MELODY EXPECTATION METHOD BASED ON GTTM AND TPS

Masatoshi Hamanaka
University of Tsukuba
1-1-1, Tenodai, Tsukuba, Ibaraki, Japan
hamanaka@iit.tsukuba.ac.jp

Keiji Hirata
NTT Communication Science Laboratories
2-4, Hikaridai, Seikacho, Kei-hanna Science City, Kyoto, Japan
hirata@brl.ntt.co.jp

Satoshi Tojo
Japan Advanced Institute of Science and Technology
1-1, Asahidai, Nomi, Ishikawa, Japan
tojo@jaist.ac.jp

ABSTRACT

A method that predicts the next notes is described for assisting musical novices to play improvisations. Melody prediction is one of the most difficult problems in musical information retrieval because composers and players may or may not create melodies that conform to our expectation. The development of a melody expectation method is thus important for building a system that supports musical novices because melody expectation is one of the most basic skills for a musician. Unlike most previous prediction methods, which use statistical learning, our method evaluates the appropriateness of each candidate note from the view point of musical theory. In particular, it uses the concept of melody stability based on the generative theory of tonal music (GTTM) and the tonal pitch space (TPS) to evaluate the appropriateness of the melody. It can thus predict the candidate next notes not only from the surface structure of the melody but also from the deeper structure of the melody acquired by GTTM and TPS analysis. Experimental results showed that the method can evaluate the appropriateness of the melody sufficiently well.

1. INTRODUCTION

We have developed a method for predicting the next notes in a melody that uses the generative theory of tonal music (GTTM) [1] and the tonal pitch space (TPS) [2]. Our melody prediction method helps a novice construct a melody or play an improvisation by displaying candidates for the next notes. This method is designed to be used with a "prediction piano" (Figure 1), which was developed to assist musical novices play improvisations. On the lid of this piano, there is a 32 × 25 full-color LED matrix that displays a piano roll view that scrolls down in time with the music.

We identified two key requirements for our melody expectation method to make it useful to musical novices playing an improvisation on the expectation piano.

Figure 1. Expectation Piano.

1) Candidate notes are predicted and output even if the input melody is novel.
2) The output is appropriate from a musical point of view.

Two approaches were considered when developing this method: statistical learning and music theory. With the statistical learning approach, the predictions depend on the characteristics of the data used for learning: composer, genre, period, country, etc. [3, 4]. Moreover, predicting candidate notes for a novel melody is problematic because the system may not be able to find a similar melody in the learning data and thus may be unable to evaluate whether the notes are appropriate or not. With the music theory approach, the predictions do not depend on the characteristics of the data used for learning. It can thus be applied to novel melodies.

Although many music theories have been proposed [5–8], GTTM is the most suitable for predicting notes in a melody because it can be used to represent the various aspects of music in a single framework. Furthermore, it includes a concept of stability, meaning that it can be used to evaluate the appropriateness of the predicted notes.

This paper is organized as follows. Section 2 briefly explains GTTM and the analyzers we constructed. Section 3 explains our melody prediction method. In Section 4, we describe the expectation piano, and, in Section 5, we present some experimental results. We conclude in Section 6 with a summary of the main points and mention of future work.

2. GTTM AND ANALYZERS

The GTTM is composed of four modules, each of which assigns a separate structural description to a listener's understanding of a piece of music. These four modules output a grouping structure, a metrical structure, a time-span tree, and a prolongational tree, respectively (Figure 2). The time-span tree is a binary tree with a hierarchical structure that describes the relative structural importance of the notes that differentiate the essential parts of the melody from the ornamentation.

There are two types of rules in GTTM, i.e., "wellformedness rules" and "preference rules". Wellformedness rules are the necessary conditions for assigning structures and restrictions on the structures. When more than one structure satisfies the well-formedness rules, the preference rules indicate the superiority of one structure over another.

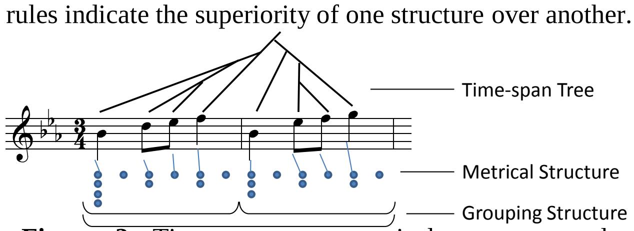

Figure 2. Time-span tree, metrical structure, and grouping structure.

2.1. Problems in implementing GTTM

In this section, we specify the problems with the GTTM rules in terms of computer implementation.

2.1.1. Ambiguous concepts defining preference rules

GTTM uses some undefined words that can cause ambiguities in the analysis. For example, GTTM has rules for selecting structures for discovering similar melodies (called parallelism), but does not have the definition of similarity itself. To solve this problem, we attempted to formalize the criteria for deciding whether each rule is applicable or not.

2.1.2. Conflict between preference rules

Conflict between rules often occurs and results in ambiguities in the analysis because there is no strict order for applying the preference rules. Figure 3 shows a simple example of a conflict between grouping preference rules (GPR). GPR3a (register) is applied between notes 3 and 4 and GPR6 (parallelism) is applied between notes 4 and 5. A boundary cannot be perceived at both 3-4 and 4-5, because GPR1 (alternative form) strongly prefers that note 4, by itself, cannot form a group.

To solve this problem, we introduced adjustable parameters that enable us to control the strength of each rule.

2.1.3. Lack of working algorithm

Knowledge represented in the rule form is in general declarative, which is advantageous in the sense that a knowledge programmer does not need to take into account an algorithm for reasoning. A system is required to perform automatic reasoning on the declaratively described knowledge.

Unfortunately, GTTM has few descriptions of the reasoning and working algorithms needed to compute analysis results.

2.2. exGTTM

In our previous work [9], we extended the GTTM theory through full externalization and parameterization and devised a machine-executable extension of GTTM, exGTTM. The externalization includes introducing an algorithm for generating a hierarchical structure of the time-span tree in a mixed top-down/bottom-up manner. Such an algorithm has not previously been represented for GTTM. The parameterization includes a parameter for controlling the priorities of rules to avoid conflicts among them as well as parameters for controlling the shape of the hierarchical time-span tree. Although it has been suggested that such parameters are required in GTTM, they were not explicitly presented.

Here, we distinguish two kinds of ambiguity in music analysis: one involves the musical understanding by humans, and the other concerns the representation of a music theory. The former kind of ambiguity derives from the ambiguity of the music itself. For the latter type of ambiguity, related to GTTM, either no concept for mechanization has been presented, or it has only been presented in an implicit way. Therefore, due to the former kind of ambiguity, we assume there is more than one correct result. We avoid the latter kind of ambiguity as much as possible by performing full externalization and parameterization.

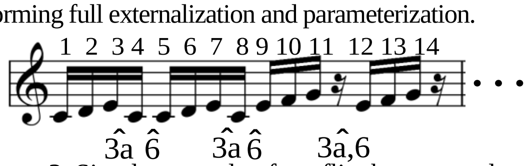

Figure 3. Simple example of conflict between rules.

2.2.1. Full externalization and parameterization

The significance of full externalization and parameterization is twofold: precise controllability and coverage of the manual results. Whenever we find a correct result that exGTTM cannot generate, we introduce new parameters and give them appropriate values so that it can generate the correct result. In this way, we repeatedly externalize and introduce new parameters until we can obtain all of the results that people consider correct. In total, we introduced 15 parameters for grouping-structure analysis, 18 for metrical-structure analysis, and 13 for time-span reduction (Table 1).

We appropriately supply lacking parameters and make implicit parameters explicit[1]. The parameters introduced by exGTTM are categorized into identified, implied, and unaware.

A parameter in the first category is identified in GTTM but it is not assigned concrete values. Hence, we valuate such a parameter. For example, since the resulting value of the GPR2a application, D_{GPR2a}, is binary, if the rule holds, D_{GPR2a} makes 1, and 0 if it does not hold. On the other hand, since GPR6 holds indefinitely, the resulting value of GPR6, D_{GPR6}, varies continuously between 0 and 1.

A parameter of the second category is implied in GTTM. Hence, we make it explicit. For example, to resolve the preference rule conflict, we introduce parameters to express the priority for each preference rule ($S^{\mathrm{GPR\ R}}$, $S^{\mathrm{MPR\ R}}$, and $S^{\mathrm{TSRPR\ R}}$ in Table 1). Since each preference rule has its own priority, all of the priority patterns are realized. This is an example of full-parameterization.

For the third category, we need to complement parameters that are not recognized in the original theory, since some of them may nearly lack any musicological meaning. For example, GPR6 in exGTTM needs to add parameters for controlling the properties of parallel segments, including the weights for pitch-oriented matching or timing-oriented matching.

We add a comment to the domain of intermediate variables, denoted as D and B. The domain of all the intermediate variables is constrained within the range of 0 to 1, and for this purpose, these variables are normalized at every computing stage. Thanks to this property, exGTTM can flexibly combine any intermediate variables (and possibly parameters) and cascade as many weighted-mean calculations as needed. This facilitates precise controllability.

2.2.2. Algorithm for acquiring hierarchy

Among the issues that require working algorithms, the problems for acquiring hierarchical structures in the grouping- and metrical-structure analyses and the time-span tree reduction can be all regarded as constraint satisfaction problems (CSP). This is because only the properties to be satisfied for the hierarchical structures are represented in the form of a rule; that is, neither constraint nor order of generating hierarchical structures is determined in advance.

The constraints stipulated by the GTTM rules are divided into two categories: local and global. The former includes GPR2 (proximity) and TSRPR1 (strong metrical position),

[1] In the paper, the word ``parameter'' is used not only for parameters used in controlling a system externally but also for internal variables (intermediated variables) connecting submodules.

	Parameters	Description
Grouping structure	$S_{GPR\,R}$ $(0 \leqq S_{GPR\,R} \leqq 1)$	Strength of each grouping preference rule. The larger the value is, the stronger the rule acts. $R \in$ { 2a, 2b, 3a, 3b, 3c, 3d, 4, 5, and 6 }
	σ $(0 \leqq \sigma \leqq 0.1)$	Standard deviation of a Gaussian distribution, the average of which is the boundary by GPR5. The larger the value is, the wider its skirt becomes.
	W_m $(0 \leqq W_m \leqq 1)$	Balance between temporal similarity of attack points and that of pitch difference in GPR6. The larger the value is, the more the system estimates the pitch difference.
	W_l $(0 \leqq W_l \leqq 1)$	Weight for the length of parallel phrases. The larger the values is, the more the length of parallel phrases is prioritized in GPR6.
	W_s $(0 \leqq W_s \leqq 1)$	Balance determining whether the note *i* becomes the ending note of a group or the beginning note of the following group in GPR6. The larger the value is, the more the note tends to be the ending note.
	T_{GPR4} $(0 \leqq T_{GPR4} \leqq 1)$	Threshold at which the effects of GPR2,3 are considered to be salient in GPR4. The smaller the value is, the more probably GPR4 is applied.
	T^{low} $(0 \leqq T^{low} \leqq 1)$	Threshold in the lower-level boundary. The smaller the value is, the more salient the boundary becomes.
Metrical structure	$S_{MPR\,R}$ $(0 \leqq S_{MPR\,R} \leqq 1)$	Strength of each metrical preference rule. The larger the value is, the stronger the rule acts. $R \in$ {1,2,3,4,5a, 5b, 5c, 5d, 5e, and 10)
	W_m $(0 \leqq W_m \leqq 1)$	Balance between temporal similarity of attack points and that of pitch difference in MPR1. The larger the value is, the more the system estimates the pitch difference.
	W_l $(0 \leqq W_l \leqq 1)$	Weight for the length of parallel phrases. The larger the value is, the more the length of parallel phrases is prioritized in MPR1.
	W_s $(0 \leqq W_s \leqq 1)$	Balance determining whether the note *i* becomes the ending note of a group or the beginning note of the following group in MPR1. The larger the value is, the more the note tends to be the ending note.
	$T_{MPR\,R}$ $(0 \leqq T_{MPR\,R} \leqq 1)$	Value of the threshold that decides whether each rule is applicable. $R \in$ { 4, 5a, 5b, 5c }
Time-span tree	$S_{TSRPR\,R}$ $(0 \leqq S_{TSRPR\,R} \leqq 1)$	Strength of each rule. The larger the value is, the stronger the rule acts. $R \in$ {1, 2, 3, 4, 5a, 5b,5c, 5d, 5e, and 10}
	W_m $(0 \leqq W_m \leqq 1)$	The balance between the temporal similarity of attack points and that of the pitch difference in TSRPR4. The larger the value is, the more the system estimates the pitch difference.
	W_l $(0 \leqq W_l \leqq 1)$	The weight for the length of parallel phrases. The larger the values is, the more the length of parallel phrases is estimated in TSRPR4.
	W_s $(0 \leqq W_s \leqq 1)$	The balance determines whether the note i becomes the ending note of a group or the beginning note of the following group in TSRPR4. The larger the value is, the more the note tends to be the ending note.

Table 1. Adjustable parameters of the exGTTM and ATTA.

and the latter GPR5 (symmetry) and MPR1 (parallelism). We need to handle global constraints carefully when generating hierarchical structures. For the example of GPR5 in Figure 4, given a group at Layer 1, an inner boundary likely occurs around the center of the group, that is, either between Notes 1 and 2 or between Notes 2 and 3. Here, we can consider two cases. In Case 1, the boundary between Notes 1 and 2 is selected, taking into account the effects of some other rules. Then in each subgroup in Layer 2, the inner boundary of the subgroup may occur on the left-hand side of a center note. On the other hand, in Case 2, the boundary between Notes 2 and 3 is selected. Therefore, the inner boundary may occur on the right-hand side of a center note. Consequently, in computing GPR5, the determined boundary position influences the identifications of remote boundaries in lower layers, and we have to take into account up-to-date global information every time. That is, a global constraint is inevitably dynamic.

In light of the above considerations, we are developing algorithms for generating hierarchical structures for exGTTM so that nodes are generated either from the bottom-most nodes

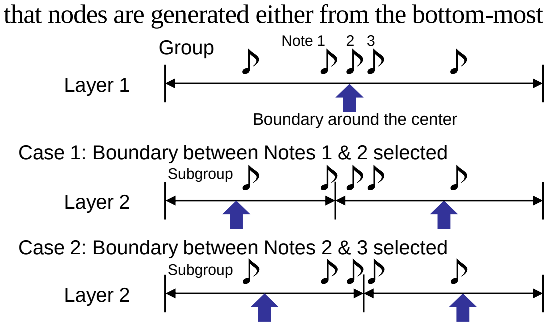

Figure 4. Simple example of conflict between rules.

or the top-most node incrementally and so that every time the nodes at a layer are calculated, global information is re-calculated before moving onto an adjacent layer.

2.3. FATTA: Fully Automatic Time-span Tree Analyzer

We implemented a time-span tree analyzer, called automatic time-span analyzer (ATTA), based on exGTTM. Although ATTA can automatically acquire a time-span tree, because the parameters are manually controlled, it takes too much time to find a set of optimal parameters. Therefore, we developed a method for automatically estimating the optimal parameters [10].

Two rules in GTTM [1] are not implemented in ATTA: GPR7 and TSRPR5.

> GPR7 (time-span and prolongational stability): prefer a grouping structure that results in a more stable time-span and/or prolongational reductions.
>
> TSRPR5 (metrical stability) In choosing the head of time-span *T*, prefer a choice that results in a more stable choice of metrical structure.

These rules require that information from later processes, such as time-span/prolongational reductions, be sent back to earlier processes.

To estimate the optimal parameters automatically, we evaluate the structural stability of the analysis results derived by ATTA. We use GPR7 and TSRPR5 to calculate the level of stability. Figure 5 shows the process flow of the FATTA, which consist of the ATTA and a loop by the GPR7 and TSRPR5.

2.3.1. Implementation of GPR7 with Tonal Pitch Space

GPR7 is applied to the loop between the time-span/prolongational reduction and grouping structure analysis. This rule leads to a preference for a grouping structure that results in a more stable time-span and/or prolongational reductions. The holding level of GPR7, which varies continuously between 0 and 1, is defined as

$$D_{GPR7} = \frac{\sum_i \text{distance}(p(i), s(i)) \times \text{size}(i)^2}{\sum_i \text{size}(i)^2}, \quad (1)$$

where i indicates the head of the time-span, which has primary and secondary branches, denoted by $p(i)$ and $s(i)$, respectively. Distance (x, y) indicates the distance between notes x and y in the tonality of the piece, which are defined using Lerdahl's tonal pitch space [2]. We normalized the distance from 0 to 1. The size(i) indicates the length of the time-span with head i. When calculating D_{GPR7}, we use the square of size(i) for the weighting for empirical reasons.

In the tonal pitch space, the distance between chord x = $C_1/\mathbf{R_1}$ and chord y = $C_2/\mathbf{R_2}$ is defined as follows:

$$\delta(x \rightarrow y) = i + j + k\,, \quad (2)$$

where i is region distance, j is chord distance, and k is basic space difference. The region distance is the smallest number of steps along the regional circle of fifth between $\mathbf{R_1}$ and $\mathbf{R_2}$. The chord distance is the smallest number of steps along the chordal circle of fifth between the roots of C_1 and C_2 within each region. The basic space distance is a specially weighted to define each chord and region. Note that pitch class only has a meaning in terms the elements of the sets that define chords and regions, and chords are always understood as functioning within some region.

2.3.2. Implementation of TSRPR5

TSRPR5 is applied to the loop between the time-span reduction and metrical structure analyzer and results in a more stable metrical structure when choosing the head of a time-span. The holding level of TSRPR5, which varies continuously between 0 and 1, is defined as

$$D_{TSRPR5} = \frac{1}{\sum_i \text{size}(i)^2} \times \sum_i \begin{cases} \text{size}(i)^2 & dot(p(i)) \geq dot(s(i)) \\ 0 & dot(p(i)) < dot(s(i)) \end{cases}, \quad (2)$$

where $dot(x)$ indicates the number of metrical dots of note x.

2.3.3. Optimization of adjustable parameters

The set of optimal ATTA parameters is obtained by maximizing the average of D_{GPR7} ($0 \leq D_{GPR7} \leq 1$) and D_{TSRPR5} ($0 \leq D_{TSRPR5} \leq 1$). The parameters and default values are $S^{rules} = 0.5$, $T^{rules} = 0.5$, $Ws = 0.5$, $Wr = 0.5$, $Wl = 0.5$, and $\sigma = 0.05$. Because there are 46 parameters, a great amount of time is needed to calculate all parameter combinations. To minimize the calculation time, we constructed an algorithm that does the following:

(1) Maximize average of D_{GPR7} and D_{TSRPR5} by changing a parameter from its minimum to its maximum value.
(2) Repeat (1) for all parameters.
(3) Iterate (1) and (2) as long as the average of D_{GPR7} and D_{TSRPR5} is higher than after the previous iteration.

3. MELODY EXPECTATION METHOD

Our melody expectation method predicts candidate notes by using the level of stability of the time-span tree defined in FATTA. Our position is that we cannot always specify a single expected, following tone; and thus, we developed an expectation piano that simply suggests multiple candidates among stable pitch events with higher stability. The functions of FATTA are restricted as follows; FATTA only treats monophonic western tonal music. Thus, our expectation method can predict only monophonic musical structures of western tonal music.

3.1. Expectation Method based on FATTA

The main advantage of our melody expectation method is that, the stability of a melody is calculated by analyzing the whole melody from the beginning note to the expected note, not from only the local melody(a few notes previous to a relevant note) ; previous melody expectation methods based on music theory (eg. Steve Larson's theory of musical forces [8]) derive the

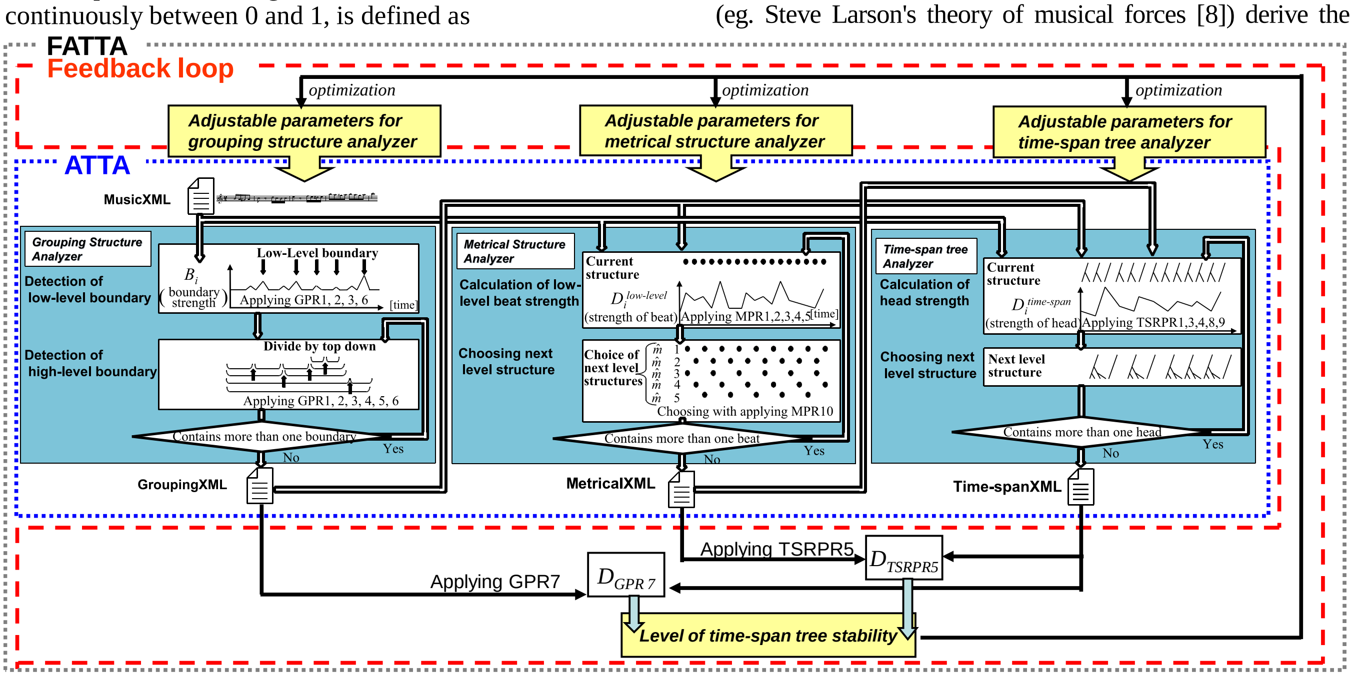

Figure 5. Processing flow of fully automatic time-span tree analyzer (FATTA).

expected note from the local melody. Music tends to be more interesting when it does not match the listener's expectations, such as a delayed note, and this may result in tension and relaxation. A composer can deliberately construct such music, which can make it difficult to predict the next notes in the melody with accuracy. For example, an ornamentation note is often inserted before the expected note. In such cases, our method can predict candidate notes fairly well because FATTA can evaluate the stability of the entire structure of the time-span trees, which includes branches connected to essential notes and leaves connected to ornament notes.

3.2. Real-time Extension for FATTA

To be able to predict notes on the basis of GTTM, FATTA must run in real time. However, FATTA needs several minutes to finish the analysis, so running in real time is difficult. Therefore, we extended the algorithm to enable real-time operation. First, to speed up the iteration described in 2.2, we use the set of optimal parameter values for the last melody, which is one note shorter than the present melody, as the initial parameter set. Second, to reduce the time used by ATTA, we introduce a window for analysis. The size of the window is the longest group length within 16 measures of the present position. This length can be acquired through preprocessing using the grouping structure analyzer in ATTA. If there is no grouping boundary in 16 measures from the present position, we use 16 measures as the window size.

3.3. Calculation Level of Stability of Melodies by FATTA

Our method evaluates the appropriateness of each candidate melody to occur after the present one by calculating its stability. We use the average of D_{GPR7} and D_{TSRPR5} as the level of stability acquired by FATTA.

Figure 6 shows the calculated level of stability from the primary note to the present note. The level of stability can be calculated after the third note because GTTM analysis needs at least four notes.

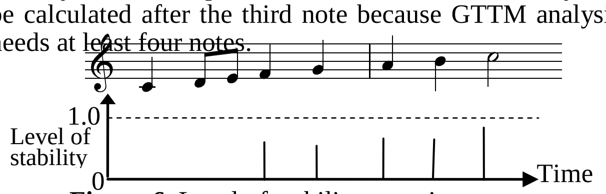

Figure 6. Level of stability over time.

In the score, the primary note is a tonic of the region, and the tail note is also a tonic of the region. This means that the levels of stability indicate a large value at the tail of the melody and a smaller value in the middle of the melody. Therefore, the higher the level of stability, the relatively closer the tonic of the region. The region of the melody and chord progression were estimated in the implementation of GPR7 by applying the tonal pitch space.

4. EXPECTATION PIANO

Our expectation piano assists novices with musical improvisation by displaying the predicted notes on the piano lid. When the novice finds it hard to continue playing the melody, she/he can continue the improvisation by playing a note displayed on the lid, without impairing tonality.

4.1. Overview

The processing flow of the prediction piano is as follows. First, the MIDI signals for the music played on the piano are sent to a computer. The signals are quantized, and a MusicXML version of the performed melody is created. A learning-based quantization method [11] is used to eliminate the deviation in onset times, which are then aligned to the normalized positions. The predicted notes are acquired by inputting the MusicXML into FATTA. Finally, the predicted notes are displayed on the piano lid.

4.2. LED Scrolling Piano Roll

The predicted notes are displayed in piano roll format within the range of view of the keyboard. The roll scrolls down at a constant speed. Below the piano lid, which is made of semitransparent acrylic resin, there is a 32 × 25 full-color LED matrix for displaying the scrolling piano roll. The 32 represents two measures when the resolution is a sixteenth note, and 25 is the number of keys on the keyboard. The color of each LED in the matrix is determined under the assumption that the onset of the next note will start at the corresponding position on the piano roll and by calculating the level of stability. When the level of stability is high, the LEDs show yellow, when it is low, they show black, and when it is neither, they show red. There is also a 32 × 20 blue LED matrix that displays the bar lines of the piano roll.

4.3. Construction

The piano is 953 mm long and 710 mm wide and resembles a grand piano. It contains a MIDI keyboard, the LED display, a microcomputer, a power supply, and four speakers. The LED display is 630 mm long and 390 mm wide. The colors of the LEDs are controlled by MAX6972 which is 16-output 12-bit pulse-width-modulation (PWM) LED drivers. There is a 5-mm gap between the LEDs and piano lid to protect the acrylic resin lid from the heat of the LEDs. A half-mirror sheet is bonded to the back side of the acrylic resin so that the lights of the LEDs show on the surface of the piano lid rather than 5 mm below it. The LED drivers are controlled using a microcomputer connected to the computer with a network cable. The computer sends the data for the LED colors by using the user datagram protocol.

5. EXPERIMENTAL RESULTS

It is difficult to compare the performance of our system with those of previous systems, which are mostly based on statistical learning, because the approaches taken are completely different. Our method, which is based on music theory, evaluates the appropriateness of the notes from a musical point of view. Therefore, we first quantitatively evaluate each step in our method and then describe an example result.

5.1. Evaluation of FATTA

We evaluated the performance of FATTA using an F-measure given by the weighted harmonic mean of precision *P* (proportion of selected groupings/dots/heads that are correct) and recall *R* (proportion of correct groupings/dots/heads that are identified). In calculating the F-measure of the grouping analyzer and time-span tree analyzer, we did not consider the possibility that a low-level error is propagated up to a higher level—we simply counted wrong answers without regard to the differences in grouping or time-span levels.

This evaluation required preparation of accurate data for the grouping structure, metrical structure, and time-span tree. We collected 100 pieces of 8-barlength, monophonic, classical music and asked people with expertise in musicology to analyze them manually with faithful regard to the GTTM. These manually produced results were cross-checked by three other experts.

The grouping, metrical, and time-span tree structures change depending on the parameter settings. To evaluate the baseline performance of the ATTA, we used $S_{\mathrm{rules}} = 0.5$, $T_{\mathrm{rules}} = 0.5$, $W_s = 0.5$, $W_r = 0.5$, $W_l = 0.5$, and $\sigma = 0.05$. The range of T_{rules}, W_s, W_r, and W_l was 0 to 1.0, and the resolution was 0.1. The range of σ was 0 to 0.1, and the resolution was 0.01.

After the set of parameters was optimized using FATTA, the average F-measure was 0.48, 0.89, and 0.49, respectively, for grouping, metrical, and time-span tree structures, all better than the baseline performance of ATTA (Table 2).

Melodies	Grouping		Metrical		Time-Span	
	Baseline	FATTA	Baseline	FATTA	Baseline	FATTA
1. Grande Valse Brillante	0.21	0.32	0.88	0.88	0.37	0.41
2. Moments Musicaux	0.24	0.60	0.95	1.00	0.58	0.74
3. TrukishMarch	0.67	0.67	0.91	0.96	0.68	0.80
4. Anitras Tanz	0.29	0.71	0.82	0.82	0.55	0.52
5. Valse du Petit Chien	0.04	0.28	0.87	0.95	0.17	0.57
	:	:	:	:	:	:
Total (100 melodies)	0.46	0.48	0.84	0.89	0.44	0.49

Table 2. F-measure for baseline and FATTA.

5.2. Evaluation of region estimation by TPS

The tonal pitch space is used by FATTA to estimate the region and chord progression. We evaluated the performance of the region estimation by using the same 100 musical pieces. The TPS correctly estimated the regions for 96 pieces.

5.3. Example of Melody Expectation

Figure 7 shows the results for Haydn's *andante*. The graph below the musical staff indicates the level of melody stability, corresponding each above note. The number under each note indicates the level of stability for selecting a pitch of 25 possible ones. Stability of a pitch event is calculated from the partial melody between its beginning and the relevant event, and not by the sole event. As a result, the F# in measure 7 does not have the lowest stability, and that of C in measure 5 is lower than that of G in the previous measure, even in C major. Although this may contradict intuition, the calculated result is faithful to our algorithm. At the end of the 4th and 8th measures, the level of stability is high. This is because a dominant chord's note that wants resolve to a tonic chord's note occurs. In contrast, at the beginning of the 5th measure, the level of stability is relatively low. This is because a tonic chord's root note at the beginning of the 5th measure occurs, and various progressions can follow the root note. These results show that our prediction method works well from a musical point of view.

6. CONCLUSION

We devised a melody expectation method that predicts candidate notes on the basis of the generative theory of tonal music (GTTM) and the tonal pitch space (TPS). It is designed to be used with an expectation piano, which displays the predicted notes on its lid, thereby supporting musical novices in playing improvisations. We experimentally evaluated our method and got the following results.

(1) The performance of the fully automatic time-span tree analyzer (FATTA) outperformed the baseline F-measure of the automatic time-span tree analyzer (ATTA).
(2) Region estimation by using a tonal pitch space (TPS) worked sufficiently well for predicting the melody.
(3) Our expectation method works well from a musical point of view.

We plan to evaluate the ability of our expectation piano to assist musical novices play improvisations with this method. We have not included the prolongational reduction in FATTA, because the search space of prolongational tree would explosively expand and this large search space would make it hard to acquire a solution within a practical time. Development of an efficient algorithm for the prolongational reduction would be one of our future tasks.

7. REFERENCES

[1] Lerdahl, F. and Jackendoff, R.. *A Generative Theory of Tonal Music*. MIT Press, Cambridge, Massachusetts, 1983.
[2] Lerdahl, F. *Tonal Pitch Space*, Oxford University Press, 2001.
[3] Mozer, M. *Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-Scale Processing*. Connection-Science. Vol. 6, No. 2-3, pp. 247-280, 1994.
[4] Yen, G., Lucas, S., Fogel, G., Kendall, G., Salomon, R., Zhang, B., Coello, C., and Runarsson, T. "Evolving Musical Sequences with N-Gram Based Trainable Fitness Functions", *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pp. 604-614, 2006.
[5] Cooper, G., and Meyer, L. B. *The Rhythmic Structure of Music*. The University of Chicago Press, 1960.
[6] Narmour, E. *The Analysis and Cognition of Basic Melodic Structure*. The University of Chicago Press, 1990.
[7] Temperley, D. *The Cognition of Basic Musical Structures*. MIT press, Cambridge, 2001.
[8] Larson, S. *Musical Forces and Melodic Expectations: Comparing Computer Models with Experimental Results*. Music Perception. 21/4, 457-498.
[9] Hamanaka, M., Hirata, K., Tojo, S.: Implementing "A Generating Theory of Tonal Music", Jornal of New Music Reserch (JNMR), Vol. 35, No. 4, pp. 249-277, 2006.
[10] Hamanaka, M., Hirata, K., and Tojo, S.: "FATTA: Full Automatic Time-span Tree Analyzer", *Proceedings of ICMC2007*, Vol. 1, pp. 153-156, 2007.
[11] Hamanaka, M., Goto, M., Asoh, H., and Otsu, N.: "A Learning-Based Quantization: Unsupervised Estimation of the Model Parameters", *Proceedings of ICMC2003*, pp. 369-372, 2003.

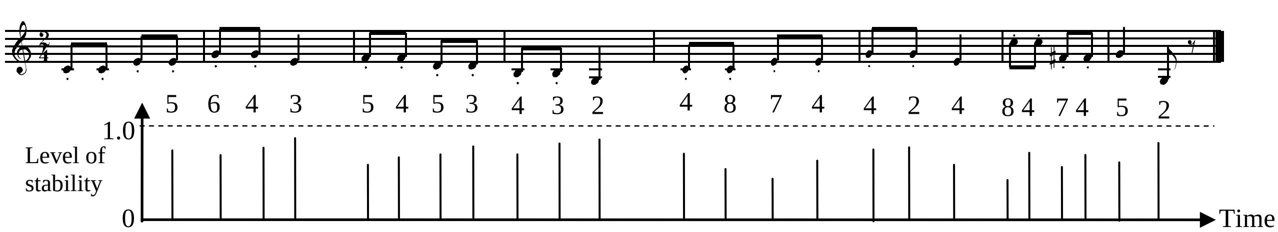

Figure 7. Example of melody expectation.