EXTENDING JAMSKETCH: AN IMPROVISATION SUPPORT SYSTEM

Akane Yasuhara, Junko Fujii, and Tetsuro Kitahara
College of Humanities and Sciences, Nihon University, Japan
yasuhara@kthrlab.jp, fujii@kthrlab.jp, kitahara@kthrlab.jp

ABSTRACT

We previously introduced JamSketch, a system which enabled users to improvise music by drawing a melodic outline. However, users could not control the rhythm and intensity of the generated melody. Here, we present extensions to JamSketch to enable rhythm and intensity control.

1. INTRODUCTION

Improvisation is an enjoyable but difficult form of music performance because musicians must create melodies while playing an instrument. Therefore, to enable non-musicians to improvise easily, various systems have been proposed [1–5]. For example, JamSketch [5] enables users to play an improvisation by drawing a curve called a *melodic outline*, which represents the overall shape of a melody, with a mouse or their finger on a piano-roll display. This approach does not require skill in playing an instrument, but the melodic outline is limited in its expressivity because it only represents how the melody moves up and down in pitch. Until now, users could not control the rhythm or intensity of the melody.

In this paper, we extend JamSketch to enable users to control the rhythm and intensity of the generated melody when drawing a melodic outline. Our intension is to add functionality while not making the system more complex or less intuitive to use. To satisfy these requirements, we adopt the following approaches:

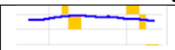
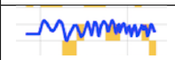
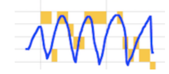
Rhythm: We support only the control of note density (how many notes appear within one bar) to keep the operation simple. Users can control the note density by changing the waviness of the melodic outline.

Intensity: Through the use of devices supporting pen pressure sensing (e.g., Microsoft Surface Pro), the system allows users to control the intensity by changing the pen pressure.

2. SYSTEM

Once the system launches, the piano-roll display with its horizontal time axis and vertical pitch axis appears. The user draws a melodic outline with a stylus pen supporting pen pressure sensing. The shape of the melodic outline is reflected in the pitch of the generated melody, and the pen pressure is reflected in the intensity of the performed melody. Also, the rhythm (note density) is controlled by drawing wavy curves.

Table 1. Waviness and note density

$0 \leq s \leq 10$	no-wave		$D = 6$
$10 < s \leq 50$	small-wave		$D = 12$
$s > 50$	large-wave		$D = 2$

2.1 Drawing a melodic outline

On the piano-roll display, the user draws a melodic outline. The melodic outline is displayed, and the line weight represents the pen pressure when the outline is drawn.

2.2 Analyzing the waviness of the melodic outline

Once the user draws a melodic outline, the system analyzes the waviness of the outline separately for each bar. Let $y(t)$ ($t = 1, \cdots, T$) be the melodic outline at a certain bar (the resolution of t depends on the screen's resolution). The system calculates a smoothed outline $\bar{y}(t) = \frac{1}{\tau_0}\sum_{\tau=0}^{\tau_0-1} y(t+\tau)$. Then, $\delta(t) = \bar{y}(t) - y(t)$ is calculated. After that,

$$s = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T-1}(\delta(t+1) - \delta(t))^2}$$

is calculated. The waviness at the corresponding bar is determined as follows (Table 1):

$$\text{waviness} = \begin{cases} \text{“no-wave”} & (0 \leq s \leq 10) \\ \text{“small-wave”} & (10 < s \leq 50) \\ \text{“large-wave”} & (s > 50) \end{cases}$$

2.3 Determining the rhythm

The rhythm is determined separately for each bar and is represented as a binary vector where 1 stands for an onset and 0 stands for a non-onset. Because the shortest duration is set to an eighth-note triplet in the current implementation, the rhythm at each bar is represented as a 12-dimensional binary vector $R = (r_0, \cdots, r_{11})$ ($r_i \in \{0, 1\}$).

First, a tentative rhythm $R' = (r'_0, \cdots, r'_{11})$ is generated from the melodic outline. The basic policy is to generate a note onset at time points when the melodic outline has a high gradient. Let $y'(i)$ be a down-sampled melodic outline (the time resolution is an eighth-note triplet). Then,

$$r'_i = \begin{cases} 1 & (|y'(i) - y'(i-1)| > \epsilon) \\ 0 & \text{(otherwise)} \end{cases}$$

is calculated, where ϵ is a threshold.

Then, the rhythm R is determined with a genetic algorithm (GA) with the fitness function defined as follows:

$$F(R) = w_0 \,\mathrm{sim}(R) + w_1 \,\mathrm{lik}(R) + w_2 \,\mathrm{dens}(R)$$

where

- $\mathrm{sim}(R)$ is the similarity to the tentative rhythm R':
$$\mathrm{sim}(R) = -\sum_{i=0}^{11} (r_i - r'_i)^2.$$
- $\mathrm{lik}(R)$ represents the musical liklihood of R:
$$\mathrm{lik}(R) = \sum_{i=0}^{11} \log P(r_i|i),$$
where $P(r_i|i)$ is the conditional probability of r_i given the time index i and is calculated from a dataset.
- $\mathrm{dens}(R)$ represents how well the melody's rhythm follows the waviness of the melodic outline:
$$\mathrm{dens}(R) = -\left(D - \sum_{i=0}^{11} r_i \right)^2,$$
where D is a preferred note density determined from the waviness of the melodic outline. In the current implementation, D is set according to Table 1.

2.4 Determining the pitch

After the rhythm of the melody is determined, the pitch (note number) of each note is determined. This is also based on a genetic algorithm with a fitness function that tries to maximize both the melody's closeness to the melodic outline and its musical likelihood calculated from a melody dataset. See [5] for details.

2.5 Determining the velocity

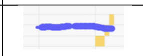
The velocity of each note is determined according to the pen pressure at the corresponding point in the melodic outline. The mapping between the pen pressure and the velocity in the current implementation is listed in Table 2.

3. EXAMPLES

An example of melody generation including different note densities is shown in Fig. 1 (left). From the 2nd to 5th measures, the melodic outline contains a small wave, and accordingly, the generated melody contains many short notes. From the 8th to 12th measures, the generated melody consists of a fewer longer notes because the melodic outline contains a large wave.

An example of controlling the velocity is shown in Fig. 1 (right). From the 4th to 5th measures, the velocity is high because the pen pressure is high (so, the curve is thick).

Table 2. Pen pressure and velocity

Pen pressure	Display	Velocity
3000 or higher		127
2000 to 3000		80
1000 to 2000		50
lower than 1000		30

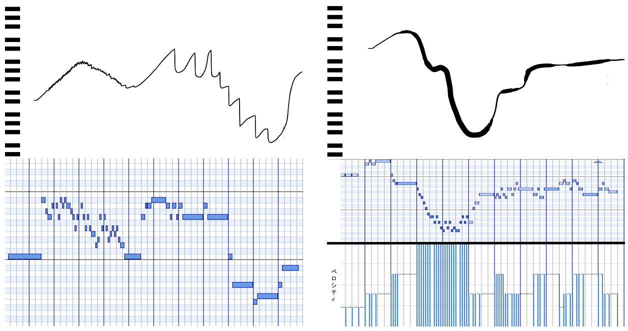


Figure 1. Examples of melodic outlines (upper) and generated melodies (lower) with rhythm (left) and intensity (right) control. The piano-roll representation includes the velocity data at the right-side figure.

4. CONCLUSION

In this paper, we presented two extensions (rhythm and intensity control) of the JamSketch system, which supports improvisation by non-musicians. We conducted experiments to confirm the effectiveness of this system. We omitted the results due to a lack of space, but we will report them in a separate paper.

Acknowledgments: This project is supported by JSPS Kakenhi JP16K16180, JP16H01744, JP16KT0136, and JP17H00749 as well as the Kawai Foundation for Sound Technology and Music.

5. REFERENCES

[1] S. Fels, K. Nishimoto, and K. Mase, "MusiKalscope: A graphical musical instrument," *IEEE Multimedia*, vol. 5, no. 3, pp. 26–35, 1998.

[2] H. Miyashita and K. Nishimoto, "Theremoscore: A new-type musical score with temperature sensation," in *Int'l Conf. New Interface for Musical Expression*, 2004, pp. 104–107.

[3] 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*, June 2004, pp. 177–180.

[4] J. Buchholz, E. Lee, J. Klein, and J. Borchers, "coJIVE: a system to support collaborative jazz improvisation," Aachener Informatik-Berichte RWTH Aachen, Department of Computer Science, http://www.informatik.rwth-aachen.de/go/id/lolj/lidx/1/file/47944, Tech. Rep. AIB-2007-04, 2007.

[5] T. Kitahara, S. Giraldo, and R. Ramírez, "Jamsketch: Improvisation support system with ga-based melody creation from user's drawing," in *Proc. of Int'l Symp. on Computer Music Multidisciplinary Research*, 2017, pp. 352–363.