MIREX 2013 QBSH TASK: NETEASE'S SOLUTION

Peng Li, Yuan Nie and Xiaoyan Li
NetEase(Hangzhou) Network Co.,Ltd
lip0620@gmail.com

ABSTRACT

This document describes our submission to QBSH task of MIREX 2013. Our algorithm adopts a two-stage cascaded solution based on Locality Sensitive Hashing (LSH) and accurate matching of frame-level pitch sequence. Firstly, LSH is employed to quickly filter out songs with low matching possibilities, resulting in a list of candidate songs for further processing. In the second stage, Dynamic Time Warping (DTW) is applied to find the N (set to 10) most matching songs from the candidate list. This approach allows for fast pruning of irrelevant songs whilst preserving the most matching ones.

1. INTRODUCTION

Given a short clip of user-hummed/sung data, a QBSH system searches in database for a list of songs that best match input query. Such a system is usually comprised of three modules: 1) an offline database building module in which songs (most commonly midi) are analyzed and song features (e.g. note sequences, local pitch features) are stored/indexed; 2) an online pitch tracking module which extracts pitch sequence from query audio; 3) an online matching module which carries out the task of calculating matching scores between query and database songs. In QBSH systems, low-complexity searching approaches are preferred due to the large number of songs in database. Therefore, most prior work consists of a coarse matching stage, aiming to quickly filter out irrelevant songs. Wang [5] and Ryynänen [4] adopted such a strategy based on note-based phrase matching and indexing techniques, respectively.

There are 2 subtasks in QBSH 2013: Classic QBSH task and Variants QBSH task. The former uses midi files as database songs whilst the latter regards user-query wav files as database songs. It has to be mentioned that our submission is dedicated to Classic QBSH task.

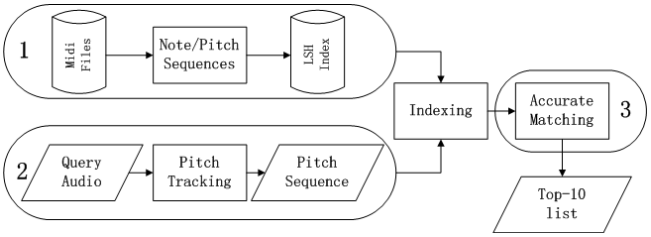

Figure 1. Framework of our QBSH algorithm

2. SYSTEM DESCRIPTION

Similarly, our approach consists of the three modules introduced in previous section (see Fig. 1). In pitch-tracking module, query audio is analyzed both in time and frequency domain to achieve accurate tracking results. In database building module, we follow the idea described in [4] and index *pitch vectors* with LSH. During online matching, candidate songs are quickly selected by LSH. Then the matching score of each candidate is assessed by asymmetric DTW [3]. Finally the top-10 list is returned.

2.1 Pitch Tracking

The accuracy of pitch tracking has a significant influence on the performance of QBSH systems. In our implementation, we have exploited temporal and spectral approaches. The main idea is to firstly detect pitch candidates for each frame in time domain based on normalized cross correlation. In frequency domain, pitch candidates of each frame are estimated with harmonic analysis. Finally, dynamic programming is employed to create the optimal pitch sequence. This sequence is further filtered in a multi-pass manner to mitigate half or double pitch errors.

2.2 LSH Indexing

Similar to [4], we use *pitch vectors* in LSH indexing [1]. The main difference between our approach and [4] is that in our implementation a *pitch vector* is extracted every ϕ (different ϕ is used for midi and query) seconds, whereas in [4] it is extracted for each note. Moreover, the extraction parameters have been tuned to give the best indexing results.

During online matching, for a given *pitch vector* of a query audio LSH looks up its matching candidates via indexing and thus filters out irrelevant songs. As a result, a small number of candidate clips are selected and delivered to accurate matching module.

2.3 Accurate Matching

Traditional matching algorithms such as Linear Scaling (LS) [2], Recursive Alignment (RA) [6] and Dynamic Time Warping (DTW) [3] have been implemented for comparison. While DTW in general outperforms the other two algorithms, it has higher complexity. Since only a small number of candidate clips need to be matched in our implementation, we select DTW as the matching engine that aligns two frame-level pitch sequences of different length and measures their similarity.

Asymmetric DTW has been used due to its better performance over other DTW variants. In addition, we define the legal region in DTW path plane. Flexible boundary constraints are employed to automatically select the best start-end points of DTW paths, which could be useful when aligning two pitch sequences with imprecise boundaries.

Once asymmetric DTW has been performed on all the candidates selected by LSH, the top-20 candidates are obtained. Then for each candidate clip we further shift its pitch sequence along pitch axis to find the position which maximizes its matching score, again, using asymmetric DTW. This procedure is adopted to refine the order of the candidates in top-20 list and thus results in a higher MRR. Finally, top-10 list is returned.

3. CONCLUSION AND DISCUSSION

In this document we present our submission to QBSH 2013. Our algorithm adopts a traditional framework which allows for fast pruning of irrelevant songs. To optimize its performance, we have run various tests and analyzed those incorrect matches. It has to be mentioned that in some cases the pitch curves of incorrect top-1 are quite similar to query pitch curves, which implies that additional information is needed to lower false alarm rate. As discussed in [5], most people hum from the beginning of a music phrase. Therefore we think the combination of LSH and phrase segmentation could be helpful in this problem.

We also analyzed previous algorithms submitted to MIREX in the past. Their test results have shown that improving QBSH performance has been a difficult task in recent years. Perhaps more sing/humming features besides pitch/note sequences could be exploited.

4. REFERENCES

[1] M. Datar, N. Immorlica, P. Indyk, and V.S. Mirrokni. Locality-Sensitive Hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational Geometry*, pages 253–262, 2004.

[2] J.R. Jang, C. Hsu, and H. Lee. Continuous HMM and its enhancement for singing/humming query retrieval. In *Proceedings of International Society for Music Information Retrieval*, pages 546–551, 2005.

[3] C. Myers, L. Rabiner, and A. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(6):623–635, 1980.

[4] M. Ryynänen and A. Klapuri. Query by humming of midi and audio using Locality Sensitive Hashing. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, pages 2249–2252, 2008.

[5] L. Wang, S. Huang, S. Hu, J. Liang, and B. Xu. An effective and efficient method for query by humming system based on multi-similarity measurement fusion. In *Proceedings of International Conference on Audio, Language and Image Processing*, pages 471–475, 2008.

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