Query by humming: automatically building the database from music recordings

Martín Rocamora[a,*], Pablo Cancela[a], Alvaro Pardo[b]

[a] *Institute of Electrical Engineering, School of Engineering, Universidad de la República, Uruguay*

[b] *Department of Electrical Engineering, School of Engineering and Technologies, Universidad Católica del Uruguay, Uruguay*

Abstract

Singing or humming to a music search engine is an appealing multimodal interaction paradigm, particularly for small sized portable devices that are ubiquitous nowadays. The aim of this work is to overcome the main shortcoming of the existing query-by-humming (QBH) systems: their lack of scalability in terms of the difficulty of automatically extending the database of melodies from audio recordings. A method is proposed to extract the singing voice melody from polyphonic music providing the necessary information to index it as an element in the database. The search of a query pattern in the database is carried out combining note sequence matching and pitch time series alignment. A prototype system was developed and experiments are carried out pursuing a fair comparison between manual and automatic expansion of the database. In the light of the obtained performance (85% in the top-10), which is encouraging given the results reported to date, this can be considered a proof of concept that validates the approach.

Keywords:
voice based multimodal interfaces, music information retrieval, query by humming, singing voice separation, melody extraction

[*] Tel. +59827110974 / Fax. +59827117435
Email addresses: `rocamora@fing.edu.uy` (Martín Rocamora), `cancela@fing.edu.uy` (Pablo Cancela), `apardo@ucu.edu.uy` (Alvaro Pardo)

1. Introduction

The constant increase in computer storage and processing capabilities has made possible to collect vast amounts of information, most of which is available online. Today, people interact with this information using various devices, such as desktop computers, mobile phones or PDAs, posing new challenges at the interface between human and machine. Yet, the most common case of information access still involves typing a query to a search engine. There is a need for new human-machine interaction modalities that exploit multiple communication channels to make our systems more usable.

Among the information available there are huge music collections, containing not only audio recordings, but also video clips and other music-related data such as text (e.g. tags, scores, lyrics) and images (e.g. album covers, photos, scanned sheet music). A query for music search is usually formulated in textual form, by including information on composer, performer, music genre, song title or lyrics. However, other modalities to access music collections can also be considered that allow more intuitive queries. For instance, to provide a musical excerpt as an example and obtain all the pieces that are similar in some sense, namely query-by-example,[1] or to retrieve a musical piece by singing or humming a few notes of its melody, which is called query-by-humming (QBH). This offers an interesting interaction possibility, in particular for small size devices such as portable audio players, and requires no music theoretical knowledge from the user. Additionally, it can be combined with traditional metadata-based search and visual user interfaces to offer multimodal input and output, in the form of visual and auditory information.

Dealing with multimodal music information requires the development of methods for automatically establishing semantic relationships between different music representations and formats, for example, sheet music to audio synchronization or lyrics to audio alignment [1]. Much research in audio signal processing over the last years has been devoted to music information retrieval [2, 3], i.e. the extraction of musically meaningful content information from the automatic analysis of an audio recording. This involves diverse music related problems and applications, from computer aided musicology [4], to automatic music transcription [5] and recommendation [6]. Many re-

[1] Audio fingerprinting techniques are used in this case, being Shazam (`http://www.shazam.com/`) probably one of the best known commercial services of this kind.

search efforts have been devoted to dealing with the singing voice, tackling problems such as singing voice separation [7] and melody transcription [8]. The incorporation of these techniques into multimodal interaction systems can lead to novel and more engaging music learning, searching and gaming applications.

Even though the problem of building a QBH system has received a lot of attention from the research community for more than a decade [9], the automatic generation of the melody database against which the queries are matched remains an open issue. In all the proposed systems - with very few exceptions - the database consists of music in symbolic notation, e.g. MIDI files. This is due to the lack of sufficiently robust automatic methods to extract the melody directly from a music recording. Although there is a great amount of MIDI files online, music is mainly recorded and distributed as audio files. Hence, the scope of this approach is limited because of the need of manually transcribing (i.e. audio to MIDI) every new song of the database. A way to circumvent this problem is to build a database of queries provided by the users themselves and to match new queries against the previously recorded ones [10]. This approach drastically simplifies the problem and is applied in music search services such as SoundHound.[2] However, the process is not automatic but relies on user contributions. Besides, a new song can not be found until some user records it for the first time. In order to extend QBH systems to large scale it is necessary to develop a full automatic process to build the database. There are only a few proposals of a system of this kind [11, 12, 13, 14] and results indicate there is still a lot of room for improvement to reach the performance of the traditional systems based on symbolic databases.

In this paper a method for automatically building the database of a QBH system is described, in which the singing voice melody is extracted from a polyphonic music recording. In our previous work [15] a technique for singing voice detection and separation was presented. The contribution of the present work is the application of this technique to a music retrieval problem involving a voice-based multimodal interface. A prototype is built as a proof of concept of the proposed method and a study is conducted that compares the performance of a previously developed QBH system [16] when using a database of MIDI files and when using melodies extracted

[2] http://www.soundhound.com/

automatically from the original recorded songs. The rest of this document is organized as follows. Next section briefly describes the QBH system used in the experiments. The method for extracting the singing voice melody from polyphonic music recordings is presented in section 3. In section 4 the experiments carried out for assessing the performance of the QBH on the automatically obtained database are described and results are reported. The paper ends with some critical discussion on the present work and conclusions.

2. Query-by-humming system

The existing QBH systems can be divided, from its representation and matching technique, basically into two approaches. The most typical solution is based on a note by note comparison [17, 18]. The query voice signal is transcribed into a sequence of notes and the best occurrences of this pattern are identified in a database of tunes (typically MIDI files). The melody matching problem poses some challenges to be considered. A melody can be identified in spite of being performed at different pitch and at different tempo. Additionally, sporadic pitch and duration errors or expressive features modify the melodic line but still allow the melody to be recognized. In the matching step, pitch and tempo invariance are typically taken into account by coding the melodies into pitch and duration contours. By means of flexible similarity rules it is possibly to achieve some tolerance to singing mistakes and automatic transcription errors. Automatic transcription of the query inevitably introduce errors that tend to deteriorate matching performance. For this reason, another usual approach avoids the automatic transcription, comparing melodies as fundamental frequency (F0) time series [19, 20]. Unfortunately, this involves working with long sequences, very long compared to note sequences, and therefore it implies high computational burden. Moreover, in many proposals the user is required to sing a previously defined melody fragment [19, 20] in order that the query exactly matches an element of the database. This is because of the difficulty of searching subsequences into sequences providing pitch and tempo invariance.

In our previous work [16], a way of combining both approaches was introduced that exploits 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. Finally, a list of musical pieces is retrieved in a similarity order. The system architecture is divided in two main stages, as depicted

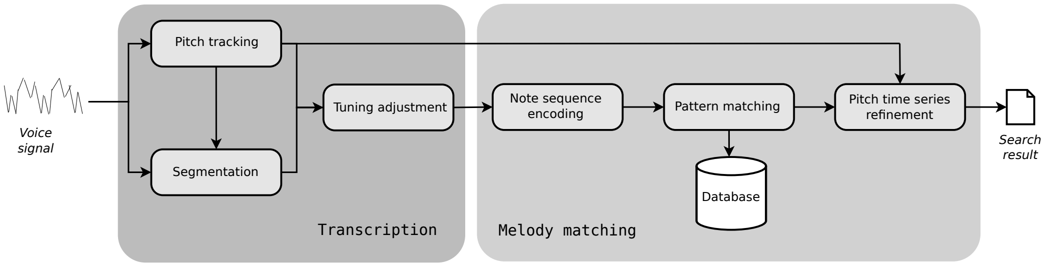

Figure 1: Block diagram of the QBH system. The input is a monophonic singing voice. The two main stages are the transcription of the query and the match of the melody pattern against the elements of the database. The output is a ranked list of songs.

in Figure 1. The first one is the transcription of the query into a sequence of notes. To do that, the F0 contour is computed using a very well know technique based on the difference function [21]. Then, the audio signal is segmented into notes by computing energy envelopes from different frequency bands and detecting salient events [22]. Besides, evident pitch changes that do not exhibit an energy increment are identified (e.g. legato notes) and considered in the segmentation. Each note is described by a pitch value, an onset time and a duration. To assign a pitch value to each note the median of its fundamental frequency contour is taken. Then the tuning of the whole sequence is adjusted by computing the most frequent deviation from the equal tempered scale, subtracting this value for every note and rounding to the nearest MIDI number [23].

In the second stage, the notes of the query are matched to the melodies of the database. The pitches sequence $A = (a_1, a_2, \ldots, a_n)$ is encoded as a sequence of intervals $\overline{A} = (a_2 - a_1, a_3 - a_2, \ldots, a_n - a_{n-1})$, so that a melody $\hat{A}$ transposition of A has the same interval representation. In a similar way, given the duration sequence, $B = (b_1, b_2, \ldots, b_n)$, a tempo invariant representation is computed as the relative duration sequence $\overline{B} = (b_2/b_1, b_3/b_2, \ldots, b_n/b_{n-1})$ [24]. When singing carelessly gross approximations in duration take place, so the inter-onset interval is used as a more consistent representation of duration and relative durations are smoothed and quantized through $q_i = \text{round}(10 \log_{10}(b_{i+1}/b_i))$, obtaining the sequence $B_q = (q_1, q_2, \ldots, q_{n-1})$ [23].

Finding good occurrences of the codified query in the database is basically an approximate string matching problem. For this task, Dynamic

Programming is used to compute an edit distance that combines duration and pitch information [25]. In this combination, pitch values are considered more important because duration information is less discriminative and not so reliable. The edit distance, $d_{i,j}$, is computed recursively as the minimum of the set of values shown in Equation 1.

$$d_{i,j} = \min \begin{cases} d_{i-1,j} + 1, & & \text{(insertion)} \\ d_{i,j-1} + 1, & & \text{(deletion)} \\ d_{i-1,j-1} + 1, & & \text{(note substitution)} \\ d_{i-1,j-1} - 1, & |\overline{a}_i - \overline{a}'_j| < 2 \text{ and} & \text{(coincidence)} \\ & |q_i - q'_j| < 2 & \\ d_{i-1,j-1} & |\overline{a}_i - \overline{a}'_j| < 2 & \text{(duration substitution)} \end{cases} \tag{1}$$

The last two values of the set are only considered if the corresponding conditions are met, where $\overline{a}$ and $\overline{a}'$ refer to the pitch interval of the query and the database element respectively, whereas q and q' correspond to their quantized relative duration. Finally, a similarity score is computed normalizing the edit distance to take values between 0 and 100,

$$\text{score} = 100\, \frac{(m-1) - d_{m,m}}{2(m-1)} \tag{2}$$

where m denotes the number of notes in the query, and $d_{m,m}$ is the final value of the edit distance between the two sequences.

As a result of the notes sequence matching, fragments similar to the query pattern are identified in the melodies of the database. Then F0 time series of this fragments are built from the matching MIDI notes, and are compared to the F0 contour of the query by means of Local Dynamic Time Warping (LDTW). The sequences are time wrapped to the same duration and pitch transposed to the same tunning. Given two m-length sequences x and y, to compute the k-th LDTW distance a matrix $\mathcal{D}(m,m)$ is built recursively by,

$$d_{ij} = \begin{cases} |x_i - y_j|^2 + \min \begin{cases} d_{i-1,j-1} \\ d_{i,j-1} \\ d_{i-1,j} \end{cases} & |i-j| \leq k \\ \\ \infty & |i-j| > k \end{cases} \tag{3}$$

for which the matrix must be initialized with $d_{1j} = |x_1 - y_j|^2$ where $j \in [1,k]$ and $d_{i1} = |x_i - y_1|^2$ where $i \in [1,k]$. The distance value is obtained as,

$d_{min} = \sqrt{\min\{d_{mj}, d_{im}\}}$ with $i, j \in [m - k + 1, m]$. The maximum allowed local time warping of a sequence relative to the other is k samples. It is easy to see that the Euclidean distance is the LDTW distance with $k = 0$. The computation of the k-th LDTW distance is implemented by also using the algorithm of Dynamic Programming but restricted to a diagonal band of width $2k + 1$ of the matrix $\mathcal{D}(m, m)$.

In this way, LDTW is applied to a small group of candidates (10 for the reported results), which is computationally efficient, and without imposing constrains to the query, since coincident fragments are identified automatically in the notes matching stage. Figure 2 shows an example of the comparison of note sequences and F0 time series between the query and an element of the database.

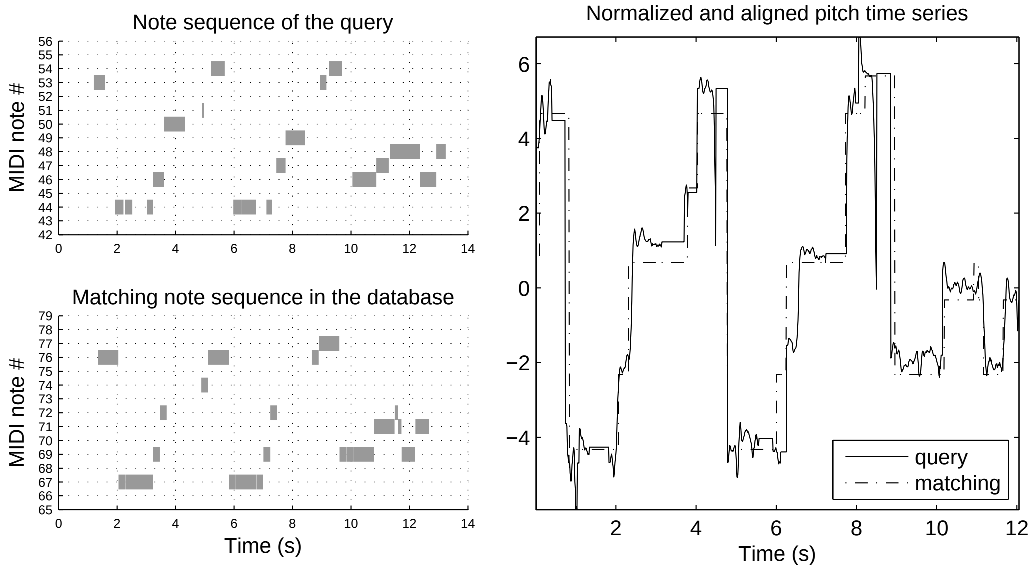

Figure 2: The piano-roll representations to the left show the transcription of the query at the top, and of an occurrence in the database at the bottom. The plot to the right depicts the corresponding F0 time series normalized and aligned by the system.

The QBH system was originally developed in Scilab and implemented as a C++ standalone application with a GUI. In this work, efforts were devoted to have a fully functional Matlab/Octave implementation and make it available for the research community.[3] Even though the search is efficient, given the two-stage matching approach, the notes matching performs an exhaustive scan of the database that can become prohibitive in a large scale scenario. This may be tackled with hashing techniques as in [26].

3. Singing voice melody extraction from polyphonic music

For building the database we focus on extracting the singing voice melody from the original polyphonic music recordings, based on the hypothesis that the melody of the leading voice is the most memorable and distinctive tune of the song and would most probably be used as a query.[4] To do that, an harmonic sound sources extraction front-end developed in previous work is applied [28, 29], which involves a time-frequency analysis, followed by polyphonic pitch tracking and sound sources separation. After that, audio features are computed for each of the extracted sounds and they are classified as being singing voice or not, as we proposed in [15]. The sounds classified as vocal are mixed in a mono channel and the transcription method used in the QBH system for transcribing the query is applied to obtain a sequence of notes and a F0 contour. This information is indexed as an element of the database. The process is depicted in Figure 3 and described in the following sections.

3.1. Harmonic sounds separation

The time-frequency analysis is based on [28], in which the application of the Fan Chirp Transform (FChT) [30] to polyphonic music is introduced. The FChT offers optimal resolution for the components of a harmonic linear chirp, i.e. harmonically related sinusoids with linear frequency modulation. This is well suited for singing voice analysis since most of its sounds have a harmonic structure and their frequency modulation can be approximated as linear within short time intervals. The FChT can be formulated as,

$$X(f,\alpha) = \int_{-\infty}^{\infty} x(t)\ \phi'_{\alpha}(t)\ e^{-j2\pi f \phi_{\alpha}(t)} dt, \tag{4}$$

[3] Available from `http://iie.fing.edu.uy/investigacion/grupos/gpa/QBH/`.

[4] According to [27], there is experimental evidence that indicates that the memory representation for lyrics seems to be tied into the memory representation for melody, providing multiple redundant constraints to assist the recall of a passage.

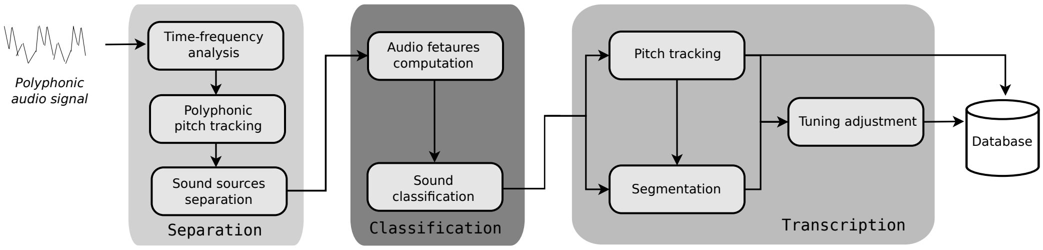

Figure 3: Block diagram of the process for automatically building the database. The system involves three main steps: Separation, Classification and Transcription. The subtasks of each step are also indicated. Note that the same monophonic transcription block used for processing queries in the QBH system is applied.

where $\phi_\alpha(t) = (1 + \frac{1}{2}\,\alpha\, t)\, t$, is a time warping function. The parameter α is the variation rate of the instantaneous frequency of the analysis chirp (see [28] for details).

In addition, based on the FChT analysis, a pitch salience representation called F0gram is proposed in [28], which reveals the evolution of pitch contours in the signal, as depicted in Figures 4 and 6. Given the FChT of a frame $X(f, \alpha)$, salience (or prominence) of fundamental frequency f_0 is obtained by summing the log-spectrum at the positions of the corresponding harmonics,

$$\rho(f_0, \alpha) = \frac{1}{n_H} \sum_{i=1}^{n_H} \log|X(if_0, \alpha)|, \tag{5}$$

where n_H is the number of harmonics considered. Polyphonic pitch tracking is carried out by means of the technique described in [29], which is based on unsupervised clustering of F0gram peaks. Finally, each of the identified pitch contours are separated from the sound mixture. To do this, the FChT spectrum is band-pass filtered at the location of the harmonics of the f_0 value, and the inverse FChT is performed to obtain the waveform of the separated sound.

3.2. Singing voice classification

The extracted sounds are then classified as proposed in [15], based on classical spectral timbre features (MFCC, see below) and some features proposed to capture characteristics of typical singing voice pitch contours. In a musical piece, pitch variations are used by a singer to convey different expressive intentions and to stand out from the accompaniment. Most typical expressive features are *vibrato*, a periodic pitch modulation, and *glissando*, a slide between two pitches [31]. Thus, low frequency modulations of a pitch contour are considered as an indication of singing voice. Nevertheless, since other musical instruments can produce such modulations, this feature is combined with other sources of information.

Mel-frequency Cepstral Coefficients (MFCC) are one of the most common features used in speech and music modeling for describing the spectral timbre of audio signals, and are reported to be among the best performing features for singing voice detection in polyphonic music [32]. The implementation of MFCC is based on [33]. Temporal integration is done by computing median and standard deviation of the frame-based coefficients within the whole pitch contour. First order derivatives of the coefficients are also included to capture temporal information, obtaining a total of 50 audio features.

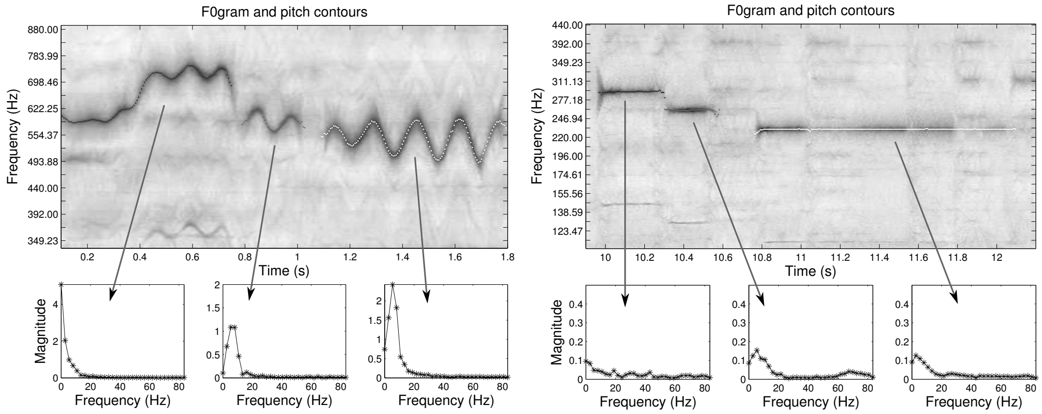

Figure 4: Vocal notes with vibrato and low frequency modulation (*left*) and saxophone notes without pitch fluctuations (*right*) for two audio files from the MIREX [34] melody extraction test set. Summary spectrum $\tilde{c}[k]$ is depicted at the bottom for each contour.

In order to describe the pitch variations, the contour is regarded as a time dependent signal $f_0[n]$ and a spectral analysis is applied using the DCT. Examples of the behaviour of the spectral coefficients, $\tilde{c}[k]$, are given in Figure 4. The two following features are derived from this spectrum,

$$\mathrm{LFP} = \sum_{k=1}^{k_L} \tilde{c}[k], \quad \mathrm{PR} = \frac{\mathrm{LFP}}{\sum_{k_L+1}^{N} \tilde{c}[k]}. \tag{6}$$

The low frequency power (LFP) is computed as the sum of absolute values up to 20 Hz ($k = k_L$) and reveals low frequency pitch modulations. The low to high frequency power ratio (PR) additionally exploits the fact that well-behaved pitch contours do not exhibit prominent components in the high frequency range. Besides, two additional pitch related features are computed. One of them is simply the extent of pitch variation,

$$\Delta f_0 = \max_n\{f_0[n]\} - \min_n\{f_0[n]\}. \tag{7}$$

The other is the mean value of pitch salience in the contour,

$$\Gamma_{f_0} = \operatorname*{mean}_n\{\rho(f_0[n])\}. \tag{8}$$

This gives an indication of the prominence of the sound source, but it also includes some additional information. As noted in [28], pitch salience computation favours harmonic sounds with high number of harmonics, such as the singing voice. Additionally, as done in [28], a *pitch preference* weighting function is introduced that highlights most probable values for a singing voice in the f_0 selected range.

The training database is based on more than 2000 audio files, comprising singing voice on one hand and typical musical instruments found in popular music on the other. For building the database the sounds separation front-end is applied (i.e. the FChT analysis followed by pitch tracking and sound source extraction) and the audio features are computed for each extracted sound. In this way, a database of 13598 sound elements is obtained, where vocal and non-vocal classes are exactly balanced. Histograms and box-plots are presented in Figure 5 for the pitch related features on the training patterns. Although these features should be combined with other sources of information, they are informative about the class of the sound. An SVM classifier with a Gaussian RBF Kernel was selected for the classification experiments, using the Weka software [35]. Optimal values for the γ kernel parameter and the penalty factor C were selected by grid-search [36].

3.3. Singing voice melody transcription

Finally, the sounds classified as singing voice are mixed in a single mono audio channel and the same transcription procedure used for processing the queries is applied. This yields the singing voice melody out from the polyphonic music recording, as a sequence of notes and as a pitch contour. Figure 6 shows the whole process for a short audio excerpt of the song *For no one* by The Beatles, which belongs to the automatically built database of the QBH system.

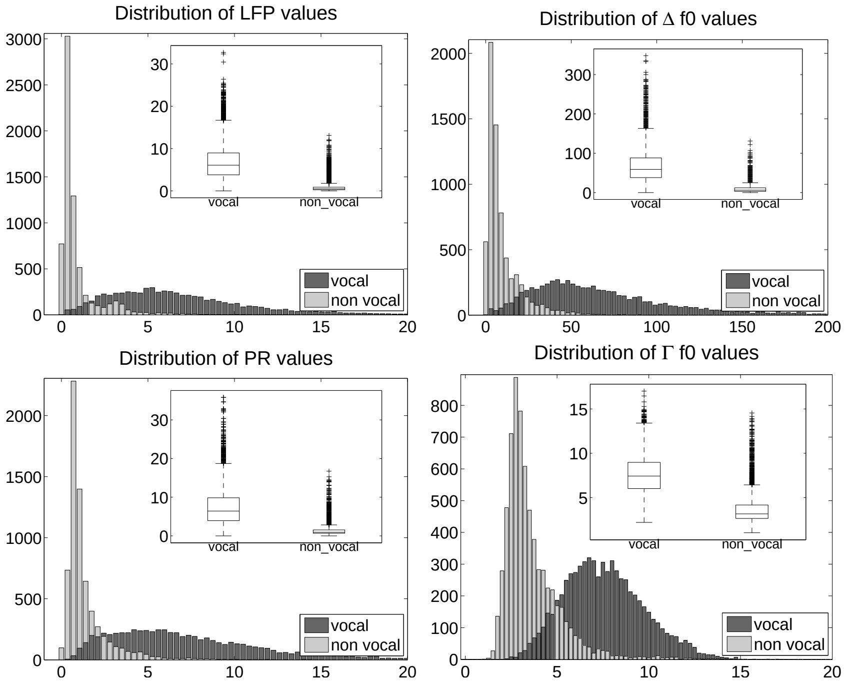

Figure 5: Histograms and box-plots of the pitch related feature values on the training database for the vocal and non-vocal classes.

4. Experiments and results

4.1. Experimental setup

The experiment is designed to evaluate the validity of extending an existing MIDI files database by using the proposed automatic method. To do that, two different datasets are used. The first one is a collection of 208 MIDI files corresponding to almost all the songs recorded by The Beatles (excluding duplicates and instrumentals) gathered from the Internet.[5] This music was selected because it is widely known making it easy to get volunteers for queries, it has generally a clear and distinctive singing voice melody, and is readily available both in audio and MIDI.

The melody of a song is assumed to be the one performed by the leading singing voice, which is usually a single MIDI channel labeled as *leading voice*

[5] From websites such as *The Beatles MIDI and video heaven*, `http://beatles.zde.cz/`.

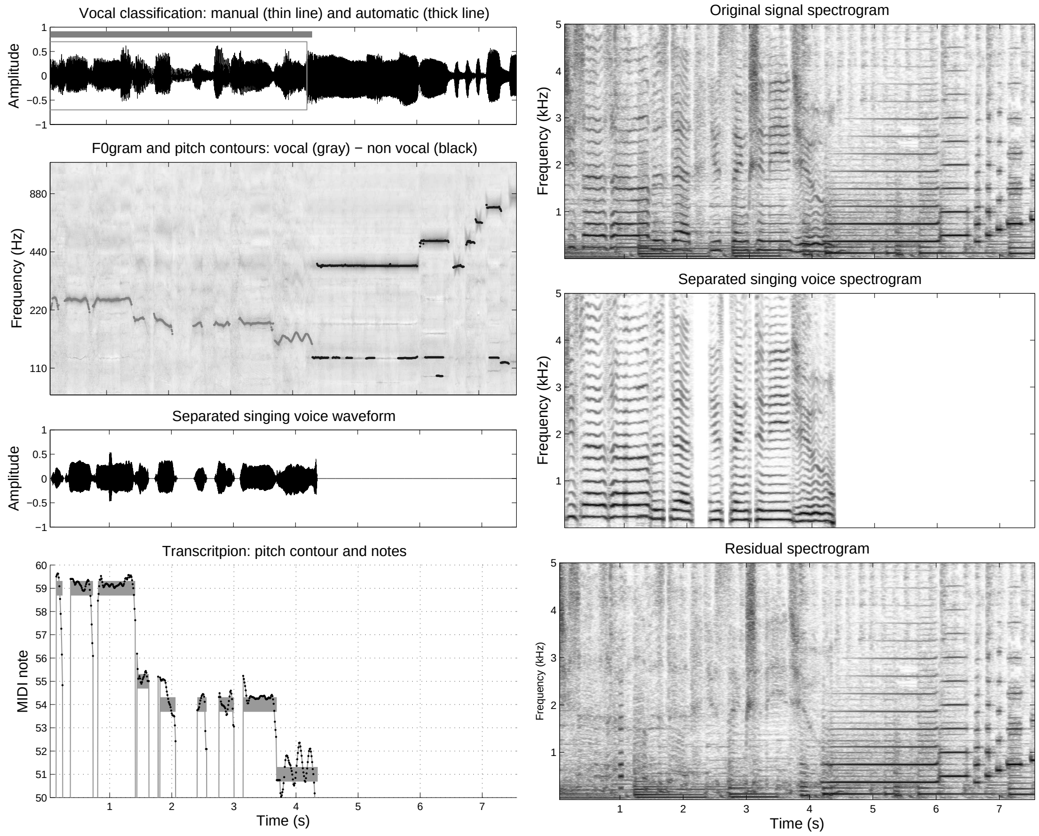

Figure 6: Example of the automatic process for building the database using a fragment of the song *For no one* by The Beatles. A singing voice in the beginning is followed by a French horn solo. There is a soft accompaniment of bass and tambourine. On the left, from top to bottom: the waveform of the recording (with manual and automatic vocal labeling), the F0gram showing both vocal and other sources pitch contours (automatically labeled), the extracted singing voice waveform, and the transcription to notes and F0 contour of the extracted singing voice. On the right, the corresponding spectrograms of the original audio mix, the extracted singing voice and the residual (the extracted singing voice subtracted from the mix).

or *melody*. This channel is manually extracted and indexed as an element of the database. To build the second database, 12 songs are selected out of this collection (which are listed in the table of Figure 7), and their melody is automatically extracted from a mono mix of the audio recording. The selection comprises different music styles and instrumentations (e.g. rock & roll, ballads, drums, bowed strings), but does not include too dense polyphonies in order that the singing main melody could be identified with no difficulty by a listener. In this case the database is modified by replacing the manually created MIDI files by the automatically extracted melodies (notes sequence and pitch contour) for the aforementioned songs.

A set of 106 sung queries corresponding to the selected songs was recorded by 10 not trained singers (6 male and 4 female), using standard desktop computer hardware. The participants were asked to sing the melody as they remembered it, with no restrictions on singing only a vocal part. They were free to sing with lyrics, hum (with syllables such as 'ta' or 'la'), or a combination of both. The mean number of notes in a query is 28, and the distribution of queries among the songs and singers is shown in Figure 7. The whole set of queries is available online, along with the mono mix and the automatic transcription of the selected songs.[6] Although including queries that do not correspond to the set of replaced songs may potentially give more insight of the QBH system, it makes the analysis of the database extension more troublesome and therefore will not be reported.

4.2. Singing voice detection evaluation

As a way of assessing the method at an intermediate step, an experiment was conducted to evaluate the degree of success on identifying the singing voice within the whole song. To do that, the 12 selected songs were manually labeled into segments containing vocals and portions with accompaniment alone. Automatic labels are obtained by applying the singing voice extraction method, as proposed in [15]. Performance is measured as the percentage of time in which the manual and automatic labeling match. The performance of a standard approach for singing voice detection in polyphonic music, i.e. MFCC of the audio mixture and an SVM classifier [32], was also computed for comparison. Results of this evaluation indicate that the proposed method for singing voice detection achieves 85.7% of correct detection. This repre-

[6] Available from `http://iie.fing.edu.uy/investigacion/grupos/gpa/QBH/`.

Song title
Blackbird
Do you want to know a secret
For no one
Girl
Hey Jude
I call your name
I've just seen a face
Michelle
Rocky raccoon
The fool on the hill
When I'm sixty four
Yesterday

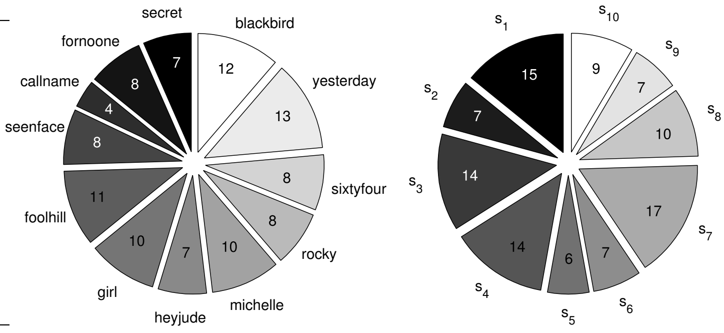

Figure 7: Experimental setup. List of the 12 selected songs whose melody is automatically obtained. The left-side chart shows the distribution of queries among the selected songs. The distribution of queries among the 10 singers s_i is depicted in the right-side chart. Note that the database is well balanced in both aspects.

sents a noticeable performance increase compared to the standard approach that yields 77.2%. Apart from the overall results, the improvement is also observable for most files of the database, as shown in Figure 8. These results are consistent with the ones reported in [15] for a different dataset, and also suggest the usefulness of the proposed pitch related features.

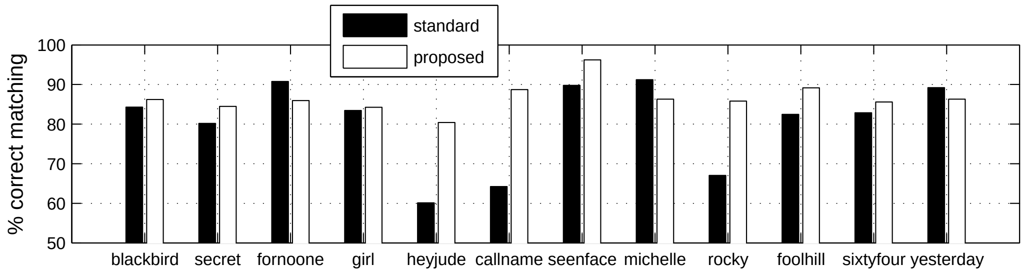

Figure 8: Singing voice detection performance as percentage of time in which the manual and automatic vocal labels match, for the proposed [15] and the standard [32] methods.

4.3. Query by humming evaluation

In order to evaluate the performance of the QBH system two standard measures are adopted: mean reciprocal rank (MRR) and top-X hit rates. Let r_i be the rank of the correct song in the retrieved list for the i-th query. Top-X hit rates are the proportion of queries for which $r_i \leq X$. Considering a set of N queries, the MRR is computed as,

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{r_i}. \tag{9}$$

Two different alternatives are considered for the audio based database. Recall that the system performs a final refinement by the direct comparison of F0 time series devised to improve matching performance. This refinement avoids errors introduced in the automatic transcription of the query. When a database of MIDI files is used, F0 time series of the matching candidates are built from the pitch of MIDI notes. In the case of the audio based database, errors are also introduced in the transcription of the singing voice melody extracted from the recording (see section 3.3). Therefore, it is preferable to perform the refinement using F0 time series computed from the extracted singing voice, rather than building it from the transcribed notes. This is confirmed by the results shown in Table 1, where the two different LDTW refinements are considered. Since the refinement is done over the 10 best matching candidates, top-10 hit rates remain unchanged.

Table 1: QBH evaluation results for MIDI and audio based databases. For the latter, the query is aligned to two different F0 time series of the matching candidate: the pitch of the transcribed notes (audio 1) and the extracted F0 contour (audio 2). Recall that the number of queries is 106 and the total number of songs (different classes) is 208.

	MRR	Top-X hit rate (%)		
		1	5	10
MIDI	0.89	88.68	89.62	91.51
audio 1	0.75	69.81	79.25	84.91
audio 2	0.76	71.70	81.13	84.91

As a way of further comparing both types of databases, an analysis is conducted considering the notes matching score assigned to the retrieved items (see equation 2). For each query, the score of the correct song is plotted against the highest score of the wrongly retrieved elements, as shown in Figure 9. This is intended to study the ability of the score to discriminate between correct and wrong retrieves. A top-1 hit result implies a correct song score higher than all the others. Thus, ideally all the query points would be located in the right-bottom triangle of the graph. For the MIDI database the vast majority of elements lie in that region, particularly for higher correct song scores. While not so markedly, the behaviour is similar for the audio based database. In the light of the above, a threshold on the score value can be useful as way of assuring confidence on the results. The thresholding determines the typical binary class scenario, resulting in True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN) regions, as depicted in Figure 9. This allows the comparison of the methods using a ROC curve, also shown in the figure. Although the MIDI database gives better results, the performance of the audio based databased is promising. As for illustrative purposes only, operating points are depicted as filled markers in the ROC (the farthest point to the diagonal), and their corresponding thresholds are plotted as vertical lines.

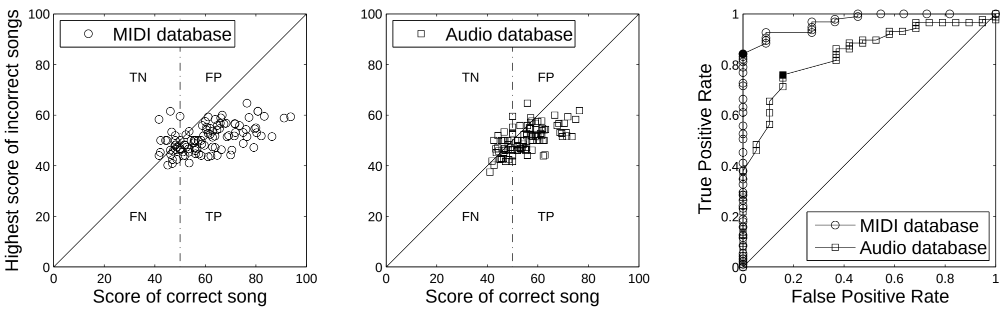

Figure 9: Analysis of the information given by the score assigned to the retrieved items for the MIDI and the audio databases. The plots to the left and center show the score of the correct song against the highest score of the wrongly retrieved elements for each query. The plot on the right shows a ROC curve for each database obtained by using different thresholds on the score value.

5. Discussion and conclusions

In this work a multimodal interface for music retrieval was considered, in which the user sings or hums a few notes of a melody as a query. The main drawback of these QBH systems is their difficult scalability, since manual annotation is required to build the database. A method was proposed to tackle this problem making it possible to extend an existing database automatically from audio recordings. A prototype of a complete system was developed in order to test the validity of the proposal. The experiments conducted show that the matching performance achieved is considerably high, obtaining 85% of the correct item in the top-10. Besides, the information provided by the scores assigned to the matching items can be exploited to determine the confidence in the retrieval.

As expected, the automatic singing melody extraction from audio recordings is not as accurate as the manual transcription, and this in turn decreases the performance of the QBH system. Nevertheless, even though the top-1 hit rate is significantly affected, the difference becomes less important for the top-10 and it is still above the reported rate for humans attempting to identify queries by ear (66%) [37]. Moreover, the evaluation of the audio based system yields an MRR of 0.76 for a database of 208 songs and 106 queries. Although a fair comparison between different experiments is not possible, the performance is encouraging given the best results for similar setups reported in other works (e.g. an MRR of 0.58 for a database of 427 songs and 159 queries [13], and an MRR of 0.56 for a database of 481 songs and 118 queries [14]). In addition, to the best of our knowledge, a direct comparison of the same QBH system based on MIDI files versus an audio based database has not been reported, which gives fairer insight on the performance gap between both approaches.

In future work further experiments should be conducted in order to assess the influence of the quality of the queries (e.g. tuning [14], length). Also, efforts must be devoted to develop a publicly available testbed for comparison of different methods, taking advantage of the existing resources, such as the ones provided by [14] and this work. In addition, there is still room for improvement in each stage of the proposed method, as shown by the singing voice detection evaluation. In particular, the analysis of the histograms and box-plots of Figure 5 suggest the use of a Gaussian modeling, such as the one proposed in [38]. In spite of the above, the current system constitutes a proof of concept that the approach of using automatic melody extraction methods seems promising, for example to increase the size of an existing MIDI based QBH system.

Acknowledgments

This work was partially supported by the R+D Program of the Comisión Sectorial de Investigación Científica (CSIC), Universidad de la República (UDELAR), Uruguay, by the CAPES/UDELAR Program of the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil, and STI-CAMSUD Mathematical Models for Visual Perception ans Subpixel Computer Vision. The authors would like to thank all the people that kindly recorded queries for the experiments.

References

[1] M. Müller, M. Goto, M. Schedl (Eds.), Multimodal Music Processing, Vol. 3 of Dagstuhl Follow-Ups, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2012.

[2] M. Müller, Information Retrieval for Music and Motion, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[3] A. Lerch, An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics, A John Wiley & Sons, Inc., publication, John Wiley & Sons, 2012.

[4] D. Leech-Wilkinson, The Changing Sound of Music: Approaches to Studying Recorded Musical Performance, Published online through the Centre for the History and Analysis of Recorded Music (CHARM), London, 2009.

[5] A. Klapuri, M. Davy (Eds.), Signal Processing Methods for Music Transcription, Springer, New York, 2006.

[6] Ò. Celma, Music Recommendation and Discovery - The Long Tail, Long Fail, and Long Play in the Digital Music Space, Springer, 2010.

[7] Y. Li, D. Wang, Singing voice separation from monaural recordings, in: Proceedings of the 7th International Conference on Music Information Retrieval, ISMIR 2006, Victoria, Canada, 8-12 October, 2006, pp. 176–179.

[8] M. Ryynänen, A. Klapuri, Transcription of the singing melody in polyphonic music, in: Proceedings of the 7th International Conference on Music Information Retrieval, ISMIR 2006, Victoria, Canada, 8-12 October, 2006, pp. 222–227.

[9] B. Pardo, J. Shifrin, W. Birmingham, Name that tune: A pilot study in finding a melody from a sung query, Journal of the American Society for Information Science and Technology 55 (4) (2003) 283–300.

[10] B. Pardo, D. Little, R. Jiang, H. Livni, J. Han, The vocalsearch music search engine, in: Proceedings of the 8th ACM/IEEE-CS Joint Conference on Digital libraries, JCDL '08, ACM, New York, NY, USA, 2008, pp. 430–430.

[11] J. Song, S. Y. Bae, K. Yoon, Mid-Level Music Melody Representation of Polyphonic Audio for Query-by-Humming System, in: Proceedings of the 3rd International Conference on Music Information Retrieval, ISMIR 2002, Paris, France, October 13-17, 2002, pp. 133–139.

[12] A. Duda, A. Nürnberger, S. Stober, Towards query by singing/humming on audio databases, in: Proceedings of the 8th International Conference on Music Information Retrieval, ISMIR 2007, Vienna, Austria, September 23-27, 2007, pp. 331–334.

[13] M. Ryynnen, A. Klapuri, Query by Humming of MIDI and Audio Using Locality Sensitive Hashing, in: Proceedings of the 2008 IEEE International Conference on Acoustics, Speech, and Signal Processing, Las Vegas, USA, March 30 - April 4, 2008, pp. 2249–2252.

[14] J. Salamon, J. Serrà, E. Gómez, Tonal representations for music retrieval: From version identification to query-by-humming, International Journal of Multimedia Information Retrieval, special issue on Hybrid Music Information Retrieval (2013) In Press.

[15] M. Rocamora, A. Pardo, Separation and classification of harmonic sounds for singing voice detection, in: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Vol. 7441 of Lecture Notes in Computer Science, Springer, 2012, pp. 707–714.

[16] E. López, M. Rocamora, Tararira: Query by singing system, in: The Second Annual Music Information Retrieval Evaluation eXchange (MIREX 2006), Abstract Collection, The International Music Information Retrieval Systems Evaluation Laboratory (IMIRSEL), Graduate School of Library and Information Science University of Illinois at Urbana-Champaign, 2006, pp. 80–83, extended abstract.

[17] A. Ghias, J. Logan, D. Chamberlin, B. C. Smith, Query by humming: musical information retrieval in an audio database, in: Proceedings of

the third ACM international conference on Multimedia, MULTIMEDIA '95, ACM, New York, NY, USA, 1995, pp. 231–236.

[18] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, S. J. Cunningham, Towards the digital music library: tune retrieval from acoustic input, in: Proceedings of the first ACM international conference on Digital libraries, DL '96, ACM, New York, NY, USA, 1996, pp. 11–18.

[19] N. Hu, R. B. Dannenberg, A comparison of melodic database retrieval techniques using sung queries, in: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries, JCDL '02, ACM, New York, NY, USA, 2002, pp. 301–307.

[20] Y. Zhu, D. Shasha, Warping indexes with envelope transforms for query by humming, in: Proceedings of the 2003 ACM SIGMOD international conference on Management of data, SIGMOD '03, ACM, New York, NY, USA, 2003, pp. 181–192.

[21] A. de Cheveigné, H. Kawahara, YIN, a fundamental frequency estimator for speech and music, The Journal of the Acoustical Society of America 111 (4) (2002) 1917–1930.

[22] A. Klapuri, Sound onset detection by applying psychoacoustic knowledge, in: Proceedings of the Acoustics, Speech, and Signal Processing, 1999. on 1999 IEEE International Conference - Volume 06, ICASSP '99, IEEE Computer Society, Washington, DC, USA, 1999, pp. 3089–3092.

[23] E. Pollastri, Processing singing voice for music retrieval, Ph.D. thesis, Universit Degli Studi Di Milano, Italy (2003).

[24] B. Pardo, W. P. Birmingham, Encoding timing information for musical query matching, in: Proceedings of the 3rd International Conference on Music Information Retrieval, ISMIR 2002, Paris, France, October 13-17, 2002, pp. 267–268.

[25] K. Lemström, String matching techinques for music retrieval, Ph.D. thesis, Department of Computer Science, University of Helsinki, Finland (2000).

[26] J. Salamon, M. Rohrmeier, A quantitative evaluation of a two stage retrieval approach for a melodic query by example system, in: Proceedings

of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe, Japan, October 26-30, 2009, pp. 255–260.

[27] D. J. Levitin, Memory for musical attributes, in: P. R. Cook (Ed.), Music, cognition, and computerized sound, MIT Press, Cambridge, MA, USA, 1999, pp. 209–227.

[28] P. Cancela, E. López, M. Rocamora, Fan chirp transform for music representation, in: Proceedings of the 13th International Conference on Digital Audio Effects, DAFx-10, Graz, Austria, September 6-10, 2010, pp. 330–337.

[29] M. Rocamora, P. Cancela, Pitch tracking in polyphonic audio by clustering local fundamental frequency estimates, in: Proceedings of the 9th Brazilian AES Congress on Audio Engineering , São Paulo, Brazil, May 17-19, 2011, pp. 80–87.

[30] L. Weruaga, M. Képesi, The fan-chirp transform for non-stationary harmonic signals, Signal Processing 87 (6) (2007) 1504–1522.

[31] J. Sundberg, The science of the singing voice, De Kalb, Il., Northern Illinois University Press, 1987.

[32] M. Rocamora, P. Herrera, Comparing audio descriptors for singing voice detection in music audio files, in: Proceedings of the 11th Brazilian Symposium on Computer Music, São Paulo, Brazil, September 1-3, 2007, pp. 187–196.

[33] D. P. W. Ellis, PLP and RASTA (and MFCC, and inversion) in Matlab, `http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/` (2005).

[34] J. S. Downie, The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research, Acoustical Science and Technology 29 (4) (2008) 247–255.

[35] I. H. Witten, E. Frank, Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[36] C. Hsu, C. Chang, C. Lin, A practical guide to support vector classification, Department of Computer Science, National Taiwan University-Online web resource: `http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf`.

[37] B. Pardo, W. P. Birmingham, Query by humming: How good can it get?, in: Workshop on the Evaluation of Music Information Retrieval Systems at SIGIR 2003, 1st August, Toronto, Canada, 2003, pp. 107–109.

[38] J. Salamon, G. Peeters, A. Röbel, Statistical characterisation of melodic pitch contours and its application for melody extraction, in: Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Porto, Portugal, October 8-12, 2012, pp. 187–192.