JamSketch Deep α: Towards Musical Improvisation based on Human-machine Collaboration

Tetsuro Kitahara and Akio Yonamine
College of Humanities and Sciences, Nihon University
Setagaya-ku, Tokyo 156-8550, Japan
{kitahara,yonamine}@kthrlab.jp

Abstract

This paper describes an improvisation support system called *JamSketch Deep* α. Its basic concept is to allow users to specify the macro structure of melodies while the micro structure is automatically generated. Once the user draws a melodic outline as the macro structure, the system genrates a melody according to the outline.

1 Introduction

Musical improvisation is indispensable for joining jam sessions, but it is not easy because players have to create melodies while playing. Therefore, there have been developed various systems for supporting unskilled people's improvisation [1, 2, 3, 4, 5, 6].

One promising approach for supporting unskilled people's improvisation is to collaboratively create and play melodies by human players (users) and computers (systems). Specifically, the process of improvisation is divided into several stages and each stage is allocated into the user or system based on its difficulty for the user. In other words, what the user can do is done by the user while what the user cannot do is done by the system.

As an instance of the above-mentioend approaches, we have been developing an improvisation suppport system called *JamSktch Deep* α, in which the macro structure of melodies are speicified by the user while the micro structure is created by the system [7]. During an improvisation, the user draws the marco stucture of a melody (called a *melodic outline*) on the piano-roll screen using the mouse or his/her finger. Once the user draws the melodic outline, the system immediately creates melodies according to the outline and plays it back in real time.

2 System

This system enables the user to darw a melodic outline and creates a melody according the outline. Once this system launches, a piano-roll screen appears and an acconpaniment prepared as a MIDI file is played back repeatedly. The user draws a melodic outline during playing back the accompaniment. The data of each bar's melodic outline with the chord progression are given to our neural-network-based model, then the model generates a melody, which is played back in real time.

Data representation The input of our model is a melodic outline with a chord progression while the output is a melody (a sequence of notes) (Figure 1 left). For simplicity, melody generation is performed independently for each measure. The melodic outline in measure m is transformed to a 16th-note-level sequence of 60-dimensional one-hot vectors representing the pitch at each frame (supported pitch range: C2 to B6). The chord progression is represented as a sequence of 12-dimensional chroma vectors. The output is a 16th-note-level sequence of one-hot vectors that represent the pitch of the generated melody at each frame. To distinguish a note of pitch p starting at time t and a note of pitch p continuing from $t-1$, they are represented as different elements.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

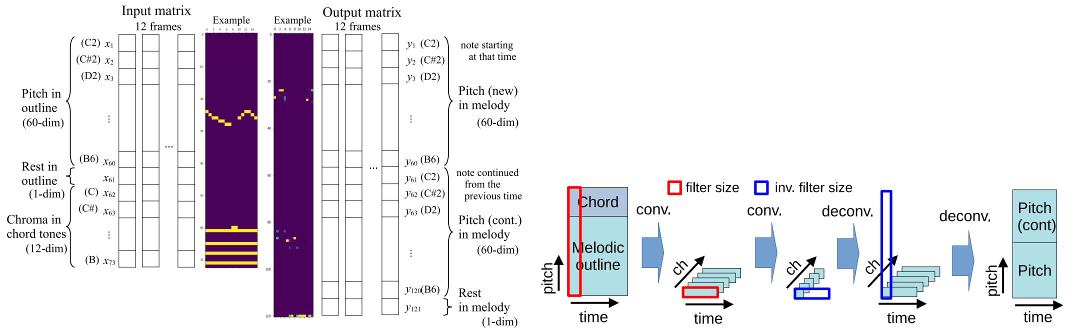

Figure 1: Data representation (left) and model architecture (right)

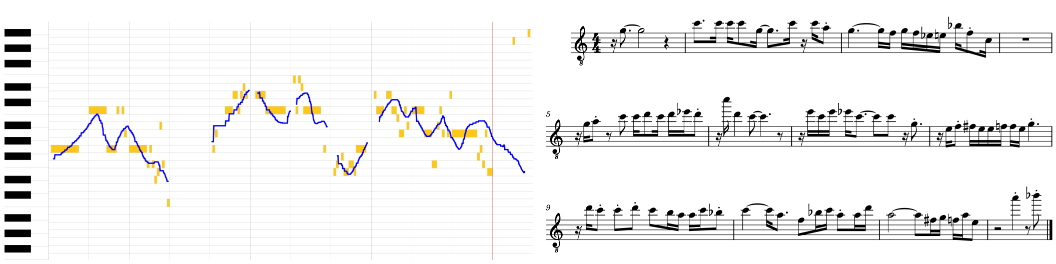

Figure 2: An example of drawn melodic outlines and generated melodies

Model architecture The model is a very simple convolutional neural network (CNN) (Figure 1 right). The first two convolutional layers convert the melodic outline (+ chord) and the following deconvolutional layers convert it to a melody.

3 Preliminary results

We implemented a prototype system, in which the model was trained with pairwise data of melodic outlines and melodies. The melodies were 96 pieces with Tonality Type of BLUES taken from the Weimer Jazz Database. The melodic outline for each melody was generated by smoothing the pitch trajectory of the melody. All melodies were transposed into the C-major or C-minor key and were quantized to 16th-note beats in advance. As an acconpaniment, a 12-bar blues chord progression ($C_7F_7C_7C_7|\ F_7F_7C_7C_7|\ G_7F_7C_7G_7$), the MIDI data of which was prepared in advance.

Figure 2 shows an example of melodic outlines that the first author drew and generated melodies. We can observe the following from this:

- The generated melody follows the meloic outline mostly.
- In the last measure, very high-pitch notes appeared even though the pitch of the melodic outline was not so high. This could be improved by introducing the loss function that evaluates the similarity of the given melodic outline and the generated melody.
- The C (tonic) and G (dominant) notes were used with long durations. Also, the E♭, F♯, and B♭ notes (blue notes) were used. These are appropriate from a musical point of view.
- The user can specify the pitch feature of the melody but cannot specify rhythmic features. This is an important future issue.

4 Conclusion

We proposed an improvisation support system, called *JamSketch Deep* α, which generates a melody according to a melodic outline drawn by the user. Generated melodies are musically appropriate to some extent but we have not confirmed if they reflect users' intention. We would like to continue the development towards collaborative improvisation by human players and computers.

Ethical Implications

This model, which generates melodies from melodic outlines, is trained with a set of melodies improvised by actual musicians. The melodic outline for each melody has been obtained by smoothing the pitch trajectory of the melody. Unless the user inputs the same melodic outline as one of the trained ones, the same melody as those included in the dataset will not be generated. However, there is no guarantee that the user will not draw such melodic outlines when a huger dataset is used.

Presentation

In the workshop, we will demonstrate our system, JamSketch Deep α. Participants can experience improvisation using this system.

Aknowledgments

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

References

[1] Homei Miyashita and Kazushi Nishimoto. Theremoscore: A new-type musical score with temperature sensation. In *Int'l Conf. New Interface for Musical Expression*, pages 104–107, 2004.

[2] Katsuhisa Ishida, Tetsuro Kitahara, and Masayuki Takeda. ism: Improvisation supporting system based on melody correction. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 177–180, June 2004.

[3] Jan Buchholz, Eric Lee, Jonathan Klein, and Jan Borchers. coJIVE: a system to support collaborative jazz improvisation. Technical Report AIB-2007-04, Aachener Informatik-Berichte RWTH Aachen, Department of Computer Science, http://www.informatik.rwth-aachen.de/go/id/lolj/lidx/1/file/47944, 2007.

[4] Emmanuel Amiot, Thomas Noll, Moreno Andretta, and Carlos Agon. Fourier oracles for computer-aided improvisation. In *Proceedings of International Computer Music Conference (ICMC 2006)*, pages 99–103, 2006.

[5] Dale E. Parson. Chess-based composition and improvisation for non-musicians. In *Proceedings of International Conference on New Interfaces for Musical Expression (NIME 2009)*, pages 157–158, 2009.

[6] Chris Donahue, Ian Simon, and Sander Dieleman. Piano Genie. In *Proceedings of the 24th International Conference on Intelligence User Interfaces (IUI 2019)*, pages 160–164, 2019.

[7] Tetsuro Kitahara and Akio Yonamine. JamSketch Deep α: A CNN-based improvisation system in accordance with user's melodic outline drawing. In *Proceedings of the 4th ACM International Conference on Multimedia in Asia (ACM MM Asia 2022)*, 2022.