JamSketch: A Drawing-based Real-time Evolutionary Improvisation Support System

Tetsuro Kitahara
College of Humanities and Sciences,
Nihon University,
Tokyo, Japan
kitahara@chs.nihon-u.ac.jp

Sergio Giraldo
Music Technology Group,
Universitat Pompeu Fabra,
Barcelona, Spain
sergio.giraldo@upf.edu

Rafael Ramírez
Music Technology Group,
Universitat Pompeu Fabra,
Barcelona, Spain
rafael.ramirez@upf.edu

ABSTRACT

In this paper, we present *JamSketch*, a real-time improvisation support system which automatically generates melodies according to *melodic outlines* drawn by the users. The system generates the improvised melodies based on (1) an outline sketched by the user using a mouse or a touch screen, (2) a genetic algorithm based on a dataset of existing music pieces as well as musical knowledge, and (3) an expressive performance model for timing and dynamic transformations. The aim of the system is to allow people with no prior musical knowledge to be able to enjoy playing music by improvising melodies in real time.

Author Keywords

Improvisation, melodic outline, melody creation

ACM Classification

H.5.5 [Information Interfaces and Presentation] Sound and Music Computing

1. INTRODUCTION

Improvisation is one of the most enjoyable forms of music performance in which musicians create music compositions in real time combining communication of emotions, instrumental technique and spontaneous response to others. There have thus been attempts to support improvisation by unskilled players using computing technologies [1, 2, 6, 7].

In this paper we present *JamSketch*, a real-time improvisation support system which automatically generates melodies according to melodic outlines drawn by the users. We developed a melody editing system that creates a melody according to a melodic outline drawn by the user [9]. Once the user draws a melodic outline, the system creates a melody according to the outline. Given the intuitive nature of drawing, several drawing-based systems have been proposed to allow people with no prior music knowledge to specify their musical ideas [3, 4, 8]. However, these systems do not create melodies in real time, so they are not unsuitable for supporting of real-time improvisation. Our system is able to generate expressive improvised melodies in real time given an outline drawn by the user.

2. SYSTEM OVERVIEW

A screenshot of our system is shown in Figure 1. Once the system is launched, a piano-roll interface is displayed on the screen. During the playback of an accompaniment (given by a MIDI file with a chord transcription), the user can draw a melodic outline on the piano-roll screen by using the computer mouse or touch pad as an input device. The melody is created individually for each measure. Once the mouse coursor enters the region of measure m and then moves out from that region (or the mouse button is released there), the creation of a melody for measure m starts. After the melody is created, the most likely expression parameters (the onset, duration, and energy for each note) are estimated. Generation of melody and expression parameters in the current implementation takes approximately 0.5 s and 0.2 s. Thus, users are required to draw their melodic outlines one bar in advance.

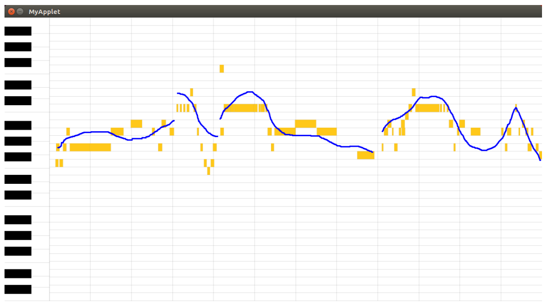

Figure 1: A piano roll with the melodic outline drawn by the user (blue) and the corresponding melody generated by the system (orange)

2.1 Drawing Melodic Outline

During the playback of the accompaniment, the user draws a melodic outline $\{y(t)\}$ on the piano-roll screen. Here, $y(t)$ represents the pitch of the outline at time t. The time resolution is an eighth-note triplet.

2.2 Determining Rhythm of Melody

When the mouse cursor moves out from measure m in the piano roll (i.e. it moves to measure m+1) or the mouse button is released, melody generation for measure m iniciates.

First, the rhythm (i.e., the melody notes' durations) is determined. The key idea is to generate a note onset at time points of high-variability of $y(t)$. This is achieved through the following steps:

1. A set of note onset candidates, $\mathcal{R}$, is defined. Each element of $\mathcal{R}$ is a 12-dimensional binary vector, where 1 stands for an onset and 0 stands for a non-onset. For example, $(1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0)$ represents a sequence of two half notes.
2. We decide a tentative rhythm R' from $\{y(t)\}$. The i-th element of R', denoted by $R'(i)$, is defined as follows:

$$R'(i) = \begin{cases} 1 & (|y(t_m+i) - y(t_m+i-1)| > \delta) \\ 0 & (\text{otherwise}) \end{cases}$$

NIME'17, May 15-19, 2017, Aalborg University Copenhagen, Denmark.

where t_m is the start time of m, and δ is a threshold.

3. We search for the closest candidate to R', that is,

$$\hat{R} = \operatorname*{argmin}_{R_k \in \mathcal{R}} ||R_k - R'||.$$

2.3 Determining Pitches

For each of the 1-value elements in $\hat{R}$, the pitch (MIDI note number) is detemined. Let L be the number of 1-value elements in $\hat{R}$. What should be determined here is $N = (n_0, \cdots, n_{L-1})$, where n_i is a MIDI note number.

To determine these pitches, we use a genetic algorithm (GA), in which $N = (n_0, \cdots, n_{L-1})$ is regarded as a chromosome. To achieve a quick melody creation, the optimization through GA is limited to 0.5 s.

2.3.1 *Initializing chromosomes*

Melodies taken from a melody corpus are memorized as a prefix tree. Initial chromosomes (sequences of note numbers) are generated by randomly proceeding in this prefix tree from the root.

2.3.2 *Calculating fitness function*

The fitness function assigns higher values to melodies which:

- are similar to the melodic outline,
- have similar characteristics to the melodies in the melody corpus,
- are not dissonant with respect to the accompaniment,
- are not too monotonous.

We therefore define the fitness function $F(N)$ as follows:

$$\begin{aligned} F(N) &= w_0 \operatorname{sim}(N) + w_1 \operatorname{seq}_1(N) + w_2 \operatorname{seq}_2(N) \\ &\quad + w_3 \operatorname{harm}(N) + w_4 \operatorname{ent}(N), \end{aligned}$$

where

- $\operatorname{sim}(N)$: Similarity to outline

$$\operatorname{sim}(N) = -\sum_{i=0}^{L-1} (n_i - y(t_i))^2,$$

in which t_i is the onset time of note n_i.

- $\operatorname{seq}_1(N)$: Pitch bigram probability

$$\operatorname{seq}_1(N) = \sum_{i=1}^{L-1} \log P(n_i \mid n_{i-1}).$$

- $\operatorname{seq}_2(N)$: Interval (pitch-motion) bigram probability

$$\operatorname{seq}_2(N) = \sum_{i=2}^{L-1} \log P(n_i - n_{i-1} \mid n_{i-1} - n_{i-2}).$$

- $\operatorname{harm}(N)$: Conditional probability for given chords

$$\operatorname{harm}(N) = \sum_{i=0}^{L-1} \log P(n_i \mid c_i, b_i),$$

in which c_i is the chord name at time t_i, and b_i is the metrical position at t_i ($b_i \in \{\text{head}, \text{on-beat}, \text{off-beat}\}$). We consider b_i because the acceptability of out-of-scale notes depends on their metrical positions.

- $\operatorname{ent}(N)$: Entropy

$$\operatorname{ent}(N) = -(H(N) - H_{\text{mean}} - \varepsilon)^2,$$

in which $H(N)$ is the entropy of $\{n_0, \cdots, n_{L-1}\}$, and H_{mean} is the averaged entropy calculated from a melody corpus. Above, ε is usually zero, but setting this to more than zero will result in more complex melodies.

Above, $P(n_i \mid n_{i-1})$, $P(n_i - n_{i-1} \mid n_{i-1} - n_{i-2})$, $P(n_i \mid c_i, b_i)$, and H_{mean} are learned from a corpus, while $w_0, \cdots, w_4$ and ε are manually set.

2.4 Estimating Expression Parameters

The expression parameters, that is, the onset deviation, duration ratio, and energy (velocity) ratio for each note are estimated with Giraldo's method [5]. Here, we use the k-nearest neighbor estimator. The upper and lower bounds, if necessary, can be given to each parameter.

3. IMPLEMENTATION AND TRIAL

We implemented this system on a touch-screen laptop PC. As a melody corpus, we used 53 melodies with the tonality of Blues taken from Weimar Jazz Database[1]. We used a 12-bar blues chord progression in the key of C, i.e.,

| C7 F7 C7 C7 F7 F7 C7 C7 G7 F7 C7 G7 |.

Preliminary tests of our system have been very positive. Users were satisfied with the usability of the system and with the quality of their improvisations, which clearly contained characteristics of blues melodies, e.g. melodies make use of notes such as $E^\flat$, $G^\flat$, and $B^\flat$ in the key of C. In the future we plan to conduct a formal perceptual test to evaluate the performance of the proposed system.

4. CONCLUSION

In this paper, we proposed an improvisation support system, *JamSketch*, in which the user can enjoy improvisation just by drawing a melodic outline. Melodies are generated in real time based on a genetic algorithm according to an outline drawn by the user and are expressively performed. Future work includes evaluations of both the user experience and the qualities of generated melodies as well as further improvements with a larger-scale melody corpus.

Acknowledgments: This work was supported by JSPS KAKENHI Grant Numbers 26240025, 26280089, 16K16180, 16H01744, and 16KT0136, as well as by the Spanish TIMuL Project (TIN2013-48152-C2-2-R), and the TELMI Project of the Horizon 2020 Research and Innovation Programme (grant agreement No. 688269).

5. REFERENCES

[1] E. Amiot, T. Noll, M. Andretta, and C. Agon. Fourier oracles for computer-aided improvisation. In *Proceedings of International Computer Music Conference*, 2006.

[2] J. Buchholz, E. Lee, J. Klein, and J. Borchers. coJIVE: a system to support collaborative jazz improvisation. In *Technical Report*, 2007.

[3] M. M. Farbood, E. Pasztor, and K. Jennings. Hyperscore: A graphical sketchpad for novice composers. *IEEE Computer Graphics and Applications*, 24(1):50–54, 2004.

[4] J. Garcia, T. Tsandilas, C. Agon, and W. Mackay. Inksplorer: Exploring musical ideas on paper and computer. In *Proceedings of International Conference on New Interfaces for Musical Expression*, 2011.

[5] S. Giraldo and R. Ramírez. A machine learning approach to ornamentation modeling and synthesis in jazz guitar. *Journal of Mathematics and Music*, 10(2):107–126, 2016.

[6] K. Ishida, T. Kitahara, and M. Takeda. ism: Improvisation supporting system based on melody correction. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 177–180, 2004.

[7] D. E. Parson. Chess-based composition and improvisation for non-musicians. In *Proceedings of International Conference on New Interfaces for Musical Expression*, 2009.

[8] J.-B. Thiebaut, P. G. Healey, N. B. Kinns, and Q. Mary. Drawing electroacoustic music. In *Proceedings of International Computer Music Conference*, 2008.

[9] Y. Tsuchiya and T. Kitahara. Melodic outline extraction method for non-note-level melody editing. In *Proceedings of the Sound and Music Computing Conference 2013*, pages 762–767, 2013.

[1]http://jazzomat.hfm-weimar.de/dbformat/dboverview.html