Two Note Based Approaches to Query by Singing/Humming

Christian Sailer
Fraunhofer IDMT
Langewiesener Str. 22
98693 Ilmenau/Germany
sar@idmt.fraunhofer.de

Abstract

This paper describes the submissions to the MIREX 2006 Query by Singing/Humming task delivered by Fraunhofer IDMT. The approach presented here is based on extracting the pitch out of monophonic singing (or humming), and hereafter segmenting and quantising it into a melody composed of discrete notes. Finally this melody is compared to a database of indexed melodies, using an error tolerant similarity search. Two algorithms have been submitted that differ in the melody extraction method, roughly characterised by the trade-off between accuracy of transcription (and therefore recall) and computing time needed. A third version accepting queries in midi format has also been submitted.

Keywords: MIREX, Query by Humming, Query by Singing, similarity search

1. Introduction

The term Query by Singing/Humming usually describes the retrieval of a musical piece containing a certain melodic theme by singing or humming the same. Several tasks are to be solved to tackle this problem: The corpus of melodies in which the query will be searched has to be acquired, the singing input has to be processed into a format that can be handled by the search algorithm, and melodies similar to the query input have to be spotted in the melody database. The last step requires a high grade of discrimination whilst being tolerant against either input errors or errors propagating from previous processing steps. As this results in many parameters to be defined in a QBSH evaluation, assessing and comparing such systems can be a tedious problem.

By presenting a data corpus in a defined format, and setting up two retrieval tasks, the MIREX 2006 QBSH task presents a frame work allowing comparison of different systems

2. Implementation Overview

The submission consists of an indexing tool (a), and three different query tools (b), (c), and (d). The query tools all use the same melody search algorithm and the same melody database, but feature different mechanisms to acquire the melody information to search for. The database is read from the file created by (a) and stored in the memory. The tools are able to read the query input as wav, aiff, mp3 or MIDI files from a given directory, match them against the database and output a list of the most similar melodies for each query file.

All algorithms are implemented in C++ and available for Windows and Linux. Approximate run times [1] are shown in table 1.

3. Indexing

For indexing, monophonic MIDI files are used. They are read by the indexer (a) and transformed into a database file that can be accessed by the query tools. This comprises just a transformation of the midi files into suitable format. Modification that may happen to the data are the elimination of polyphony, if overlapping notes lead to MIDI files that are slightly polyphonic. The behaviour for massively polyphonic midi files is not defined.

4. Query

The general approach to querying by audio files is extracting the melody from the audio itself. In this case, discrete note representation of the melody are extracted, as can for example be found in MIDI files. Two different algorithms to extract melodies are submitted separately. The third submitted algorithm reads in pre-extracted melodies from monophonic MIDI files.

In a subsequent step, the extracted melodies are compared to the melodies in the indexed database. The look-up

[1] All run times are measured on a 3GHz Intel Pentium IV system.

Table 1. Run times for the different algorithms. N denotes number of indexed songs, l the length of the query

Algorithm	Run Time	Scaling
Indexing	1-2s/1000 songs	$O(N)$
DB Look Up	2s/1000 songs	$O(lN)$
Extraction Warp	about $\frac{1}{5}$ Realtime	$O(l)$
Extraction Ear	about $1.5 - 2$ Realtime	$O(l)$

Table 2. Overview of the results of QBSH task. See text for explanation, entries discussed here in bold.

	AU1	AU2	**ear**	**midi**	**warp**	FH	NM	RJ	RL	RT1	RT2	XW1	XW2
Task 1	0.205	0.288	**0.568**	**0.283**	**0.348**	0.218	0.688	0.883	0.800	0.196	0.390	0.926	0.900
Task 2	0.163	0.238	**0.587**	**0.649**	**0.415**	0.309	0.722	0.926	n.e.	0.468	0.401	n.e.	n.e.

part is exactly the same for all query by singing/humming variants presented here.

4.1. Melody Extraction using a Physiological Ear Model

The first algorithm uses a physiological ear model to achieve a transcription that is as close to human hearing as possible. This yields a very precise melody transcription of monophonic audio, and therefore the best accuracy in querying that is achievable, but due to its high complexity, computing time exceeds real time.

The algorithm used here is based on the implementation of Heinz [1, 2] and has undergone minor modifications for bug fixes and stability improvement.

4.2. Melody Extraction using a Warped FFT

A second, much faster algorithm uses a warped FFT [3] to transform the time signal into a spectrogram with sufficient temporal resolution and sub half-note bin width throughout the spectrum. With an algorithm inspired by PreFest [4, 5], a salient pitch line is extracted from the spectrogram. In a melody segmentation process that has been developed based on the works of Heinz [2], a sequence of temporally discrete note objects is derived from the pitch line in conjunction with further spectral information. These note objects are then quantised to a discrete 12-tone note grid, resulting in a sequence of discrete musical notes.

4.3. Melody Similarity Search

The look-up is carried out as string alignment process [6], which has been adapted for melody search [7] on discrete note representations of the melodies. As basic search alphabet, the relative change of a melody over time, i.e. not notes, but descriptions of note transitions are used, represented by the note intervals and ratio of inter-onset intervals. This makes the search algorithm independent from absolute tempo and pitch.

As further investigations have shown, human individuals tend to render melodies in about the correct tempo, so the absolute note length has been added to the evaluation criteria of the melody search [8].

The alignment is carried out as a semi-local alignment, meaning that the whole query string must match any part of the reference string, and returns a value that increases with the similarity of the query to the reference.

In a post processing step, the contours of the M best matching melodies are compared to the contour of the query, and a correction of the alignment values is carried out. In the current implementation, $M = 50$ is used.

As the resulting alignment values depend on the size of the query string (the longer the string, the greater the maximum possible value), the values have to be normalised to allow an assessment of the alignment quality.

5. Results and Discussion

The data used consists of about 2000 MIDI noise files, 48 ground truth MIDI files and 2797 renditions of these 48 melodies as wave [2], pitch vector and MIDI files. Task 1 uses the noise files and the ground truth files as database and the vocal renditions as queries and measures reciprocal rank of the matching ground truth file. Task 2 uses the noise files, the ground truth files and the renditions as database and the renditions as query and measures the recall rate of versions of the query song among the top 10 results.

An overview [3] of the results of the QBSH tasks can be found in table 2.

As could be expected, the physiological ear model outperformed the warped FFT extraction in both tasks, while both are no match for some of the best algorithms. The warped FFT extraction algorithm is also known for having some problems with distorted files, and may have had some problems with the 8-bit quantisation of the query files.

Surprisingly, the version using midi files performed very weak in task 1, which may be explained by the mediocre quality of the query midis, which where generated automatically from pitch vector files [4]. This assumption is also supported by the increase of performance in task 2 – possibly the query MIDI files were more similar to each other than to the respective ground truth files.

One main problem of the described algorithms is probably the similarity search engine that has only been developed and optimised until 2004. Recent developments on other features and more elaborate search strategies proved to be more successful on this task.

6. Conclusive Remarks

The algorithms solely depending on note-quantised melodies are clearly outperformed by algorithms using multiple stage search algorithms and/or pitch vectors to represent singing queries. This allows at least the assumption that pitch vectors are a better representation for sung inputs than quan-

[2] 8kHz, 8bit, mono

[3] See http://www.music-ir.org/mirex2006/index.php/QBSH:_Query-by-Singing/Humming_Results for full results and information on participants

[4] See QBSH discussion page on http://www.music-ir.org/mirex2006 for details

tised melodies. This may prove to be an important result for further development on QBSH systems.

7. Acknowledgements

We would like to thank the ISMIRSEL team for their effort put into the organisation and the running of this task.

Work on the Query by Humming algorithms used in this approach was partially funded by the EU-Project Semantic Hifi (FP6-507913) [5] .

References

[1] T. Heinz and A. Brückmann, "Using a physiological ear model for automatic melody transcription and sound source recognition," in *Proceedings of the 114th Audio Engeering Society's Convention*, 2003.

[2] T. Heinz, "Ein physiologisch gehörgerechtes verfahren zur automatisierten melodietranskription," Ph.D. dissertation, Technische Universität Ilmenau, 2006.

[3] A. Härmä, M. Karjalaunen, L. Savioja, V. Välimäki, U. K. Lane, and J. Huopainiemi, "Frequency-warped signal processing for audio applications," *Journal of the Audio Engineering Society*, vol. 48, no. 11, pp. 19–22, November 2000.

[4] M. Goto, "A robust predominant-f0 estimation method for real-time detection of melody and bass lines in cd recordings," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, Istanbul, 2000, ordner-Nr: A-13.

[5] ——, "A predominant-f0 estimation method for cd recordings: Map estimation using em algorithm for adaptive tone models," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 5, 2001, pp. 3365–3368, ordner-Nr: A-14.

[6] D. Gusfield, *Algorithms on Strings, Trees and Sequences*. Cambrigde University Press, 1997.

[7] C. Sailer, "Using string alignment in a query-by-humming system for real world applications," Talk at 150th ASA Fall Meeting, October 2005.

[8] M. Dittrich, "Untersuchung und Optimierung eines musikalisch fundierten Suchverfahrens zur Melodieerkennung," Master's thesis, Technische Universität Ilmenau, 2003.

[5] see http://shf.ircam.fr