Tararira: Query By Singing System

Ernesto López, Martín Rocamora
Instituto de Ingeniería Eléctrica
Facultad de Ingeniería de la Universidad de la República
Julio Herrera y Reissig 565 – (598) (2) 711 09 74, Montevideo, Uruguay
elopez,rocamora@fing.edu.uy

Abstract

This extended abstract details a submission to the Music Information Retrieval Evaluation eXchange in the Query by Singing/Humming task. The problem of query by singing consists of building a machine capable of simulating the cognitive process of identifying a musical piece from a few sung notes of its melody. In this work, the algorithms of pitch tracking, onset detection and melody matching used in the system Tararira [1] are briefly described. Much effort has been put on automatic transcription of singing voice as it is a key factor in the overall performance. A novel way of combining note by note matching with the approach based on pitch time series matching is introduced.

Keywords: QBH, MIREX, melody matching.

1. Introduction

Through the last decade, different approaches to face the query by singing problem were considered. In all the proposals, the database consist of music in symbolic notation, generally MIDI, instead of raw or compressed audio as there is no sufficiently robust automatic way to extract the melody directly from a recording to compare it with the query.

The systems proposed can be divided, from its representation and matching technique, basically into two approaches. The traditional approach is based on note by note comparison [2][3], whereas a more recent approach utilizes the comparison of fundamental frequency time series [4][5]. The first approach consist of transcribing the voice signal into a sequence of notes and searching for the best occurrences of this pattern on database of melodies. Due to the performance decrease produced by transcription errors, the other approach avoids the automatic transcription, comparing melodies as fundamental frequency time series. Unfortunately, this implies working with long sequences (very long compared to sequences of notes) therefore computational time becomes prohibitive. Moreover, it is necessary to require the user to sing a previously defined melody fragment [4][5].

In the system Tararira [1] a novel way of combining both approaches is introduced, that preserves the advantages of each of them. Firstly, the system selects a reduced group of candidates from the database, using note by note matching. Then, the selection is refined using fundamental frequency time series comparison.

The system architecture is divided in two main stages, as depicted in figure 1. The first one is the transcription of the query into a sequence of notes. In the second one, this sequence is matched to the melodies stored in the database, and a list of musical pieces is retrieved, in a similarity order.

The transcription stage involves the following tasks:

- To estimate the fundamental frequency contour to set the note pitches.
- To segment the audio signal in order to establish the onset time and duration of notes.
- To perform a melodic analysis to adjust the note pitches to the equal tempered scale.

The tasks of the matching stage are:

- To codify the note sequence so as to obtain key and tempo transposition independence in the matching.
- To set flexible similarity rules to take into account query ornaments or mistakes, and automatic transcription errors.
- To refine the candidates selection, avoiding automatic transcription errors, by comparing fundamental frequency time series.

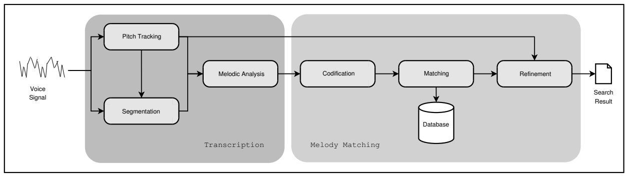

Figure 1: Block diagram of the system.

Work partly supported by Comisión Sectorial de Investigación Científica (CSIC).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.
© 2006 University of Victoria

2. Singing voice transcription

The goal of the automatic transcription is to extract from the audio signal the sequence of notes that best represents the sung melody. To do that, the events with greater probability to correspond to notes are identified. Each event is characterized by three values: pitch, onset time and duration.

In this work, much effort is put on the automatic transcription of the singing voice, as it remains an unsolved problem. The singing voice is one of the most difficult musical instruments to deal with.

2.1. Pitch tracking

To establish the notes pitch, the evolution of the fundamental frequency (F0) of the singing voice must be estimated. There are very well known techniques to do that. The implemented algorithm utilizes the Difference Function [6], a variation of the Autocorrelation Function.

2.2. Audio segmentation

The transcription of the query requires to establish each note onset time and duration. This problem is known as automatic audio segmentation into notes, and is the most difficult part of the automatic transcription. The singing voice has a set of features that make the note boundaries diffuse, and so, hard to identify.

It is possible to distinguish different note onsets in a singing voice signal. For notes sung with syllables starting with occlusive phonemes (such as, /ta/), the sudden energy release produces hard onsets, that are shown as a big signal energy increments. When a note starts with a gradual energy increase (for example, a nasal consonant), the onset is softer and therefore difficult to establish.

The algorithm implemented, in order to get a robust segmentation, looks for signs of events in the amplitude envelope as well as in the fundamental frequency contour. In the first stage, events corresponding to energy changes are detected. The algorithm calculates energy envelopes from different frequency bands [7], as generally the events appear more clearly in some band than in the envelope of the signal. The most salient ones are considered genuine note onsets. On a second stage, the weaker events are validated if they show a pitch change. Finally evident pitch changes that do not show an energy increment are identified (e.g. legato). However, this is not an easy task because the expressiveness of the performance and the lack of training of inexperienced singers introduce a set of features in the frequency contour that can be wrongly considered as additional notes (soft transition, spikes, instabilities, vibrato) [8].

2.3. Adjustment to the equal tempered scale

To assign a pitch value to each note, first of all it is necessary to approximate the fundamental frequency contour to a single frequency value. Then, a note pitch from the equal tempered scale is associated to this frequency value. Singers, specially untrained ones, are unable to sing according to a tuning system, so the query does not respect the reference and intervals of the equal tempered scale. It becomes necessary to adjust the natural deviation between the sung melody and the equal tempered scale.

The adjusting technique used assumes the hypothesis that when singing a melody a reference tone is held in mind, and that tempered note intervals relative to this reference are sung [8]. The method consists in estimating the reference tone through the most frequent deviation from the equal tempered scale, and in this way adjusting the pitch of the sung notes. Estimating the reference tone through the most frequent deviation takes into account that besides the deviation from the absolute equal tempered scale, in some note intervals an additional error may exist that is related to the difficulty of singing them.

3. Melody matching

A melody can be identified in spite of being performed at different pitch and at different tempo. However, some changes modify the melodic line but still allow the melody to be recognized, like sporadic pitch and duration errors or expressive features. The independence to the specific pitch and tempo is carried out in the note's encoding. By means of flexible similarity rules in the matching stage it is possibly to achieve tolerance to modifications due to ornaments or mistakes of the query, and automatic transcription errors.

In the system developed, a two stage approach is performed: firstly, a small group of candidates is selected based on notes comparison and then the search is refined using pitch time series.

3.1. Note sequences matching

Working with note sequences, the melody matching problem is basically an approximate string matching problem.

Encoding

The pitch transposition invariance is obtained by encoding the pitches sequence $A = (a_1, a_2, \ldots, a_n)$ as the sequence of intervals $\overline{A} = (a_2 - a_1, a_3 - a_2, \ldots, a_n - a_{n-1})$. It is evident that a sequence A' transposition of A has the same interval representation.

Ideally, tempo invariance should be obtained by normalizing the notes duration to a tempo invariant reference duration, for example, the duration of a beat, as in written notation. Unfortunately, it is not always possible to automatically estimate the tempo from a sung melody. A simple substitute for the beat is the duration of the previous event [9]. Given the duration sequence, $D = (d_1, d_2, \ldots, d_n)$, the tempo invariant representation utilized is the relative duration sequence $\overline{D} = (\frac{d_2}{d_1}, \frac{d_3}{d_2}, \ldots, \frac{d_n}{d_{n-1}})$. This sequence is quantized to a discrete alphabet. Due to the gross approximations in duration that are committed when singing carelessly, the inter onset interval is used as a more consistent representation of durations.

Matching

The matching step consists in finding good occurrences of the codified query in the database. For this task, the Edit Distance is calculated using the algorithm called Dynamic Programming [10], combining duration and pitch information. In this combination, pitch is considered more important because it is more discriminative than duration. Moreover, duration information is less trustful when singing carelessly.

Once all the database elements were compared with the encoded query, the best occurrences of the pattern are selected according to the Edit Distance (see figure 2).

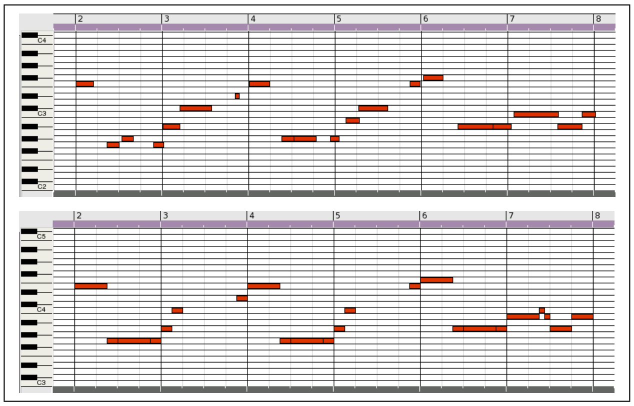

Figure 2: Transcription of the query (top) and an occurrence in the database (bottom).

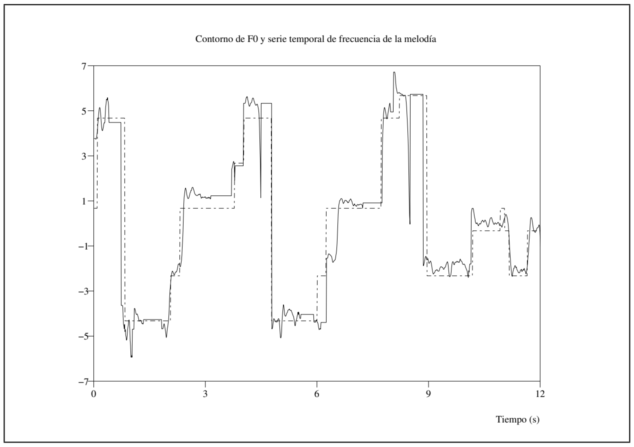

Figure 3: The corresponding pitch time series of the query and the occurrence in figure 2, normalized and aligned by the system.

3.2. Pitch time series matching

As a way of avoiding the automatic transcription errors, a recent approach compares the F0 contour of the query with melodies codified as pitch time series, by means of Local Dynamic Time Warping (LDTW).

However, this approach has some restrictions. Besides it high computational cost, an element of the database must match exactly the query, as it is not possible to search subsequences into sequences providing pitch and tempo invariance [4][5]. For this reason, the database building process is troublesome, because it is necessary to identify from the original melody those fragments likely to be sung.

In the note sequence matching stage, fragments similar to the query are identified in the melodies of the database. Then pitch time series of this fragments are build and are compared to the F0 contour of the query (see figure 3). In this way LDTW is applied to a small group of candidates, without imposing constrains to the query.

4. Evaluation Task

The system was submitted to the MIREX Evaluation eXchange in the Query by Singing/Humming Task 1, known as Known-Item Retrieval. In this task, submitted systems take a sung query as input and return a list of songs from the test database. Mean reciprocal rank (MRR) of the ground truth is calculated over the top 20 returns. The test database consists of 48 ground-truth MIDIs + 2000 Essen Collection MIDI noise files. The query database consists of 2797 sung queries. The sung queries were represented by audio (.wav), pitch vector (.pv) and MIDI (.mid) files transcribing the pitch vectors. Participants could choose whether to use the audio, pitch vector or MIDI files for querying. The system submitted uses the audio queries in .wav format.

4.1. Results

In the table of figure 4 the main results of the MIREX QSBH Task 1 are presented: Mean Reciprocal Rank, machine type [1] and runtime in seconds. Runtime is specified separately for the index and query stage or as a unique value representing both parts. More details of the results are available online [2] .

4.2. Analysis

The submitted system achieved a good performance, placing 4th among 13 evaluated systems. Significance tests show that the results of the first four algorithms are grouped closely.

The data set used in the evaluation has the remarkable characteristic that every query starts from the beginning of the corresponding song. Although this information could be used as prior knowledge, the submitted system does not take it into account (to the best of our knowledge at least one of the three best ranked algorithms do it). The hypothesis of match from the beginning would increase the performance of any system, as it reduces false positives. With regards to the assumption of this hypothesis in a real world system, it can be claimed that as long as each melody can be cut into phrases in advance, every query can be considered to match from the beginning. However, there is no guarantee that an

[1] Machine type A is a Dual AMD Opteron 64 1.6GHz, 4GB RAM, CentOS. Machine type B is an Intel P4 3.0GHz, 3GB RAM, XP.

[2] http://www.music-ir.org/mirex2006/index.php/MIREX2006_Results

		CPU	Runtime (s)	
Participant	MRR	type	index	query
Wu, Li (1)	0.926	B	63	2502
Wu, Li (2)	0.900	B	63	2817
Jang, Lee	0.883	B	25637	
López, Rocamora	**0.800**	**A**	**6**	**20604**
Lemström et al.	0.688	A	8302	
Sailer (1)	0.568	A	3	56560
Typke et al. (2)	0.390	A	23442	4629
Sailer (3)	0.348	A	3	4618
Uitdenbogerd (2)	0.288	A	8	140
Sailer (2)	0.283	A	3	608
Ferraro, Hanna	0.218	A	89239	
Uitdenbogerd (1)	0.205	A	8	166
Typke et al. (1)	0.196	A	23442	2034

Figure 4: QBSH Task 1 results.

arbitrary query starts at the beginning of a phrase. Moreover, to do this an automatic melody segmentation system is needed, introducing another source of error. In general, imposing this kind of constrains to the problem limits the scope of the system.

Regarding the runtime, although no particular attention was paid on efficiency, the processing time performed was reasonable. Further work should consider improving it.

Finally, an interesting conclusion that can be drawn from the results of this contest is that the state of the art in the query by humming problem shows that, although being still an open problem, it is feasible to face real world situations. Much effort has to be put on automatically building the database. In this sense, results on the MIREX Audio Melody Extraction Task are promising.

References

[1] E. López and M. Rocamora, "Tararira: Sistema de búsqueda de música por melodía cantada," *Proc. of the X Brazilian Symposium on Computer Music*, pp. 142–153, 2005.

[2] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, "Query by humming: Musical information retrieval in an audio database," *Proc. ACM Multimedia*, pp. 231–236, 1995.

[3] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Cunningham, "Towards the digital music library: Tune retrieval from acoustic input," *Proc. of the ACM Digital Libraries*, pp. 11–18, 1996.

[4] R. B. Dannenberg and N. Hu, "A comparison of melodic database retrieval techniques using sung queries," *JCDL*, pp. 301–307, 2002.

[5] D. Shasha and Y. Zhu, "Warping indexes with envelope transforms for query by humming," *Proc. of the 2003 ACM SIGMOD Conference on Management of Data*, pp. 181–192, 2003.

[6] A. de Cheveignè and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *JASA*, vol. 111, pp. 1917–1930, 2002.

[7] A. P. Klapuri, "Sound onset detection by applying psichoacoustic knowldege," *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1999.

[8] E. Pollastri, *Processing Singing Voice for Music Retrieval*. PhD thesis, Università Degli Studi Di Milano, 2003.

[9] B. Pardo and W. Birmingham, "Encoding timing information for musical query matching," *ISMIR*, pp. 267–268, 2002.

[10] K. Lemström, *String Matching Techinques for Music Retrieval*. PhD thesis, Department of Computer Science, University of Helsinki, 2000.