QUERY BY HUMMING USING LOCALITY SENSITIVE HASHING FOR MIREX 2008

Matti Ryynänen and Anssi Klapuri
Department of Signal Processing, Tampere University of Technology, Finland
matti.ryynanen@tut.fi, anssi.klapuri@tut.fi

ABSTRACT

We submit a method for the MIREX 2008 task "Query by Singing/Humming" based on locality sensitive hashing (LSH). The method constructs an index of melodic fragments by extracting pitch vectors from a database of melodies. In retrieval, the method automatically transcribes a sung query into notes and then extracts pitch vectors similarly to the index construction. For each query pitch vector, the method searches for similar melodic fragments in the database to obtain a list of candidate melodies. This is performed efficiently by using LSH. The candidate melodies are ranked by their distance to the entire query and returned to the user.

1 INTRODUCTION

Query by humming (QBH) refers to music information retrieval systems where short audio clips of singing or humming act as queries. In a normal use case of QBH, a user wants to find a song from a large database of music recordings. If the user does not remember the name of the artist or the song to make a metadata query, a natural option is to sing, hum, or whistle a part of the melody of the song into a microphone and let a QBH system to retrieve the song.

The QBH task can be broadly divided into two subproblems: i) converting a query into a format which enables robust searching and ii) matching the query with melodies in the database. The former problem is often associated with automatic transcription of a query into temporally segmented note events or into frame-wise measured pitch trajectory, whereas the latter concentrates on measuring melodic similarity. See [4] for an overview of music information retrieval systems and research on melodic similarity.

2 METHOD DESCRIPTION

The submitted method has been published by the authors in [3], and the method is here only briefly outlined in the following. For details, please see the original publication [3].

Figure 1 shows a block diagram of the method. Given a database of melodies in MIDI format, the method constructs an index of melodic fragments by extracting pitch vectors. A pitch vector stores an approximate representation of melody contour within a fixed-length time window starting at each MIDI note onset (see Fig. 2). In retrieval, the method automatically converts a query into MIDI notes and then extracts pitch vectors. For each query pitch vector, the method searches for nearest neighbors in Euclidean space from the index of database melody fragments to obtain melody candidates and their matching positions in time. This can be performed very efficiently by using locality sensitive hashing (LSH) [1, 2]. Final ranking of candidates is done by comparing the whole transcribed query to each candidate melody segment by using recursive alignment proposed by Wu et al. [5]. Due to the melodic fragment index, the method manages long database melodies directly, without having to segment melodies into phrases. Also, the queries do not have to start from the beginning of a melodic phrase. Using LSH provides a significant speed-up and without losing accuracy in retrieval performance. In particular, the method is designed for matching MIDI melodies in contrast with pitch-trajectory based methods.

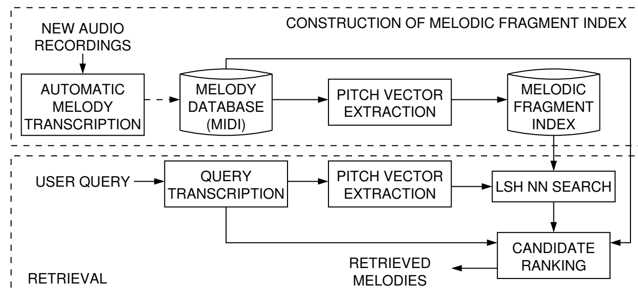

Figure 1. A block diagram of the method.

3 ABOUT THE IMPLEMENTATION

The method has been implemented as Matlab M-files, MEX-files, and Linux executables, and runs in Linux Matlab versions 6.5, 7.3, and 7.4. The LSH codes have been adapted from the implementation package E2LSH by the Alexandr Andoni (see http://web.mit.edu/andoni/www/LSH).

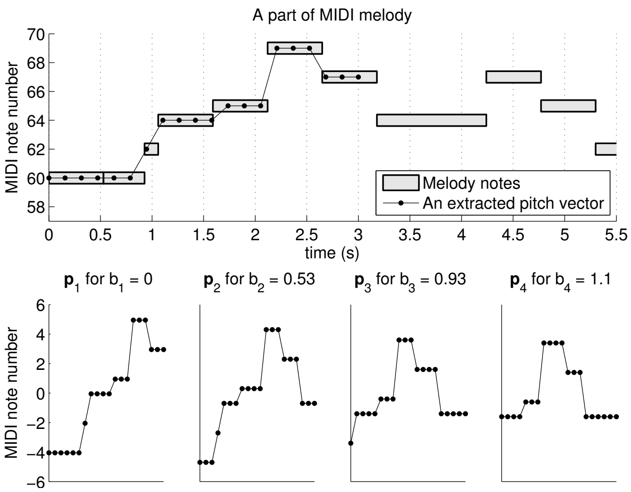

Figure 2. Example of pitch vector extraction.

4 RESULTS AND DISCUSSION

The submitted method obtained mean-reciprocal ranks (MRR) of 93% (rank #1 together with the same result as the method by Wang) and 89% (rank #3) for the subtasks 1 and 2, respectively. For the first subtask, the results are similar to our simulations in [3]. The run-time of the method is moderate (approximately the second fastest depending on submissions). The fastest method by Wu & Li performed best in the second subtask.

In general, it seems that improving the very good retrieval results from MIDI melody databases is already quite hard. To make the task even more challenging, perhaps we could have a QBH task where the search space consists of audio recordings instead of MIDI databases during the following MIREX evaluations. In [3], we also carried out tests on this and some examples are given at

http://www.cs.tut.fi/sgn/arg/matti/demos/qbh.

5 ACKNOWLEDGMENTS

The authors would like to thank the IMIRSEL group for running the evaluations and organizing MIREX in general, the task organizers and data providers, people involved in the discussions, and all the participants.

6 REFERENCES

[1] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 459–468, 2006.

[2] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. ACM Symposium on Computational Geometry*, pages 253–262, 2004.

[3] M. Ryynänen and A. Klapuri. Query by humming of MIDI and audio using locality sensitive hashing. In *Proc. 2008 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2249–2252, Las Vegas, Nevada, USA, Apr. 2008.

[4] R. Typke. *Music Retrieval based on Melodic Similarity*. PhD thesis, Universiteit Utrecht, 2007.

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