2006 IEEE International Conference on
Systems, Man, and Cybernetics
October 8-11, 2006, Taipei, Taiwan

Computer Music Composition Based on Discovered Music Patterns

Shih-Chuan Chiu and Man-Kwan Shan

***Abstract*—Computer music composition has been the dream of the computer music researcher. In this paper, we investigated the approach to discover the rules of music composition from given music objects, and automatically generate a new music object style similar to the given music objects. The proposed approach utilizes the data mining techniques to discover the rules of music composition characterized by the music properties, music structure, melody style and motif. A new music object is generated based on the discovered rules. To measure the effectiveness of proposed computer music composition approach, we adopted the method similar to the Turing test to test the discrimination between machine-generated and human-composed music. Experimental results showed that it is hard to discriminate. Another experiment showed that the style of generated music is similar to the given music objects.**

I. INTRODUCTION

COMPOSING music by formal processes of machine has been investigated for a long time. Current research on computer composition may be classified into two approaches according to the way of composition rule generation. In the explicit approach, the composition rule is specified by humans while in the implicit approach the composition rule is learned from sample music. Training data is required, in the implicit approach, to discover the composition rules. In this paper, we investigated the implicit approach of computer music composition based on the discovered music patterns from training data. The developed approach will take a set of user-specified music as input and generate the music with music style similar to the user-specified music set.

There are four design issues regarding the implicit approach, feature extraction, feature analysis, rule learning and music generation, as shown in Figure 1. Feature extraction concerns the extraction of low-level music features from sample music. Feature analysis obtains the high-level semantic information from low-level music features. Rule learning discovers the patterns (compositional rules) in terms of the high level semantic information from the set of sample music. Music generation employs the discovered patterns to generate music.

The process of popular music production consists of two major steps, composition, arrangement and record. Composers create original melody with chord in the basic structure. Arrangers rewrite and adapt the original melody written by composers by specifying harmonies, instrumentation, style, dynamics, sequence, et al. After these two steps, performance, recording, mixing, and audio mastering are conducted to produce the music.

Existing work on the implicit rule approach generates the melody only, ignoring the consideration of chord. In this paper, we proposed a new framework addressing music composition with the consideration of both melody and chord. Especially, the proposed framework is developed based on the data mining techniques.

In the next section, we review previous work on computer music composition. Section 3 gives the system architecture and feature extraction of proposed approach. Feature analysis and rule learning is described in section 4. Section 5 presents the music generation method. Experiments are shown in section 6. The conclusion is made in section 7.

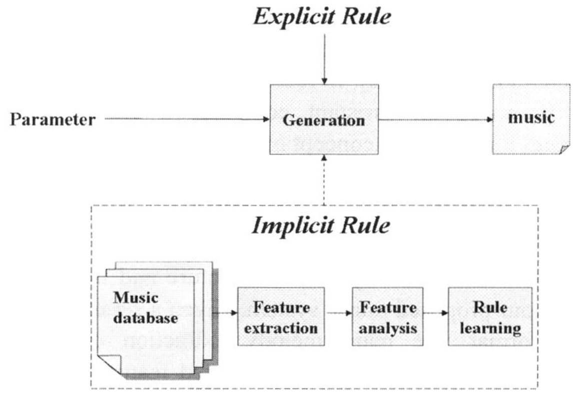

Fig. 1. Flow chart of computer music composition.

II. RELATED WORK

Early work on computer music generation focuses on the explicit approach that the composition rules are specified by composer. Examples are three approaches introduced in the classical book of computer music "Composing Music with Computers," where the probability model, grammar model and automata model are employed to model the music composition rules elicited from musicians [9].

Recent work on computer music composition tries to develop the implicit rule approach. D. Cope separated a set of music into small segments. A new music object is generated by analyzing and combining these small segments [2]. Y. Marom used Markov Chain to model melody [8]. At the IRCAM research center, S. Dubnov et al. constructed a model for simulating the performed style of great master by utilizing the approaches of incremental parsing (IP) and predict suffix

Shih-Chuan Chiu is with the Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan 300, ROC (e-mail: g9222@cs.nccu.edu.tw).

Man-Kwan Shan is with the Department of Computer Science, National Chengchi University, Taipei City 11605, Taiwan ROC (e-mail: mkshan@cs.nccu.edu.tw).

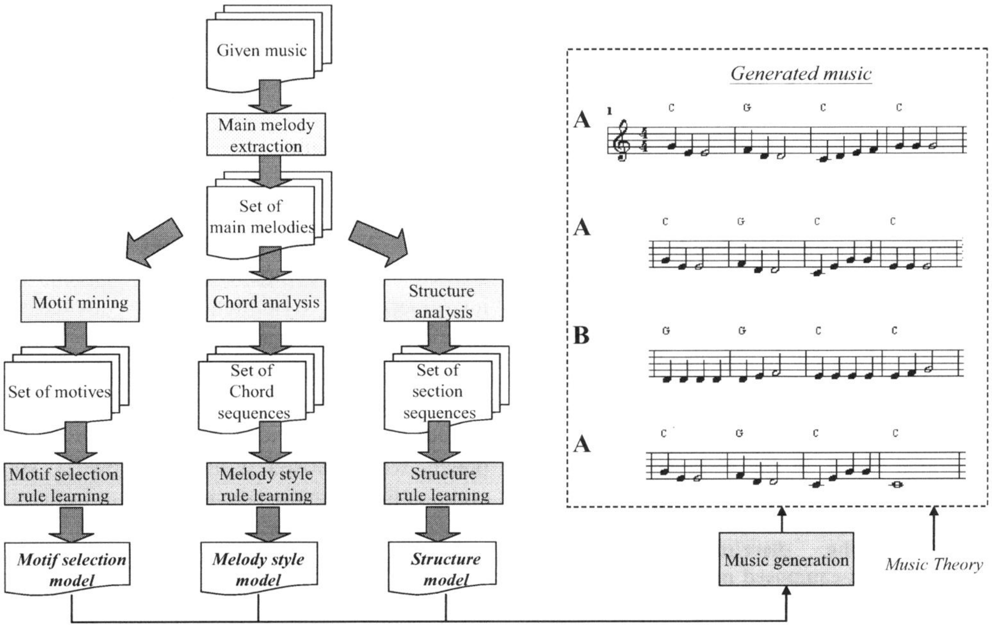

Fig. 2. The system architecture and process flow of the proposed approach.

tree (PST) coming from statistics and information theory areas [3]. At CMU, B. Thom proposed a real-time interaction system that generates a response solo according to a solo of user and play style model. The system models play style by using the concept of an expectation-maximization algorithm to train a lot of bars [12]. In recent years, M. M. Farbood at MIT proposed an assistant composition system which generates music by the concept of painting [4].

III. System Architecture and Main Melody Extraction

Figure 2 shows the system architecture and the process flow of our proposed music system. Given a set of music in MIDI format, the main melody extraction component extracts the main melody and associated features for each music object. Then, each extracted melody is analyzed by the motif mining component, the chord analysis component, and the structure analysis component respectively. The motif mining component finds the set of motives which constitute the candidates for the motif selection learning component. The chord analysis component produces the set of chord sequences for the melody style learning. The structure analysis component generates the set of section sequences for structure learning. After these analysis and learning processes, three models, music structure model, melody style model and motif selection model are established. In the music generation component, a new music object is generated based on these three models.

Melody is the essential element for music composition. The main melody extraction component consists of two steps. In the first step, quantization corrects the onset time and duration of notes. This comes from the fact that in some music of MIDI format, it is possible that notes do not appear in appropriate position. The next step extracts the monophonic melody from the polyphonic melody. Uitdenbogerd et al. [13] have presented the melody extraction methods, namely, All-mono, Entropy-channel, Entropy-part and Top-channel. According to their experimental result, All-mono obtains the best accuracy. The basic idea of All-mono is to merge all tracks contained in a MIDI file. The main melody is extracted by keeping the note with the highest pitch from those pitches occurring at the same time.

IV. Analysis and Learning

4.1 Music Structure Analysis and Rule Learning

Music structure can be regarded as a hierarchical structure similar to the structure of an article. In our approach, a music object is composed of sections and a section is composed of one or more phrase. The structure analysis component discovers the section-phase hierarchical structure of a music object while the structure learning component mines common characteristics from structures of several music objects.

There are five steps for the music structure analysis. In the first step, pitch and duration information of each note is extracted from the main melody. The main melody is a note sequence where a note can be parameterized with several property values such as pitch, duration, velocity, etc. Velocity is only considered in music performance, therefore only pitch and duration are considered for the structure analysis.

Then, the repeating pattern technique is employed to discover the repeating patterns of pitch-duration sequence. There exist the algorithms based on suffix tree and correlative matrix [6] to discover the repeating pattern in the research field of bioinformatics and music mining

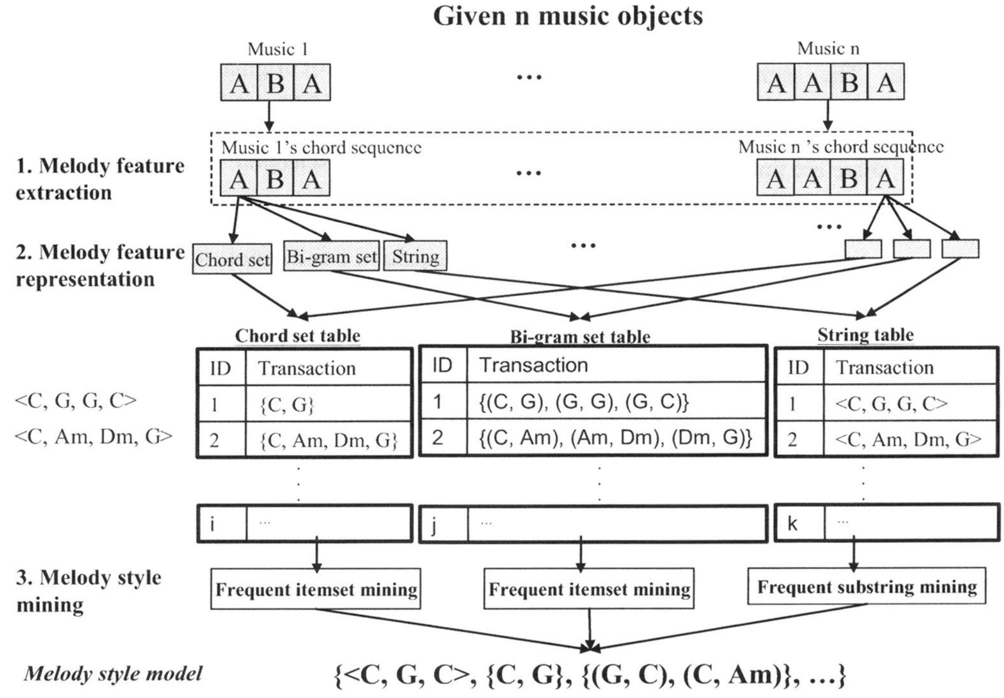

Fig. 4. Flow chart of melody style mining.

respectively.

After the repeating pattern mining process, a music object may contain more than one repeating pattern. Each repeating pattern appears as several instances. Figure 3 shows an example of the instances after repeating pattern mining on music "Little Bee." In this figure, a strip denotes an instance of non-trivial repeating pattern abbreviates NTRP. There are six non-trivial repeating patterns.

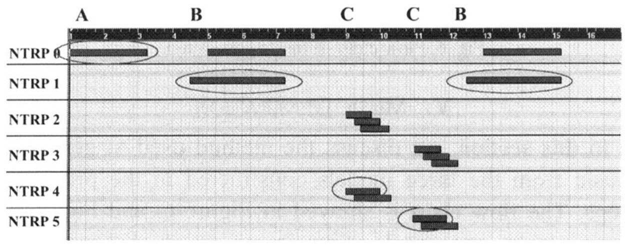

Fig. 3. An example of the instances after repeating pattern mining on music "Little Bee."

Not all instances of repeating pattern are required for analysis, therefore we have to select appropriate instances. In our approach, firstly, all the instances of the repeating patterns with length shorter than two bars are filtered out. Then we transfer the problem into the exon chaining problem in Bioinformatics [7]. We wish to find the set of non-overlapping repeating pattern instances such that the total length of the selected instances is maximized.

Given a set of weighted intervals in a chain, the exon-chaining problem tries to find a set of maximum weight of non-overlapping intervals. This problem can be solved by dynamic programming. We can modify this algorithm for the pattern selection problem by replacing weight of interval with duration. Figure 3 shows the discovered repeating patterns and corresponding instances of the music "Little Bee." The five circled instances are selected by the selection algorithm modified from the exon chaining algorithm.

Each of the selected instance therefore corresponds to a section. We labeled each selected instances such that the instances of the same repeating pattern are labeled with the same symbol. For the example of Figure 3, the labeled sequence becomes ABCCB.

The next step of music structure analysis is to discover the phrase structure for each section. We use the approach of LBDM (Local Boundary Detection Model) developed by Cambouropoulos et al.[1] to segment a section into phrases. LBDM extracts the pitch interval sequence, the inter onset interval sequence and reset sequence from main melody. These three sequences are integrated into the sequence of boundary strength values measured by the change rule and the proximity rule. Peaks of the boundary strength value sequence are therefore the boundaries of segments.

The structure analysis component outputs a section sequence where the section is parameterized by *label*, *occurrence*, *numOfPhrase* and *length*. Attribute *label* denotes which label it is. Attribute *occurrence* denotes the number of appearances of the same label. Attribute *numOfPhrase* denotes the number of phrases in this section. Attribute *length* denotes the length of the section. In the learning step, the statistical analysis of the set of section sequences is derived to capture the common patterns of the music structure.

4.2 Chord Analysis and Melody Style Rule Learning

After the analysis of the music structure, the melodies are segmented into sections. The segmented melodies are collected for music style mining. We have proposed the music style mining technique to construct the melody style model [11]. As Figure 4, there are three steps for proposed melody style mining technique, melody feature extraction, melody feature representation, and melody style mining. The basic idea is to extract the chords accompanied with melody as the feature of melody. A chord is a number of pitches sounded simultaneously. The *chord assignment* algorithm is based on the harmony theory. A detailed algorithm can be referred to from our previous work [11]. After determining the chord, the feature of a melody can be represented as a set of chords, a set of bi-grams and a sequence of chords. For instance, a chord sequence is <C, G, G, C>. This sequence represented by set of chord is {C, G}, represented by bi-gram of chord is {(C, G), (G, C)}, represented by sequence of chord is still <C, G, G, C>.

To obtain the hidden relationships between chord and music styles, we employ mining methods with respect to the three representations of melody feature. If the representation of melody feature is a set of chords or a set of bi-grams of chords, frequent itemset mining algorithm is utilized. If the feature of the melody is represented as a sequence of chords, frequent substring mining algorithm modified from the sequential pattern mining is employed. The discovered frequent patterns in terms of chords constitute the music style model.

Fig. 5. Examples of the development of motif: (1) Repetition, (2) Sequence, (3) Contrary Motion, (4) Retrograde, (5) Augmentation and Diminution.

4.3 Motif Mining and Motif Selection Rule Learning

A motif is a reoccurring fragment of notes that may be used to construct the entirety or parts of theme. Based on music theory, there are several ways for developing a motif. The major ways of the motif development are repetition, sequence, contrary motion, retrograde, augmentation and diminution. Figure 5 shows these five developments of motif variations. The first segment rounded by the block is the original motif, and the following segments are the developments of original motif. We have modified the repeat pattern algorithms based on the development of motif to discover the motives [5].

The motif selection model describes the importance of motifs. Let $Freq_{m,music}$ denotes the frequency of a motif m appearing in music object *music*. We normalize the formulas as equation 1 and denote it as *Support(m,music)*. For a motif m in the given set of given music object DB, we sum up its support and denote it as *ASupport(m,DB)*. Finally, we normalized the *ASupport* as equation 3 and denoted it by *NSupport(m,DB)*, where *Min(DB)* and *Max(DB)* represent the minimum and maximum *ASupport* of the motif in DB.

$$Support(m, music) = Freq_{m,music} \Big/ \sum_{\forall motif \in music} Freq_{motif,music} \quad (1)$$

$$ASupport(m, DB) = \sum_{\forall am \in DB} Support(m, am) \quad (2)$$

$$NSupport(m, DB) = (ASupport(m, DB) - Min(DB) + 1) / (Max(DB) - Min(DB) + 1) \quad (3)$$

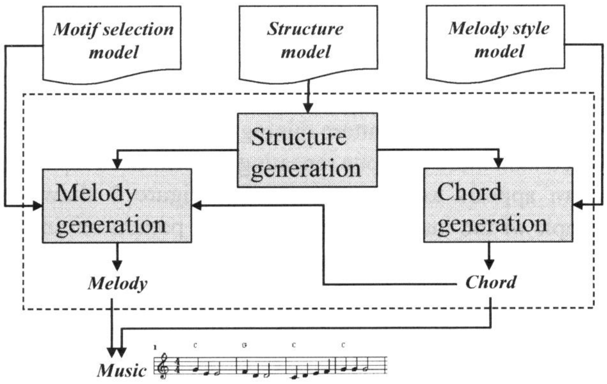

Fig. 6. Flow chart of music generation.

V. MUSIC GENERATION

In this section, we discuss the method used to generate music from the three models constructed in the previous steps. The flow chart is showed in Figure 6, and Figure 7 demonstrates an example of music generation. According to probabilistic distribution in the music structure model, the system generates the music structure expressed as a sequence of sections. For each section in the section sequence, the heuristic algorithm, Phrase-Arrangement shown in Figure 8, decomposes the section into one or more variable-length phrases for constitution of the second level structure.

Then the chord generation component generates the chord for each bar based on the music style model. As stated in section 4.2, the music style model consists of the frequent patterns in terms of chords. The chord generation component randomly generates several chord sequences. The more patterns of the music style model contained in a randomly generated chord sequence, the higher the score of this chord

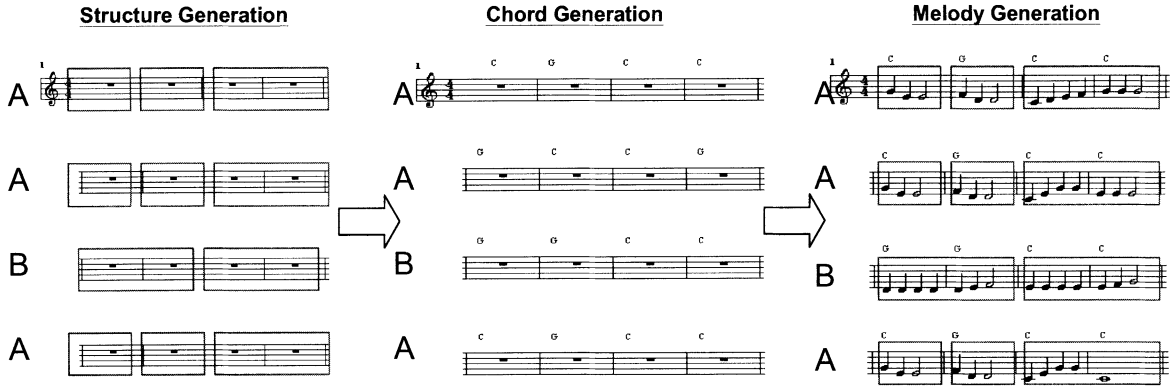

Fig. 7. An example of music generation.

Algorithm Phrase-Arrangement
Input: the length of the section (*sLength*) and the number of phrase in this section (*numLength*)
Output: phrase arranged

1) initialize all *pLength* in this section to *zero*
2) $i=0$
3) $avePhrase = sLength / numPhrase$
4) $e = \arg\min_{x} \left| 2^{x} - avePhrase \right|$
5) $pLength = 2^{x}$
6) while($sLength - pLength > 0$) or ($numPhrase \neq 1$)
7) $\quad$ *phrase[i].pLength=pLength*
8) $\quad$ 60% set *isMotivic* is *true*, 40% is *false*
9) $\quad$ *sLength=sLength-pLength*
10) $\quad$ $numPhrase = numPhrase - 1$
11) $\quad$ *i++*
12) *phrase[i]=sLength*
13) except the first phrase, set parameter *isVariance* of all phrase with the same length to *true*

Fig. 8. Phrase-Arrangement algorithm.

sequence. The chord sequence with the highest score is assigned to the respective bar.

After the structure and chord information are determined, the melody generation component works as follows. For each phrase, the melody generation component selects a motif from the motif selection model. In general, the duration of a motif is shorter than that of a phrase. The selected motif is developed (repeated) based on the major ways of motif development mention in section 4.3.

To ensure that the developed sequence of motives is harmonic to the determined chord sequence, an evaluation function is employed to measure the harmonization between a motif sequence and a chord sequence. This evaluation function is, in fact, the inverse function of chord-assignment algorithm mentioned in section 4.2. In melody style mining, given a melody, the chord-assignment tries to find the best accompanied chord sequence. Here, given a chord sequence, the evaluation function tries to find the best accompanied motif sequence. If the developed motif sequence is evaluated to be disharmonious, the melody generation component selects another motif from the motif selection model and develops the motif variation. This process is repeated until a harmonic motif sequence is produced.

Note that, from the music structure point of view, some sections are associated with the same label. An example shown in Figure 9 is the section one and three. These two sections are all associated with label "A." For those phrases contained in the repeated section, the motif sequences are simply duplicated from the motif sequences generated in the phrases of previous section of the same label.

Finally, the melody generation component concatenates the motif sequences along with the corresponding chord sequences to compose the music.

VI. Experiments

To evaluate the effectiveness of the proposed music generation approach, two experiments were performed. One experiment is the discrimination test for discrimination the machine-generated music from human-composed music. The other experiment is to test whether the music style of the generated music is similar to that of the given music objects. It is difficult to evaluate the effectiveness of a computer music composition system because the evaluation of effectiveness in works of art often comes down to individual subjective opinion. In 2001, M. Pearce addressed this problem and proposed a method to evaluate the computer music composition system [10]. Here, we adopt this method to design our experiments.

In the first experiment, the performance of the generated music object was tested by the approach similar to the Turing Test. Subjects were asked to discriminate between the music composed by composer and that generated by the proposed system. Our system was considered to succeed if subjects cannot distinguish the system-generated music from the human-composed music. There were 36 subjects including four well-trained music experts. The prepared dataset consists of 10 system-generated music objects and 10 human-composed music objects. The human-composed music are "Beyer 8", "Beyer 11", "Beyer 35", "Beyer 51", "Through All Night", "Beautiful May", "Listen to Angle Singing", "Melody", "Moonlight", and "Up to Roof."

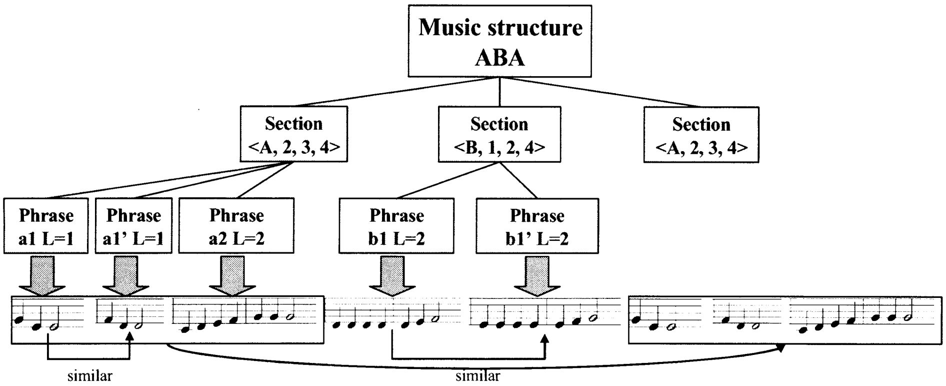

Fig. 9. An example of melody generation.

These music objects are all piano music containing melody and accompaniment. These 20 music objects were sorted randomly and displayed to subjects. Subjects were asked to listen each music object and answer whether it is system-generated or human-composed music. The proportions of correctly discriminated music were calculated from the obtained result (Mean is the average of the accuracy). The significant test is performed with the one-sample t-test against 0.5 (the expected value if subjects discriminated randomly).

TABLE I.
THE RESULT OF THE DISCRIMINATION TEST

	Mean	SD	DF	t	P-value
All subjects	0.522	0.115	35	1.16	0.253
All subjects except experts	0.503	0.106	31	0.166	0.869

SD: *the standard deviation*, DF: *the degree of freedom*, t: *t statistic*.

The result of the experiment test is shown in Table I. The result shows that it is difficult to discriminate the system-generated music objects from the human-composed ones. All subjects (including experts) have little higher discrimination because some of them possess well-trained music background.

In the second experiment, we try to evaluate whether the music style of the system-generated music is similar to that of the given music. We demonstrated our system in the web page,http://avatar.cs.nccu.edu.tw/~stevechiu/cms/experiment2/index.cgi, for subjects. For each round of music generation, subjects were asked to give the score, from 0 to 3, to denote the degree they subjectively feel, from dissimilar to similar. Each subject repeated three times. There were totally 31 subjects to perform this test. The mean of the score is 1.405 and standard deviation is 0.779.

VII. Conclusions

In this paper, we proposed a new framework for a music compositional system. Data mining techniques were utilized to analyze and discover the common patterns or characteristics of music structure, melody style and motif from the given music objects. The discovered patterns and characteristics constitute the music structure, the melody style, and the motif selection model. The proposed system generates the music based on these three models. The experimental results show that the system-generated music is not easy to be discriminated from the human-composed music.

References

[1] E. Cambouropoulos, "The Local Boundary Detection Model (LBDM) and its Application in the Study of Expressive Timing," *Proc. of the International Computer Music Conference, ICMC'01,* 2001.

[2] D. Cope, "Computer Modeling of Musical Intelligence in EMI," *Computer Music Journal, Vol. 16, No. 2,* 1992.

[3] S. Dubnov, G. Assayag, O. Lartillot and G. Gejerano, "Using Machine-Learning Methods for Musical Style Modeling," *IEEE Computer, Vol. 36, No. 10,* 2003.

[4] M. Farbood, "Hyperscore: A New Approach to Interactive, Computer-Generated Music," Master Thesis, Department of Science in Media Arts and Sciences, Massachusetts Institute of Technology, USA, 2001.

[5] M. C. Ho, "Theme-based Music Structural Analysis," Master Thesis, Department of Computer Science, National Cheng Chi University, 2004.

[6] J. L. Hsu, C. C. Liu and Chen, A. L. P., "Efficient Repeating Pattern Finding in Music Database," *In Proc. of IEEE Transaction on Multimedia,* 2001.

[7] N. C. Jones and P. A. Pevzner, "An Introduction to Bioinformatics Algorithms," The MIT Press, 2004.

[8] Y. Marom, "Improvising Jazz with Markov Chains," Ph. D. Thesis, Department of Computer Science, Western Australia University, Australia, 1997

[9] E. R. Miranda, Composing Music with Computers, Focal Press, 2001.

[10] M. Pearce and G. Wiggins, "Towards A Framework for the Evaluation of Machine Compositions," *In Proc. of AISB'01 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, 2001.

[11] M. K. Shan and F. F. Kuo, "Music Style Mining and Classification by Melody," *IEICE Transactions on Information and System, Vol. E86-D, No. 4,* 2003.

[12] B. Thom, "BoB: An Improvisational Music Companion," Ph. D. Thesis, Department of Computer Science, Carnegie Mellon University, USA, 2001.

[13] A. L. Uitdenbogerd and J. Zobel, "Manipulation of Music For Melody Matching," *In Proc. of ACM International Conference on Multimedia, MM'98*, 1998.