Using Quantized Melody Contour to Discover Significant Repeating Figures in Classic Music

Chuan-Wang Chang and Hewijin Christine Jiau
Department of Electrical Engineering
National Chen Kung University
No1.Ta-Hsueh Rd., Tainan City, Taiwan, ROC

Contact:
Chuan-Wang Chang
Department of Electrical Engineering
National Chen Kung University
E-mail: chuan@eembox.ncku.edu.tw
Tel:886-6-2757575 Ext. 62400-717
Fax: 886-6-2345482

Abstract

To develop a content-based music database, it is very important to extract music features from the raw data of music object and organize them as the music index for music retrieval. A repeating pattern is a series of notes which appears more than once in a music object. The musicologists believe repetition is a universal characteristic in music forms. Most of the repeating patterns are themes or easy to remember tone for people. Hence, the themes and other repeating patterns are important music features that can be used for both music data analysis and content-based music retrieval. In this paper, we use the idea of interval between two adjacent notes to form *quantized melody contour* for representing music objects. This representation differs from existing method that uses note to form a melody string. A *figure* is defined as a melody contour of a musical segment. A *sequence pattern* is a melody segment that has the same figure with other melody segment. From musicologist's view, figures are more meaningful than repeating patterns. Hence, we also propose a method to find significant *repeating figures*. These figures could cover most of the repeating patterns got by noted melody string. In addition, the number of repeating figures is less than the number of repeating patterns. As a result, the searching cost including execution time and memory space can be dramatically decreased.

Keywords

Content-based music retrieval, repeating patterns, sequence patterns, repeating figures,

melody contour

1 Introduction

With the rapid growth of digital music/audio data, rapid access of such data is strongly desired, especially for large music database. In the past, the problem of music retrieval based on familiar metadata such as music's title, composer, singer or lyrics has been solved by existing bibliographic retrieval systems [1-5]. However, techniques for retrieval of a piece of music based on the music contents, especially based on an incomplete remembrance, has not yet reach maturity. Research and development related to content-based retrieval of music/audio has been receiving increasing attention in recent years [6-12].

Some early works on content-based of music/audio retrieval are based on signal processing. Wold *et al.* [9] proposed an approach to retrieve audio objects based on the acoustical features of the audio object, such as loudness, pitch, brightness, bandwidth, and harmonicity. In this approach, an N-vector is constructed to classify sounds for similarity searching. Scheirer [10] focused on pitch and rhythmic analysis, segmentation, as well as music similarity estimation at higher level such as genre classification. Tzanetakis *et al.* [11] built tools to distinguish speech from music, and to do segmentation and simple retrieval tasks. Foote [12, 13] experimented with music similarity detection by matching power and spectrogram values over time using a dynamic programming method.

Query by humming is another popular content-based retrieval method for large music database. This kind of system can take a user's acoustic input (a short clip of singing or whistling) via a microphone and then retrieve the desired song from a music database. It is very useful when you want to find a song from music library but forget its title or artist. Ghias *et al.* [14] transform a music object into a string which consists of three kinds of symbols ("U," "D," and "S" which stand for a note is higher than, lower than, or equal to the previous note, respectively). The string can regards as a coarse melody contours of the music object. Then, the problem of music retrieval is transformed into that of approximate string matching. But it simplifies the melody so much that it cannot distinguish music very well, especially when the music database is large. To represent the melody more appropriately, accurately and discriminatively, new features have been proposed. Chou *et al.* [15] use chordal reductions based on melodic line to represent the musical data. Chen *et al.* [16] use rhythmic information for retrieving pieces of music

No matter what the content-based music retrieval is based on, the string matching is the main technique for the music query processing whose performance is dependent on the length of music objects to be matched. For this reason, if the music objects are large, the execution time for query processing may become unacceptable. Hsu[17] proposed an approach for discovering nontrivial repeating patterns in music data. A repeating pattern is a series of notes appearing more than once in a music object. For example, the well-known motive “sol-sol-sol-mi” in Beethoven’s Symphony No. 5 repeatedly appears in the music object. The musicologists believe that repetition is a universal characteristic in music forms [18-22]. Due to the length of repeating patterns is much shorter than a whole music object, to present the music object by its repeating patterns satisfy both semantic and efficient requirements for content-based music retrieval.

However, a motive is often extended by a repetition on different scale degrees. This kind of repetition is called *sequence patterns*. In Hsus’ method [17], the sequence either been regarded as a long repeating pattern gathered with other motives or as an independent motive without any repetition.

By examining the score of a classic music object, we find many sequences in the music object. Since the sequence pattern is a motive that repetition on different scale degree, it has the same figure as other melody patterns. We can treat them as an alternative repetition of repeating patterns. It is useful to develop a music database system, especially a query by humming system that permits key-independent queries.

If the repeating patterns and sequence patterns can be found, the number of repeating patterns can be further reduced. As a result, the search cost including execution time and memory space can be dramatically decreased. Therefore, technique for discovering the sequence patterns from a series of notes of a music object is needs to be developed.

The rest of this paper is organized as follows: In the next Section we present various music features and introduce the semantics of the repeating patterns. We will review some other works of music feature extraction in Section 3. Section 4 gives an overview of the algorithm for finding significant repeating figures. Eventually, in Section 5 we present some accomplished experiments with our method and we conclude the paper with discussion in Section 6.

2 Music Features and Repeating Patterns

To develop a content-based music database, it is very important to extract music features from the raw data of music object and organize them as the music index for music retrieval.

2.1 Three Fundamental Elements in Music

Melody, harmony, and rhythm are three fundamental elements in music object [22]. We will explain briefly these terms in this subsection. For a more comprehensive exposition of music theory, psychoacoustics and psycho-musicology please referred to [18-22].

Melody: a series of musical notes arranged in succession, in a particular rhythmic pattern, to form a recognizable unit.

Harmony: the combining of notes simultaneously, to produce chords, and their successive use to produce chord progressions.

Rhythm: the subdivision of a span of time into perceptible sections; the grouping of musical sounds, principally by means of duration and stress.

Melody is traditionally considered, along with rhythm and harmony, as one of the three fundamental elements in music. It is an oversimplification to regard them as independent, however. Rhythm is an important element within melody itself, not only because each note of the melody has a duration but also because larger-scale rhythmic articulation gives shape and vitality to a melody; while, at least in Western music, harmony often plays a fundamental role in determining the contour and direction of a melodic line, and the harmonic implications of a line of melody may accordingly give it life.

Rhythm is an important element in melody, it affects the progression of harmony, and has a role in such matters as texture, timbre and ornamentation. It is fundamental to the dance; dance patterns, derived from natural rhythms of bodily motion, have dictated many of the rhythmic patterns that pervade Western music.

2.2 Repeating Patterns

Throughout the history of music, composer have employed certain fundamental and basic principles of organization and structure which serve as a mold or framework for the presentation of the materials of music—rhythm, melody and harmony.[18] Even though every composition has its own unique and individual qualities, and is the result of the nature development of composers' ideas, each is based upon the underlying principle of unity and variety, which is achieved by the repetition of musical elements either exact or modified and presentation of new material.

Almost all music is based on the extension and expansion of a brief group of notes called a *motive*. A motive is a short harmonic, melodic, or rhythmic fragment or figure from which a theme, melody, or entire composition is developed. For example, in Fig 1, two groups of sixteenth notes are followed by an exact repetition. Another example is shown as Fig. 2, Respighi repeats a one-measure motive three times in this excerpt.

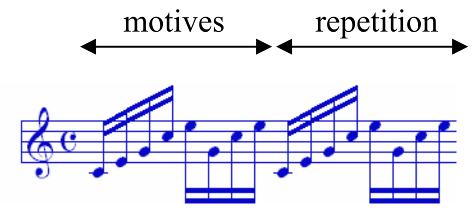

Fig.1 Bach: The Well Tempered Clavier, Book I, Prelude No. 1 in C Major

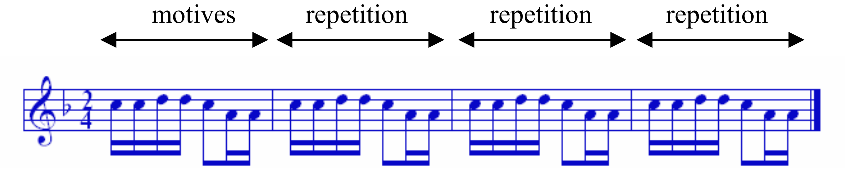

Fig. 2 Respighi: Pines of Rome, Pines of Villa Borghese

However, a motive is often extended by a repetition on different scale degrees. This kind of repetition is called *sequence* and may be an exact or modified duplication involving melody, rhythm, or harmony, separately or together. For example, in Fig. 3, measure 2 is a melodic and rhythmic sequence of the motive in measure1. In Fig. 4, the theme start with a four-note motive, which is repeated in sequence (measure 2 and 3). In the third sequence the note has somewhat altered.

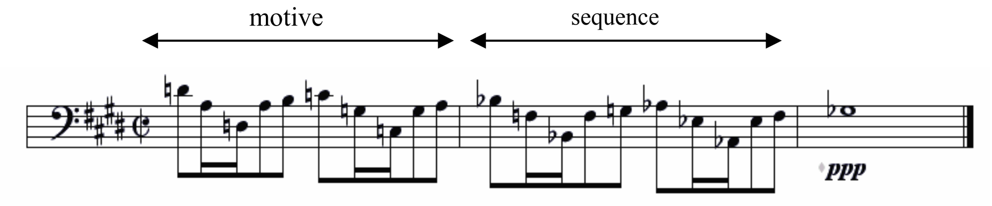

Fig. 3 Anton Bruckner: Symphony No. 7

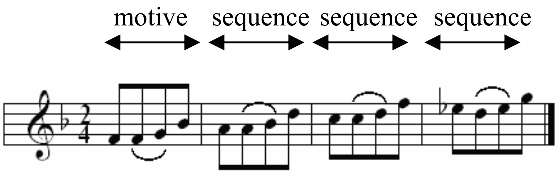

Fig. 4 Beethoven: Symphony No. 6

3 Related Works

As discussed above, the repeating patterns are important features of the music object. We can use the theme or the repeating patterns to construct indices that can speedup music retrieval. A nontrivial repeating pattern discovering method is proposed by Hsu *et al.* [17]. Nontrivial repeating patterns exclude those patterns which are all contained in other longer patterns, such that they can reduce the redundancy of the repeating patterns and speed up music search.

In order to make a description of Hsus' Algorithm, we excerpt a phrase with 16 notes from Tchaikovsky Pathetic, Symphony No. 6, as shown in Fig. 5. Its corresponding melody string ***S*** is "F5-G5-A5-G5-A5-B5-C6-D6- B5-C6-A5-B5-C6-B5-C6-D6-E6-F6-D6-E6." The processes of finding nontrivial repeating patterns are described as follows.
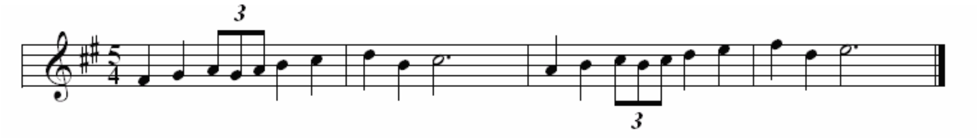
Fig. 5 Tchaikovsky: Pathetic, Symphony No. 6

Let $\boldsymbol{S_i}$ denote the *i*th note in the music string ***S*** of length *n*. The first step of discovering the repeating patterns in the music string $\boldsymbol{S_i}$ is to construct a correlative matrix ***T***. The correlative matrix ***T*** is initialized to an $n \times n$ zero matrix. $\boldsymbol{T_{i,j}}$ indicate the cell of the *i*th row and *j*th column in the matrix ***T***. Then, for the first row of ***T***, if $\boldsymbol{S_1}$=$\boldsymbol{S_j}$, $\boldsymbol{T_{1,j}}$ is set to one, for $2 \le j \le n$. Next, every remaining cell $\boldsymbol{T_{i,j}}$ of the upper triangular part of ***T*** is filled by the following equations:

$$T_{i,j} = \begin{cases} T_{i-1,j-1} + 1, & if \quad S_i = S_j \qquad for \quad 2 \le i \le (n-1), 3 \le j \le n, and \quad i < j \\ 0, & otherwise \end{cases}$$

If the value recorded in $\boldsymbol{T_{i,j}}$ is *n*, it indicates a repeating pattern of length *n* appearing in the positions *(j-n+1)* to *j* in ***S***. Table I shows the result after all notes are processed. For example, by observe the correlative matrix ***T***, we can find a repeating pattern of length 2 in the position 4 to 5 in the melody string ***S***.

TABLE I

Correlative Matrix after all notes are processed

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	F5	G5	A5	G5	A5	B5	C6	D6	B5	C6	A5	B5	C6	B5	C6	D6	E6	F6	D6	E6
F5	—																			
G5		—		1																
A5			—		2						1									
G5				—																
A5					—						1									
B5						—			1			2		1						
C6							—			2			3		2					
D6								—								3			1	
B5									—			1		1						
C6										—			2		2					
A5											—									
B5												—		1						
C6													—		2					
B5														—						
C6															—					
D6																—			1	
E6																	—			2
F6																		—		
D6																			—	
E6																				—

When the correlative matrix is constructed, the next step is to generate all repeating patterns and calculate their repeating frequencies. For every cell $\boldsymbol{T}_{i,j}$ in the correlative matrix $\boldsymbol{T}$, if $\boldsymbol{T}_{i,j}$ is greater than zero, it means that the corresponding pattern $\boldsymbol{P} = \boldsymbol{S}[(j-\boldsymbol{T}_{i,j} + 1): j]$ and all its substrings are repeating patterns.

The candidate set, denoted ***CS***, is employed for finding all repeating patterns and their frequencies. Each element in the ***CS*** is represented by (*pattern, rep_count, sub_count*), where *pattern*, *rep_count*, and *sub_count* represent a repeating pattern, the count of matching to repeating pattern, and the number of the repeating pattern being a proper substring of the other repeating patterns, respectively. The *rep_count* is used to calculate the repeating frequency. The *sub_count* is used to examine whether this repeating pattern only appears as a proper substring of another repeating pattern (i.e., *rep_count* = *sub_count*). They use four cases to illustrate there idea.

Case 1: ($\boldsymbol{T}_{i,j}$=1 and $\boldsymbol{T}_{(i+1),(j+1)}$=0). For example, the value of $\boldsymbol{T}_{3,11}$ is one, which indicates there is a matching to the corresponding substring “A5.” Moreover, the value of $\boldsymbol{T}_{4,12}$ is zero, which means “A5” is not a proper substring of another repeating pattern at this position. We insert (“A5,” 1, 0) into candidate set ***CS***.

Case 2: ($\boldsymbol{T}_{i,j}$=1 and $\boldsymbol{T}_{(i+1),(j+1)} \neq 0$). For example, the value of $\boldsymbol{T}_{2,4}$ is one, which indicates there is a matching to the corresponding substring “G5.” Moreover, the value of $\boldsymbol{T}_{3,5}$ is two, which means “G5” is a proper substring of another repeating pattern (“G5-A5”) here, we insert (“G5-A5,” 1, 0) into the ***CS***.

Case 3: ($\boldsymbol{T}_{i,j}$>1 and $\boldsymbol{T}_{(i+1),(j+1)} \neq 0$). For example, the value of $\boldsymbol{T}_{6,12}$ is two, which indicates the substrings “A5-B5” and “B5” repeatedly appear here, we insert (“A5-B5,” 1, 0) and (“B5,” 1, 0) into the ***CS***. But, since $\boldsymbol{T}_{7,13}$ is three, which indicates the two repeating patterns “A5-B5” and “B5” are proper substrings of the repeating pattern “A5-B5-C6.” We insert (“A5-B5,” 1, 1) and (“B5,” 1, 1) into the ***CS***.

Case 4: ($\boldsymbol{T}_{i,j}$>1 and $\boldsymbol{T}_{(i+1),(j+1)}$=0). For example, the value of $\boldsymbol{T}_{7,13}$ is three, which indicates all of the three substrings “A5-B5-C6,” “B5-C6,” and “C6” repeatedly appear here. Since $\boldsymbol{T}_{8,14}$ is zero, which indicates the repeating pattern “A5-B5-C6” is not a proper substring of another repeating pattern here. We insert (“A5-B5-C6,” 1, 0), (“B5-C6,” 1, 1) and (“C6,” 1, 1) into the CS.

The repeating frequency *f* of the repeating pattern can be calculated by the following equation:

$$rep\_count = \frac{1}{2} f(f-1), i.e., \quad f = \frac{(1+\sqrt{1+8\times rep\_count})}{2}$$

The final step is to remove all of the trivial repeating patterns in the candidate set **CS**. If a repeating pattern has the same *rep_count* and *sub_count*, it will be removed from the **CS**. The remainders in the **CS** are nontrivial repeating patterns. In this example, we find all the nontrivial repeating patterns of the music string ***S***, which are shown as in Table II.

TABLE II

Nontrivial Repeating Patterns Derived From The Phrase Excerpted From Tchaikovsky: Pathetic, Symphony No. 6

No.	Nontrivial Repeating Patterns	Frequency
1	G5-A5	2
2	A5	3
3	A5-B5-C6	2
4	B5-C6	4
5	B5-C6-D6	2
6	D6-E6	2
7	D6	3

Lo *et al.* [23] proposed a fast pattern extracting technique (FastPAT) which can find all nontrivial repeating patterns in music objects speedily and efficiently. The basic idea is based on the correlative matrix proposed by Hsu *et al.* [17]. They use some skills to speedup the process of finding all nontrivial repeating patterns.

4 Proposed Method

Using the correlative matrix approach been illustrated in previous section, we can derive seven nontrivial repeating patterns from the melody string ***S***, as shown in Table II. When we look into the score in Fig. 5, we notice that two melody segments that are composed by the first note to the tenth note and the eleventh to the twentieth, respectively, have the same melody contour. However, using Hsu's algorithm, we could not find any more meaningful repeating patterns from the melody segment.

Here, when we developing a query-by-humming system in a key-independent circumstance. We use the idea of intervals between a series of adjacent notes to make the analysis has the property of key-independency. The melody segments have the same contour could be considered as a repeating figure. By the method of using intervals between adjacent notes to describe music feature, hopefully, we are going to find more significant repeating figures. Eventually, reduce the number of repeating patterns and promote the query efficiency.

In this section, the basic idea of our approach will be illustrated through an example. Please refer to the melody segment in Fig. 5. Let S_i denotes the *i*th note in the music string. First, we can get a *quantized melody contour string* ***QS*** which has the length of 19. Each element in the quantized melody contour string is derived by S_i-S_{i-1}, i.e., the difference between two adjacent notes. The melody may be represented as a string of pitch intervals. For example, the first element of ***QS*** is the result by subtracting the first note (F5) from the second note (G5). The corresponding melody contour string ***QS*** is ”1-1-(-1)-1-1-1-1-(-2)-1-(-2)-1-1-(-1)-1-1-1-1-(-2)-1.”

Secondly, we construct a $(n-1)\times(n-1)$ correlative matrix ***T***. $T_{i,j}$ indicate the cell of the *i*th row and *j*th column in the matrix ***T***. In order to simplify the calculation, we need only take care of the upper triangular part of ***T*** (*i.e.* those cells of $T_{i,j}$ where i < j). The procedure of constructing the correlative matrix is followed the pseudo code in Fig. 6. The result is shown in Table III.

```
// Create Correlative Matrix
// i: row number; j: column number; n: the length of melody contour
for (i = 1; i <= n-1; i++)
  {   for (j = i+1; j<= n-1; j++)
      {   if ($S_i \neq S_j$)
                  $T_{i,j}$ =0;
          if ($S_i$=$S_j$)
              {   if (i=1)
                      $T_{1,j}$=1;
                  else
                      $T_{i,j}$ =$T_{i-1,j-1}$+1;
              }
      }
  }
```

Fig. 6 Pseudo code for constructing correlative matrix

TABLE III

Correlative Matrix After All Intervals Are Processed

position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
intervals	1	1	-1	1	1	1	1	-2	1	-2	1	1	-1	1	1	1	1	-2	1
1	—	1		1	1	1	1		1		1	1		1	1	1	1		1
1		—		1	2	2	2		1		1	2		1	2	2	2		1
-1			—										3						
1				—	1	1	1		1		1	1		4	1	1	1		1
1					—	2	2		1		1	2		1	5	2	2		1
1						—	3		1		1	2		1	2	6	3		1
1							—		1		1	2		1	2	3	7		1
-2								—		2								8	
1									—		3	1		1	1	1	1		9
-2										—								2	
1											—	1		1	1	1	1		3
1												—		1	2	2	2		1
-1													—						
1														—	1	1	1		1
1															—	2	2		1
1																—	3		1
1																	—		1
-2																		—	
1																			—

Then, beginning from the first row, we search for the cell **$T_{i,j}$**, where **$T_{ij,}$** satisfy the following conditions: **$T_{i,j}$** = 1, $T_{i+l-1,j+l-1} \geq 2$, and $T_{i+l,j+l} = 0$, in which the l represent the length of repetition.

However, a short repeating pattern is not significant enough. It appears frequently in most musical melody and could not provide meaningful information for analysis. Hence, we ignore the patterns with $l < 2$. For example, **$T_{1,2}$** = 1 and **$T_{2,3}$** = 0, we get l=1. It means that there is a repeating pattern with length l+1=2, and the corresponding notes are G5 and A5.

In this example, when we checked out the values of **$T_{i,j}$**= **$T_{1,4}$** = 1, in the correlative matrix, we found that **$T_{2,5}$** = 2, and **$T_{3,6}$** = 0; therefore, we got the result of l = 2.

Then, we go through every cells on row (i+l-1) started from **$T_{i+l-1,i+l}$** to the right hand side, in order to check out if there is any cell with length greater than or equal to l. For the purpose of finding so called *sequence,* we discard those repetitions formed by overlapping to reduce the times of comparison in correlative matrix. For example, when we started the search from **$T_{i+l-1,i+l}$ = $T_{1+2-1,1+2}$ = $T_{2,3}$** to find out is there any cell with length greater than or equal to l (>=2, in this example).

In order to determine whether any overlapping of repetition exists, we use *co* to stands for the distance of one cell from its previous cell. Fig.7 shows the procedure of checking that is there any overlapping figure exist. If (**$T_{i+l-1,k}$** >= l) ***AND*** (co >= l), it means we get a repeating pattern with length of l+1 started from the position of k-l+1. The k indicates the position starting from column (j+2) that when we check the ith row of the correlative matrix. For example, when we check the cell **$T_{1,4}$** =1, according to the previously statements, for **$T_{2,5}$** = 2, we get that there is a repeating figure on the position of 4 with length of l+1=3. When we checked out that is there any cell where the value is greater than or equal to 2, started from the position of **$T_{2,5}$** to the right, we found that **$T_{2,6}$** = **$T_{2,7}$**=2. However, none of them satisfied the restriction $co \geq l$. That means the repeating figure from 5 to 6 and the repeating figure from the position of 4 are overlapped. Here, we could ignore those cells with co < l. After examining those cells with value greater than or equal to 2 in row 2, we found that the melody segment which have the length of 3 and the same melody contour are started from position 1, 4, 11, and 14.

The result could be used to determining if we will go further checking in other rows. For example, when we found out that **$T_{2,12}$** =2 and satisfied the condition of ($T_{i+l-1,k}$ >= l) ***AND*** (co >= l). We set the value of **$T_{1,11}$** to -1. It means that there is a repeating pattern with the length of 2+1=3 on position 11. Therefore, we do not need to do any further work when we found that $T_{4,11} = 1$ and $T_{5,12} = 2$. It means that there has exists a repeating pattern with length 3 already.

```
//check overlapped figure from $T_{i+l-1, i+1}$ t0 $T_{i+l-1, |QS|}$
for (k= j + 2; k <= n - 1; k++)
   {
      co = 0;                    //check_overlap
      if ($T_{i+l-1, k}$  < $l$ )
         co = co + 1;
      if (($T_{i+l-1, k}$ >= $l$ ) AND (co >= $l$ ))
         {
            //Get a repeating figure with length of $l$+1 started from the position of
               starting from $k-l+1$ //
            $T_{i,k-l+1}$ = -1
            co = 0;
         }
      if (($T_{i+l-1, k}$ >= $l$ ) AND (co < $l$ ))
         {
            co = co + 1;
            $T_{i,k-l+1}$ = -1
         }
   }
```

Fig. 7 Pseudo code for checking overlapped figures

The whole process of finding significant repeating figures is summarized in Fig. 8.

```
// The whole process of our proposed method
  {
     Transfer the melody string S into quantized melody contour string QS;
     Construct the correlative matrix T of QS;
     for ($i$ = 1; $i$ <= |QS|; $i$ ++)
     {
        for ($j$ = $i$ + 1; $j$ <= |QS|; $j$++)
           {
              if ($T_{i,j}$ = 1)
                {
                    Check the length (say $l$) of repletion from $T_{i\,j}$ to $T_{i+l-1,j+l-1}$;
                    If ($l$ >=2)
```

```
                {
                  if ($T_{i,j} \neq -1$) OR (Find_Figure(fig, $l$) in FigureSet = FALSE)
                    Check overlapped figures from $T_{i+l-1, i+l}$ to $T_{i+l-1, |QS|}$;
                    Add_to_FigureSet(fig, $l$, $S_{j-l+1}$);
                }
              }
          }
    }
}
```

Fig. 8 The whole process of our proposed method

By the derivation based on our algorithm, we produced repeating figures. The homologous patterns represented by every figure as a result shown in Table IV. It is worthy to be mentioned that there are 5 repeating figures listed in Table IV, which including the figures with length 1. They could represent every pattern that Hsu’s algorithm could do and actually, using our method, the result is 2 less than that of Hsu’s algorithm did in Table II. Moreover, if we ignore those figures with length 1, there are only 4 meaningful figures left. It means that we have the deduction of 43%. The Bold words in Table IV represented those patterns found by Hsu's method. In Table IV, the repeating figures can cover every repeating pattern that Hsu’s got except the repeating patterns with length of one. Besides, in this example, we found a more meaningful repeating figure: (1, 1, -1, 1, 1, 1, 1, -2, 1), it could be used to represent two different melody fragments. They are "F5-G5-A5-G5-A5-B5-C6-D6-B5-C6" and "A5-B5-C6-B5-C6-D6-E6-F6-D6-E6".

TABLE IV

The Repeating Figures and There Homologous Patterns

No.	Repeating Figures	Homologous Patterns
1	1	F5-G5, **G5-A5**, A5-B5, **B5-C6**, C6-D6, **D6-E6**, E6-F6
2	1, 1	F5-G5-A5, G5-A5-B5, **A5-B5-C6**, **B5-C6-D6**, C6-D6-E6, D6-E6-F6
3	1, 1, 1	G5-A5-B5-C6, A5-B5-C6-D6, B5-C6-D6-E6, C6-D6-E6-F6
4	1, -2, 1	C6-D6-B5-C6, B5-C6-A5-B5, E6-F6-D6-E6
5	1, 1, -1, 1, 1, 1, 1, -2, 1	F5-G5-A5-G5-A5-B5-C6-D6-B5-C6, A5-B5-C6-B5-C6-D6-E6-F6-D6-E6

5 Experiments

We analyze music structure using data on the MIDI format. MIDI is an encoding format which encodes and relays performance-related information about digital music. A MIDI file contains instructions to perform particular commands, such as note on, note off, events and timing information [24].

All MIDI files are parsed to extract their corresponding melody strings. The note count is defined as the number of distinct notes appearing in a music file. Here, we implement both Hsu's method and ours. The result of the experiment is listed in Table V. We list five familiar music objects including nursery rhyme that everybody familiar with and classic music composed by Haydn, and Bach. The length of those music objects are ranging from 49 to 455. We compared the number of repeating patterns and repeating figures generated by both methods, respectively. Obviously, our method turns out to get less repeating figures, as shown in the 4th field in Table V. The result could be even better if we eliminate short repeating figures with length $l<2$, as shown in the 5th field in Table V. Besides, by checking these figures one by one, we also illustrated that the repeating figures we got cover every repeating pattern that Hsu's got. That means using our method, we could find less but typical repeating figures and more than that it is more meaningful.

TABLE V

Information of Song	Note count	No. of RPs by Hsu's method	No. of RPs by our method ($l>=1$)	No. of RPs by our method ($l>=2$)
Little Bee (nursery rhyme)	49	20	15	10
SONATE, Opus 2 Nr. 2, Joseph Haydn	129	55	34	24
SONATE, Opus 2 Nr. 1, Joseph Haydn	132	51	43	36
INVENTIO 8, J. S. Bach	289	101	64	53
FUGA XV, J. S. Bach	455	166	113	100

RPs: Repeating patterns

6 Conclusions and Future Work

In this paper, we propose a method to find significant repeating figures in classic music. The repeating figures can be organized as the music index for music retrieval. In our method, we first transfer the melody string into a quantized music contour string by subtract the previous note from the adjacent later note. A sequence pattern is a motive which extended by a repetition on different scale degrees. Then, we make use of the correlative matrix to generate all significant repeating figures.

The result of our experiment illustrates that using the idea of sequence patterns could greatly reduce the number of repeating patterns. As you can see in our experiment results, the number of the repeating figures we got is dramatically less than any of that other researcher got. In addition, the repeating patterns we found are more meaningful than other's.

Even though, there are various features in music. Since repetitions may have slight variations to enrich the melody. The *sequence of a node* is only one of those features. We found that in our method, *strict sequence* will give us a better result. Therefore, we believe that if we add the feature of rhythm into our consideration, hopefully we will find more useful repeating patterns. Hence, in the future work, we are going to do some efforts on integrating some more music features such as melody and rhythm into our experiment to find the *variant sequence* in music. Finding more significant repeating patterns is our destination. On the other hand, combine the technique of audio signal processing and the technique of music feature extraction and apply them on the field of MP3 and real time music is also our future goal of efforts.

References

[1]Van Rijsbergen, C.J., A non-classical logic for information retrieval. Computer Journal, 29(6):481-485, 1986.

[2]Biswas, G., J.C. Bezdek, V. Subramanian and M. Maerques, Knowledge-assisted document retrieval: II. The retrieval process. Journal of American Society of Information Science, 38(2):97-110, 1987.

[3]Turtle, H.R. and W.B. Croft, Inference networks for document retrieval. Proc. of ACM Conf. On Research and Development in Information Retrieval, pp.1-24, 1990.

[4]Turtle, H.R. and W.B. Croft, A comparison of text retrieval models. The Computer Journal, 35(3):279-290, 1992.

[5]Ramesh Jain, InfoScope:Multimedia Information Systems. http://vision.ucsd.edu/papers/infoscope/ infoscope.html, 1995.

[6]Thom Blum, Doug Keislar, James Wheaton, and Erling Wold, Audio Analysis for Content-Based Retrieval. http://www.musclefish.com/cbr.html, 1996

[7]Silvia Pfeiffer, Stephan Fischer and Wolfgang Effelsberg, Automatic Audio Content Analysis. In Proceedings of the Fourth ACM International Multimedia Conference, pp.21-30, 1996.

[8]Y.H. Tseng, Content-Based Retrieval for Music Collections. In Proceedings of ACM SIGIR'99, pp.176-182, 1999.

[9]Wold, E., Blum, T., Keislar, D., and Wheaton, J., Content-based Classification, Search, and Retrieval of Audio. IEEE Multimedia, Vol.3, No.3, pp.27-36, 1996,.

[10]E. D. Scheirer, Pulse Tracking with a Pitch Tracker", in Proc. Workshop on Applications of Signal Processing to Audio and Acoustics, 1997.

[11]G. Tzanetakis and P. Cook, "Audio Information Retrieval (AIR) Tools", in International Symposium on Music Information Retrieval, 2000.

[12]J. Foote, An Overview of Audio Information Retrieval. Multimedia Systems, pp.2-10, ACM Press & Springer-Verlag, 1999.

[13]J. Foote, "ARTHUR: Retrieving Orchestral Music by Long-Term Structure", in International Symposium on Music Information Retrieval, 2000.

[14]A. Ghias, J. Logan, D. Chamberlin, and B.C. Smith, Query by Humming: Musical Information Retrieval in an Audio Database, Proc. of ACM Multimedia, pp.231-236, 1995.

[15]T. -C. Chou, A. L. P. Chen, and C. –C. Liu, Music databases : indexing techniques and implementation, in Proceedings of the IEEE International Workshop on Multimedia Data Base Management Systems, 1996.

[16]J. C. C. Chen and A. L. P. Chen, Query by rhythm- an approach for song retrieval in music databases, in Proceedings of the Eight International Workshop on Research Issues in Data Engineering, 1998.

[17]J.-L. Hsu, C.-C. Liu, and Arbee L.P. Chen, Discovering Nontrivial Repeating Patterns in Music Data. IEEE Transactions on Multimedia, Vol.3, No.3, pp.311-325, 2001.

[18] Charles W. Walton, Basic Forms in Music, Alfred Publishing, 1974

[19]C. L. Krumhansl, Cognitive Foundations of Musical Pitch. London, U. K.: Oxford Univ. Press, 1990.

[20]E. Narmour, The Analysis and Cognition of Basic Melodic Structures, Chicago, IL: Univ. of Chicagi Press, 1990.

[21]V. Bakhmutova, V. D. Gusev, and T. N. Titkova, "The search for adaptations in song melodies," Computer Music Journal, Vol. 21, No. 1, pp. 58-67, 1997.

[22]The Classical Music Pages, http://w3.rz-berlin.mpg.de/cmp/classmus.html

[23]Y.-L. Lo, H.-C. Yu, and M.-C. Fan, FastPET: A Fast Non-trivial Repeating Pattern Extracting Technique for Music Data. National Computer Symposium, 2001.

[24]Dustin Caldwell, "Standard MIDI File Format."