# MIREX2014:QUERY BY HUMMING/SINGING SYSTEM

**Yinxin Hou, Minhui Wu, Dadong Xie and Hailong Liu**
Tencent technology (Beijing) Co., Ltd.
{yixinhou,sofiawu,dadongxie,hailongliu}@tencent.com

## ABSTRACT

This extended abstract describes our submission to the QBSH (Query by Singing/Humming) task of MIREX (Music Information Retrieval Evaluation eXchange) 2014. Our system takes advantage of a two-stage match method based on the Hierarchical K-means Tree (HKM) and accurate matching of the note-based dynamic programming matching technique. The HKM method constructs an index of melodic fragments by extracting pitch vectors from a database and is employed to quickly select a list of candidate. Then the accurate matching is applied to find the ones with highest scores.

## 1. INTRODUCTION

A typical QBSH system is designed to take an audio query sung/hummed by a user as input, and produce the output of a song list ranked by the level of melody similarity with the query. There are 2 subtasks in QBSH 2014: Classic QBSH evaluation and Variants QBSH evaluation. The former one uses midi files as database songs and .wav format human singing/humming query whilst the latter one builds database based on .wav format human singing/humming snippets. The methods for both subtasks (classic and variant) are briefly introduced in the following section.

## 2. SYSTEM DESCRIPTION

Figure 1 shows the flowchart of the proposed system. Our system consists of three modules: 1) an online pitch tracking module which extracts pitch sequence from query audio and transcribes it into temporally segmented note events with pitch value and duration pairs; 2) an offline database build module in which songs (midi files in subtask1 and wav files in subtask2) are analyzed and then pitch vectors are indexed by HKM tree; 3) an online matching module which calculates matching scores between query and database songs and produce the final result.

In pitch-tracking module, query audio is analyzed in time domain to achieve a quick extraction whilst database audio of subtask2 is processed in frequency domain to achieve accurate tracking results. In database building module, we adapt the idea described in [4] and index the pitch vectors with HKM tree. During online matching, candidate songs are quickly selected by HKM. Then the matching score of each candidate is assessed by note-based dynamic programming matching technique.

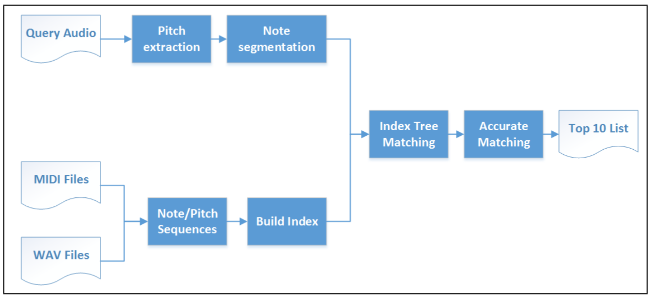

**Figure 1**. The block diagram of the proposed QBSH system.

### 2.1 pitch tracking

In our implementation, we have exploited temporal approaches for query audios and spectral approaches for database audios. The time domain approach of the Average Squared Mean Magnitude Difference Function (ASMDF) [1] is applied for query audio to extract the pitch value of each frame. In frequency domain, pitch candidates of each frame are estimated with harmonic analysis and dynamic programming is employed to find the best path with these candidates and get the optimal pitch sequence [2,3]. The audio data may include a lot of noise such as vibration and shaking of the user's voice and the surrounding which could degrade matching accuracy. Therefore, the reference and query data are normalized by mean-shifting, median filtering, average filtering and min-max scaling before matching procedure.

### 2.2 HKM Indexing

Given a database of MIDI files or wave files, we extract pitch vectors similar to [4] to construct an index of HKM tree. The main difference is that we apply Hierarchical K-means Tree (HKM) instead of Locality Sensitive Hashing (LSH). Besides, we preserve more the smallest-distance matches for each query point and adopt adaptive weights for the corresponding song.

During the online matching, the pitch sequence of a query is converted into note events and extracts pitch vectors. For each query pitch vector, the HKM tree searches for nearest neighbors in L1 distance from the database with adaptive weights. The scores of the database songs are obtained as a weighted summation. As a result, a small number of candidates are selected and delivered to accurate matching module.

### 2.3 Accurate Matching

Although we noticed that the frame-based dynamic time warping (DTW) algorithm is more robust and precise, yet it is considerably slower than note-base approaches. Inspired by [5] , we use the dynamic programming matching method to compute the minimum edit distance between the query note sequence and the note sequence from survival candidates.A natural way to quantify the differences between two note sequences is to count the minimal number of transformation applied to the first sequence in order to obtain the second.Typical transformations include deletion of a term from a sequence, insertion of a term to a sequence and replacement of one term by another. Besides, it is necessary to allow two more kinds of transformation, similar to the compressions and expansions, called consolidations and fragmentations, involving combining multiple notes to form a single one and segmenting one note into multiple note. These concepts can be generalized by associating a weight to each type of transformation. Thus, we do not necessarily count for each transformation but a predefined value related to the kind of transformation and to the elements involved in the transformation

## 3. CONCLUSION

In this document, we proposed an efficient and practical QBSH system, which enables fast melody matching.The proposed fast melody comparison method is combined index structure with note-based dynamic programming technique to attain satisfactory efficiency and effectiveness. It is very practical for very large music database since the running-time of match is limit and the online pitch tracking is very fast.

## 4. REFERENCES

[1] Chakraborty, Roudra and Sengupta, Debapriya and S-inha, Sagnik: “Pitch tracking of acoustic signals based on average squared mean difference function” *ignal, Image and Video Processing*, pp. 319–327, 2009.

[2] D. Hermes: “Measurement of pitch by subharmonic summation,” *Journal of Acoustical Society of America*,pp. 257–364, 2009.

[3] H.C. Huang and F. Seide: *Pitch tracking and tone features for Mandarin speech recognition*, ICASSP, 2000.

[4] M. Ryynnen and A. Klapuri: *Query by humming of midi and audio using locality sensitive hashing*, ICASSP, page 2249-2252. IEEE, (2008).

[5] Mongeau, M. and Sankoff, D.: *Comparison of Musical Sequences*, Computers and the Humanities,24(3):161-175 (1990)