The Effectiveness Study of Various Music Information Retrieval Approaches*

Jia-Lien Hsu+, Arbee L.P. Chen, Hung-Chen Chen and Ning-Han Liu
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan 300, R.O.C.
alpchen@cs.nthu.edu.tw

ABSTRACT
In this paper, we describe the Ultima project which aims to construct a platform for evaluating various approaches of music information retrieval. Two kinds of approaches are adopted in this project. These approaches differ in various aspects, such as representations of music objects, index structures, and approximate query processing strategies. For a fair comparison, we propose a measurement of the retrieval effectiveness by *recall-precision curves with a scaling factor adjustment*. Finally, the performance study of the retrieval effectiveness based on various factors of these approaches is presented.

Keywords
music database, retrieval methods, retrieval effectiveness, evaluation platform, performance study.

1. Introduction
With the growth of music objects available, it is getting more attention on the research of constructing music information retrieval systems. To provide an efficient and effective content-based retrieval of music objects, various approaches have been proposed in which the music representations, index structures, query processing methods, and similarity measurements are key issues.

Considering the issue of music representation, several approaches are introduced to model various features of music content, such as pitch, rhythm, interval, chord, and contour. To efficiently process user queries, different kinds of techniques are proposed, including string matching methods, dynamic programming methods, n-gram indexing methods, and list-based and tree-based indexing structures with the corresponding traversal procedures. Due to the great number of approaches proposed, a quantitative and qualitative comparison of these approaches becomes needed [7].

In the traditional information retrieval area, the techniques involved in the evaluation of retrieval systems and procedures have been investigated. The most common evaluation criteria have also been identified, such as precision and recall, response time, user effort, form of presentation, and collection coverage [21].

We initiate the Ultima project to build a platform for the evaluation of music information retrieval systems in terms of their retrieval efficiency and effectiveness. Considering the retrieval efficiency, we focus on the performance study of music representations, indexing and query processing which involve a wide range of techniques [13]. In this paper, we focus on the retrieval effectiveness study.

The rest of this paper is organized as follows. In Section 2, we describe our project for evaluating music information retrieval approaches. The issues of system design, data set, query set generation, and efficiency and effectiveness are also introduced in this section. We introduce the experiment setup and the measurement of retrieval effectiveness, and present the experiment results in Section 3. Section 4 concludes this paper and points out our future directions.

1.1 Related work
Selfridge-Field [22] provides a survey on conceptual and representational issues of music melody. Research works involved in MIR systems are introduced as follows. Ghias *et al*. [10] propose an approach for modeling the content of music objects. A music object is transformed into a string which consists of three kinds of symbols, 'U', 'D', and 'S' which represent a note is higher than, lower than, or the same as its previous note, respectively. The problem of music data retrieval is then transformed into that of approximate string matching.

In [1][6], a system supporting the content-based navigation of music data is presented. A sliding window is applied to cut a music contour into sub-contours. All sub-contours are organized as an index structure for the navigation. Tseng [24] proposes a content-based retrieval model for music collections. The system uses a pitch profile encoding for music objects and an *n*-gram indexing for approximate matching. In [26], a framework is proposed in which the music objects are also organized as an *n*-gram structure for efficient searching. Similar techniques of *n*-gram indexing have also been employed in [8][28][29]. Furthermore, Downie and Nelson [8] provide an effectiveness evaluation of an *n*-gram based MIR system by using statistical analysis.

CIKM'02, November 4-9, 2002, McLean, Virginia, USA.

* This work was partially supported by the R.O.C. Program for Promoting Academic Excellence of Universities (MOE 89-E-FA04-1-4) and the National Science Council (NSC 90-2213-E-007-049).
+Currently affiliated with Department of Computer Science and Information Engineering, Fu Jen Catholic University, Taipei Hsien, Taiwan 242, R.O.C.

McNab *et al*. [20] use dynamic programming techniques to match melodic phrases. The issues of melody transcription and matching parameters are discussed and the relationship between the matching criteria and retrieval effectiveness is shown. Also using dynamic programming techniques, Lemstrom and Perttu [17] present a bit-parallel algorithm for efficiently searching melodic excerpts, which leads to a better performance. Clausen *et al*. [5] design a web-based tool for searching polyphonic music objects. The proposed algorithm is a variant of the classic inverted file index for text retrieval.

To develop a content-based MIR system, we have implemented a system called *Muse* [3][4][16]. In this system, various methods are applied for content-based music data retrieval. The rhythm, melody [25], and chords of a music object are treated as music feature strings and a data structure called *1D-List* is developed to efficiently perform approximate string matching [16]. Moreover, we consider music objects and music queries as chord strings [4] and *mubol* strings [3]. A tree-based index structure is developed for each approach to provide an efficient matching capability. In [3], we propose an approach for retrieving music objects by rhythm. Instead of using only melody [1][4][6][10][16] or rhythm of music data, we consider both pitch and duration information plus the music contour, coded as *music segment*, to represent music objects [2]. Two index structures, called *one-dimensional augmented suffix tree* and *two-dimensional augmented suffix tree*, are proposed to speed up the query processing. By specifying the similarity thresholds, we provide the capability of approximate music information retrieval. When considering more than one feature of music objects simultaneously for query processing, we propose multi-feature index structures [15]. With the multi-feature index, both exact and approximate search functions on various music features are provided. Following these research works on MIR systems, we also present a study on quantitative comparison of efficiency for various retrieval techniques [14].

2. The ULTIMA project

The Ultima project is established with the goal to make a comprehensive and comparative assessment of various MIR approaches. Under the same environment and real data sets, a series of experiments can be performed to evaluate the efficiency and effectiveness of the MIR approaches. Some issues can be explored in depth, such as the threshold setting, the query specification, and the most influential factors which dominate the system performance. Furthermore, heuristics for choosing appropriate representation schemes, indexing structures, and query processing methods when building an MIR system can be provided with the support of the performance study. In this project, the platform will be continuously maintained and served as the testbed whenever new approaches for content-based music information retrieval are proposed.

2.1 System design and implementation

The system is implemented as a web server, which runs on the machine of Intel Pentium III/800 with 1GB RAM on MS Windows 2000 by JDK 1.3. For posing queries at the client end, we provide the ways of humming songs, playing the piano keyword, uploading MIDI files, and using the computer keyboard and mouse. The server end consists of a mediator, four modules, and a data store, as shown in Figure 1.

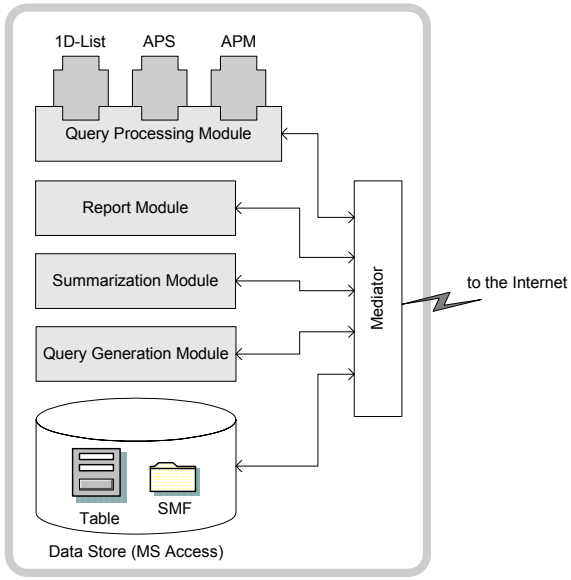

Figure 1: The function blocks of the server in the Ultima project.

The mediator receives user queries and coordinates with other modules. The music objects and the corresponding information, such as title, composer, and genre, are organized as standard MIDI files and relational tables, respectively. The summarization module aims to resemble and visualize query results. The query generation module aims to generate parameterized user queries for performance evaluation, as discussed in Section 2.3. The report module aims to monitor and assess the performance of the system, such as the elapsed time of query processing, storage space of indices, as well as the precision and recall of each MIR approach. The query processing module aims to resolve queries from the client end or the query generation module. The query processing module is designed as a "container" to which each query processing method can be "plugged-in." Whenever a new method is proposed, it can be easily plugged into the module for performing experiments under the same environment. Currently, four methods are considered, *i.e.*, 1D-List [16], APS [2], APM [3], and APC [4]. Among them, two methods are adopted in this project as shown in Section 2.5.

2.2 Data set

The MIR approaches implemented in the platform are all designed for monophonic music objects. Therefore, the testing data (contributed by CWEB Technology, Inc.) is a collection of 3500 single track and monophonic MIDI files. Most of them are Chinese and English pop songs in various genres.

The average size of the music objects in the database is 328.05 notes. When coding these objects by music segments, the average object size is 272 segments. Based on the statistics of the CWEB data set, we estimate that one music segment corresponds to 1.21 notes. The *note count* is defined as the number of distinct notes appearing in a music object. According to the MIDI standard [19], the alphabet size is 128. Therefore, the note count of every melody string is between 1 and 128. For the CWEB data set, the average note count is 13.46.

2.3 Query set generation

In traditional information retrieval research, there exist standard testing data, queries and the associated answers [9][27]. Therefore, a fair performance evaluation can be easily done. In this project, we will also investigate a standard procedure for generating parameterized queries and the associated answers from a data set. The parameterized queries can be tailored for a certain application, targeted user, scenario, and environment. With the variety of queries, a more accurate performance study can be achieved.

2.4 Efficiency and effectiveness studies

In the efficiency study, we design and perform a series of experiments to evaluate the methods of indexing and query processing [13]. Factors influencing the system performance are identified, such as query length, database size, and query approximation degree. The measurement of performance for efficiency is based on memory usage, retrieved candidates and elapsed time for efficiency. In the effectiveness study, a series of experiments are also designed and performed on the same platform. The details of the experiment setup and results are to be presented in Section 3.

2.5 Description of the approaches

We adopted our list-based and tree-based approaches, namely, 1D-List and APS, respectively, in this project. The two approaches cover various methods of music representation, indexing, and query processing, as summarized in Table 1. Due to space limitation, the description of the approaches is skipped. The detailed algorithms of the two approaches can be found in [16] and [2], respectively.

Table 1: The representations and index structures of the two approaches.

Approach	Representation	Index structure
1D-List	Melody string	List-based
APS	Sequence of Music segments	Suffix tree-based

3. The effectiveness study

In this section, we first discuss the measurement of retrieval effectiveness and propose *a recall-precision curve with scaling factor adjustment* for comparing various approaches. Then, the experiment setup of the effectiveness study is described, including the factors to be explored, query samples, and the procedure for performing experiments. Finally, we illustrate the experiment results of the effectiveness study for the two approaches, *i.e.*, 1D-List and APS. In the APS family, 1-D AST (duration), 1-D AST (pitch), and 2-D AST are all implemented.

3.1 Measure of effectiveness

Traditional measures of retrieval performance are *precision* and *recall*, defined as follows [27].

$$precision = \frac{number\ of\ retrieved\ references\ that\ are\ relevant}{number\ of\ references\ that\ are\ retrieved}$$

$$recall = \frac{number\ of\ retrieved\ references\ that\ are\ relevant}{number\ of\ relevant\ references}$$

For better illustration, the retrieval effectiveness in terms of precision and recall will be shown as a diagram of recall-precision curve in which precision is plotted as a function of recall.

Provided that a benchmark (the data set, the query set, and the associated answers) exists, a fair comparison of the retrieval effectiveness among various approaches can be achieved. However, there is no such benchmark dedicated to MIR systems yet. Therefore, in this effectiveness study, we prepare our own queries, and the answers have to be determined by users based on the user perception (we call this a *relevance judgment* from the user).

Example 1:

Suppose there are twenty objects in the database. In Table 2, the first column shows the rank (according to the associated similarity to the query) of objects in the database, and the second column indicates whether the object is relevant to the query, which is decided by the user. In this example, ten objects are confirmed as relevant in the entire database. The third and fourth columns show the corresponding recall and precision, and the associated recall-precision curve (indicated by 'whole') is plotted in Figure 2.

Table 2: The whole ranked results with the associated relevance judgment, recall and precision.

rank	relevance	recall	precision
1	Y	0.1	1
2	Y	0.2	1
3	Y	0.3	1
4		0.3	0.75
5	Y	0.4	0.80
6		0.4	0.67
7	Y	0.5	0.71
8		0.5	0.63
9	Y	0.6	0.67
10	Y	0.7	0.70
11		0.7	0.64
12		0.7	0.58
13	Y	0.8	0.62
14		0.8	0.57
15		0.8	0.53
16	Y	0.9	0.56
17		0.9	0.53
18		0.9	0.50
19	Y	1	0.53
20		1	0.50

According to the definition of recall, the number of relevant references is required. However, to use recall to assess the retrieval quality is infeasible in our experiments because it is unrealistic for the user to make relevance judgments to all the music objects in the database. To solve this problem, *relative recall* was proposed for comparing a number of different retrieval strategies using database and query set [23][11]. The restriction of the relative recall is its inability to compare results from one experiment or database with another. To overcome this problem, we propose a *scaling factor technique* to predict the recall for plotting the recall-precision curve.

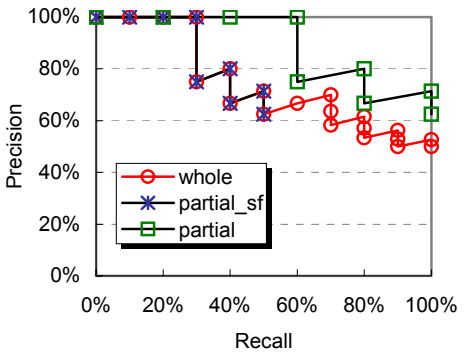

Figure 2: The diagram of recall-precision curves for Example 1.

Suppose that only the top eight objects are retrieved and displayed by a certain approach, as shown in the non-shaded cells in Table 2. There are two ways to predict the total number of relevant objects:

1) optimistic prediction

In this prediction, those unretrieved objects are assumed to be irrelevant. In other words, the total number of relevant objects in Table 2 is assumed to be five. The recall is re-computed and shown in Table 3(a). However, by only considering partial information in the optimistic prediction, the associated recall-precision curve (indicated by 'partial' in Figure 2) may be misleading.

2) scaling prediction

In this prediction, the number of retrieved objects will be considered for a better prediction. We first define the *scaling factor* as follows.

DEFINITION 1: *scaling factor* (sf_x)

Denote AS_x the set of relevant objects from the top x ranked results. The *scaling factor* for AS_x, denoted by sf_x, is defined as follows: $sf_x = |AS_x| / |AS_{|DB|}|$,

where $|AS_{|DB|}|$ means the total number of relevant objects in the entire database. By properly estimating sf_x, we can get $|AS_{|DB|}|$ for computing the recall. Assume that $sf_8 = 0.5$ in Table 2, the total number of relevant objects is 10. The recall is shown in Table 3(b), and the associated recall-precision curve (indicated by 'partial_sf') is plotted in Figure 2. Note that we have no idea about the precision when the recall is more than fifty percent because the real distribution of the $AS_{|DB|}$ is unknown.

Table 3: The top eight ranked results.

(a) The optimistic prediction: total number of relevant objects = 5

Rank	relevance	**recall**	Precision
1	Y	**0.2**	1
2	Y	**0.4**	1
3	Y	**0.6**	1
4		**0.6**	0.75
5	Y	**0.8**	0.80
6		**0.8**	0.67
7	Y	**1**	0.71
8		**1**	0.63

(b) The scaling prediction: total number of relevant objects = 10 ($sf_8 = 0.5$)

Rank	relevance	**recall**	precision
1	Y	**0.1**	1
2	Y	**0.2**	1
3	Y	**0.3**	1
4		**0.3**	0.75
5	Y	**0.4**	0.80
6		**0.4**	0.67
7	Y	**0.5**	0.71
8		**0.5**	0.63

The optimistic prediction can be considered as a special case of the scaling prediction with $sf_8 = 1$. As shown in Figure 2, the 'partial_sf' curve is closely approaching to the 'whole' curve, rather than the 'partial' curve. Apparently, the accuracy of the predicted curve relies on the scaling factor. Unless checking the entire database, there is no way to obtain the scaling factor. Accordingly, we provide the following assumptions to estimate the scaling factor.

ASSUMPTION 1:

Denote RS_x the set of the top x ranked results retrieved by a certain approach. Assume

$$|RS_x| = f(|AS_x|), \text{ for } 1 \le x \le |DB|, \text{ and } |AS_1| = 1.$$

Assumption 1 says that the number of the retrieved results is a function of the number of retrieved relevant objects. Moreover, the first retrieved result is always relevant. Based on our observation in the experiments, the function can be estimated as stated in Assumption 2.

ASSUMPTION 2:

$|RS_x| = B^{|AS_x|} - (B-1)$, where B is a positive integer.

Under the two above assumptions and Definition 3, the scaling factor, sf_x, can be derived as follows:

$$sf_x = |AS_x| / |AS_{|DB|}| = \log_B(|RS_x| + (B-1)) \,/\, \log_B(|DB| + (B-1))$$

When setting $B = 3$ and $|DB| = 3500$, Table 4 shows some examples of sf_x derived from $|RS_x|$. Therefore, for each query, we can predict the total number of relevant objects by using the known values of $|RS_x|$ and $|AS_x|$. Note that in Assumption 2, the function and parameter settings are dependent on the data and approach.

Table 4: The scaling factor sf_x as B=3 and $|DB|$=3500.

|RSx|	sfx
1	0.135
2	0.170
50	0.484
100	0.567
400	0.735
800	0.819
1000	0.847
2000	0.931
3500	1

3.2 Experiment setup

From the data set, ten music objects are randomly chosen for generating query samples. Based on the location and the length of the query sample, four query samples are generated for each music object. The *incipit* denotes that this query sample is taken from the start of a music object. The other situation is that the query sample is taken from the *refrain*. The length of query samples is 6 or 10 music segments for the APS family, and 8 or 12 notes for 1D-List, respectively. When posing queries for each approach, three thresholds are given. All these factors are summarized in Table 5.

The procedure of performing experiments is introduced as follows. As illustrated in Figure 3, the first two steps (denoted by circle 1 and circle 2) of sampling queries and setting thresholds are described in the previous paragraph. After posing queries with various thresholds for each approach, the retrieved results are displayed, in which the most similar result is ranked as the top one, and so on. In the third step (denoted by circle 3), based on user perception of listening, the relevant objects will be marked. Hence, the effectiveness in terms of precision and recall can be obtained, as well as the predicted recall-precision curve.

3.3 Experiment result

First of all, Table 6 shows the sf_x derived from our experiment results with respect to various setups, which is used to plot the recall-precision curves.

Due to the space limitation, the illustration of each recall-precision curve of our experiments is not included in this paper. Table 7 shows the average precision with various experiment setups. In general, the precision of 1D-List is better than the APS family.

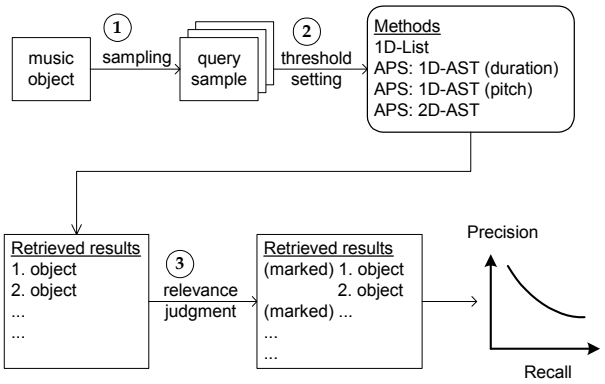

Figure 3: The procedure of performing experiments.

More precisely, 1D-List achieves a high precision in the limited range of recall, while a moderate precision for the APS family can be obtained in the wide range of recall (*cf.* Figure 4 and Figure 5). This is because the music objects in 1D-List are coded as melody strings, and the matching criterion of 1D-List is stricter than APS. Although the precision of the APS family is lower than that of 1D-List, it does allow a less precise query specification, and is therefore more suitable for the non-experienced users.

When comparing the APS family, the precision of the three approaches in a descending order is 1-D AST (pitch), 2-D AST, and 1-D AST (duration). The rationale of this result is as follows. Reported by the users who performed the experiments, the dominant criterion for the relevance judgment is based on the melody (*i.e.*, pitch information). The rhythm, tempo, instruments, and the number of voices are considered for relevance judgment only when the melody is similar. However, in the experiment settings, the pitch and duration information of 2-D AST are treated equally (w_{pitch} = 0.5 and $w_{duration}$ = 0.5). That is, the parameter setting of 2-D AST does not correspond to the user perception. As a result, 2-D AST does not perform better than 1-D AST (pitch). We believe that the effectiveness of 2-D AST would be better when setting appropriate parameters (*e.g.*, w_{pitch} = 0.8 and $w_{duration}$ = 0.2).

In the following, we discuss the impact of each factor on retrieval effectiveness.

- The factor of location (refrain or incipit)

In average, the effectiveness of 'incipit' query is better than 'refrain' query (*cf.* Figure 6 and Figure 7). In 1D-List, there is no difference between 'incipit' query and 'refrain' query. In the APS family, the effectiveness of 'incipit' query is better than 'refrain' query.

From the algorithmic viewpoint of query processing, it should have no difference between the two kinds of queries. However, the results of the APS family do not support the above statement. This phenomenon comes from our design of the user interface. As shown in Figure 8, each retrieved music object is associated with the scroll bar in which a marker indicates the matching location. Users have to manually scroll or click on the location for listening to the matching music fragment of the retrieved objects. As reported by the users, it is not easy to precisely locate the position, although there is a maker and a numeric value of the position. If the users do not listen to the retrieved object at the right position, the similar objects may be determined as irrelevant. The precision can therefore decrease.

In addition, the refrain is usually a theme of the music object. It is well known that the theme is the most memorable part of a music object. Therefore, the retrieved results tend to be marked as irrelevant unless the retrieved results perfectly match what the users keep in mind. On the contrary, when making the relevance judgment for incipit queries, the users may listen to the retrieved results more carefully such that a fair judgment could be achieved.

From the above discussion, a well-designed user interface will be useful to enhance the retrieval effectiveness.

- **The factor of the threshold (low, mid, or high)**
  As expected, the precision decreases when increasing the thresholds for all approaches. Note that for the APS family, the precision with 'mid' threshold (Threshold_2) is the same as the one with 'high' threshold (Threshold_3). That is, Threshold_2 has been set to an improper high value in our experiments such that the retrieval effectiveness of 'mid' threshold and 'high' threshold cannot be distinguishable.
- **The factor of query length (shorter or longer)**
  For 1D-List approach, the precision of longer-length queries is constantly better than that of shorter-length queries. For other approaches, this regularity cannot be achieved, and the rationale needs further investigation.

Table 8: The notations used in the following figures.

Notation	Explanation
R	query is a refrain
In	query is an incipit
1D-AST_Px	th_p = x, for APS 1D-AST using pitch information
1D-AST_Dx	th_d = x, for APS 1D-AST using duration information
2D-AST_PDx	Th_p = x and th_d = x, for APS 2D-AST
1D-List_Kx,y	K = x, for 1D-List (L = 8); K = y, for 1D-List (L = 12)

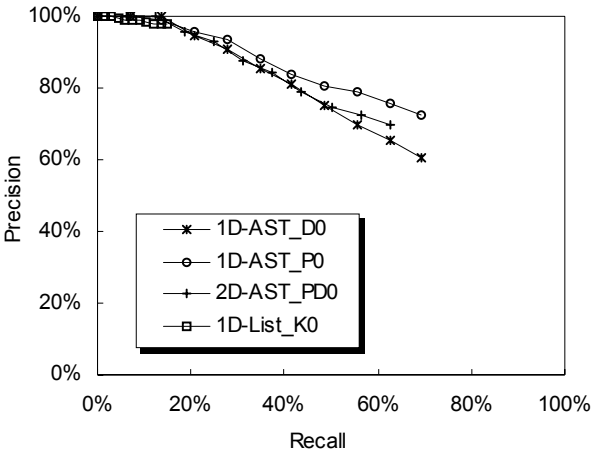

Figure 4: The comparison when setting low thresholds (Threshold_1).

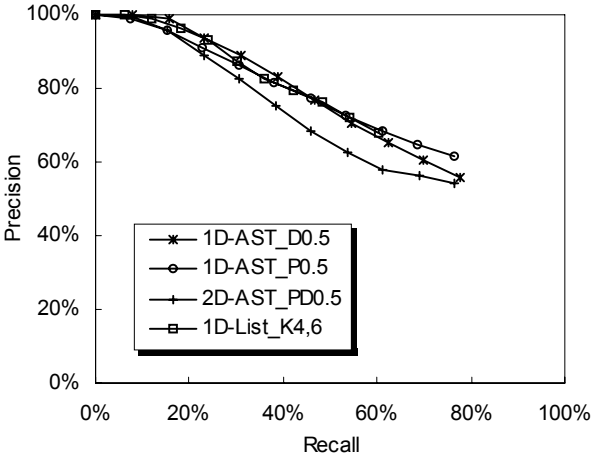

Figure 5: The comparison when setting middle thresholds (Threshold_2).

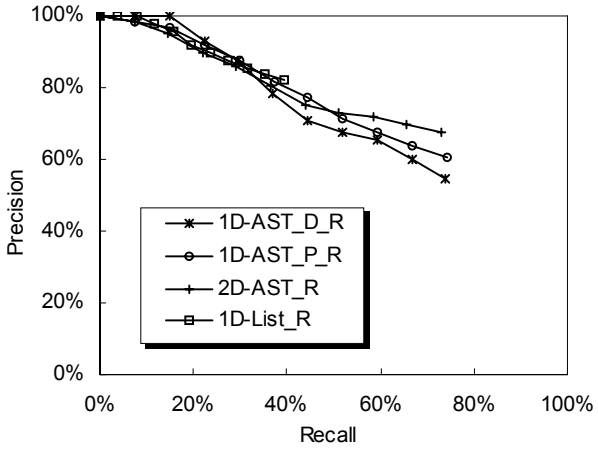

Figure 6: The comparison when queries are refrains.

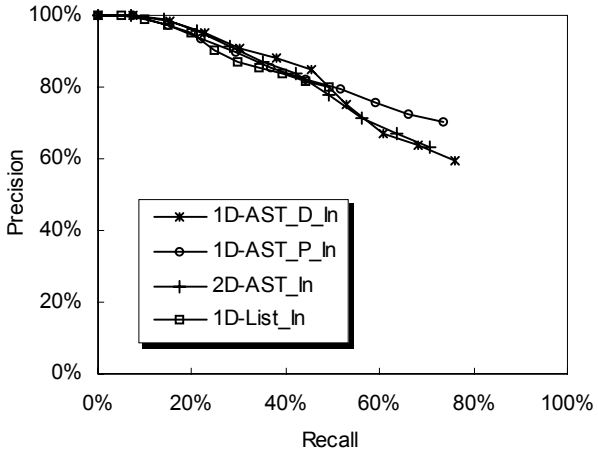

Figure 7: The comparison when queries are incipits.

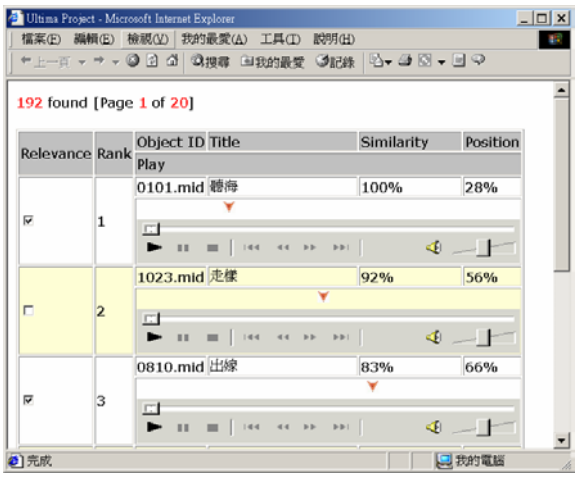

Figure 8: The user interface showing retrieval results.

4. Conclusion

In this paper, we describe the ULTIMA project that aims at building a platform for evaluating the performance of various approaches for music information retrieval. The issues of system design, query set generation, and retrieval effectiveness are discussed. The list-based and augmented suffix tree-based approaches are considered. The factors of the location, the query length and the thresholds for approximate query processing are investigated. To achieve a fair comparison among different approaches, we propose a new measurement of effectiveness, *i.e.*, the recall-precision curve with the scaling factor technique. The experiment results are further analyzed.

Future works include the design and implementation of the summarization module as well as the query generation module. While more and more polyphonic music retrieval methods are proposed, we also plan to extend our project for evaluating these methods.

Acknowledgments

We would like to thank the members of the MAKE (Multimedia and Knowledge Engineering) Lab at National Tsing Hua University, including Chung-Wen Cho, Chang-Rong Lin, Yong-Shun Tzeng, Shih-Hsin Shen, Chia-Ming Chiang, Min-Hong Jian, Xinyi Jiang, Chia-Hsiung Lee, and Amanda Chang, for implementing the methods, constructing the platform, building up the server, and performing the experiments. Also, we would like to thank the CWEB Technology, Inc., for sharing us the data set used in our experiments.

References

[1] Blackburn, S. and DeRoure, D. (1998). A tool for content-based navigation of music. In Proceedings of the ACM Multimedia Conference, (pp. 361-368).

[2] Chen, A. L. P., Chang, M., Chen, J., Hsu, J. L., Hsu, C. H. and Hua, S. Y. S. (2000). Query by music segments: An efficient approach for song retrieval. In Proceedings of IEEE International Conference on Multimedia and Expo. New York.

[3] Chen, J. C. C. and Chen, A. L. P. (1998). Query by rhythm: An approach for song retrieval in music databases. In Proceedings of the 8th International Workshop on Research Issues in Data Engineering, (pp. 139-146).

[4] Chou, T. C., Chen, A. L. P., and Liu, C. C. (1996). Music databases: Indexing techniques and implementation. In Proceedings of IEEE International Workshop on Multimedia Data Base Management System.

[5] Clausen, M., Engelbrecht, R., Mayer, D. and Smith, J. (2000). PROMS: A web-based tool for searching in polyphonic music. In Proceedings of ISMIR.

[6] DeRoure, D. and Blackburn, S. (2000). Content-based navigation of music using melodic pitch contours. Multimedia Systems, 8(3), Springer. (pp. 190-200).

[7] Downie, S. (2000). Thinking about formal MIR system evaluation: Some prompting thoughts. Available at http://www.lis.uiuc.edu/~jdownie/mir_papers/downie_mir_eval.html.

[8] Downie, S. and Nelson, M. (2000). Evaluation of a simple and effective music information retrieval method. In Proceedings of ACM SIGIR, (pp. 73-80).

[9] Frakes, W. B. and Baeza-Yates, R. (1992). Information retrieval: Data structures and algorithms, Prentice-Hall.

[10] Ghias, A., Logan, H., Chamberlin, D., and Smith, B. C. (1995). Query by humming: Musical information retrieval in an audio database. In Proceedings of the ACM Conference on Multimedia, (pp. 231-236).

[11] Goncalves, P.F., Robin, J. T.L.V.L., Miranda, O.G., Meira, S.L. (1998). Measuring the Effect of Centroid Size on Web Search Precision and Recall. In Processings 8th Annual Conference of the Internet Society.

[12] Gusfield, D. (1997). Algorithms on strings, trees, and sequences. Cambridge University Press.

[13] Hsu, J. L. and Arbee L.P. Chen. (2001). Building a platform for performance study of various music information retrieval approaches. In Proceedings of the 2nd Annual International Symposium on Music Information Retrieval (ISMIR 2001). Bloomington, Indiana, USA.

[14] Hsu, J. L. and Arbee L. P. Chen. A quantitative comparison of various techniques for content-based music information retrieval. In Stephen Downie, Tim Crawford, and Don Byrd (eds.). Music information retrieval: audio, midi, and score. Kluwer Academic Publishers (to appear).

[15] Lee, W. and Chen, A. L. P. (2000). Efficient multi-feature index structures for music data retrieval. In Proceedings of SPIE Conference on Storage and Retrieval for Image and Video Databases.

[16] Liu, C. C., Hsu, J. L., and Chen, A. L. P. (1999). An approximate string matching algorithm for content-based music data retrieval. In Proceedings ICMCS.

[17] Lemstrom, K. and Perttu, S. (2000). SEMEX: An efficient music retrieval prototype. In Proceedings of ISMIR.

[18] McCreight, E. M. (1976). A space economical suffix tree construction algorithm. Journal of Assoc. Comput. Mach., 23, 262-272.

[19] MIDI Manufactures Association (MMA), MIDI 1.0 Specification, http://www.midi.org/.

[20] McNab, R. J., Smith, L. S., Witten, I. H., and Henderson, C. L. (2000). Tune retrieval in the multimedia library. Multimedia Tools and Applications, 10(2/3), Kluwer Academic Publishers.

[21] Salton, G. and McGill, M. (1983). Introduction to modern information retrieval. MaGraw-Hill Book Company.

[22] Selfridge-Field, E. (1998). Conceptual and representational issues in melodic comparison. In Hewlett, W. B. and Selfridge-Field E. (Ed.), Melodic similarity: Concepts, procedures, and applications (Computing in Musicology: 11), The MIT Press.

[23] Tague-Sutcliffe, J. (1992). The pragmatics of information retrieval experimentation, revisited. Journal of Information Processing and Management, (pp. 467-490).

[24] Tseng, Y. H. (1999). Content-based retrieval for music collections. In Proceedings of ACM SIGIR.

[25] Uitdenbogerd, A. and Zobel, J. (1998). Manipulation of music for melody matching. In Proceedings of the 6th ACM International Multimedia Conference, (pp. 235-240).

[26] Uitdenbogerd, A. and Zobel, J. (1999). Melodic matching techniques for large music databases. In Proceedings of the 7th ACM International Multimedia Conference, (pp. 57-66).

[27] Witten, I. H., Moffat, A., and Bell, T. C. (1994). Managing gigabytes: compressing and indexing documents and

images, Van Nostrand Reinhold, International Thomson Publishing company.

[28] Yanase, T. and Takasu, A. (1999). Phrase based feature extraction for musical information retrieval. In Proceedings of IEEE Pacific Rim Conference on communications, Computers, and Signal Processing.

[29] Yip, C. L. and Kao, B. (2000). A study on *n*-gram indexing of musical features. In Proceedings of IEEE International Conference on Multimedia and Expo. New York.

Table 5: The factors in the experiment setting.

Factor \ Method	APS			1D-List
	1-D AST (duration)	1-D AST (pitch)	2-D AST	
Number of music objects for generating queries	10			10
Is the query sample a refrain or an incipit?	refrain/incipit			refrain/incipit
Length of query sample, denoted L	6/10 (segment)			8/12 (note)
Number of query samples per music object	4			4
Threshold setting of approximation for a query sample	th_d = 0, 0.5, 1.0 th_p = 0, 0.5, 1.0			K=0, 4, 7 (for L=8) K=0, 6, 11 (for L=12)
Total number of posing queries	120			120

Table 6: The sf_x for various experiment setups.

Setting \ Method		APS						1D-List	
		1-D AST (duration)		1-D AST (pitch)		2-D AST			
Is query a refrain or incipit?		R	In	R	In	R	In	R	In
L=6 (seg.) L=8 (note)	Threshold_1	0.81	0.83	0.82	0.79	0.76	0.70	0.16	0.18
	Threshold_2	0.88	0.89	0.88	0.89	0.89	0.89	0.70	0.69
	Threshold_3	0.88	0.89	0.88	0.89	0.89	0.87	0.71	0.70
L=10 (seg.) L=12 (note)	Threshold_1	0.57	0.57	0.60	0.57	0.55	0.49	0.13	0.13
	Threshold_2	0.65	0.69	0.64	0.64	0.64	0.64	0.51	0.51
	Threshold_3	0.65	0.68	0.64	0.64	0.65	0.64	0.53	0.53

Note that L is the length of query sample.

The threshold setting of approximation is described as follows.

	1-D AST (duration)	1-D AST (pitch)	2-D AST	1D-List (L=8)	1D-List (L=12)
Threshold_1	th_d=0	th_p=0	th_d, th_p=0	K=0	K=0
Threshold_2	th_d=0.5	th_p=0.5	th_d, th_p=0.5	K=4	K=6
Threshold_3	th_d=1.0	th_p=1.0	th_d, th_p=1.0	K=7	K=11

Table 7: The average precision for various experiment setups.

Setting \ Method		APS						1D-List	
		1-D AST (duration)		1-D AST (pitch)		2-D AST			
refrain (R) or incipit (In)		R	In	R	In	R	In	R	In
L=6 (seg.) L=8 (note)	Threshold_1	0.837	0.860	0.808	0.908	0.835	0.897	0.981	0.975
	Threshold_2	0.836	0.845	0.773	0.843	0.756	0.755	0.818	0.805
	Threshold_3	0.836	0.845	0.773	0.843	0.756	0.755	0.803	0.792
L=10 (seg.) L=12 (note)	Threshold_1	0.787	0.871	0.871	0.931	0.840	0.905	1	1
	Threshold_2	0.743	0.825	0.832	0.819	0.804	0.744	0.938	0.909
	Threshold_3	0.743	0.782	0.832	0.819	0.804	0.744	0.883	0.891