QBSH System for MIREX

Xiao Wu
HCCL Lab, Institute of Acoustics,
Chinese Academy of Sciences
xwu@hccl.ioa.ac.cn

Ming Li
HCCL Lab, Institute of Acoustics,
Chinese Academy of Sciences
mli@hccl.ioa.ac.cn

Abstract

This paper describes HCCL lab's submission to the Query-by-Singing/Humming(QBSH) task of Music Information Retrieval eXchange(MIREX) 2006. As we do not participate in the second sub-task, this paper will only deal with the Known-Item Retrieval sub-task. In the submitted system, we apply a novel algorithm called Recursive Alignment(RA) to compute the similarity score between query and candidates. We also employ the multilevel filter strategy to reduce the running time. Finally, we give the evaluation results of the presented system.

Keywords: MIREX, QBSH, Music Information Retrieval

1. System Overview

This section gives a overview of the submitted system.Figure1 presents the framework of the submitted system which originates from [1]. The system consists of four stages: (1) feature extraction (2) filters using part of the query (3) filters using the whole query and (4) final rescoring. Inspired by Viola who introduces cascade filters to detect human faces [2], the system employs seven level filters to efficiently eliminate unlike candidates. Basically the former filters are more efficient but less accurate than the latter ones. Top-down fashioned similarity measure algorithms are selected for the final rescorer and most of the filters. We believe such category of algorithms are more robust to local mismatches caused by note-segmentation erorrs, inaccurate singing and grace notes in the reference. The following sections will describe these stages in detail.

2. Database Preprocess

The database are constructed with monophonic midi files. Common used information such as pitch value, note duration and onset time is included in the database. Besides, we also perform note compression, music phrase segmentation and pentanotes clustering while building the database.

2.1. Note Compression

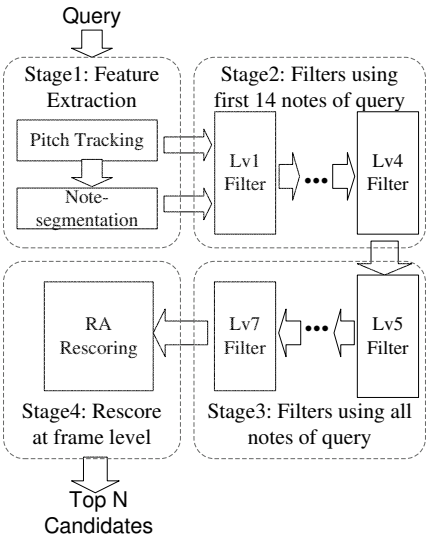

Figure 1. System Framework

As we find in practice that most people shorten the long notes when they sing, we compress the duration d which is longer than the song average duration $\overline{d}$ to

$$\overline{d} + log\left[1 + \alpha(d - \overline{d})\right]/\alpha$$

where α is a predefined constant.

2.2. Music Phrase Segmentation

Our analysis on real world queries shows that more than 98% of them start from the beginning of music phrases. Thus we give each note a weight value to tell how probable this note could be the beginning of certain music phrase. While computing the similarity score, this weight will be considered. The value of the weight is decided by the context such as neighboring rest notes, duration of the previous note and repeating pattern.

2.3. Pentanotes Clustering

We cluster all the neighboring 5 notes into 128 classes using K-Mean algorithm. Meanwhile a forward table and a inverse table are constructed to provide indexing between classes and pentanotes.

3. Features Extraction

Feature extraction includes pitch tracking and note segmentation. The input query is 16bit encoded linear PCM with 8kHz sample rate.

3.1. Pitch Tracking

Pitch value is computed for each window of 25ms where adjacent windows overlap by 15ms. Improved sub-harmonic summation is adopted as the pitch tracking method[3]. Spectral energy is normalized by the average energy around the frequency, which lowers gross errors. Post processing such as median filtering, linear filtering are adopted too.

3.2. Note Segmentation

We segment notes by using an energy-based approach [4]. It is processed as the following procedure. Firstly, voiced sections and unvoiced sections are discriminated apart by adaptive energy threshold. Secondly, notes are segmented by the fluctuation of the harmonic energy and wave energy. Thirdly, notes are split within which pitch fluctuation is beyond a semitone. Finally, notes whose duration is too short are deleted or merged to its adjacent notes.

4. Filtering and Similarity Measure

The system introduces 7-level filters to efficiently eliminate unlike candidates. Each filter keeps very high recall. The former filters are more efficient but less accurate than the latter ones. Candidates which survive cascade filters are passed to rescorer to re-compute the similarity score. A novel algorithm called Recursive Alignment(RA) [1], which outperforms all other competitor algorithms in our experiment for its high precision, is applied in pitch contour for rescoring. Variations of RA which run much faster at the expense of less accuracy are selected by some of the filters.

As is shown in Figure 1, the QBSH system has two filtering stages. In stage 2 only the first few notes of the query (usually the first 14 segmented notes) are used to generate 6000 most probable candidate melody sections. Then the whole query is used to select 500 survivors out of the 6000 candidates in stage 3. The reason we do not use the whole query in stage 2 comes from the consideration of runtime efficiency. Table 1 lists all the filters used in the submitted system.

4.1. Key Detection

Since the query and the candidate are usually from the different keys, we always subtract their own mean pitch during the similarity computation. Furthermore, finer tuning is made in the final rescore stage to determine the best key transposition.

4.2. Pentanote Indexing

Pentanotes indexing is performed before all other filters. Firstly part of the whole query are selected for stage 2. Then all pentanote clusters are compared with the head and tail of the part-query using frame-based RA algorithm. The 25% most similar clusters are kept. With the index table we can map these clusters to pentanotes in the database and construct melody candidates. Usually several million candidates are constructed after this filter.

Table 1. Cascade filters

LV	Feature(s)	Algorithm
Stage2: using part of the query		
1	pitch contour	pentanote indexing
2	pitch contour	RA VarⅢ
3	variance, highest pitch, etc.	linear classifier
4	segmented notes	RA VarⅡ
Stage3: using the whole query		
5	pitch histogram distance	linear classifier
6	segmented notes	RA VarⅡ
7	segmented notes	RA VarⅡ

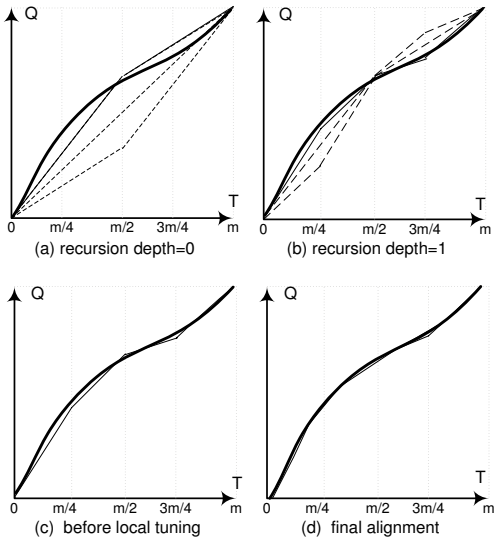

Figure 2. RA Alignment with 1 Recursion and 3 Possible Scale Tries

4.3. Linear Classification with Simple Features

The second and the fifth filters are linear classifiers using simple features such as pitch variance, highest pitch value and pitch histogram etc. Compared with either pitch contour or note sequence, they are one dimension features which need few computation to compute or to classify. And also compared with N-grams, they are global view features which seem to be robust to local errors. Here Manhattan distance is adopted to calculate the similarity between two histograms. The classification thresholds are predefined constants.

4.4. RA and RA variations

Recursive Alignment(RA) [1] inspired by J.Jang's LS [5] is a top-down algorithm to match query and melody candidate at frame level. The basic idea of RA comes from the fact

that the query and the candidate are similar if and only if they roughly share the same shape in the global view. The algorithm divides the candidate melody into 2^N parts recursively and each part uses a linear alignment scale. After that local tuning is applied to get the final alignment path. Figure2 gives an example of RA. The main difference between RA and other frame level alignment algorithms such as DTW is RA's top-down fashion. In RA higher level decision is always made ahead of lower ones, that is, global scale factor which is determined before local ones will restrict the local scales within a reasonable range. We believe such top-down style can handle long-distance information (rhythm and duration for example) better.

RA variations are employed by some filters. RA VarⅡ uses segmented notes instead of frames while computing score which reduces complexity magnitude from frame order to note order. RA Ⅲ divides frequency space into several bands and binarizes the pitch value in each sub-band, so the score of several frames can be computed in parallel utilizing the 32-bit bandwidth of computers.

5. System Implementation

The system is implemented with C++ and is built in Win32 environment with Intel C++ Compiler 9.0. We submit two executable files based on different assumptions. The first one assumes that the queries are always from the beginning of the targets, which fits the case of the evaluation. The second one allows the user start from any position of the target song, which is relatively slower and less accurate but we think it has more practical value.

6. Evaluation Result

The queries are 2797 wave files and the database containing 48 ground truth MIDI files along with 2000 Essen Collection noise MIDI files. Before recognition starts, we convert all queries into 16bit linear encoded format with 8K Hz sample rate.

We achieved the best results among all contestants in the subtask we participated. For the system "match from beginning", the Mean Reciprocal rank (MMR) is 0.926. For the system "match from anywhere", the MMR is 0.900. We think the submitted system mainly benefits from two things: firstly, introducing top-down fashioned RA algorithm which considers long-distances shape of pitch contour while optimizing the alignment; secondly, employing carefully selected filters which greatly reduce the search space while keeping high recall. The system gives a good performance even if there is no assumption of singing from beginning because it is designed for this intention. In the future, we may focus on revising the MIDI database to make the reference more similar to human singing. Perhaps some statistical technologies are needed.

7. Acknowledgments

This work is supported by National Natural Science Foundation of China (10574140, 60535030), Chinese 973 program (2004CB318106). Many thanks to IMIRSEL for their great effort in the organization and evaluation of MIREX2006. We are also grateful to Stephen Downie and Jyh-Shing Jang who lead the QBSH task and prepare the evaluation data.

References

[1] Wu,X., Li,M., Liu,J., Yang,J., Yan,Y., "A top-down approach to melody match in pitch contour for query by humming," in *Proc of International Conference of Chinese Spoken Language Processing*, 2006.

[2] Viola,P., Jones,M., "Robust real-time object detection," in *International Journal of Computer Vision*, 2002.

[3] Li,M., Wen,Y., Yu,T., "High efficient pitch tracking method for tonal feature extraction," in *Proc of International Conference of Chinese Computing*, 2001.

[4] Li,M., Yan,Y., "An humming based approach for music retrieval," in *Proc of National Conference on Man-Machine Speech Communication*, 2005.

[5] Jang,J., Hsu,C. and Lee,H., "Continuous hmm and its enhancement for singing/humming query retrieval," in *Proc of ISMIR*, 2005.