MIREX 2010: Query by Singing/Humming

Tzu-chun Yeh and Yi-Fan Fang

MIR Lab, Department of Computer Science
National Tsing Hua University, Taiwan
{Tzu-chun.Yeh, Yi-Fan.Fang}@mirlab.org

ABSTRACT

This extended abstract describes the submission to the QBSH (Query by Singing/Humming) task of MIREX (Music Information Retrieval Evaluation eXchange) 2010. The methods for both subtasks (classic and variant) are briefly introduced in the following sections (be more specific). More detailed analysis of the results retrieved in the evaluation will be arranged and given in the revised version.

Keywords: MIREX, Query by Singing/Humming (QBSH), LS (Linear Scaling), DTW (Dynamic Time Warping).

1. INTRODUCTION OF MIREX QBSH TASK

The goal of QBSH task is to evaluate music information retrieval systems which use singing or humming queries from users. In MIREX 2010, there are two different subtasks for QBSH task:

- Subtask 1: Classic QBSH evaluation
  - Queries: human singing/humming periods in the form of WAV files. These files include both Roger Jang's and ThinkIT's collections.
  - Database: ground-truth and noise MIDI files in monophonic form. These files include 48+106 Roger Jang's and ThinkIT's ground-truth files with Essen Database which includes 2000+ MIDI files.
  - Evaluation: top-10 hit rate which scores 1 point for a hit in the top 10 list and 0 otherwise.
- Subtask 2: Variant QBSH evaluation
  - Queries: the same as subtask 1.
  - Database: the same database as what is used in subtask 1 with additional WAV files in all available corpora. (without WAV files that use in queries)
  - Evaluation: the same as subtask 1.

2. THE PROPOSED APPROACHES

2.1 Methods

The methods we used for the QBSH task are shown below:

- DTW: Dynamic Time Warping [2, 3]
- LS: Linear Scaling [4]
- PLS: Partial Linear Scaling

PLS is an improved method of LS by using the beginning part of input signal to perform LS, then we obtain a weighted distance from the PLS and LS distance by a given weighting function. This method is used to solve the problem that some people tend to sing or hum well in the beginning but poorly at the following parts. In our observation, some people changed tempo or keys of the song during their singing or humming. Original LS cannot solve this problem since the input signal is scaled by several fixed ratio in LS and shifted by a fixed value for key transposition. Although DTW may solve this problem, the execution time of PLS is much faster because PLS only needs to perform LS twice. We use several different weighting function on the combination of PLS and LS. The comparison of results obtained from these functions is given in the evaluation result section.

2.2 Feature set selection

We evaluate our methods by using Roger Jang's collection [1], which contains 4431 WAV files and human-labeled PV (pitch vector) files with 48 MIDI ground truth files and additional 2000+ Essen Database noise MIDI files. Since these corpora are all available, we have 3 kinds of feature set for input: WAV, PV, and MIDI files. However, we did not use MIDI files on our evaluation because our methods are all based on frame-based pitch vectors, .

Our evaluation of LS and PLS methods is shown in Figure 1. The results show that PLS slightly improves the hit rate in top-1 and top-10 list. Moreover, PV set performs better than WAV set, which generated by a robust pitch tracking method based on dynamic programming. The parameters and the weighting function which mentioned in this figure are explained in the next section.

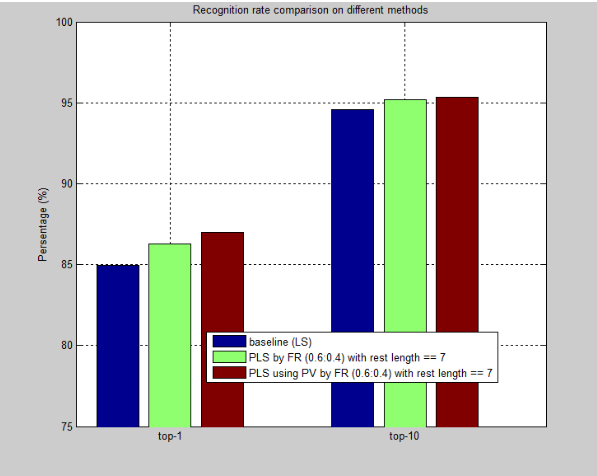

Figure 1. A comparison by LS and PLS performance using WAV and PV files

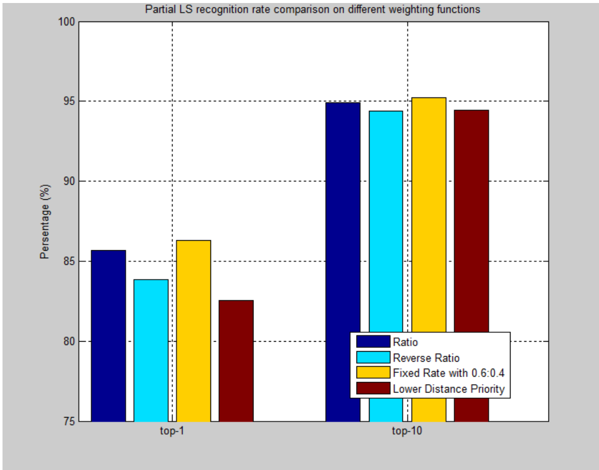

Figure 2. Recognition rates among different weighting functions in PLS

3. EVALUATION RESULTS

In above section, we mentioned that there are several different weighting functions for combining of LS and PLS distances. Here, we show four weighting functions that we used on PLS evaluation: Ratio (R), Reverse Ratio (RR), Fixed Ratio (FR), and Lower Distance Priority (LDP). Ratio function is based on the ratio between LS and PLS distances. For example, if the distance of LS is 3 and the distance of PLS is 1 for a given pitch vector, the Ratio weight will be 3:1 = 0.75:0.25 and the Ratio function distance will be 0.75*3 + 0.25*1 = 2.5. By using the same example on RR, we get 0.25*3 + 0.75*1 = 1.5. Both methods are considered to enlarge the distance between the similar and dissimilar comparisons for recognition. Fixed Rate function sets a rate *N* for the weight of LS and (1-*N*) for the weight of PLS. For the above example again, if we set *N* = 0.6, the distance of FR function for the given pitch vector is 0.6*3 + (1-0.6)*1 = 2.2. The last function, LDP, gives a threshold *d* to restrict the combination of the distance of LS and PLS. For example, if *d* is set to 0.3 and the distance of LS or PLS is below 0.3, the distance of LDP will be the one below the threshold. Otherwise, if both distances are below or above the threshold, the distance of LDP will be the mean of the distances of LS and PLS. This approach aims to keep the lower distance retrieved from recognition since it might be the appropriate answer to the query input.

The evaluation results of PLS with these four weighting functions are shown in Figure 2. It shows that the FR function performs the highest recognition rate in PLS. This is because PLS may fit all the similar parts of ground truth due to the use of shorter information than original LS, and some of these are not the true answers to the query input but have lower distances. This may confuse all the weighting functions other than FR function since it has a fixed ratio to both LS and PLS, and this prevents the interference of the above problem.

Since the execution time is not the main issue of the QBSH task, we try PLS on the subtask 1, and DTW on the subtask 2. The resulted figure will be added in the revised version of this extended abstract.

4. REFERENCES

[1] J.-S. Roger Jang, "QBSH: A Corpus for Designing QBSH (Query by Singing/Humming) Systems", available at the "QBSH corpus for query by singing/humming" link at the organizer's homepage at http://www.cs.nthu.edu.tw/~jang.

[2] J.-S. Roger Jang and Ming-Yang Gao, "A Query-by-Singing System based on Dynamic Programming", International Workshop on Intelligent Systems Resolutions (the 8th Bellman Continuum), PP. 85-89, Hsinchu, Taiwan, Dec 2000.

[3] J.-S. Roger Jang, Hong-Ru Lee, "Hierarchical Filtering Method for Content-based Music Retrieval via Acoustic Input", The 9th ACM Multimedia Conference, PP. 401-410,Ottawa, Ontario, Canada, September 2001.

[4] J.-S. Roger Jang, Hong-Ru Lee, Ming-Yang Kao, "Content based August Music Retrieval Using Linear Scaling and Branch-and-bound Tree Search", IEEE International Conference on Multimedia and Expo, Waseda University, Tokyo, Japan, 2001.

[5] J.-S. Roger Jang, Chao-Ling Hsu, Hong-Ru Lee, "Continuous HMM and Its Enhancement for Singing/Humming Query Retrieval", International Symposium on Music Information Retrieval 2005, London, UK, Sept 2005.

[6] J.-S. Roger Jang, Jiang-Chun Chen, Ming-Yang Kao, "MIRACLE: A Music Information Retrieval System with Clustered Computing Engines", 2nd Annual International Symposium on Music Information

Retrieval 2001, Indiana University, Bloomington, Indiana, USA, October 2001.