A Non-notewise Melody Editing Method for Supporting Musically Untrained People's Music Composition

Yuichi Tsuchiya[1] and Tetsuro Kitahara[1] *

College of Humanities and Sciences, Nihon University, Japan
kitahara@kthrlab.jp

Abstract. There have been many systems developed that automatically create a melody. However, when the created melody is not satisfactory, it is difficult for musically untrained people to edit it manually on a conventional MIDI sequencer. Therefore, we propose a melody editing method based on a *melodic outline*, which represents the overall shape of a melody. Given a melody, its melodic outline is obtained by applying the Fourier transform to the melody's pitch trajectory and extracting low-order Fourier coefficients. After the outline is redrawn, it is transformed into a note sequence by the inverse procedure of the extraction and a hidden Markov model. Experimental results showed that (1) for novice participants, our system was easier than the conventional piano-roll interface; (2) generated melodies were satisfactory for both novice and intermediate participants; and (3) novice participants' ideas about what melody they want became clearer as they experienced melody editing every day.

Keywords: Melody editing, computer-aided music composition, melodic outline, Fourier transform, hidden Markov model

1 Introduction

Automatic music composition systems (Hiller & Isaacson, 1958; Cope, 1991; Ames & Domino, 1992; Biles, 1994; Ando, Dahlstedt, Nordahl, & Iba, 2006; Fukayama, Nakatsuma, Sako, Nishimoto, & Sagayama, 2010) provide users with original music without requiring them to perform musically difficult operations. These systems automatically generate melodies and backing tracks based on the user's input such as style parameters. In some cases, however, the generated pieces do not completely match those desired or expected by users because it

This paper is an extended edition of the papers presented at SMC 2013 (Tsuchiya & Kitahara, 2013) and ICMC/SMC 2014 (Kitahara & Tsuchiya, 2014).

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

is difficult to express the desire as style parameters. The common approach for solving this problem is to edit the generated pieces manually with a MIDI sequencer, but it is not an easy operation for musically untrained people.

The goal of this study is to achieve an environment that enables musically untrained users to explore satisfactory melodies by repeated trial-and-error editing of melodies generated by automatic music composition systems. There are two reasons why it is difficult for musically untrained people to use a conventional MIDI sequencer. The first reason is that such people are likely to understand music without mentally representing it as a sequence of notes (Goto, 2004). The melody representation for melody editing should therefore not be based on musical notes; it should instead capture the coarse structure of the melody that an untrained person would recognise. The second reason is that it is difficult for untrained people to avoid dissonant notes in a MIDI sequencer. A certain support is therefore needed to avoid such notes using a computing technology.

Here, we propose a new sub-symbolic melody representation called a *melodic outline*. The melodic outline represents only the coarse temporal characteristics of the melody; the notewise information from the melody is hidden. Once a melody is specified in the MIDI format, its melodic outline is automatically obtained and displayed on a piano-roll interface. The user can directly redraw this melodic outline in part, and for that redrawn part, a new melody is automatically generated, avoiding notes dissonant against the accompaniment.

The use of such drawing or similar approaches has been attempted so far. *Hyperscore* is a graphical music composition tool, in which different colors are assigned to different motives and the user can compose by drawing lines with various colors (Farbood, Pasztor, & Jennings, 2004). *Hand Composer* recognises a gesture by the user using Leap Motion and creates music based on the gesture (Mandanici & Canazza, 2014). *InkSplorer* is a tool for composing music in a contemporary style based on hand-written gestures (Garcia, Tsandilas, Agon, & Mackay, 2011). *Music Sketcher* is also a drawing-based user interface for electroacoustic music (Thiebaut, Healey, & Kinns, 2008). *JamSketch* enables users to improvise by drawing melodic outlines (Kitahara, Giraldo, & Ramírez, 2017). *UPIC*, developed by Xenakis, is also a music composition tool based on drawing, in which the user draws on a built-in board and the system interprets the drawing as dynamic envelopes, scores in the time-pitch domain, or other musical parameters (Xenakis, 1985). However, these studies do not deal with the mutual transform between a melody (a sequence of notes) and its melodic outline. On the other hand, our system is capable of transforming a given melody to a melodic outline, allowing the user to edit this melodic outline, and transforming again the edited outline back to a sequence of notes.

The rest of this article is organised as follows. In Section 2, we describe the concept of the melodic outline. In Section 3, we present our melody editing system as well as a method for extracting a melodic outline from a sequence of notes using the Fourier transform, and a method for generating a sequence of notes from the melodic outline redrawn by the user. In Section 4, we report the results of a short-term experiment, which was conducted to compare the usability of our melody editing interface with a conventional piano-roll interface. We hypothesised that novice participants (those without the experience of music composition) would favour our interface, although participants who had composed with a MIDI sequencer would not. Our results imply that this hypothesis is true. In the same section, we also show the results of a long-term experiment, which was conducted to investigate how the behaviour of musically untrained participants' melody editing changes over time. We here illustrate that the desire of melody editing (i.e., which notes users want to edit and how) becomes more specific over time. In Section 5, we introduce a system in which the user can edit melodies using our melody editing interface and a conventional piano-roll interface seamlessly. We also report the results of an experiment for confirming the usability of this system. Finally, we conclude the article in Section 6.

2 Basic Concept of Melodic Outline

A melodic outline is a melody representation in which the melody is shown as a continuous curve. An example is shown in Fig. 1. A melodic outline is mainly used for editing a melody with a three-step process: (1) the target melody represented as a sequence of notes is automatically transformed into a melodic outline; (2) the melodic outline is redrawn by the user; and (3) the redrawn outline is transformed into a note sequence. The key technology for achieving this is the mutual transform of a note-level melody representation and a melodic outline. We believe that this mutual transform should satisfy the following requirements:

1. A melodic outline does not explicitly represent the pitch and note value of each note.
2. When a melodic outline is inversely transformed into a note sequence without any editing, the result should be equivalent to the original melody.
3. When a melodic outline edited by the user is transformed into a note sequence, musically inappropriate notes (e.g., notes causing dissonance against the accompaniment) should be avoided.

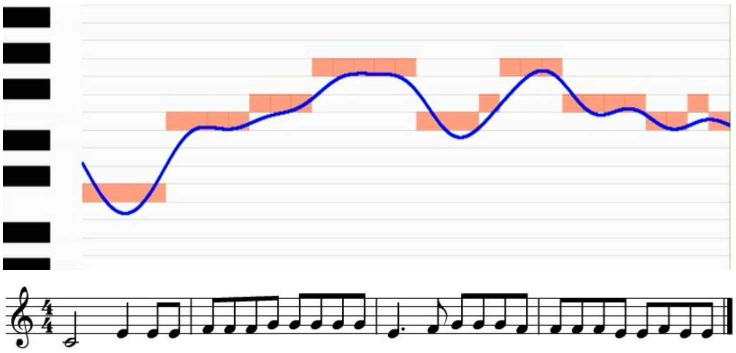

Figure 1. Example of a melodic outline. The curve superimposed on the piano-roll interface is the melodic outline.

No previous studies have proposed melody representations satisfying all these three requirements. Various methods for transforming a melody to a lower-resolution representation have been proposed (Marlot, 2008), but these representations are designed for melody matching in query-by-humming music retrieval, so they cannot be inversely transformed into a sequence of notes. *OrpheusBB* (Kitahara, Fukayama, Katayose, Sagayama, & Nagata, 2011) is a human-in-the-loop music composition system, which enables users to edit automatically generated content when it does not satisfy their desire. When the user edits some part of the content, this system automatically regenerates the remaining part, but the editing is performed at the note level.

The flow of the melody editing is shown in Fig. 2. The method supposes that the user composes a melody with an automatic music composition system. The melody is transformed into a melodic outline with the method described in Section 3.2. The user can freely redraw the melodic outline as described in Section 3.3. Then, the melodic outline is inversely transformed into a note sequence using the method described in Section 3.4. If the user is not satisfied with the result, the user again edits the melodic outline. The user can repeat the editing process until a satisfactory melody is obtained.

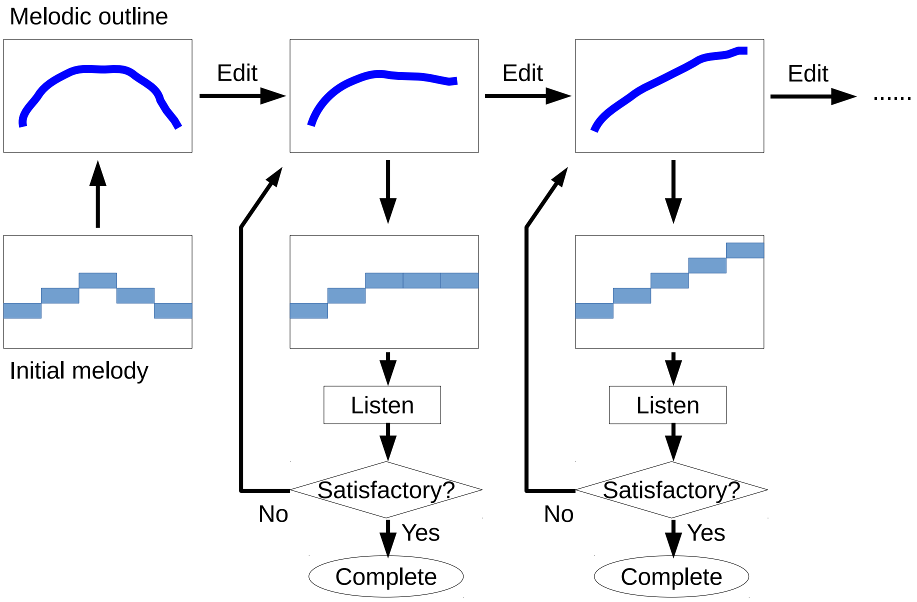

Figure 2. Flow of melody editing.

3 Melody Editing System based on Mutual Transform of Melodic Outline and Note Sequence

We here describe a system that enables users to edit melodies using melodic outlines. Our melody editing system consists of three steps: (1) transform of a note sequence into a melodic outline; (2) editing of the melodic outline; and (3) inverse transform of the edited melodic outline back into a note sequence.

3.1 System Overview

In our system, the original melody is assumed to be an output of *Orpheus* (Fukayama et al., 2010). *Orpheus* is a web-based automatic music composition system in which users can input Japanese lyrics of musical pieces that they want. Once the user inputs the lyrics and some parameters in a web browser, the system searches for the melody that has the closest pitch motion to the prosody of the given lyrics based on a probabilistic model with manually tuned probability parameters. After the user creates a melody using *Orpheus*, they input the melody's ID, given by *Orpheus*, into our system. Then, our system obtains a MIDI file from the *Orpheus* web server, and displays the melody both in a note-level representation and as a melodic outline (Fig. 3 (a)). Once the user redraws the melodic outline, the system immediately regenerates the melody and updates the display. If the user is not satisfied having listened to the regenerated melody, they can redraw the melodic outline repeatedly until a satisfactory melody is obtained. Fig. 3 (b) shows a screenshot of a melody editor based on the conventional piano-roll interface, which was used for comparison with our system in experiments.

3.2 Transform of a Note Sequence into a Melodic Outline

The given note sequence of a melody (Fig. 4 (a)) is transformed into a pitch trajectory (Fig. 4 (b)). The pitch is represented logarithmically, where middle C is 60.0 and a semitone is represented by 1.0. (The difference from note numbers is that non-integer values are acceptable.) Regarding the pitch trajectory as a periodic signal, the Fourier transform is applied to this trajectory. Note that the input to the Fourier transform is not an audio signal, so the result does not represent a sound spectrum. Because the Fourier transform is applied to the pitch trajectory of a melody, the result represents the feature of temporal motion in the melody. Low-order Fourier coefficients represent slow motion in the melody while high-order Fourier coefficients represent fast motion. By extracting low-order Fourier coefficients and applying the inverse Fourier transform to them, a rough pitch contour of the melody, i.e., the melodic outline, is obtained (Fig. 4 (c)).

The *fineness* of the melodic outline can be controlled by changing how many Fourier coefficients are extracted (Fig. 5).

3.3 Editing of the Melodic Outline by the User

Once the melody is transformed to a melodic outline, the user can edit the melodic outline by redrawing it with the mouse. Immediately after the user redraws the melodic outline, a new melody is generated by means of the method described in Section 3.4.

3.4 Inverse Transform of a Melodic Outline into a Note Sequence

Once part of the melodic outline is redrawn, the redrawn outline is transformed back into a note sequence. For simplicity, in the current implementation, the rhythm of the note sequence is assumed to be same as in the original note sequence; only the pitch for each note is changed via the editing. An overview of the procedure of the transform is shown in Fig. 6.

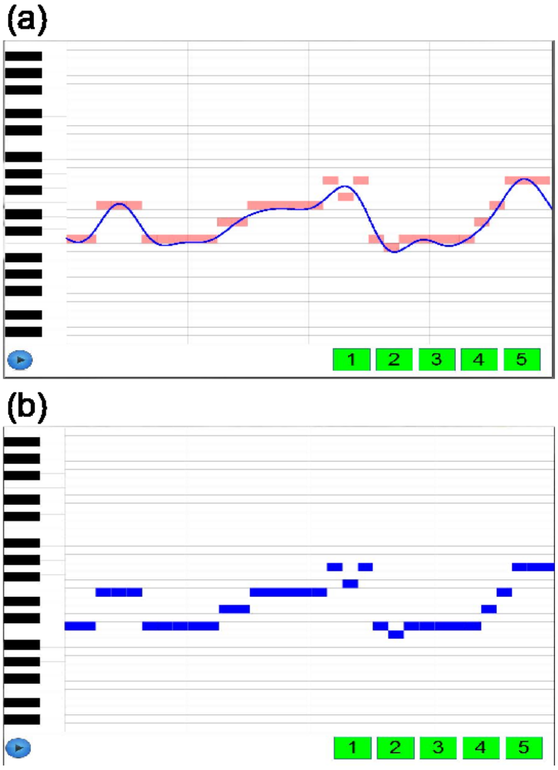

Figure 3. The user interface for the melody editing system. (a) Melody editing interface based on a melodic outline (our system); (b) Conventional piano-roll interface. For both interfaces, buttons labelled 1 to 5 were installed for recording the user's ratings of the generated melody and were used in the experiments. Note that these buttons are used only for recording the user's ratings and thus have no effect on the melody generation process.

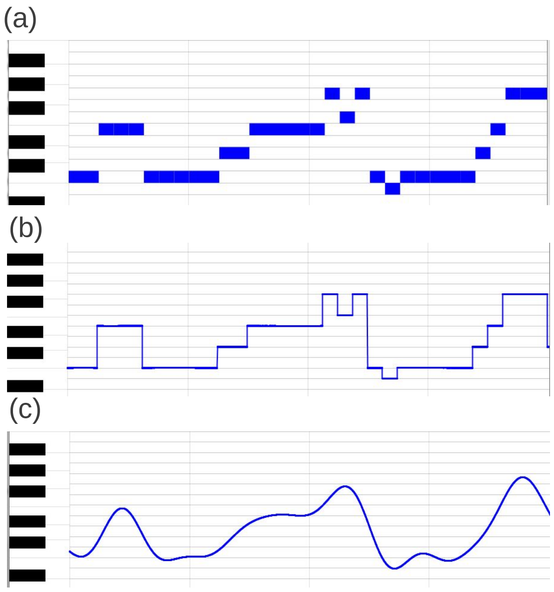

Figure 4. Flow of extracting melodic outline. (a) Note sequence of melody; (b) Pitch trajectory of melody; (c) Melodic outline.

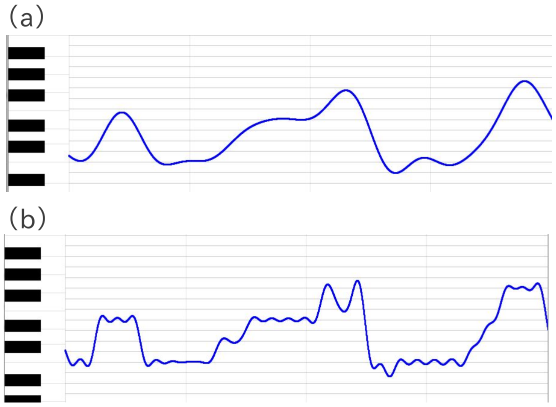

Figure 5. Examples of melodic outline with different settings of fineness.

First, the Fourier transform is applied to the redrawn outline (Fig. 6 (a)). Then, the higher-order Fourier coefficients of the original pitch trajectory, which had been removed when the melodic outline was extracted, are added to the Fourier coefficients of the redrawn outline to generate the same pitch trajectory as the original melody from the non-redrawn part of the melodic outline. Next, the inverse Fourier transform is applied, producing the post-edit pitch trajectory (Fig. 6 (b)).

Next, the pitch trajectory is transformed into a note sequence. In this process, notes that cause dissonance with the accompaniment are avoided, which is achieved using a hidden Markov model. The HMM used here is shown in Fig. 7. This model is formulated based on the idea that the observed pitch trajectory $O = o_1 o_2 \cdots o_N$ is emitted with random deviation from a hidden sequence of note numbers $H = h_1 h_2 \cdots h_N$ that does not cause dissonance.

The HMM consists of hidden states $\{s_i\}_{i_\mathrm{L} \leq i \leq i_\mathrm{H}}$, each of which corresponds to the note number i (therefore, each h_n takes an element of $\{s_i\}$). Each state s_i emits a value for pitch following a normal distribution $N(i, \sigma^2)$. For example, the state s_{60}, corresponding to the note number 60, follows the normal distri-

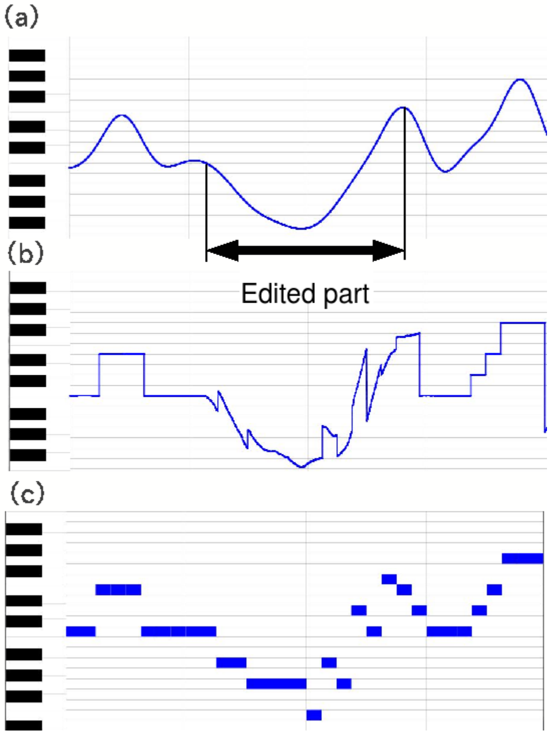

Figure 6. Flow of generating note sequence from melodic outline. (a) Redrawn melodic outline; (b) Pitch trajectory obtained with the Fourier transform; (c) Note sequence generated with the HMM.

bution with a mean of 60.0 and a variance of σ^2. The variance σ^2 is common among all states and is experimentally determined; it is set to 13.0 in the current implementation. In the current implementation, 36 states from s_{48} to s_{84} are used.

The transition probability $P(s_j|s_i)$ is determined as follows:

$$P(s_j|s_i) = p_1(j)\ p_2(|j-i|), \tag{1}$$

where $p_1(j)$ is the probability that a note number j appears in the target key (C major in the current implementation). This is defined based on the idea of avoiding non-diatonic notes as follows:

$$p_1(j) = \begin{cases} 16/45 & \text{(C)} \\ 2/45 & \text{(D)} \\ 8/45 & \text{(E)} \\ 3/45 & \text{(F, A)} \\ 12/45 & \text{(G)} \\ 1/45 & \text{(B)} \\ 0 & \text{(Non-diatonic notes)} \end{cases} \tag{2}$$

In addition, $p_2(x)$ is the probability that a pitch interval x (that is, an interval between note numbers i and j) appears as an interval of two successive notes.

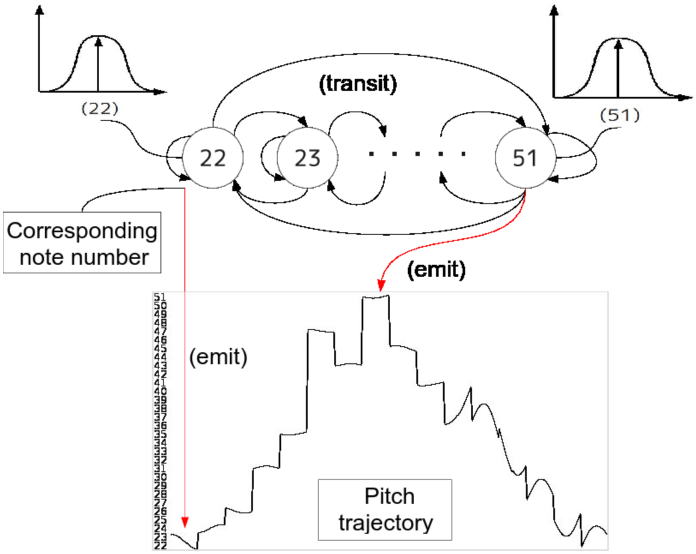

Figure 7. Overview of HMM for estimating note sequence from post-edit pitch trajectory.

This probability is defined as follows:

$$p_2(x) = \begin{cases} 1/63 & \text{(Augmented fourth,} \\ & \text{Diminished fifth} \\ & \text{Major sixth, Minor seventh)} \\ & \text{Major seventh)} \\ 2/63 & \text{(Perfect unison}^1\text{)} \\ 4/63 & \text{(Minor sixth)} \\ 6/63 & \text{(Perfect fourth, Perfect fifth)} \\ 10/63 & \text{(Minor second, Major second,} \\ & \text{Minor third, Major third)} \end{cases} \tag{3}$$

Because the rhythm of the generated melody is assumed to be same as that of the original melody, the state transitions occur only at each note boundary. As described above, the transition probabilities are manually determined so that non-diatonic notes in the C major scale are avoided. However, the transition probabilities can be learned using a melody corpus. If the transition probabilities are learned with melodies of a particular genre (e.g., jazz), they would reflect the melodic characteristics of that genre.

By using the Viterbi algorithm on this HMM, we obtain a sequence of note numbers $H = h_1 h_2 \cdots h_N$ (which would not contain dissonant notes) from the pitch trajectory $O = o_1 o_2 \cdots o_N$. Finally, the result is output in the MIDI format.

4 Experiments

We conducted both short-term and long-term experiments.

4.1 Short-term Experiment

The short-term experiment aimed to compare the usability of our melody editing method with the piano-roll interface. We hypothesised that our melody editing method would be more useful for musically untrained people (such as those who have not composed using a conventional MIDI sequencer) than the piano-roll interface, because our method does not require the user to consider consonance with the backing harmony when selecting notes for the melody. On the other hand, we hypothesise that, for users who have composed using a conventional MIDI sequencer, our melody editing method will *not* be more useful, because they can manipulate each note directly with the piano-roll interface.

Experimental Conditions We asked the participants to edit a melody using both our system and the piano-roll interface. The melody was prepared by providing the lyrics "*Boku ga makura wo shite neru toki mo / Ohisama wa nezu*

[1] Here, *perfect unison* includes the octave as well as the same pitch.

ni guruguru to / Chikyu no mawari wo mawatte wa / Asa mata asa wo koshiraeru"[2], an excerpt of a poem entitled *Ohisama no tabi*, taken from *Sekai Doyo Shu* (a collection of Japanese translations of children's songs of the world) (Saijo & Mizutani, 1991), set to *Orpheus.* The participants were 12 students (nine males and three females). The musical experience of the participants is listed in Tab. 1.

Table 1. Musical experience of participants in the short-term experiment (A–L represent the label of each participant).

	Performance	Composition	Group
A	Electone, 10 years	Yes	Intermediate (INT)
B	Piano, 16 years	Yes	
C	Piano, 9 years	Yes	
D	Piano, 7 years	No	
E	Piano, 7 years	No	
F	Piano, 2 years	No	Novice (NOV)
G	No	No	
H	No	No	
I	No	No	
J	No	No	
K	No	No	
L	No	No	

Table 2. Results of short-term experiment (*No answer).

Participants		Our system					Pianoroll		
		Q1	Q2	Q3	Q4	Editing time	Q1	Q2	Editing time
Intermediate (INT)	A	6	3	2	6	3 min 00 s	6	6	1 min 40 s
	B	6	7	7	7	2 min 00 s	3	1	4 min 00 s
	C	6	3	5	6	6 min 00 s	6	2	8 min 30 s
	D	3	3	5	5	6 min 00 s	5	6	5 min 00 s
	E	6	2	5	6	4 min 40 s	5	5	5 min 30 s
	Median	6	3	5	6	—	5	5	—
	Mean	5.4	3.6	4.8	6.0	—	5.0	4.0	—
Novice (NOV)	F	6	7	6	6	3 min 50 s	6	2	8 min 00 s
	G	1	2	*	3	8 min 10 s	7	2	9 min 00 s
	H	5	5	6	5	10 min 45 s	2	2	14 min 00 s
	I	6	6	6	7	4 min 50 s	6	6	9 min 20 s
	J	6	3	5	5	10 min 30 s	5	6	7 min 40 s
	K	6	6	5	5	4 min 10 s	6	2	7 min 00 s
	L	7	7	1	6	5 min 30 s	3	4	6 min 20 s
	Median	6	6	6	5	—	6	2	—
	Mean	5.3	5.1	4.8	5.3	—	5.0	3.4	—

[2] This is a Japanese translation of "The sun is not a-bed, when I / At night upon my Pillow Lie; / Still round the earth his way he takes, / And morning after morning makes" from *The Sun's Travels* written by Robert Louis Stevenson.

First, the participants practised with both systems. With those participants who belong to the department of computer science,[3] we spent ten minutes explaining how to use both systems and then allowed free practise for three minutes. With the remaining participants, we spent 15 minutes explaining how to use both systems and then allowed free practise for five minutes.

In the next phase, we gave the participants instructions on how they should edit a given melody, namely: "make notes from the second to the third bars gradually lower in pitch, then make notes in the third bar gradually raise." The participants used our system to edit the given melody according to these instructions, and then answered the following questions:

Q1 Were you satisfied with the output melody?
Q2 Was editing easy?
Q3 Did the generated melody match what you expected when drawing the outline?
Q4 Do you think that you were able to draw a melodic outline as instructed?

The answers to these questions were given using a scale of one to seven. The time for editing was not limited; participants were allowed to edit the melody until satisfied.

In the final phase, the participants used the piano-roll interface to edit the melody according to the same instructions, and then answered Q1 and Q2.

Experimental Results The experimental results are listed in Tab. 2. The results are summarised as follows (below, the intermediate and novice groups are abbreviated to INT and NOV, respectively):

Q1 Most participants in both INT and NOV expressed high satisfaction with the generated melodies for both systems. On the scale of 1 to 7, the median for our system was 6 in both INT and NOV, and the mean was 5.4 in INT and 5.3 in NOV. For the piano-roll interface, the median was 5 in INT and 6 in NOV, and the mean was 5.0 in both INT and NOV. Note that Participants A to C, who have experience in composition, also rated highly (6 on the scale of 1 to 7) the melodies generated by our system.

Q2 In NOV, the ease of our system (mean: 5.1) was rated as superior to using the piano-roll interface (mean: 3.4). On the other hand, INT rated the ease of using the piano-roll interface (mean: 4.0) superior to our system (mean: 3.6). Participant A rated our system low because this participant had experience using the piano-roll interface built into a commercial MIDI sequencer. Participants D and E gave our system a low rating. Because these participants have extensive (i.e., 7 years) experience in piano performance, they were considered to have ample skill at understanding a melody as a note sequence. This could explain why they were comfortable with directly manipulating notes, even though they had no experience with composition. In contrast,

[3] This category is not related to the participant grouping (INT and NOV) based on musical experience.

participants in NOV gave the piano-roll interface a low rating. This was because they had difficulty understanding a melody as a sequence of notes. In the interview, some of these participants commented that they could not imagine how a given melody could be edited to obtain their desired melody using the piano-roll interface.

Q3 Most ratings for this question were higher than the mid-point of 7 (median: 5 in INT and 6 in NOV; mean: 4.8 in both INT and NOV). This result implies that our melody generating method can, to some extent, accurately generate expected melodies from given melodic outlines.

Q4 Most ratings for this question were also higher than the middle score (median: 6 in INT and 5 in NOV; mean: 6.0 in INT and 5.3 in NOV). This result shows that melodic outlines are a melody representation that is usable for musically untrained users to express their desire for melody generation.

Editing time The editing time was shorter with our system in most participants in both groups. This is because (1) users do not have to repeat trial-and-error to avoid musically inappropriate notes since such notes are automatically avoided; and (2) users can edit multiple notes with a single short operation.

Future Issues Revealed through the Experiment One of the most important issues to consider in the future is to enable users to adjust the parameters for melody generation. In particular, the degree of appearance of non-diatonic notes, which is defined by the state transition probabilities in the HMM, is an important parameter. An intuitive graphical user interface for controlling such a parameter will widen the variability of generated melodies. In fact, Participant B commented that he did not obtain a desired melody although he tried editing repeatedly. This comment implies a demand for greater ability to adjust parameters. Moreover, there was a comment that it was not easy to draw a melodic outline with a mouse. This problem can be solved by using a tablet PC.

4.2 Long-term Experiment

We conducted a long-term experiment to observe how the opinions of musically untrained people changed during repeated sessions with melody editing. We were particularly interested in changes in participants' standard of satisfaction with melodies, their desire for melody generation, and their thoughts while editing melodies.

Experimental Conditions The experiment was divided into practise and test phases:

Practise Participants were required to practice melody editing using a tablet PC every day.[4] This practise could be done anywhere and at any time.

[4] Although we used a PC in the short-term experiment, we used a tablet PC here to decrease the participants' burdens because they were requested to use our system every day.

Test Once a week, participants were required to come to our laboratory and edit a specified melody using the same tablet PC. During editing, the participants' screen was captured and recorded as a video. Afterward, the experimenter interviewed participants individually while watching the video together.

The experiment was conducted for one month. Only our system (Fig. 3 (a)) was used; no comparison with the piano-roll interface was conducted. The participants for this experiment were three students (age: 22–24), none of whom had experience playing an instrument or composing.

The instructions for each phase were as follows:

Practise Procedures during the practise phase are as follows:

1) Listen to three melodies. (The three melodies were prepared in advance using *Orpheus* with the same lyrics and different parameters (harmony/rhythm). The lyrics were taken from *Sekai Doyo Shu* at random.)
2) Choose two of the three melodies to edit.
3) Edit each of the chosen melodies as instructed below:
   - 3-1) Launch our system. A text box will appear.
   - 3-2) In the text box, input your intention for melody editing (i.e., the characteristics of the melody desired after editing); use natural language (specifically Japanese). (This input is regarded as a tentative goal of melody editing, but the actual goal can change during the editing.)
   - 3-3) Start editing. (During editing, the participant can play back the current melody any time. After listening to the current melody, the participant is asked to click one of the rating buttons 1 to 5. These rating buttons are used only to express satisfaction/dissatisfaction with the result of editing, and have no effect on the melody generation process.)
   - 3-4) Finish editing when you are satisfied with the generated melody.

Test Procedures during the test phase are same as during the practise phase. In this phase, after completing Steps 1 to 3, the experimenter interviewed each participant while watching the video of the screen captured during melody editing. Participants were particularly asked to articulate their thoughts while editing and their feelings when listening to the generated melodies. Participants were also asked the following questions:

Q1 Were you satisfied with the output melody?
Q2 Did the generated melody match what you expected when drawing the outline?
Q3 Was editing easy?

The answers to these questions were given using a scale of one to seven.

Experimental Results The ratings by the participants and average editing time for each week are listed in Tab. 3.

In the last week of the experiment, all participants rated all questions equal to or higher than in the first week. These findings show that the participants became familiar with our system and learned how to achieve melodies they liked with our system.

However, the results in the third week for Participants A and C and in the second week for Participant B were lower than in the previous weeks. This is because they achieved a stronger sense of what they wanted as they became familiar with melody editing. In fact, the participants often edited specific notes again and again, and said in the interview that they wanted to achieve a particular melody. This implies that the experience of melody editing with our system arouses musically untrained people to explore melodies that they want and makes their desire to create certain melodies clearer.

When asked whether anything changed during the one-month period, Participant A answered that he learned to anticipate the melody simply by seeing the melodic outline on the screen and to expect what melody would be generated when drawing changes in the outline. These results imply that our system is useful in improving the musical ability of untrained people, as well as supporting the development of such people's compositional skills.

Future Issues Revealed through the Experiment The most salient comment in the interviews was the demand to be able to edit specific notes. Because such comments were few in the first week, this demand was clearly enhanced by the experience of editing melodies at the outline level. In addition, there was a comment that the outline-level editing was suitable as the first step, but the participant came to want note-level editing as the experiment progressed. These comments show that the best solution will be an integration of outline-level and note-level editing strategies. We therefore have developed a prototype

Table 3. Results of the long-term experiment (Participants A–C are different participants from those in the short-term experiment).

		Q1	Q2	Q3	Avg. editing time
A	1st week	6	6	6	3min30s
	2nd week	7	7	6	3min20s
	3rd week	5	5	5	3min00s
	4th week	6	6	7	2min40s
B	1st week	5	5	6	6min40s
	2nd week	3	4	3	4min40s
	3rd week	6	5	6	3min40s
	4th week	6	5	7	3min30s
C	1st week	6	5	3	5min30s
	2nd week	5	6	6	7min30s
	3rd week	6	5	5	4min10s
	4th week	7	6	7	4min50s

system where users can seamlessly switch between the two editing strategies. We introduce this system in the next section.

Sometimes the generated melodies did not match the melodic outlines that participants drew. This is caused by constraints in the Viterbi search of the HMM. In the current settings of the state transition probabilities, the possibility of the appearance of C and G is high. In addition, the probability of transiting from a note to the same note is set low to avoid monotonous melodies. These settings sometimes make the generated melody deviate far from the drawn outline. A mechanism for adjusting such parameter settings would solve this problem.

5 Application to a Hybrid Melody Editor based on Notewise and Non-notewise Strategies

The experiments reported in the previous section showed that musically untrained people come to want to edit specific part of a melody as they experience non-notewise melody editing. One solution to satisfy this demand is to develop a melody editing system in which notewise editing and non-notewise editing can be seamlessly switched. We therefore developed such a system and confirmed its effectiveness through an experiment.

5.1 System Overview

This system has the following features:

- The melodic outline and the piano-roll display are overlayed. The user can switch between the outline mode and the piano-roll mode. In the outline mode, the outline is overlayed on the piano-roll display, while the piano-roll display is overlayed on the outline in the piano-roll mode.
- Once the melodic outline is edited, the piano-roll display is immediately updated; and *vice versa*.
- Once the user edits a certain note in the piano-roll mode, this note will be *fixed* on that pitch. After being fixed, the note will not be changed by editing it in the outline mode (Fig. 8). This is in order to avoid notes edited in the piano-roll mode being overridden by editing in the outline mode; this is important because editing in the piano-roll mode is considered more specific than editing in the outline mode.
- The user can manually fix and unfix notes as intended.

The typical usage of this system (Fig. 9) is as follows:

1. The user tries to edit a melody at the outline level repeatedly.
2. If he/she wants to edit specific notes of the melody, he/she switches the system to the piano-roll mode and edits those notes directly.
3. If he/she listens to the edited melody and finds satisfactory notes, he/she fixes such notes.

This system was implemented based on the method described in Section 3. The difference is not to update notes when the user redraws the melodic outline if those notes are fixed.

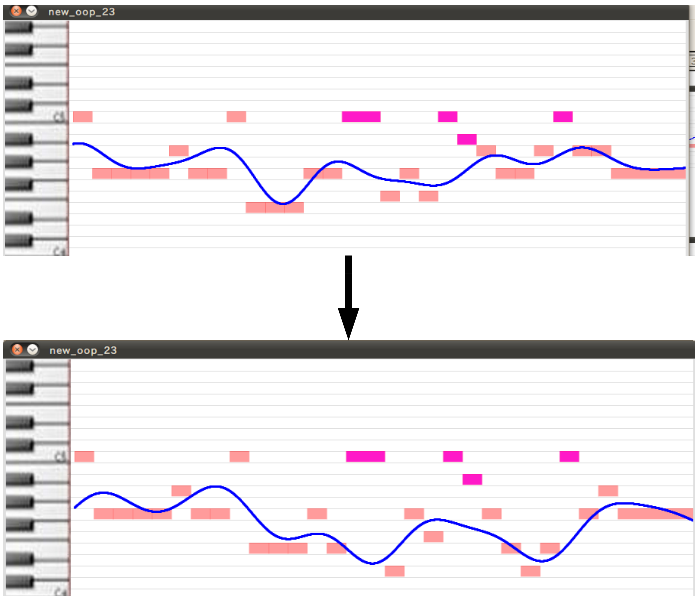

Figure 8. Example of melody editing in the outline mode in which some notes are fixed on certain pitches. It can be seen that fixed notes (shown in pink) are not changed, even though the melodic outline is redrawn.

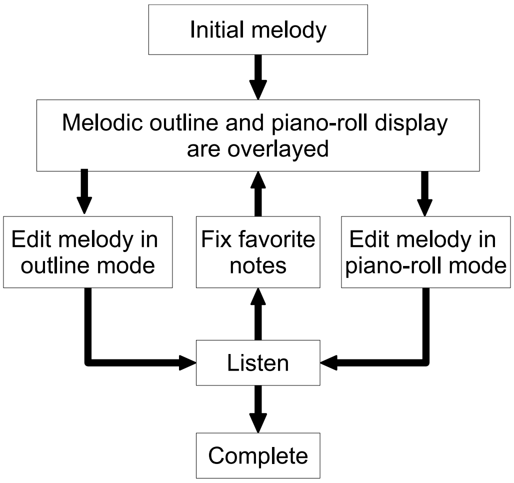
Initial melody
Melodic outline and piano-roll display are overlayed
Edit melody in outline mode
Fix favorite notes
Edit melody in piano-roll mode
Listen
Complete

Figure 9. The typical usage of the hybrid melody editor.

5.2 Experimental Conditions

An experiment was conducted in a similar way to the long-term experiment (Section 4.2) with the same three participants. First, we asked the participants to practise melody editing with this system for a week to learn how to use it. Then, we asked them to come to our laboratory to participate in the experiment. The experiment was conducted according to the following procedure:

1) Listen to a melody. The melody was generated using *Orpheus*.
2) Launch our system. A text box will appear.
3) In the text box, input your intention of melody editing (i.e., the characterstics of the melody desired after editing); use natural language (specifically Japanese).
4) Start editing.
5) Finish editing when you are satisfied with the generated melody.

During editing, we did not specify which mode (outline or piano-roll) should be used or in what order. The participants freely chose which mode they used as they like. Unlike the long-term experiment reported in Section 4.2, we used a laptop PC because note-wise editing was considered easier with a mouse than with a finger on a touch screen. The screen was recorded as a video during the whole editing session. The video was used in an interview after the test. After the test, the participants were asked the following questions, with ratings given on a scale of one to seven (except Q5):

Q1 Were you satisfied with the output melody?
Q2 Did you feel that the output melody uses musically inappropriate notes?
Q3 Was editing easy?
Q4 Was the note fixing function convenient?
Q5 Which out of this (hybrid) system and the previous (outline-only) system was better?
Q6 How do you rate the output melody?

5.3 Experimental Results

Tab. 4 lists the texts that the participants input about how they wanted to edit the given melody.

The results of the questionnaire are listed in Tab. 5. For Q1, Q3, Q4, and Q6, every participant gave a score of 5 or higher. For Q5, every participant answered that the hybrid system was better than the outline-only system. Participants A and C answered that this was because the note fixing function was useful and because they could edit particular notes directly. Participant B, however, answered that he sometimes created dissonance with the hybrid system. This is because he had to consider which notes made dissonance himself in the piano-roll mode.

5.4 Qualitative Discussion through Interview

We conducted an interview with each participant. While the experimenter (the first author) watched a video of the editing screen together with the participant, he asked the participant's intentions, impression of the output melodies, system usability, etc.

Fig. 10 shows an excerpt of the interview with Participant A.[5] Fig. 11 (a) shows the original melody as it was depicted on the screen, and Fig. 11 (b) shows it in an early stage of editing. In the latter, the outline in the second bar was redrawn, and the melody was changed accordingly. Fig. 11 (c) shows the melody in a later stage of editing, where the participant used the note fixing function. Finally, Fig. 11 (d) shows the melody after editing. In the interview, Participant A (1) used both the outline and piano-roll modes; (2) used the note fixing function; and (3) felt some notes were strange (or inappropriate) when edited at the note level. From a musical point of view, notes in the second bar became higher, and the entire melody consisted only of diatonic notes. The participant attempted to fulfill intentions (to make the second bar more exciting) by increasing the pitch, and this goal was met according to the questionnaire.

Fig. 12 shows an excerpt of the interview with Participant B. Fig. 13 (a) shows the original melody and as it was depicted on the screen, and Fig. 13 (b) shows it in an early stage of editing. In the latter, the participant changed the last note to a high pitch and then fixed it, as mentioned in the interview. Fig. 13 (c) shows the melody in a later stage of editing, and Fig. 13 (d) shows it after editing. This figure shows that the participant finished editing using the piano-roll mode (note-level editing). In the interview, Participant B (1) considered whether notes were dissonant during editing; (2) thought he could judge dissonance himself to some extent; and (3) tended to use outline-level editing to avoid dissonance. The melodies shown in Fig. 13 consist of diatonic notes only.

Table 4. Texts given by participants about how they wanted to edit the melody. Note that the original texts were written in Japanese. The following are translations by the authors.

A	Make the second bar more exciting
B	Make the third and fourth bars more bright
C	Make the pitches of the third bar higher

Table 5. Experimental results for the hybrid melody editor (For Q5, H: hybrid system, O: outline-only system).

	Q1	Q2	Q3	Q4	Q5	Q6	editing time
A	6	3	6	6	H	6	3 min 30 s
B	6	6	7	6	H	5	2 min 40 s
C	7	3	7	6	H	7	4 min 10 s

[5] All interviews were conducted in Japanese. The descriptions in the figures are translations by the authors.

The edited melody satisfied the participant's intention (to brighten the third and fourth bars), assuming that higher notes tend to be more bright.

Fig. 14 shows an excerpt of the interview with Participant C. Fig. 15 shows the melody before, during and after editing. This participant wished to restore the melody and used note-level editing to do so. This was possible because she remembered the original melody at the note level. This usage was not expected (even though this would be unnecessary if the system had an UNDO function). While editing, notes in the third bar (D4–G3–G4) were fixed. In the interview, this participant indicated that melodies generated using outline-level editing exceeded expectations.

To summarise, the three participants generally tended to:

- Use both outline-level and note-level editing.
- Consider whether melodies were dissonant while editing and felt that dissonant notes were sometimes created using note-level editing.
- Use outline-level editing to avoid dissonance or in anticipation of melodies that they hoped would exceed expectations.
- Sometimes have clear intentions to move specific notes to specific pitches and used note-level editing to accomplish this.

6 Conclusions

In this paper, we proposed a method enabling musically untrained people to edit a melody at the non-note level by transforming the melody to a melodic outline. The melodic outline is obtained by applying the Fourier transform to the pitch trajectory of the melody and extracting only low-order Fourier coefficients. After the outline is redrawn by the user, it is transformed into a note sequence. In this transform, a hidden Markov model is used to avoid notes dissonant against the accompaniment. An experiment showed that our method was easier for novice participants than the conventional piano-roll interface and that generated melodies were satisfactory for both novice and intermediate participants.

In addition to this experiment, we conducted a long-term experiment, in which we requested participants to use our system every day for one month. Through this experiment, we found that, although participants who had no experience in music composition had only abstract ideas about what kinds of melodies they want in the beginning, the ideas became clearer and more specific as they experienced melody editing. Based on this finding, we also developed a hybrid melody editor, in which the user seamlessly switches between the outline-level and note-level editing paradigms.

In the content design field, it is said that controllers for editing content should be based on the cognitive structure of the content and at an appropriate abstraction level (Katayose & Hashida, 2007). When a user interface for editing content satisfies this requirement, it is called *directable*. Melodic outlines are designed based on the insight that non-professional listeners cognise melodies

— *How many times did you use the system before the test?*
About five times.
— *I felt you were accustomed to use the system.*
Thank you.
— *You used a tablet PC last time. Which was easier to use, the tablet PC or the laptop PC?*
The laptop PC was better because the response was quick.
— *When you first edited the melody [Fig. 11 (b)], did you feel anything wrong in the sound?*
The beginning of the 2nd bar was fairly good, but I was not satisfied with the last half at all.
— *After that, you edited a particular note again and again with the outline mode. Why didn't you try the piano-roll mode?*
Because I just used the outline mode from habit. Also, when I edit a melody myself (with the piano-roll mode), I sometimes select inappropriate notes. This is also why.
— *When you edit the melody furthermore [Fig. 11 (c)], what impression did you get?*
I felt that the note next to the fixed note in the 2nd bar was not so good, so I edited it.
— *I see. How was the note fixing function?*
Easy to use. It was useful when I found notes I don't want to change.
— *How was the note-level editing?*
Almost no melodies output from melodic outlines were dissonant, but I sometimes feel notes are strange when I edit them at the note level.
— *Which were better, melodies output from the outline-level editing or ones edited at note level?*
If I have a clear intention like "this note should be moved to here", the note-level editing was better.
— *I see. Thank you very much.*

Figure 10. Interview with Participant A after the experiment with the hybrid melody editor.

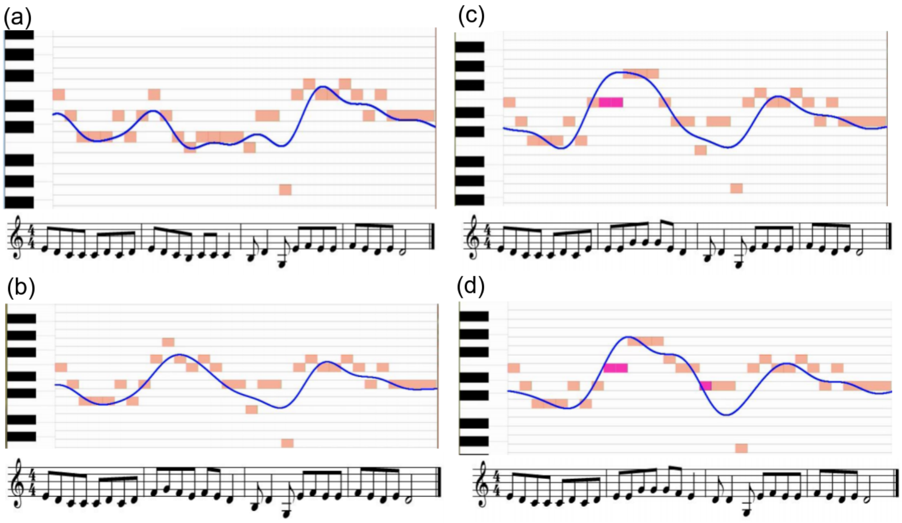

Figure 11. Melody editing by Participant A during the experiment with the hybrid melody editor. (a) Before editing; (b)(c) While editing; and (d) After editing.

— *First of all, could you tell me what points about this new melody editor were good, compared with the outline-level editor you used last time?*
With the outline-level editor, I had to edit notes again and again when the output notes were not satisfactory. With this new editor, I can directly move such notes. This is the best point.
— *You answered, in the questionnaire, that it became easier to use but notes causing dissonance increased.*
When I edited melodies at the note level, I felt dissonant notes were generated.
— *In early editing [Fig. 13 (b)], you moved the last note up and fixed it. What was your intention to do so?*
I wanted a melody in which the pitch is getting higher little by little towards the last note.
— *What did you feel when you edited the melody at the note level?*
I edited the last note very well.
— *Was the note fixing function easy to use?*
Yes, it was. I can keep notes I don't want to change.
— *What did you feel at the following note-level editing?*
I felt the result was somewhat wrong. In the previous long-term experiment, I didn't feel dissonance because I always did outline-level editing. But this time, I sometimes feel dissonance when I edit melodies at the note level.
— *Were you able to distinguish dissonant notes from non-dissonant notes yourself?*
I learned to do that to some extent.
— *You corrected those dissonant notes. After that, you edited one or two notes at the outline level. Why didn't you try note-level editing?*
It might have been better. Outline-level editing doesn't generate dissonant notes, so I tend to rely on it.
— *Next, you edited two notes at the note level. What was your intention to do so?*
I wanted to make the notes lower.
— *What did you feel when you listened to the result of the note-level editing?*
It had no dissonance. I felt I did it well.
— *I see. Thank you very much.*

Figure 12. Interview with Participant B after the experiment with the hybrid melody editor.

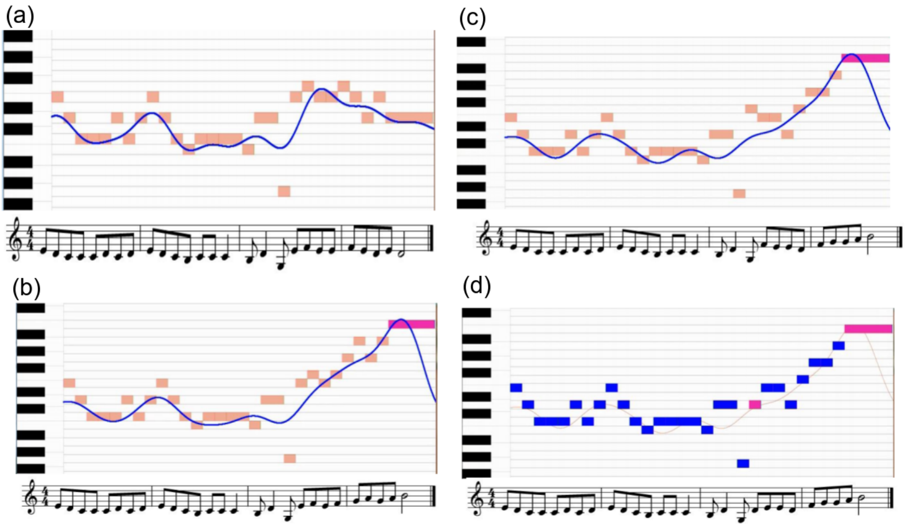

Figure 13. Melody editing by Participant B during the experiment with the hybrid melody editor. (a) Before editing; (b)(c) While editing; and (d) After editing.

— *Let's start. You edited the melody with the intention to make notes in the 3rd bar higher. How clear was this intention?*
It was really a vague idea. It was not a clear idea like "this note should be in this pitch". But when I listened to the original melody, I liked low-pitch notes in the 3rd bar. I didn't want to change them.
— *I see. You edited the melody at the outline level at first [Fig. 15 (b)]. What did you feel when you listened to the output?*
I found the melody I first listened to was better, so I restored the melody with the note-level editing.
— *I see. How did you use the note fixing function?*
By fixing notes I don't want to edit, I can keep them even if I edited the melody at the outline level. It was helpful.
— *According to the video, you seemed to fix notes in turn in part where you want to edit.*
Yes. I do so.
— *How do you use the outline-level and note-level editing?*
When I clearly want to put a note in a particular pitch, I use note-level editing.
— *What do you think about the output of these two editing methods?*
It is sometimes not so good with note-level editing. With outline-level editing, I got many melodies that were different from my imagination but which were good beyond expectation.
— *I see.*
I sometimes edit melodic outlines in a different way from my imagination in anticipation of this "beyond expectation".
— *How do you judge if melodies are good or bad?*
I judge it by listening. It is difficult to imagine melodies at the note level.
— *I see. Let's end the interview.*
Thank you very much.

Figure 14. Interview with Participant C after the experiment with the hybrid melody editor

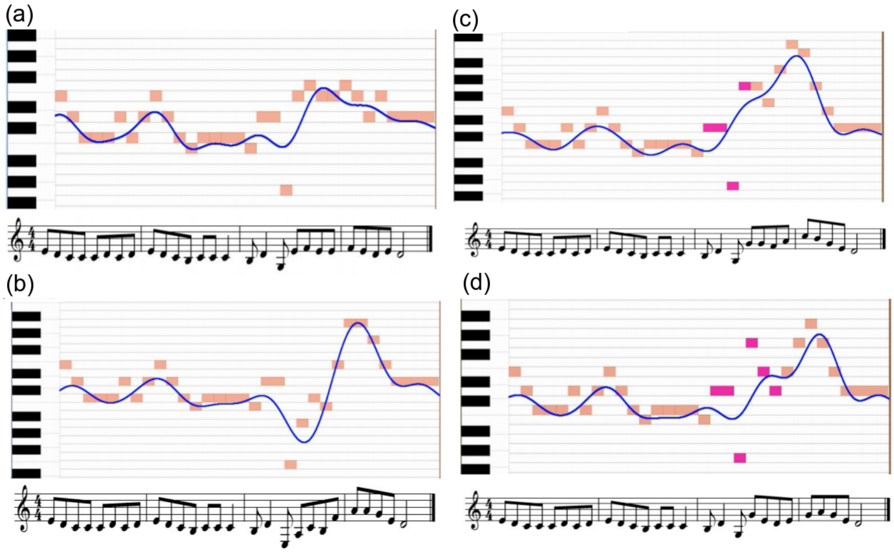

Figure 15. Melody editing by Participant C during the experiment with the hybrid melody editor. (a) Before editing; (b)(c) While editing; and (d) After editing.

without mentally obtaining note-level representations. The melody editing interface based on melodic outlines is therefore considered to achieve directability in editing melodies.

This study has various limitations. First, the sample size (particularly in the long-term and hybrid melody editing studies) was very small. While our experimental results were promising, a larger sample size would allow more robust conclusions. Second, this melody editor cannot edit the rhythm of a melody. For simplicity, the edited melodies are assumed to have retained their rhythms. To enable users to edit the rhythm of a melody, we must design a user interface that allows inputting the desired rhythm as well as the desired pitch. This is not a trivial design issue: it requires a trade-off between the degrees of freedom and design simplicity. Third, this implementation and these experiments rely on *Orpheus*. Because *Orpheus* finds a melody that matches the prosody of a Japanese lyric given by the user, that melody tends to have comparatively simple pitch patterns, for example, pitch repetition and a relatively narrow range. To apply our melody editing approach to more general styles of melodies, we must resolve challenges such as retaining the style of the original melody. Possible solutions could include learning the note transition probabilities in the HMM from existing melody data, or adopting more sophisticated models of melody representing a specific style of music.

Through the long-term experiment, we found that all participants could express their desired melodies as melodic outlines. We believe that by exploring their own melodies through this system, participants are provided with enjoyment that differs from fully automatic melody composers, such as those that generate melodies by simply clicking a "generate" button. Through further developments of our system, we aim to enable everyone who enjoys music to create music by utilising computer-based creativity support technologies.

References

Ames, C., & Domino, M. (1992). Cybernetic composer: An overview. In M. Balaban, K. Ebcioglu, & O. Laske (Eds.), *Understanding music with AI* (pp. 186–205). Palo Alto, CA: AAAI Press.

Ando, D., Dahlstedt, P., Nordahl, M. G., & Iba, H. (2006). Computer aided composition by means of interactive GP. In *Proceedings of the 2006 international computer music conference* (pp. 254–257). New Orleans, LA: International Computer Music Association.

Biles, J. A. (1994). Genjam: A genetic algorithm for generating jazz solos. In *Proceedings of the 1994 international computer music conference* (pp. 131–137). Aarhuas, Denmark: International Computer Music Association.

Cope, D. (1991). *Computers and musical style.* Oxford: Oxford University Press.

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

Fukayama, S., Nakatsuma, K., Sako, S., Nishimoto, T., & Sagayama, S. (2010). Automatic song composition from the lyrics exploiting prosody of the Japanese language. In *Proceedings of the 2010 sound and music computing conference* (pp. 299–302). Barcelona.

Garcia, J., Tsandilas, T., Agon, C., & Mackay, W. (2011). Inksplorer: Exploring musical ideas on paper and computer. In *Proceedings of the 2011 international conference on new interfaces for musical expression* (pp. 361–366). Oslo.

Goto, M. (2004). A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Communication*, *43*(4), 311–329.

Hiller, L., & Isaacson, L. (1958). Musical composition with a high-speed digital computer. *Journal of the Audio Engineering Society*, *6*(3), 154–160.

Katayose, H., & Hashida, M. (2007). Discussion on directability for generative music systems. In *SIG technical reports of IPSJ* (Vol. 2007-MUS-71, pp. 99–104). Tokyo: IPSJ. (in Japanese)

Kitahara, T., Fukayama, S., Katayose, H., Sagayama, S., & Nagata, N. (2011). An interactive music composition system based on autonomous maintenance of musical consistency. In *Proceedings of the 2011 sound and music computing conference.* Padova.

Kitahara, T., Giraldo, S., & Ramírez, R. (2017). JamSketch: Improvisation support system with GA-based melody creation from user's drawing. In *Proceedings of the 2017 international symposium on computer music multidisciplinary research* (pp. 352–363). Matoshinhos, Portugal.

Kitahara, T., & Tsuchiya, Y. (2014). Short-term and long-term evaluations of melody editing method based on melodic outline. In *Proceedings of the joint conference of the 40th international computer music conference and the 11th sound and music computing conference* (pp. 1204–1211). Athens.

Mandanici, M., & Canazza, S. (2014). The "Hand Composer": Gesture-driven music composition machines. In *Proceedings of 13th conference on intelligent autonomous systems* (pp. 553–560). Padova: Springer.

Marlot, M. (2008). A mid-level representation for melody-based retrieval in audio collections. *IEEE Transactions on Multimedia*, *10*(8), 1617–1625.

Saijo, Y., & Mizutani, M. (1991). *Sekai doyo shu.* Tokyo: Fuzanbo. (in Japanese)

Thiebaut, J.-B., Healey, P. G., & Kinns, N. B. (2008). Drawing electroacoustic music. In *Proceedings of the 2008 international computer music conference.* Belfast: International Computer Music Association.

Tsuchiya, Y., & Kitahara, T. (2013). Melodic outline extraction method for non-note-level melody editing. In *Proceedings of the 10th sound and music computing conference* (pp. 762–767). Stockholm.

Xenakis, I. (1985). Music composition treks. In C. Roads (Ed.), *Composers and the computer* (pp. 172–191). Los Altos, CA: William Kaufmann.