A Real Time Common Chord Progression Guide on the Smartphone for Jamming Pop Song on the Music Keyboard

Conference Paper · July 2014

DOI: 10.13140/RG.2.1.3854.3202

1 author:

Simon Lui
Singapore University of Technology and Design

A Real Time Common Chord Progression Guide on the Smartphone for Jamming Pop Song on the Music Keyboard

Simon Lui
Singapore University of Technology and Design
20 Dover Drive, #03-R17, Singapore, 138682
simon_lui@sutd.edu.sg

ABSTRACT

Pop music jamming on the keyboard requires massive music knowledge. Musician needs to understand and memorize the behavior of each chord in different keys. However, most simple pop music follows a common chord progression pattern. This pattern applies to most simple pop music on all the 12 keys. We designed an app that can reduce the difficulty of music jamming on the keyboard by using this pattern. The app displays the current chord in the Roman numeral and suggests the expected next chord in an easy to understand way on a smartphone. This work investigates into the human computer interaction perspective of music performance. We use a smartphone app as a bridge, which assists musician to react faster in music jamming by transforming the complex music knowledge into a simple, unified and easy to understand format. Experiment result shows that this app can help the non-keyboardist musician to learn pop music jamming. It also shows that the app is useful to assist keyboardist in making key transpose and playing music in the key with many sharps and flats.

Keywords

Chord progression, pop music jamming, human computer interaction on music, music performance guiding tool

1. INTRODUCTION

Many people know how to play a musical instrument as well as understand the basic music theory. However not all of them play the keyboard. Moreover, not many keyboardists can jam and improvise simple music. It is well known that most pop music follow the common chord progression as shown in Figure 1. We call it the *13642571-pattern*. The pattern fits in most simple pop music [1]. It is not a strict pattern. Musician can skip some intermediate states in the pattern. They can make changes in the chord inversion or adding 7th, 9th or 6th note. A few chords may not follow the *13642571-pattern*, but the whole song usually follows the pattern in general. Some pop music is not composed in the *13642571-pattern* originally. However the melody part of that song can usually fit in this pattern as well.

The *13642571-pattern* can be used for three purposes. Firstly, it can be used for causal enjoyment. The *13642571-pattern* can fit in most simple pop music. Musicians with some basic music theory background can jam most simple pop music in this pattern without a score. Secondly, it can be used as a music jamming learning tool. It is the first step towards learning improvisation. Musicians can first practice simple music jamming with the pattern. Then the musicians will be familiar with it and hence able to add transition chord within the pattern. Eventually they will be able to jam outside this basic pattern. Thirdly, it can be used as a real-time assisting tool for keyboard performance. For example, in a party or a causal performance event, musicians are often requested to play some enjoyable music. It could be a keyboard solo or singing accompaniment. The main challenge is how to perform immediately with no score in hand and having no time for preparation. On the other hand, the audience usually doesn't know the original chord progression of the song. They only know the melody. They just want to listen to some good listening music with a good chord progression. In this case, the *13642571-pattern* can help the musician to jam a good listening performance instantly. Moreover, a well-experienced keyboardist might also find it difficult to jam a song with a lot of sharps and flats. For example, an untrained male amateur singer usually prefer to sing in the C# major instead of D major. It is because both low tonic note C#3 (138.6Hz) and D3 (146.8Hz) are easy to sing, but the high tonic note C#4 (277.2Hz) is easier to singer than D4 (293.7Hz) for male amateur singer. However, C# major have 7 sharps which is not easy to jam without preparation. The *13642571-pattern* provide a unified and simple guide for the keyboardist to jam instantly to overcome the clumsy mess of a lot of sharps and flats.

The *13642571-pattern* is good enough for jamming simple pop music. However there are many possible complex chord transition patterns in the other kinds of music. There were a lot of previous works researched on this. For example, Paiement [2] proposed a complete probabilistic model for chord progression using the expectation–maximization (EM) algorithm. Absolu [3] summarized a set of chord progression data with the top-k queries method. In this paper we focus on the presentation of chord progression on a smartphone to assist performance. Our interface and system design can work on any kind of chord progression. So we first work on the *13642571-pattern*. We can use the same system to guide the user to perform music in the other chord progressions.

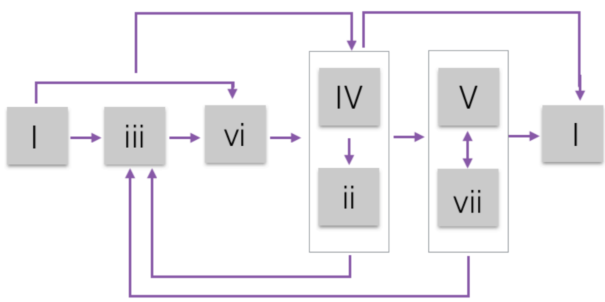

Figure 1. The common chord progression for simple pop music

2. TARGET AUDIENCE

There are three kinds of target audiences. 1) Non-keyboardist musicians who know music theory. 2) Keyboardists who need a chord progression guide during their performance. This is especially important when they play music with key transposition or in the key having a lot of sharps and flats. 3) Anyone who wants to learn to play the common chord progression on the keyboard that can fit in most simple pop music.

3. DESIGN PRINCIPLE

This is a smartphone app for real-time and instant usage. Hence the system display must be simple, easy and quick to read. Firstly, we only display one single note per chord and we choose to display the root note of each chord, since it is the most essential and simplest way to represent a chord. We expect the professional musician to be able to derive and come up with the rest of the notes of the chord, no matter it is a major, minor, or 7^{th} chord. For amateur user, jamming by just playing the root note is also adequate for the enjoyment and learning purpose. For the medium level amateur user, jamming with a *root note, a 5^{th} note, and then an 8^{th} note* pattern by the left hand also sounds very good for the causal performance purpose, which is independent of the chord type but just depends on the root note of the chord. Secondly, to make the app quick to read, we avoid additional text description. We present every instruction graphically and unified as a single symbol. For example, the app highlights the next expected chord with a rotating circle with size contrast, and the Roman chord numeral is embedded on the key surface in the app. This is to make sure that the user can read as many information as possible on one sight. Thirdly, we minimize the number of control steps such that the user can easily access it during the performance, and yet make the app customizable. For example, we only provide one single key transpose button. No other button is provided in order not to mess up the GUI design. Also, the screen can be swapped horizontally. Some users might want to have the root note displayed on the leftmost of the screen, while some user might want to display the F note or C note on the leftmost instead. Fourthly, we display the next expected chord in the simplest way. We avoid additional arrow and text description in order not to mess up the display design. We just highlight the current chord in blue and the next expected chords with a rotating circle spot.

4. IMPLEMENTATION

We implement the app on an iPhone running iOS7, since iPhone is portable, widely used, and supported by a lot of MIDI I/O devices. We use a Kawai CL20 digital keyboard with MIDI I/O for music performance input. We use the iRig MIDI interface that supports the iOS CoreMIDI framework to connect iPhone and Keyboard.

Firstly, the app initializes the key. User can either input the key manually or use the key estimation mode: user can play a melody and the app will estimate the key of the piece by analyzing the melody notes. We use the Krumhansl–Schmuckler key-finding algorithm [4] to estimate the key. After the key initialization, the app will display the Roman chord numeral on the root note of every chord respectively as shown in Figure 2. The user can drag the red "key" icon left and right to change the key at any time.

Then, the user can play music on the keyboard. The app will display the root note of the current playing chord and the expected next chord. For simplicity in this first implementation, we request the users to play all chords in the root position only. So the root note of the current chord can be easily measured, which is the note with the lowest pitch. There are many chord estimation algorithms available. For example, Scholz [5] performs chord recognition from symbolic data using pattern matching and a rule-based optimization technique. Pardo [6] recognizes chord by a graph search based approach. We will use one of the most suitable algorithms afterwards. It doesn't affect the design of the app interface, which is the aim of this paper. The app displays several next expected chord. They are highlighted with rotating circle in different sizes according to their transition probability. It is as shown in Figure 3 and Figure 4.

Figure 2. After the key initialization.

Figure 3. User is playing a vi chord in a music of D major. The app suggests the user to play a IV or ii chord afterwards.

Figure 4. User is playing a IV chord in a music of A major. The app suggests the user to play a iii, V, I or ii chord afterwards.

For acoustic piano with no MIDI I/O port, user can manually set the key of the piece and simply use the Roman chord numeral as shown in Figure 2 for performance reference. In the future we will work on recognizing chord from acoustic piano in real-time with a smartphone microphone. For example, Yoshioka [7] recognizes musical chords from real-world audio signals in compact-disc recordings by estimating chord boundary by a hypothesis algorithm. He gets an accuracy of 77%. Mauch [8] achieves an accuracy of 71% of chord identification with a multi-layered Bayesian network. Sheh [9] is able to identify 75% chord correctly by using the expectation–maximization (EM) algorithm and train it with the HMM model. We believe the technique of real-time acoustic chord recognition will be mature soon and will have much better accuracy. However it is out of the scope of this paper.

5. EXPERIMENT

We invited 10 keyboardists, 10 non-keyboardist musicians, and 10 non-musicians to participate in the experiment. First, we provided a brief introduction about the app followed by a short training of 5 minutes. They had 3 minutes to try out the app before the experiment. We prepared 5 simple songs in major key. All songs have 16 bars with sol-fa name melody and Roman chord numeral written on a paper. The list of songs is as shown in Table 1, which require two *13642571-patterns* to play each song. The participants were invited to pick two songs and choose a key. Then they performed it with the app. The keyboardist players randomly chose from all 12 possible keys, while the other participants only random chose a key with less than or equal to 2 sharps or flats. They were asked to sing the melody and play the accompaniment part on the keyboard. After playing the songs, they were invited to rate our app on three questions in the scale of 1-7 as shown in Table 2. The questions are as shown in Table 3. Figure 5 to 7 show the rating of the participants in a histogram.

We found that the non-musicians have much less *enjoyment* on using the app compared with musicians. Non-musicians reflected that the Roman chord numeral is difficult to understand. The non-keyboardist musicians gave the highest rating in *learning*. The app design is helpful for non-keyboard musicians who understand music theory to pickup keyboard jamming quickly. We also find that the keyboardist players gave a high rating for the app as a *performance-assisting tool*. They reflected that the app is very helpful for them especially when they need to play music in the key with a lot of sharps and flats. Overall, non-musicians seem not to like the app very much. This app is perceived as a great learning tool for the non-keyboardist musicians and a good performance-assisting tool for the keyboardists.

We also invited the participants to give us further comments after the experiment. A professional keyboardist was asked to perform *Moon River* in *B major* with 5 sharps. He found that the app really helped him in locating chord when there were so many sharps to handle. A non-keyboardist musician suggested that we should also display the *13642571-pattern* flow chart as in Figure 1 in the app interactively. However he had no idea of how to combine the chart into the app in a unified way. A non-musician participant commented that he needed more real-time guide on using the app. For example, guide him playing the music with an arrow or with a voice instruction. Since he had difficulty to look at both the app and the keyboard key at the same time. He suggested that we could use a projector to project the Roman chord numeral on the keyboard, or design a transparent sheet and stick it on the keyboard. One of the non-musicians has difficulty to use the app after more than 10 minutes of training. Hence we simplified the task and she just need to play the root note of each chord with one finger. She eventually can play music with the app and found it fun. Now she can sing along and play the keyboard accompaniment with one finger in C Major. She can do it for a lot of simple pop music without a score.

Table 1. List of songs

Love me tender
Try to remember
Moon river
The way we were
My heart will go on

Table 2. Rating interpretation

Value	Interpretation
7	Entirely good
6	Very good
5	Quite good
4	Somewhat good but somewhat bad
3	Quite bad
2	Very bad
1	Entirely bad

Table 3. Participant average rating

Questions	Keyboardist rating	Non-keyboardist musician rating	Non musician rating
Enjoyment	5.4	5.8	3.9
Learning	4.2	6.3	4.6
Performance-assisting tool	5.9	4.6	3.6

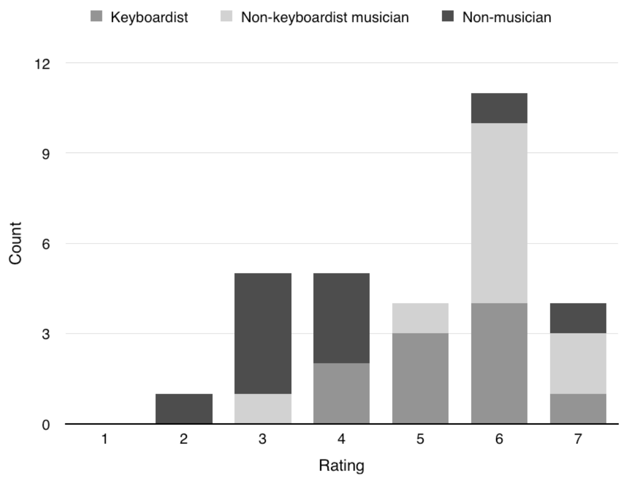

Figure 5. Participants' rating of the app for *Enjoyment*.

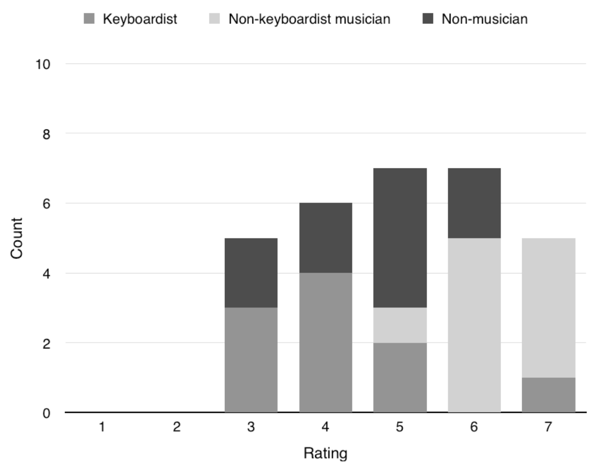

Figure 6. Participants' rating of the app as a *Learning Tool*.

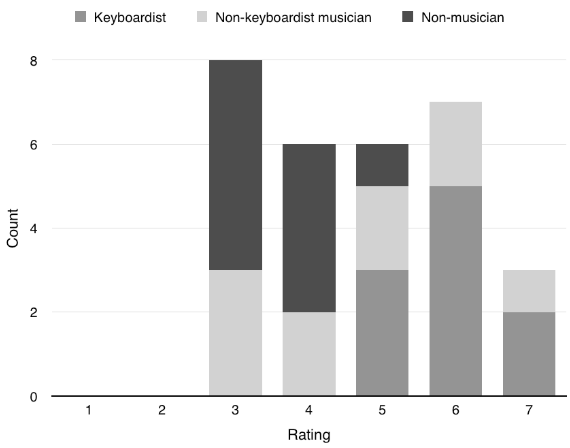

Figure 7. Participants' rating of the app as a *Performance Assisting Tool*.

6. FUTURE WORK

Our app will support other chord progressions, for example, the simple jazz progressions. In order to display a jazz chord progression, we need to provide additional note information other than the root note. We will develop the app on an iPad such that more information can be displayed on a larger screen. We will work on the music in minor key. We will further investigate into the HCI perspective of music. For example, how to adjust the design of showing the expected next chord in different color and size contrast; how to display more chord information in a single unified symbol such as combining *7th* chord and *sus* chord information with the roman chord numeral and the key; how to show the next expected chord in the best way, especially to link the app and keyboard in a better way such that the user doesn't need to switch their eye between the app and the keyboard frequently. We will also look into the possibility of embedding this system into a digital keyboard. For example, compact the GUI design and show it on a tiny LED display, or display it with a dim light symbol on the key. It will be an interesting extra feature for the digital keyboard. We will also invite musician to evaluate the performance of non-musician in using this app. The comment from people that are not themselves users of the app can help us to further improve the app design.

7. ACKNOWLEDGEMENT

This work is supported by the SUTD-MIT International design center (IDC) Research Grant (IDG31200107 / IDD11200105 / IDD61200103). Special thanks for the reviewer's detailed and invaluable comment.

8. REFERENCES

[1] Green, L. (2007). How popular musicians learn: A way ahead for music education. Ashgate Publishing, Ltd.

[2] Paiement, J. F., Eck, D., & Bengio, S. (2005). A probabilistic model for chord progressions. The 6th International Conference on Music Information Retrieval, London, United Kingdom.

[3] Absolu, B., Li, T., & Ogihara, M. (2010). Analysis of chord progression data. In Advances in Music Information Retrieval (pp. 165-184). Springer Berlin Heidelberg.

[4] Lui, S., Horner, A., & So, C. (2010). Retargeting expressive musical style from classical music recordings using a support vector machine. Journal of the Audio Engineering Society, 58(12), 1032-1044.

[5] Scholz, R., & Ramalho, G. (2008). COCHONUT: Recognizing Complex Chords from MIDI Guitar Sequences. In ISMIR (pp. 27-32).

[6] Pardo, B., & Birmingham, W. P. (2001). The chordal analysis of tonal music. The University of Michigan, Department of Electrical Engineering and Computer Science Technical Report CSE-TR-439-01.

[7] Yoshioka, T., Kitahara, T., Komatani, K., Ogata, T., & Okuno, H. G. (2004). Automatic Chord Transcription with Concurrent Recognition of Chord Symbols and Boundaries. In ISMIR (pp. 100-105).

[8] Mauch, M., & Dixon, S. (2010). Simultaneous estimation of chords and musical context from audio. Audio, Speech, and Language Processing, IEEE Transactions on, 18(6), 1280-1289.

[9] Sheh, A., & Ellis, D. P. (2003). Chord segmentation and recognition using EM-trained hidden Markov models. ISMIR 2003, 185-191.