Int. Computer Symposium, Dec. 15-17, 2004, Taipei, Taiwan.

A Practical Music Retrieval System Based on Sliding Melodic Contour

Chuan-Wang Chang, Chuan-Yu Chang[†], and Hewijin Christine Jiau
Department of Electrical Engineering, National Cheng Kung University
†Department of Electronic Engineering, National Yunlin University of Science and Technology
chuan@ee.ncku.edu.tw

Abstract*- In this paper, we proposed a contour-based music retrieval system that can efficiently and properly retrieves music according to user's query. To enhance the fault tolerance of user's query, we use interval instead of note as the basic unit of music. We also present an index structure by using sliding window technology. Moreover, we analyzed the effects caused by varied length of query and segmentation. Experiments show the satisfying and exhilarating results in the proposed system. The average time for music retrieval is less than one second.*

Keywords: music retrieval system, sliding window

1. Introduction

The success of multimedia techniques and the growth of the Internet have ensured that huge volumes of multimedia data are available all around us. However, how to access these multimedia data effectively is an important issue. Many countries invest a lot of time and money in developing digital library to help multimedia data retrieval. Content-based access is one of the most important ways to retrieve objects of a digital library. The music information retrieval (MIR) systems had been designed to carry out content-based music retrieval.

Music retrieval is preceded by some kind of feature extraction that is done over entire collection of music data. The extracted features, such as melody, rhythm, and chord, were used as description for the music data and define the search space. In the retrieval phase, features are extracted from the input query and compared with the extracted data of the music collection.

Wold *et al*. [1] proposed a music retrieval system for short sound files. They extracted loudness, pitch, brightness, and harmonic from music data for further processing. Pfeiffer *et al*. [2] developed a query by example system that extracted the fundamental frequency from music data and presented a correlation operation as retrieve processing. However, it is difficult to express these signal features definitely, especially for people without music training.

Ghias *et al.* [3] transform a music object into a string which consists of three kinds of symbols ("U", "D", and "S" stand for a note which is higher than, lower than, or equal to the previous note, respectively). The string can be regarded as a coarse melodic contour of the music object. Then the problem of music retrieval is transformed into string matching. Chen *et al*. [4-8] proposed an algorithm to find repeating patterns in music data and discussed many techniques for music retrieval such as indexing, querying and string matching.

Most of the prior works suffer from many common difficulties such as the decision of the starting note for searching, increasing of data complexity, and management of database. All of these difficulties affect the retrieve efficiency seriously.

In our previous works [9, 10], we proposed contour-based repeating figure finding algorithms to reduce the volume of music and promote the tolerance of input query. We also built a prototype [11] to investigate related techniques for music retrieval. In [11], a sliding window segmentation technique is proposed. Besides, the pitch score of query segment was measured, and this score was used to rank the retrieved candidates. This system overcomes an important issue that the input query must be the beginning of music. That is we can input any part of music as query. However, the ranked list of retrieved music sometimes is inconsistency with expectation. In this paper, a more precise measurement method based on Euclidean distance was proposed to overcome the drawback in [11]. Experimental results show the satisfying and exhilarating results in the proposed system. We also investigate four critical issues that influence music retrieval. From the experiments, we argue that if the length of input query is longer than 7, we will get a better result.

The rest of this paper is organized as follows. In Section 2, we will outline the architecture of proposed music retrieval system. Section 3 gives a detail description of related techniques that are utilized in proposed system. The accomplished experimental results are presented in Section 4. Finally, we conclude this paper and state the future research in Section 5.

2. Outline of Music Retrieval System

The music format of the proposed contour-based music information retrieval system is represented with MIDI [12, 13]. In this system, songs were represented by a sequence of notes that is obtained from the main melody track of the MIDI file. Figure 1 shows the outline of the proposed system.

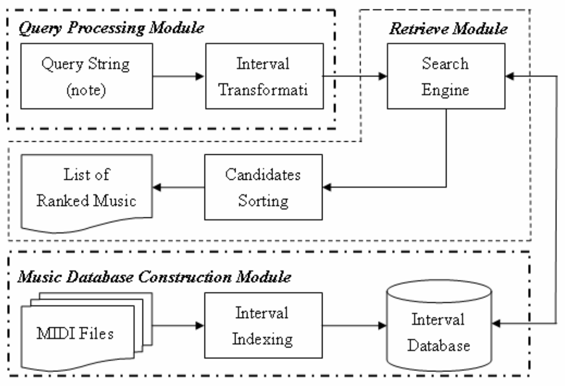


Figure 1. Outline of the music retrieval system

This system composed of three main modules: *music database construction module*, *query processing module*, and *retrieve module*.

In music database construction module, songs stored in the database were segmented into specific length. In this module, a sliding window indexing approach was applied to speedup the retrieval processing from large music database. Details about segmentation and indexing are described in Section 3.2.

In query processing module, user's query is obtained by keying part of the target music using a keyboard-like interface. Then, the input query is transformed into interval string.

The system then retrieves music according to the user's query. In retrieve module, the Euclidean distance between input query and songs in database was calculated. The retrieved songs were ranked based on the similarity.

3. Proposed Method

In the following subsections, we describe some important techniques used in our proposed system such as melodic contour transform, database construction, searching schemes, and candidates ranking.

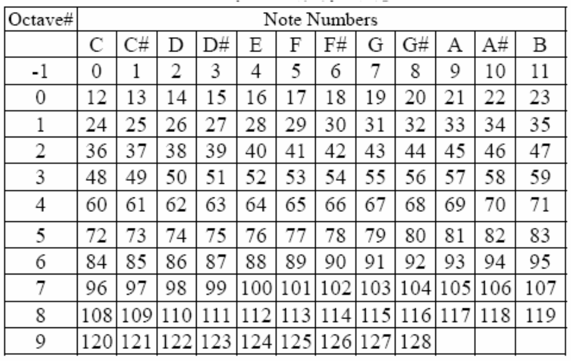
3.1 Melodic contour

Because of incomplete or imperfect recall, the key of user's query may different from the original music. To enhance the fault tolerance, all the original songs were transformed into melodic contours.

Let N_i denotes the *i*th note of the melody string. The note is represented as a note number according to the standard MIDI format [12, 13], as shown in Table 1. Based on the Ghias's method [3], a melodic contour describes a series of relative pitch transitions. A note of music is compared with a previous note. The representation of melodic contour in our system is defined as

$$S_i = \begin{cases} 1 & if \quad N_{i+1} > N_i \\ 0 & if \quad N_{i+1} = N_i \\ -1 & if \quad N_{i+1} < N_i \end{cases} \qquad (1)$$

Table 1. Note Number of Standard MIDI



Octave#	Note Numbers											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
-1	0	1	2	3	4	5	6	7	8	9	10	11
0	12	13	14	15	16	17	18	19	20	21	22	23
1	24	25	26	27	28	29	30	31	32	33	34	35
2	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59
4	60	61	62	63	64	65	66	67	68	69	70	71
5	72	73	74	75	76	77	78	79	80	81	82	83
6	84	85	86	87	88	89	90	91	92	93	94	95
7	96	97	98	99	100	101	102	103	104	105	106	107
8	108	109	110	111	112	113	114	115	116	117	118	119
9	120	121	122	123	124	125	126	127	128			

S1

S2

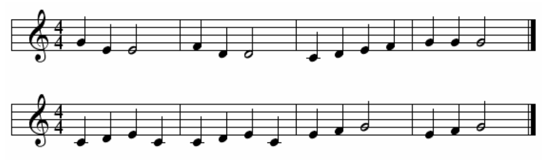


Figure 2. Two excerpts of nursery rhyme

For instance, in Figure 2, there are two excerpts of nursery rhyme. They can be represented by notes as S1:G4 E4 E4 F4 D4 D4 C4 D4 E4 F4 G4 G4 G4 and S2: C4 D4 E4 C4 C4 D4 E4 C4 E4 F4 G4 E4 F4 G4, respectively. By Equation (1), they are transformed into melodic contour as S1: -1 0 1 -1 0 -1 1 1 1 1 0 0 and S2: 1 1 -1 0 1 1 -1 1 1 1 -1 1 1.

3.2 Segmentation and Storing

In order to allow the user's query can be any part of melody, we partition the melodic contour into many successive segments with specific lengths. Meanwhile, we store the corresponding position and the title of song into interval blocks. Consider a melodic contour with length *n*. If we partition the melodic contour with length *m* which $n \geq m$, we may get $n-m+1$ segments.

Figure 3 shows a case of segmentation. In this case, we can derive nine segments with length 4 from a melodic contour with length 12. Then, information of these nine segments will be stored to corresponding interval blocks. Figure 4 makes a description of storing. The title and position of the first partitioned segment is stored to an interval block with index "-1 0 1 -1". "S1#1" denotes the title of song is S1 and the occurrence position of the partitioned segment is 1.

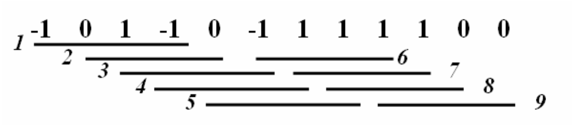


Figure 3. Segmentation with specific length

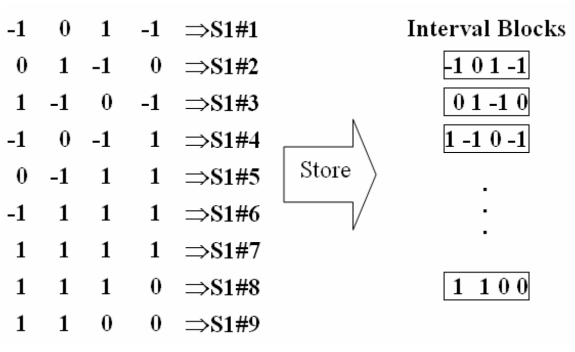


Figure 4. Store segments to corresponding blocks

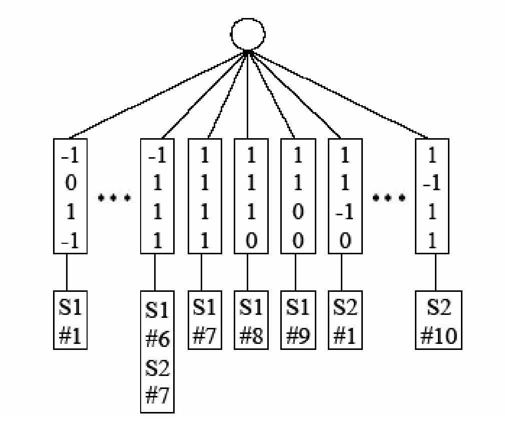


Figure 5. Example of storing structure

Let us illustrate how our segmentation and storing method works by applying it to the music excerpts in Figure 2. If the length of segmentation is 4, we partition each music contours and store the corresponding information to the interval blocks. The storing structure is shown as Figure 5.

It is worthwhile to point out that the complexity and required interval blocks will exponentially increase when the segment length is longer. Consider the case of $m = 4$, there are at most $3^4=81$ interval blocks. If $m = 5$, the number of interval block will become $3^5=243$. To show the influence of segment length and how to decide the appropriate segment length, we conduct many experiments. Related results and discussions will present in Section 4.

3.3 Searching

In our system, there are two searching schemes: ***exactly matching*** and ***similarity matching***. Let us explain these schemes by giving the following four music excerpts:

S3 : e4 d4 d4 c4 e4 e4 d4

S4 : e6 d6 d6 f#5 a5 a5 f#5

S5 : e4 d4 d4 a3 b3 b3 a3

S6 : g4 f4 f4 d#4 f4 f4 c4

All the transformed contour string of these four music excerpts are "-1 0 -1 1 0 -1". Suppose we segment the contour string with length 4, the derived interval strings are "-1 0 -1 1", "0 -1 1 0", and "-1 1 0 -1". We store the corresponding information to the interval blocks, as shown in Figure 6.

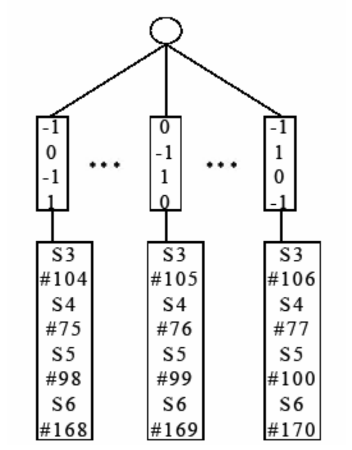

-1 0 -1 1
0 -1 1 0
-1 1 0 -1
S3 #104 S4 #75 S5 #98 S6 #168
S3 #105 S4 #76 S5 #99 S6 #169
S3 #106 S4 #77 S5 #100 S6 #170

Figure 6 . The storing structure of S3~S6

In case of the input query is same as S5, we can also derive three interval strings: "-1 0 -1 1", "0 -1 1 0", and "-1 1 0 -1". Furthermore, we can find four candidates: S3, S4, S5, and S6.

3.3.1 Exactly Matching

The goal of exactly matching is to retrieve the songs that contain identical pitches with the queried segment.

Let us assume that a note-based query pattern $\boldsymbol{N} = \boldsymbol{n}_1\boldsymbol{n}_2 \cdots \boldsymbol{n}_m$ with length *m* is transformed into a contour-based query pattern $\boldsymbol{I} = \boldsymbol{i}_1\boldsymbol{i}_2 \cdots \boldsymbol{i}_{m-1}$ with length *m-1*. If the contour-based query pattern ***I*** is equal to the index of interval block *id* and the corresponding notes of ***I*** are the same as candidate c_k, we say that the queried pattern ***N*** occurs in c_k .The exactly matching problem is to finding all candidates where the given query pattern ***N*** occurs in the database.

In the above example, we merely care for the candidates: S3, S4, S5, and S6. Because the first segmented interval string of these four songs are the same. The corresponding locations of the first interval block are S3#104, S4#75, S5#98, and S6#168. We shall match the corresponding note of the location with the queried note. We can find that the 98th to 104th notes of S5 is equal to the query string "e4 d4 d4 a3 b3 b3 a3". Hence, S5 is retrieved.

3.3.2 Similarity Matching

In this system, the adoption of melodic contour is to enhance the fault tolerance caused by imperfect memory or perceptive error. Sometimes, many songs representing the same melody will not necessarily be identical. However, they have the same melodic contour. Therefore, the purpose of the similarity matching is to retrieve all songs in the database that have the same melodic contour of query.

We introduce the similarity-matching scheme by an example. If the query notes were "e4 d4 d4 a3 b3 b3 a3", the query is transformed into melodic contour string: "-1 0 -1 1 0 -1". We will derive 4 candidates S3~S6, as shown in Figure 6. Then, we check whether the locations of segmented contour string in corresponding interval blocks are successive. The interval blocks "-1 0 -1 1", "0 -1 1 0", and "-1 1 0 -1" are derived from the input query. We can observe that S3~S6 simultaneously exist in 4 interval blocks and the corresponding locations are successive such as S3#104, S3#105, and S3#106. The same phenomenon occurs in S4~S6. The candidates S3, S4, S5, and S6 are retrieved because they have the same melodic contour.

3.4 Candidates Ranking

In similarity matching, although the contour-based melody permits key–independent query, there are many retrieved music have the same interval-based melodic contour. It is important to rank the retrieved music according to the similarity between retrieved music and the query. In our prior work [11], we have proposed a method to measure the similarity between query and candidates by using segmented pitch score.

In this paper, the similarity is measured by the Euclidean distance which is defined as the following equation:

$$d(Q,C)=\sum_{i=1}^{n}(q_i-c_i)^2 \ , \qquad (2)$$

where $Q=\{q_1,q_2,...,q_n\}$ denotes the query string with length of n and $C=\{c_1,c_2,...,c_n\}$ denotes the retrieved music that has the same interval string of query string. The q_i and c_i refer to the ith note of query string and retrieved music, respectively. The corresponding values of q_i and c_i are defined as Table 1. The smaller Euclidean distance means the higher similarity.

For instance, the Euclidean distance between candidates (i.e., S3, S4, S5, and S6) and the query string "e4 d4 d4 a3 b3 b3 a3" are 84, 3578, 0, and 184, respectively. Therefore, these candidates can be ranked as S5, S3, S6, and S4.

4. Experimental Results

We collected more than 1000 MIDI files from Internet [14, 15] to form the experimental dataset. Each MIDI file was transformed into melodic contour and construct the corresponding database with specific segment length.

In our experiments, we mainly focus on four critical issues that influence the retrieve efficiency.

(1) Comparison between segmented pitch score and Euclidean distance.

We conduct an experiment to investigate the effects that caused by segmented pitch score [11] and Euclidean distance method.

(2) Discussion on the efficiency of varied length of segmentation.

To understand what length of segmentation can reach a better query efficiency, we construct a database by segment all music with varied length. We also carry out many retrieval processes to know the influence of varied segment lengths.

(3) Comparison between contour-based searching and note-based searching.

In our method, note-based melody is transformed into contour-based melody. To show the effectiveness and excellences of our method, a comparison between contour-based searching and note-based searching was accomplished.

(4) Discussion on the efficiency by using varied length of query.

Because the length of query is one of the most important factor that influence the retrieving efficiency. Experiments on different query length were conducted to know what length of user's query could get a better efficiency.

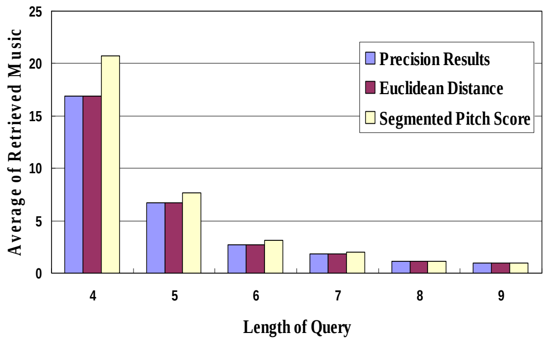


Figure 7. A comparison between different methods

4.1 Segmented Pitch Score vs. Euclidean Distance

To show that the Euclidean distance method may effectively filter out the dissimilar candidates, we randomly select 10 music excerpts as queries for further validation. In query processing module, these excerpts are processed by the method mentioned above. In the retrieve module, the probable candidates are ranked by the segmented pitch score and the Euclidean distance method, respectively. Figure 7 is a comparison between these two methods and precision results. We can find that the numbers of retrieved music are the same as precision result when the similarity matching adopts the Euclidean distance method. However, when using the segmented pitch score, we shall retrieve more impertinent music. In this experimental result, we can know the proposed Euclidean distance method can effectively filter out the dissimilar candidates.

4.2 Efficiency of Varied Length of Segmentation

In music database construction module, the length of segmentation is an important issue that affects the follow-up music retrieval seriously. In this experiment, we analyze the searching time needed by varied length of segmentation, as shown in Figure 8. We can see that the searching time is dramatically reduced as the length of segmentation grows. Both similarity matching and exactly matching schemes can retrieve music in one second when the length of segmentation is equal to or greater than seven.

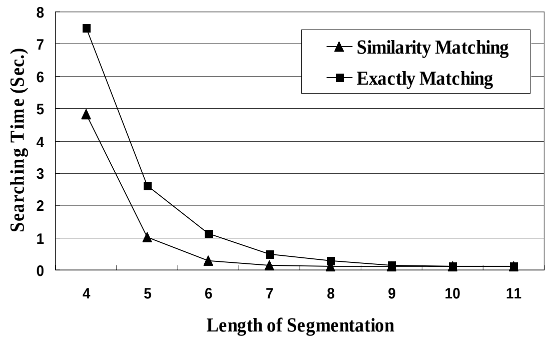


Figure 8. The searching time needed by varied length of segmentation

4.3 Contour-based Searching vs. Note-based Searching

In order to present the benefit of our proposed method, we conduct two experiments that using two kinds of melody representation: contour-based representation and note-based representation. Meanwhile, the query processing is also processed by the same way. We randomly select 10 music excerpts as test queries. Figure 9 shows the average searching time of different methods by using varied length of segmentation. The results show that music retrieval based on contour-based searching performed better than note-based searching.

4.4 Efficiency of Varied Length of Query

In retrieve module, an experiment was carried out to know the influence of varied length of query. We are interested in comparing the number of retrieved music caused by different searching schemes with varied length of query. Obviously, if the length of query is shorter, there are a great many candidates in the corresponding interval blocks. When adopts similarity matching, a shorter length of query may retrieves a great deal of possible music, as shown in Figure 10. In exactly matching, the number of retrieved music is dramatically reduced, as shown in Figure 11.

It is deserves to mention, the number of retrieved music is decreased when the length of query is longer.

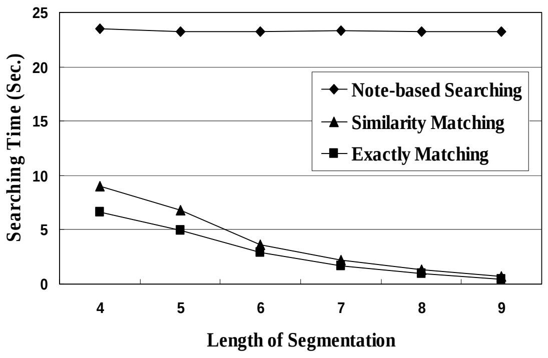


Figure 9. The searching time of different data representation methods

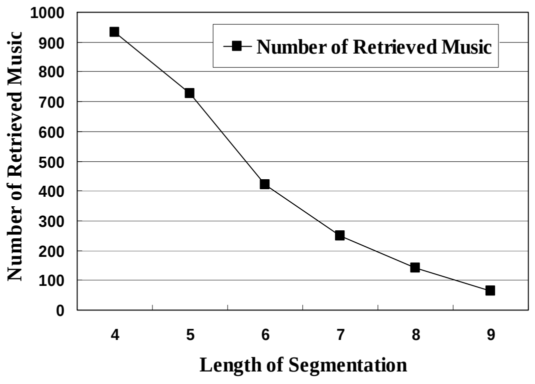


Figure 10. The number of retrieved music by using similarity matching

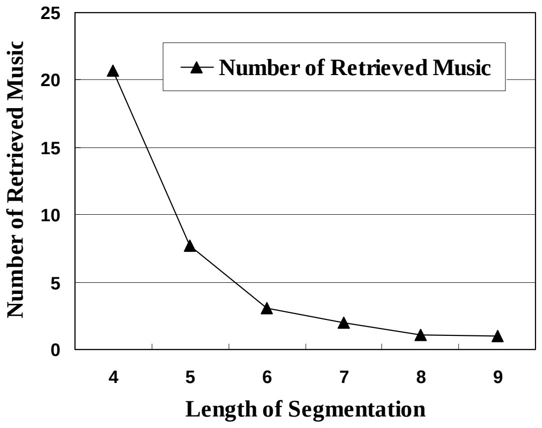


Figure 11. The number of retrieved music by using exactly matching

5. Conclusion

Due to the rapid developing in computing and Internet technologies, human beings are constantly being inundated by multimedia information. The technologies for handling multimedia data are most important and most challenging.

In this paper, we focused on content-based music information retrieval. For the content-based retrieval of a large music database, a unified feature space is desirable. In such a unified feature space, music items can be represented and measures of similarity can defined.

We have proposed a system for matching pieces of music according to similarity. We argue that the success of the matching requires effective extraction, indexing, and similarity techniques. The goal of the proposed melodic contour transformation is to enhance the fault tolerance for input query. Indexing technique can speed up the retrieval. The task of similarity measure is to score the similarity of input query and music items in the database.

The experimental results show the satisfying and exhilarating results in the proposed system. In the proposed system, the average time for music retrieval is less than one second. Besides, four critical issues that influence music retrieval are also investigated. We argue that if the length of input query is longer than 7 notes, we will get a better result.

Reference

[1] 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.

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

[3] Ghias. A, H. Logan, D. Chamberlin, and B.C. Smith, "Query by Humming：Musical Information Retrieval in an Audio Databases," *Proceedings of the 3rd ACM International Conference on Multimedia*, pp231-236, 1995.

[4] A L.P. Chen, M. Chang, J. Chen, J.L. Hsu, C.H. Hsu, and S.Y.S., Hua, "Query by Music Segments：An Efficient Approach for Song Retrieval, " *Proceedings of IEEE International Conference on Multimedia and Expro*, pp. 873-876, 2000.

[5] C.C. Liu, J.L. Hsu, and A.L.P. Chen, "An Approximate String Matching Algorithm for Content-Based Music Data Retrieval, " *Proceedings of IEEE International Conference on Multimedia Computing and Systems*, pp.451-456, 1999.

[6] J.L. Hsu, C.C. Liu, and A.L.P. Chen, "Discovering Nontrivial Repeating Patterns in Music Data," *IEEE Transactions on Multimedia*, pp.311-325, 2001.

[7] T.C. Chou, A.L.P. Chen, C.C. Liu, "Music Databases: Indexing Techniques and Implementation, "*Proceedings of the International Workshop on Multi-Media Database Management Systems*, pp.46-53, 1996.

[8] W. Lee, and A.L.P. Chen, "Efficient Multi-Feature Index Structures for Music Data Retrieval," *Proceedings of SPIE Conference on Storage and Retrieval for Image and Video Databases*, pp.177-188, 2000.

[9] C. W. Chang and H. C. Jiau, "Using Quantized Melody Contour to Discover Significant Repeating Figures in Classic Music," *Proceedings of the International Computer Symposium*, pp. 1641-1648, 2002.

[10] C.W. Chang and H.C. Jiau, "Extracting Significant Repeating Figures in Classic Music by Using Quantized Melody Contour," *Proceedings of IEEE International Symposium on Computers and Communications*, pp. 1061–1066, 2003.

[11] C.Y. Chang and J.F Chang, "Apply Interval Indexing to Music Information Retrieval," *Proceedings of the Symposium on Digital Life and Internet Technologies*, 2004.

[12] MIDI Manufactures Association (MMA), MIDI1.0 Specification, http://www.midi.org

[13] Standard MIDI-File Format Spec.1.1, updated,ht-tp://www.csw2.co.uk/tech/midi2.htm #BMA1_3

[14] CMIDI.com,http://www.cmidi.com/.

[15] MIDI Music Center, http://members.at.infoseek.co.jp/midicentre/index2.html.