrument playing skills. Although recent online services offer a variety of training materials, the volume of such resources is overwhelming, and learners may get lost in which to practice. We develop models and algorithms to identify important and easy-to-learn chords given song genres and artists. We instantiate them in Strummer, an interactive system for guitar chord practice tailored toward novice learners. Strummer provides step-by-step tutorials to make practice manageable. Our user study confirmed that participants were able to continue their practice over five days and showed positive attitudes toward our Strummer system.

9. REFERENCES

[1] A. Glise, *Classical Guitar Pedagogy: A Handbook for Teachers*, Mel Bay Publications, Inc., 1997.

[2] G. Percival et al., "Effective use of multimedia for computer-assisted musical instrument tutoring," in *Proc. EMME*, 2007, pp. 67–76.

[3] R. Dannenberg et al., "A computer-based multi-media tutor for beginning piano students," *Journal of New Music Research*, vol. 19, no. 2–3, pp. 155–173, 1990.

[4] Y. Takegawa et al., "Design and implementation of a piano practice support system using a real-time fingering recognition technique," in *Proc. ICMC*, 2011, pp. 387–394.

[5] R. Johnson et al., "Real-time feedback for learning the violin," in *Proc, BCS-HCI*, 2011, pp. 535–538.

[6] K. Huang et al., "Pianotouch: A wearable haptic piano instruction system for passive learning of piano skills," in *Proc. IEEE ISWC*, 2008, pp. 41–44.

[7] Y. Motokawa and H. Saito, "Support system for guitar playing using augmented reality display," in *Proc. IEEE ACM ISMAR*, 2006, pp. 243–244.

[8] M. Barthet et al., "Music recommendation for music learning: Hotttabs, a multimedia guitar tutor," in *Proc. WOMRAD*, 2011, pp. 7–13.

[9] J. A. Burgoyne et al., "An expert ground truth set for audio chord recognition and music analysis," in *Proc. ISMIR*, 2011, pp. 633–638.

[10] C. Harte et al., "Symbolic representation of musical chords: A proposed syntax for text annotations," in *Proc. ISMIR*, 2005, pp. 66–71.

[11] M. Goto et al., "Songle widget: Making animation and physical devices synchronized with music videos on the web," in *IEEE ISM*, 2015, pp. 85–88.

[12] P. M. Fitts and M. I. Posner, *Human performance.*, Brooks/Cole, 1967.