An Auditory Model Based Transcriber of Singing Sequences

L. P. Clarisse[1], J. P. Martens[1], M. Lesaffre[2], B. De Baets[3], H. De Meyer[4] and M. Leman[2]

[1] Department of Electronics and Information Systems (ELIS), Ghent University;
Sint-Pietersnieuwstraat 41, 9000 Gent (Belgium). martens@elis.rug.ac.be, 0032-09-2643395.
[2] Institute for Psychoacoustics and Electronic Music (IPEM), Ghent University.
[3] Department of Applied Mathematics, Biometrics and Process Control, Ghent University.
[4] Department of Applied Mathematics and Computer Science, Ghent University.

ABSTRACT

In this paper, a new system for the automatic transcription of singing sequences into a sequence of pitch and duration pairs is presented. Although such a system may have a wider range of applications, it was mainly developed to become the acoustic module of a query-by-humming (QBH) system for retrieving pieces of music from a digitized musical library. The first part of the paper is devoted to the systematic evaluation of a variety of state-of-the art transcription systems. The main result of this evaluation is that there is clearly a need for more accurate systems. Especially the segmentation was experienced as being too error prone (≈ 20 % segmentation errors). In the second part of the paper, a new auditory model based transcription system is proposed and evaluated. The results of that evaluation are very promising. Segmentation errors vary between 0 and 7 %, dependent on the amount of lyrics that is used by the singer. The paper ends with the description of an experimental study that was issued to demonstrate that the accuracy of the newly proposed transcription system is not very sensitive to the choice of the free parameters, at least as long as they remain in the vicinity of the values one could forecast on the basis of their meaning.

1. INTRODUCTION

It sounds appealing to have the possibility of retrieving a musical piece from a musical database, just by singing or humming an excerpt from that piece. In general, the proposed retrieval methodology is called Query-by-Humming (QBH). Both academic interest and practical appeal have encouraged the development of QBH systems over the last decade.

In this paper, we only consider singing sequences, be it that we make a distinction between singing with lyrics (i.e. singing the words), or singing without lyrics (i.e. singing with meaningless syllables like /da/, /na/, /du/, etc). Most dedicated state-of-the-art QBH systems were specifically designed for and tested on singing without lyrics. Some systems even put additional restrictions on the type of syllables that can be used (mostly /da/).

Nearly all QBH system consist of two parts: (i) an acoustic module for converting the acoustic input into a sequence of segments (time intervals) with associated discrete frequencies (notes), and (ii) a pattern matching module for matching this sequence to the musical data in a database. In case the acoustic signal is a singing sequence, notes cannot overlap in time. The result of the transcription system should thus be a segmentation of the signal into successive notes, optionally separated by white-spaces.

Most QBH systems (see for instance [10, 15, 19, 25]) are dedicated systems whose acoustic module always produces a result meeting this constraint. However, some systems use a general purpose wav-to-midi converter instead (see for instance [8, 14]). Such a converter may also produce overlapping notes, which may be resolved by a proper post-processing of the output before supplying it to the QBH pattern matcher.

In this paper we are solely dealing with the acoustic module of a QBH system. It is expected though, that the performance of the QBH system as a whole is highly dependent on the quality of the transcription provided by this module. This quality can be expressed in terms of the number of segmentation errors (deleted or inserted notes), substitution errors (the note was incorrect in terms of its frequency), and time alignment errors (the detected segment has different endpoints than the correct segment). The substitution errors mainly affect the transcribed melody, whereas the other errors mainly affect the rhythm.

Some QBH systems do not perform a segmentation (see for instance [5, 9, 18, 21]) and just convert the acoustic input into a pitch contour (e.g. one pitch sample per frame of 10 ms). It is our conviction however that similarity matching on the basis of pitch only is not powerful enough. In fact, an obvious objection is that it relies on the weakest point of a mediocre singer, namely the correctness of his pitch contour. Rhythm is also considered an important aspect of human music recognition, especially for the recognition of music with a less expressive pitch pattern, and there is no reason to believe that rhythm would be unimportant for an automatic QBH system. Therefore, all systems described in this paper intend to perform a segmentation.

The structure of this paper is as follows. In section 2 we outline the general principles underlying the acoustic modules of some state-of-the-art QBH systems. Then we describe our methodology for evaluating the transcriptions provided by such a module, and we present the results of an evaluation of 8 modules. Following the results of this evaluation we have developed a more accurate transcription system, as described in section 3. The evaluation of this new system is presented and discussed in section 4. The paper ends with some conclusions.

2. THE ACOUSTIC MODULE OF A QBH SYSTEM

The acoustic module of a QBH system always contains an acoustic front-end to transform the acoustic signal into a parametric representation of the time-frequency information carried by this signal. This parametric representation is then analyzed in detail by the transcription module, in order to produce the requested transcription of the acoustic signal.

2.1 The acoustic front-end

The acoustic front-end aims to extract features that are relevant for the transcription process. The main features usually are the energy (or some more complex estimate of the loudness of the signal), the pitch and the degree of voicing. The features are determined per frame of a certain length, and subsequent frames are typically shifted over 10 ms. If frames are chosen longer than 10 ms, subsequent frames overlap.

As far as we know, only the Haus and Pollastri system has extracted the degree of voicing. The extraction is based on the mean and standard deviation of the energy and the zero-crossing rate of the derivative of the background noise and the signal. Using this information, the system tries to discriminate between vowels, voiced consonants and unvoiced sounds.

Traditionally, pitch detection has received most attention in the acoustic front-end of a QBH system. By far the mostly used pitch determination method is the autocorrelation method (see for instance [5, 8, 21]). Meldex on the other hand uses the Gold-Rabiner algorithm [20].

2.2 The transcription system

Transcription systems consist of two parts: a segmentation part, in which the audio input is divided into note segments and white-spaces, and a note assignment part, in which a single note (frequency) is assigned to every note segment. The methods of doing such, vary widely from system to system. We will summarize the methods adopted by two well documented systems.

Meldex [15, 16, 17] uses a segmentation which is purely based on the root mean square (RMS) - the square root of the energy - as a function of time. A note onset is recorded when the RMS exceeds some threshold and a note offset is recorded when the RMS drops below a second lower threshold. The thresholds were respectively set at 35% and 55% of the mean RMS over the entire signal. A note is assigned to the segment by identifying the highest peak in the histogram of the frame-level pitch frequencies found in the segment, and by computing the average of the pitches lying in that bin. The pitch is then converted to a MIDI note using a scale which is adapting to the intonation of the user. The idea is to keep track of the bias in the computed frequency of the singer, and to subtract this before performing the note assignment. As shown in [10] however, simply rounding the computed to the closest note frequency yields a better performance.

The system of Haus and Pollastri [10] is more elaborate. The segmentation process starts with a first estimation of segment boundaries based on signal/noise discrimination, with the noise level set to 15% above the RMS of the first 60 ms of the input. Next, the on/offset estimation is refined by incorporating the detection of vowels, voiced consonants and unvoiced sounds. The pitch of a segment is computed on the basis of the frames labeled as vowel in this segment. After the fundamental frequencies have been detected, they are median filtered (mediating three subsequent frames) and checked for octave errors. Four adjacent frames with similar fundamental frequencies are grouped into a block. Legato is detected when subsequent blocks have pitches more than 0.8 semitones apart. In this case additional segment boundaries are inserted. Just like in the Meldex system one aims at capturing the intention of the singer. Conversion from frequencies to the equally tempered scale incorporates a relative scale. The relative scale is based on the assumption that each singer has a reference tone in mind and that the other notes are sung relative to the scale constructed on that tone. The first thing Pollastri tries to do is to look for the reference tone. Global pitches of the segments are compared to an absolute scale and differences are represented in a histogram of overlapping bins of 0.2 semitones. The prominent peak is identified and an average is made over this winning bin to find the shift transforming the absolute scale into the user scale. Shifting the absolute scale by this amount minimizes the deviation error and thereby it is claimed that the user scale has been found. Further refinements are made on the basis of additional rules.

3. EVALUATION OF TRANSCRIPTION SYSTEMS

In order to evaluate the quality of a system for the transcription of singing sequences one needs (i) a representative corpus of singing sequences by naive singers, (ii) a reliable reference transcription of these sequences, and (iii) a good method for measuring the discrepancies between the generated and the reference transcriptions in a quantitative manner.

3.1 Corpus collection

Five men and six women of different ages (between 23 and 51 years old) were asked to sing two excerpts from two different songs. They were free to choose how they would sing: with or without lyrics. All subjects were unexperienced singers. They were free to choose a melody from a list of 50 which they had in front of them. The subjects were invited in the room where the computer was, they were given the list, they decided what tunes they would sing, and they immediately started to sing. The recordings were made in a normal office room with a home PC and a hand-held microphone (type Sony ecm ms907). The samples were recorded at a sampling frequency of 22.05 kHz with a resolution of 16 bit, and saved as a PCM wav file.

A typical phenomenon was that the volume (loudness) was quite large at the beginning, but much lower at the end. It also happened frequently during singing with lyrics that the subjects fell out of words and continued by singing parts without lyrics.

In total, 22 samples were recorded. Two recordings, one of a male and one of a female subject, were taken out for algorithm development and tuning, the remaining 18 samples (7 without and 11 with lyrics) were considered as an evaluation corpus. This corpus consisted of 150 seconds of acoustic signal, containing approximately 300 notes. Obviously, this corpus is too limited to be really representative, but it was considered large enough to yield at least good indications of expected system performances.

3.2 Making the reference transcriptions

In order to get a reliable reference transcription, a musical expert was asked to segment the speech into notes and white-spaces. It was found convenient to use the open source tool PRAAT [4] for this purpose. The musical expert had to introduce time markers indicating the beginning or end of a note, and to assign a note or white-space label between two time markers. For doing this, the expert had a visual image of the signal on the screen (see figure 1 which shows a screen dump), and the ability to listen repeatedly to any fragment of the signal. The note labeling is found to be the most time consuming part of the task.

Once the note labeling was ready, it was saved in the TextGrid format of Praat, and subsequently converted to a MIDI format [24], the format that is used by most transcription systems.

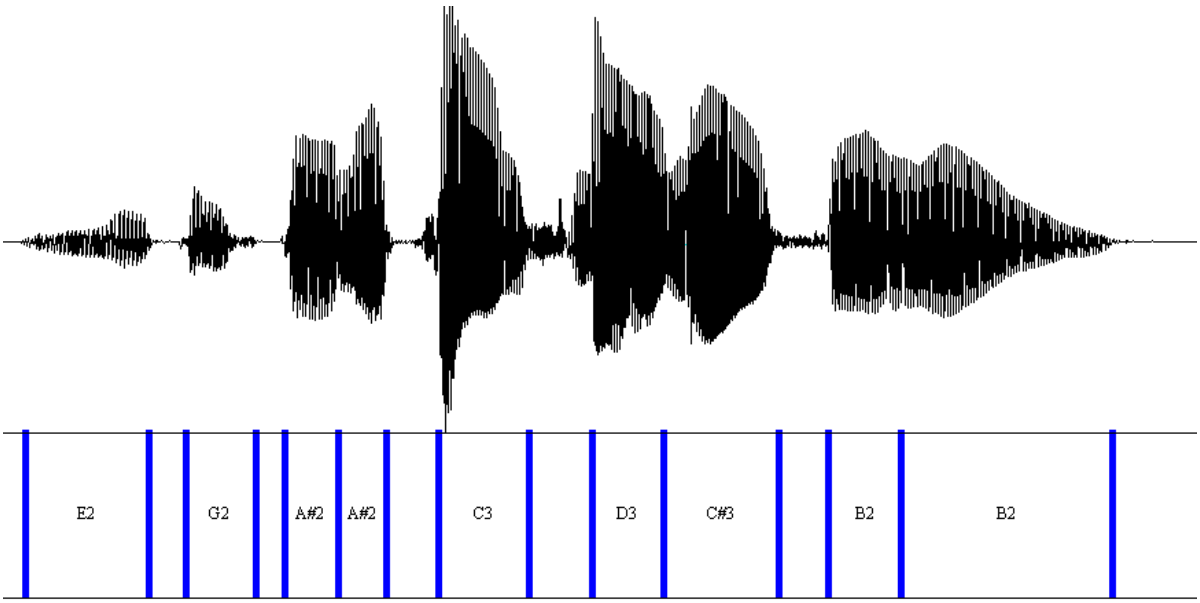

Figure 1: A screen dump of the image in front of the musical expert after he has introduced the note boundaries and the note labels, according to the annotation scheme of the Autoscore system.

3.3 Evaluation methodology

The goal of the evaluation is to compare a computed transcription with the reference transcription of the signal. As both can consist of a different number of segments (notes and white-spaces), a direct comparison is not straightforward. However, a simple solution is offered by the Dynamic Time Warping algorithm (DTW) [23].

If the computed and reference transcriptions are characterized by N_c time markers $t_{c,i}$ and N_r time markers $t_{r,j}$ respectively, we want DTW to align each $t_{c,i}$ with a $t_{r,j}$ in such a way that the accumulation of local costs attached to these alignments is minimized. I.e., DTW must identify the warping path $\hat{\mathbf{w}}$ satisfying

$$\hat{\mathbf{w}} = \arg\min_{\hat{\mathbf{w}}} \sum_{i=1}^{N_c-1} c(t_{c,i}, t_{c,i+1}, t_{r,w_i}, t_{r,w_i+1})$$

The pairs $(i, j = w_i)$ can be represented as points on a path from (1,1) to (N_c, N_r) in a two-dimensional trellis (see figure 2).

The path consists of subsequent transitions characterized by displacements $\Delta i = 1$ and $\Delta j = w_i - w_{i-1}$. In order to obtain a sensible path, $\Delta j \in (0, 4)$ was imposed as a constraint. Obviously, the definition of the local cost contributions will determine the properties of the alignment one obtains for a specific transcription pair. Our first goal was to penalize the time differences between the computed and their associated reference time markers. The note frequency discrepancies were considered as a secondary criterion. This way, the alignment does not depend too much on the quality of the pitch detector. The local cost contribution of a transition $(\Delta i, \Delta j)$ was therefore determined on the basis of the following considerations:

- $\Delta j = 0$ means that two computed time markers are assigned to the same reference marker. This points to an inserted time marker in the computed transcription, and is penalized with an insertion cost $c_{ins} = 0.95$.
- $\Delta j > 0$ means that a new computed time marker is assigned to a new reference marker. In this case one considers two discrepancies: a discrepancy in the timing, and a discrepancy in the note frequencies of the segments starting at these time markers (a different note is considered as a note substitution error). The timing cost c_{time} is equal to the absolute time difference divided by some ΔT_{max} which was set to 0.2s. The substitution cost c_{sub} is equal to the minimum of 0.5 semitones and 0.25 times the note frequency difference in semitones. The substitution cost for assigning a note to a white-space is set to 2 semitones.
- $\Delta j > 1$ means that some reference time marker is not assigned to any computed marker. This is penalized by an extra deletion cost c_{del} multiplied by the number of deletions $(\Delta j - 1)$.

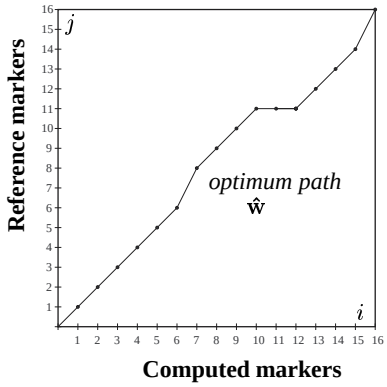

Figure 2: A warping path $\hat{\mathbf{w}}$ representing an alignment of the automatic and the reference transcription of a singing sequence.

Once the alignment between the transcriptions is available, one can easily determine the number of deletions and insertions along the warping path. For determining the number of substitutions, we distinguished between exact or not, and between within a semitone or not. In case two or more computed segments were assigned to the same reference segment, the decision was based on a comparison of the frequency of the computed segment that had the largest overlap with the reference segment.

4. EVALUATION OF STATE-OF-THE-ART

In this section we describe an experimental evaluation of 8 different systems which are assumed to represent the state-of-the-art in transcribing singing sequences.

Before reviewing the systems that were tested, we recall that some of them allow the user to specify a lot of free parameter settings. In all cases we used the preferred settings specified in the manual. If the note range could be specified it was always set to (C2,C6) = (65 Hz,1000 Hz).

Some programs seemed to introduce some delay. For that reason we allowed transcriptions to be shifted in time before supplying them to the DTW algorithm. The results presented later always correspond to the time shift producing the lowest alignment cost.

4.1 Evaluated transcription systems

Some of the systems that were tested are commercial systems, which are not well documented in terms of underlying methodologies. However, where references to scientific publications can be made, they are included. Let us look at the list of the five systems for which detailed results are provided in table 1:

Meldex This is maybe the most famous QBH system. For a recent and detailed overview, we refer to [16].

Pollastri The system of Haus and Pollastri [10] was developed in the context of query by humming, with the term humming referring to singing without lyrics. In this case, the transcription of our material was performed by the author himself. We got the assurance from Pollastri that the conversion was made under the same conditions as specified in [10].

Akoff Composer This is a shareware program by Andrei Kovalev [1] for the conversion of monophonic music waves to a MIDI file format.

Widi This is a polyphonic music recognition system developed by Russian students in physics [28]. It has a monophonic mode, and it is in this mode that we tested it.

Autoscore This is another off-the-shelf monophonic music to MIDI converter [3]. This system has already been used for query by humming by Naoko Kosugi [14].

Three other systems that were tested are the commercial packages **Audioworks** [2], **Digital Ear** [7] and **Intelliscore** [12]. These systems performed (according to our tests) worse than Akoff, Widi or Autoscore, and were therefore not included in table 1.

4.2 Detailed evaluation results

The evaluation results are summarized in table 1. The results are separated according to the singing mode: with or without lyrics. Two important conclusions can be drawn from these results:

1. All systems make a considerable amount of deletion and insertion errors, and singing with lyrics seems to be much more difficult to segment than singing without lyrics.

2. Although exact note recognition is low for all systems, most systems provide a within 1 semitone note recognition accuracy of 85.00 % or more. Especially Widi seems to incorporate an excellent pitch extractor. However, this is not necessarily true since Widi produces many short (inserted) notes for unstable segments, and consequently there is a high chance that the longest segment in the more stable part of the note has the correct pitch.

Note that for the published systems, our evaluation results appear to be significantly worse than those reported in these publications. One likely explanation is that the system performances depend too much on the recording conditions (volume, noise, room acoustics) and the parameter settings. Another explanation may be that we used naive singers, and that for singing without lyrics, we did not force them to use a particular syllable (e.g. the /ta/-syllable, probably the most easy one to analyze).

Listening to the transcribed sequences convinced us of the absolute need for more accurate segmentations. Even the best system (Pollastri) is usually unable to provide a sufficiently accurate segmentation of the singing with lyrics sequences. That is why we have conceived a new transcription system that is described in the next section.

5. A NEW TRANSCRIPTION SYSTEM

The acoustic module of our QBH system comprises an auditory model which is essentially the same model as that published in [26]. However, it is the first time we have used it for the analysis of human singing. Our main motivations for preferring an auditory model over a more standard acoustic front-end are the following:

Table 1: Overview of the results obtained by comparing computed and reference transcriptions using the methodology outlined in section 3.

	Akoff	Autoscore	Meldex	Widi	Pollastri
Singing without lyrics					
notes deleted	6.72 %	7.26 %	37.31 %	5.22 %	4.76 %
notes inserted	11.19 %	14.29 %	4.48 %	64.18 %	7.94 %
notes deleted + inserted	17.91 %	21.55 %	41.79 %	69.40 %	12.70 %
exact note recognition error	40.71 %	54.26 %	53.73 %	31.15 %	48.31 %
note recognition error $>$ 1 semitone	4.42 %	10.64 %	28.36 %	1.64 %	10.37 %
Singing with lyrics					
notes deleted	18.50 %	22.95 %	52.46 %	18.50 %	13.66 %
notes inserted	30.00 %	12.02 %	3.28 %	60.50 %	5.46 %
notes deleted + inserted	48.50 %	34.97 %	55.74 %	79.00 %	19.13 %
exact note recognition error	48.34 %	44.27 %	66.23 %	34.72 %	58.39 %
note recognition error $>$ 1 semitone	13.91 %	15.27 %	31.17 %	6.25 %	16.79 %

1. We were able to prove that the speech loudness pattern emerging from the model provides excellent cues for the phonetic segmentation of speech [27].

2. The built-in pitch extractor, called AMPEX (Auditory Model based Pitch Extractor) has been proven to be among the best pitch extractors available for speech analysis [11, 22].

3. Since the pioneering work of Davis and Mermelstein [6], the perceptually based MFCCs (Mel-Frequency Cepstral Coefficients) have become the standard parametric representation for speech recognition applications.

In the subsequent sections we first describe our auditory model and the improvements that were made since the original publication of the model. Then we introduce the segmentation and pitch assignment algorithms that were developed to produce the envisaged transcriptions.

5.1 The auditory model

A general outline of the auditory model is depicted in figure 3. The acoustic signal is first filtered by a band-pass filter that models the sound transmission in the outer and middle ear. The filtered signal is then supplied to a cochlear processing block which models the conversion of the acoustic signal into neural firing patterns observed in groups of auditory nerve cells. Each group represents nerve cells connected to neighboring hair-cells somewhere along the Basilar Membrane (BM) in the cochlea. The number of cells in a group is assumed to be large enough so as to make it sensible to characterize the group response by means of a time pattern representing the neural firing density as a function of time. Each pattern is obtained by one analysis channel consisting of a band-pass filter with a unique tuning frequency, a non-linear hair-cell model and an envelope extractor. In agreement with physiological measurements [13], the neural fibers do not transmit modulation frequencies that are much larger than 500 Hz.

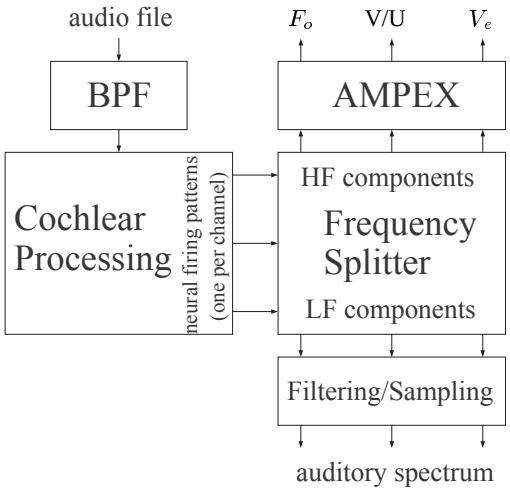

Figure 3: Architecture of the auditory model front-end.

Each neural firing pattern is then split into a low and a high-frequency component by means of a frequency splitter with a characteristic frequency of 20 Hz. The low-frequency components are decomposed of their spontaneous activity (value in the absence of any signal), further low-pass filtered and down-sampled to 100 Hz so as to form the components of an auditory spectrum. The latter represents the short-term neural activity (loudness) distribution across channels. The high-frequency components are supplied to a pitch extraction module, called AMPEX (Auditory Model based Pitch EXtractor). It produces one pitch per frame, and it is consists of three major parts:

1. A *pseudo-autocorrelation analysis* of the individual high-frequency components $f_{hm}(t)$: $f_{hm}(t)$ is replaced by a sequence of pulses occurring at the positions of its maxima, and a function $R_m(\tau)$ very much similar to a short-time autocorrelation function is derived from this signal. The channel contributions are then accumulated to a global pseudo-autocorrelation function $R(\tau)$.

2. A *pitch candidate extraction algorithm* that identifies all relevant peaks (larger than a small threshold) in $R(\tau)$, and thus produces a set of pitch candidates T_k and their corresponding evidences $E_k = R(T_k)$ for each frame.

3. A *pitch continuity analysis* to retrieve the best pitch T_o, its corresponding voicing evidence V_e, and a voiced/unvoiced decision for each frame. If T_{jk}, E_{jk} $(k = 1, .., N_j)$ represent the pitch candidates and their evidences hypothesized in frame j, and if the frame rate is 10 ms, the voicing evidence of a pitch candidate T hypothesized in frame n is computed as
$$V_e(T) = \sum_{j=n-2}^{n+2} \sum_{k=1}^{N_j} E_{jk}\ \delta(\frac{|T - T_{jk}|}{T + T_{jk}} < \epsilon_T) \qquad (1)$$
with $\delta(.)$ being 1 if the condition is satisfied and 0 otherwise, and with ϵ_T being a coincidence threshold. The pitch candidate with the highest V_e is selected as the pitch, and a voiced/unvoiced decision is made on the basis of this evidence (see [26]).

The auditory model is designed in such a way that it can process a continuous audio stream. Obviously, due to the pitch continuity analysis, there is a delay of 20 ms between the acoustic input and the model output. When aligning the auditory features with the acoustic signal, one has to compensate for this delay.

Since its publication in [26], AMPEX was further improved in the following ways:

1. In order to make the voiced/unvoiced decision less dependent on the signal level, the evidence assigned to a pitch candidate T during the pitch candidate extraction stage is no longer not $R(T)$ but $R(T)/[R(0) + \epsilon M]$, with M being the number of channels in the auditory model.

2. In order to reduce the number of harmonic pitch errors, the pitch evidences computed in the pitch continuity analysis were multiplied by $0.5 + 0.1T$ (T in ms) so as to compensate for the tendency of the algorithm to produce somewhat larger evidences for smaller values of T.

3. The pitch continuity analysis continues to seek for the pitch candidate T getting the highest evidence according to equation (1), but it then determines the effectively generated pitch hypothesis as
$$T_o(T) = \frac{\sum_{j=n-2}^{n+2} \sum_{k=1}^{N_j} T_{jk} E_{jk}\ \delta(\frac{|T - T_{jk}|}{T + T_{jk}} < \epsilon_T)}{\sum_{j=n-2}^{n+2} \sum_{k=1}^{N_j} E_{jk}\ \delta(\frac{|T - T_{jk}|}{T + T_{jk}} < \epsilon_T)} \qquad (2)$$

With these improvements, the total pitch and V/U error rate for the speech database used in [26] was reduced from 5.1 to 3.7 %, and there is also a much better balance between the performance for male and female voices now.

So as to reduce the CPU time, different channels are operated at different sampling frequencies. The auditory model therefore contains a decimation unit to supply down-sampled copies of the input signal to these channels. This unit was enhanced in two ways with respect to [26]:

1. In order to prevent aliasing products of high-frequency tones to produce activity in low-frequency channels, a higher order decimation filter (with a high-frequency suppression of more than 66 dB) was introduced.

2. In order to prevent harmonics, introduced by the half-wave rectifier in the hair-cell models, to produce low-frequency aliasing products in the hair-cell outputs, the sampling frequency in a channel has to be larger than 7.2 times the center frequency of the channel bandpass filter (see [26]). In order to satisfy this condition for the high frequency channels, the decimation unit was extended to produce an up-sampled version of the input signal as well.

In all the experiments reported in this paper, the auditory model comprises 23 channels covering the frequency range from 140 Hz to 6 kHz, and it produces one acoustic parameter vector per 10 ms. Each vector consists of an auditory spectrum (23 values), a voiced/unvoiced decision, a voicing evidence, a loudness value and a pitch frequency (zero if the frame is unvoiced).

It is important to emphasize that all the free parameters of the auditory model were optimized for normal speech processing. They were not changed for the analysis of the singing sequences appearing in the present study.

5.2 The segmentation algorithm

To begin with, the auditory spectrum components of a frame are accumulated across channels to produce the so called loudness of that frame. The pitch (F_o), loudness (L) and voicing evidence (V_e) pattern for a two-seconds extract from a singing sequence is depicted on figure 4. These are the patterns which are further analyzed by our segmentation system.

The segmentation is primarily based on the loudness function, whose deep minima are supposed to delimit the note segments. In order to obtain a robust decision, the deep minimum detection algorithm must be able to deal with loudness fluctuations which are not referring to note boundaries.

We have implemented a robust extremum detection algorithm which assumes that there is some silence at the beginning of the file. The algorithm goes from left to right, it starts by searching for a maximum and it proceeds according to the following principles.

1. *While searching for a maximum*
   Keep track of the position and the value of the largest loudness (stored as the potential maximum), and consider a maximum found at the moment the actual value is sufficiently lower than the stored maximum. When a maximum is found, store the position and loudness of the actual frame as a potential minimum and start looking for a minimum.

2. *While searching for a minimum*
   Keep track of the position and the value of the smallest loudness (stored as the potential minimum), and consider a minimum found at the moment the actual value is sufficiently higher than the stored minimum. When a minimum is found, generate a new note segment (starting at the previous minimum), store the position and loudness of the actual frame as a potential maximum and start looking for a maximum.

To determine what sufficiently higher/lower is, we adopt Weber-Fechner's law of psycho-acoustics [29]. It states that equal increments of sensation due to some energy variable are associated with equal increments of the logarithm of that variable supplemented with some bias. Consequently, if L_b is the loudness bias, loudness L_2 is sufficiently higher/lower than L_1 if $|L_2 - L_1|/(L_1 + L_b)$ exceeds some threshold ϵ_L.

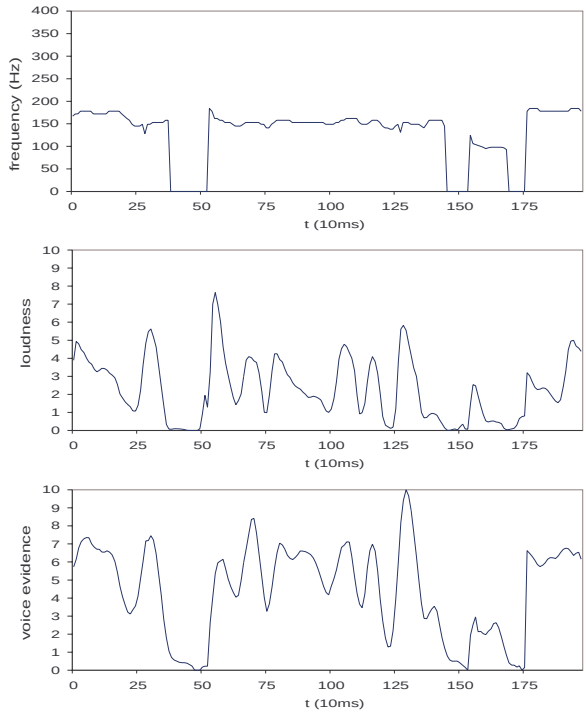

Figure 4: The pitch, loudness and voicing evidence patterns emerging from the auditory model.

In order to detect white-spaces too, the extremum detection algorithm is further extended as follows. When the loudness goes under some white-space threshold L_{ws} in more than 2 successive frames, a note segment is generated and the search for extrema is inhibited until 2 successive frames with a loudness above L_{ws} are encountered. At that moment, a white-space segment is generated, and a new search for a maximum is started.

The white-space threshold can be made adaptive and proportional to the lowest loudness found over the last two seconds, but as we normalized the energy of the singing sequences before analyzing them, it was possible to select a fixed threshold throughout the experiments.

5.3 Post-processing the segments

It happens that low energy segments like breaths and noises appear as note segments in the computed segmentation. In a segment post-processing stage, we relabel them as white-spaces as soon as they satisfy one of the following conditions:

1. the maximum voicing evidence is smaller than V_{min}
2. the maximum loudness is smaller than αL_{ws} ($\alpha > 1$).

This post-processing stage completes the segmentation process.

5.4 The pitch assignment algorithm

In order to determine the note label of a note segment, the pitch contour is analyzed in the center part of that segment. The onset

and offset of a note segment are excluded because pitch algorithms have a tendency to make pitch errors in these areas. On the other hand, the more pitch values one can retain, the more accurate the computed pitch is going to be. We choose to consider the first and last 2 frames as the onset and offset of the note segment. The note label is obtained in two steps.

Step 1: segmental pitch determination

The segmental pitch is computed as the average of F_o over the frames of the segment (central part). To cope with possible octave errors this average is iteratively improved by eliminating those frames whose pitch deviates more than a certain ΔF_o from the actual average, and by computing a new average on the basis of the remaining frames. The process is stopped as soon as the segmental pitch does not change anymore. Usually this happens after one or two iterations.

If one wants to maximize the note recognition within a semitone, one intuitively feels that ΔF_o should be smaller than $\sqrt[6]{2} - 1 \approx 0.12$. We have not tried to optimize this value, and used $\Delta F_o = 0.10$ in all our experiments.

In some exceptional cases, a segment may contain so many octave errors that there are almost no pitch values within ΔF_o of the first segmental pitch approximation. To get the right frequency in this case, an escape route is followed. It consists of constructing a histogram of the frame pitches and selecting the most likely value as the segmental pitch.

Step 2 : note labeling

Once the segmental pitch is determined, it can be converted to a MIDI note using the equally tempered frequency scale. Using the conventions that A4 corresponds to 440Hz and that MIDI note zero corresponds to C-1, one readily finds that

$$\text{MIDI-note}(F_o) = \frac{\log(F_o/F_{ref})}{\log\ \sqrt[12]{2}}; \quad F_{ref} \approx 8.1758 \text{ Hz} \qquad (3)$$

We always round the frequency to the nearest MIDI note. No attempt is made to adjust to the scale of the user. For the moment we are only interested in transcribing the sequences as precisely as possible, disregarding the intention of the singer.

6. EXPERIMENTAL RESULTS

Our system was evaluated in exactly the same way as the state-of-the-art systems were in section 4.

6.1 Parameter tuning

The free parameters of the algorithm were optimized on the recordings of one male and one female singer which did not contribute to the evaluation corpus (see section 3.1). In table 2 we have listed the parameters, their meaning and their values. The parameters are grouped according to their appearance in the segmentation, the segment post-processing and the note assignment stages.

Table 2: Internal parameters and their settings found by empirical evaluation.

parameter	meaning	value
ϵ_L	min. loudness deviation	35%
L_b	loudness bias	2.5% of maximum
L_{ws}	white-space threshold	2.5% of maximum
V_{min}	min. note voicing evidence	15% of maximum
α	min. note loudness vs L_{ws}	3
ΔF_o	max. frequency deviation	10%

6.2 Evaluation results

The results of our evaluation are labeled MAMI (after the name of our project: Musical Audio Mining) in table 3. They are presented in opposition to the results of the best state-of-the-art system according to our previous tests.

Table 3: Transcription results for the proposed system (MAMI) as compared to the results of the Pollastri system.

	MAMI	Pollastri
Singing without lyrics		
notes deleted	0.00 %	4.76 %
notes inserted	2.24 %	7.94 %
notes deleted + inserted	2.24 %	12.70 %
exact note recognition error	35.88 %	48.31 %
note recognition error $>$ 1 semitone	1.53 %	10.37 %
Singing with lyrics		
notes deleted	4.92 %	13.66 %
notes inserted	2.19 %	5.46 %
notes deleted + inserted	7.10 %	19.13 %
exact note recognition error	42.01 %	58.39 %
note recognition error $>$ 1 semitone	6.51 %	16.79 %

Apparently, both types of singing sequences are much better transcribed by the MAMI system. The remaining 2.24% segmentation errors in the singing without lyrics sequences all appear in one short sequence which is sung with very unstable notes. The exact note recognition errors are spread over the files. The note recognition within a semitone is always very high (98.5% on average), ensuring enough precision for a QBH application. Five of the seven singing without lyrics sequences were transcribed 100% correctly.

Segmenting singing with lyrics has also reached an acceptable level now (about 7% segmentation errors on average). The note recognition, although not as good as for singing without lyrics, is also quite reliable (about 93.5 % on average).

It appears that over the whole set of 18 files no octave errors have been made. The largest note deviation is 4 semitones, and it occured only once.

6.3 Sensitivity to parameter settings

The main parameter for controlling the segmentation algorithm is ϵ_L. It was verified experimentally that the total deletion+insertion error rates are not much affected as long as ϵ_L stays in the range of 25% to 45%. In this range, loudness fluctuations due to legato/vibrato usually do not emerge in inserted note boundaries.

The only parameter that controls the pitch assignment is ΔF_o. Changing this parameter from 10 to 100% resulted in an increase of the note recognition error within 1 semitone of only 2%. This is owed to the large robustness of the AMPEX pitch detector.

Omitting the segment post-processing stage shows a 3 % increase of the insertion error rate. Especially in the longer sequences breath removal seems to be necessary.

The bottom line is that the settings of the free parameters are not critical, and the optimal settings are very much in line with what one would expect on the basis of their meaning.

6.4 Limitations of the present system

As AMPEX analyzes temporal fluctuations in the envelope patterns of the auditory model hair-cell outputs, it cannot detect a pitch much larger than 500 Hz. This means that whistling and singing with a

high pitch cannot be handled by the present system. In spite of this we obtain good results because our corpus contains only one file with some whistling in it.

Monophonic instruments can in principle be handled adequately by AMPEX as long as their pitch remains below 500 Hz. However, we did not perform any test to confirm this.

So as to attain a higher applicability of the system, we are currently developing a frequency-based pitch extractor to complement the time-based AMPEX algorithm. The frequency-based extractor will identify maxima in the auditory spectrum, and use them to derive a best pitch estimate and its evidence. Using this extension, it should also become possible to handle whistling and monophonic instruments with a high pitch.

7. CONCLUSIONS

We have established that most transcription systems are incapable of accurately transcribing singing sequences of naive singers. Three problems were identified: (i) they offer but a poor segmentation, (ii) they can only handle singing with specified syllables (e.g. /ta/), and (iii) their performance is very sensitive to the choices of the free parameters. Some systems even require training from the user.

Astonished by this result, we have developed a new auditory model based transcription system that seems to perform an acceptable segmentation and note labeling of free singing (with or without lyrics, and without any restrictions on the syllables used in singing without lyrics). In addition, the performance of the algorithm is not very sensitive to the settings of its free parameters.

8. ACKNOWLEDGMENTS

We thank Gaetan Martens, Koen Tanghe and Dirk Van Steelant for valuable discussions on the subject. We also acknowledge Emanuele Pollastri for testing his system on our corpus.

This research was supported by project ”Musical Audio Mining” (010035-GBOU) which is funded by the Flemish Institute for the Promotion of the Scientific and Technical Research in Industry.

9. REFERENCES

[1] Akoff music composer 2.0. Akoff Sound Lab. http://www.akoff.com.

[2] Audioworks 2.15. http://www.audioworks.com.

[3] Autoscore Deluxe 2.0. Wildcat Canyon Software. http://www.wildcat.com

[4] Boersma P. and Weenink D. Praat. A system for doing phonetics by computer. Report 132, Institue of Phonetics (Amsterdam). (http://www.praat.org)

[5] Dannenberg R.B., Mazzoni D. (2001). Melody Matching Directly From Audio. Procs ISMIR 2001, 17-18.

[6] Davis S, Mermelstein P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans. ASSP 28, 357-366.

[7] Digitalear. Epinoisis Software. http://www.digital-ear.com

[8] Francu C., Nevill-Manning C.G. (2000) Distance metrics and indexing strategies for a digital library of popular music. Proc. IEEE Int. Conf. on Multimedia and Expo, 889-892.

[9] Ghias A., Logan J., Chamerlin D., Smith B.C. (1995). Query By Humming. Musical Information Retrieval in an Audio Database. Procs ACM Multimedia 1995. 231-236.

[10] Haus G., Pollastri E. (2001). An Audio Front End for Query-by-Humming Systems. Procs ISMIR 2001, 65-72.

[11] Hermes D. (1992). Pitch Aanalysis. Visual representations of speech analysis (edts M. Cooke, S. Beet). Wiley & Sons.

[12] Intelliscore 4.0, Innovative Music Systems Inc. http://www.intelliscore.net

[13] Johnson D. (1980). The relationship between spike rate and synchroning in responses of auditory-nerve fibers to single tones. J. Acoust. Soc. Am. 68, 1115-1122.

[14] Kosugi N., Nishihara Y., Sakata T., Yamamuro M. and Kushima K. (2000). A Practical Query-By-Humming System for a Large Music Database. Procs ACM Multimedia 2000, 333-342.

[15] McNab R.J., Smith, L.A. and Witten I.H. (1996). Signal Processing for Melody Transcription. Australian Computer Science Conference, 301-307.

[16] McNab R.J., Smith L.A., Witten I.H. and Henderson C.L. (2000). Tune Retrieval in the Multimedia Library. Multimedia Tools and Applications, 113-132.

[17] Meldex is a part of the New Zealand Digital Library project. Webpages www.nzdl.org/musiclib and www.nzdl.org.

[18] Nishimura T., et al.(2001). Music Signal Spotting Retrieval by a Humming Query Using Start Frame Feature Dependent Continuous Dynamic Programming. Procs ISMIR 2001, 211-218.

[19] Prechelt L., Typke R. (1998) An Interface for Melody Input. Unpublished (see http://wwwipd.ira.uka.de/tuneserver).

[20] Rabiner L.R. et al. (1976). A comparative performance study of several pitch detection algorithms. IEEE Trans ASSP 24, 399-418.

[21] Roger Jang J.-S., Chen J.-C., Gao M.-Y. (2000). A Query-by-Singing System based on Dynamic Programming. Int. Workshop on Intelligent Systems Resolutions (8th Bellman Continuum), 85-89.

[22] Rouat J., Liu Y., Morisette D. (1997). A pitch and voiced/unvoiced decision algorithm for noisy speech. Speech Communication 21, 191-207.

[23] Sakoe H. and Chiba S. (1978). Dynamic programming algorithms optimization for spoken word recognition. IEEE Trans ASSP 26, 43-49.

[24] Sapp C.S. Improv software MIDI-library. http://improv.sapp.org.

[25] Sonoda T., Goto M., Muraoka Y. A WWW-based Melody Retrieval System (1998). Procs ICMC 1998, 349-352.

[26] Van Immerseel L. and Martens J.P. (1992). Pitch and voiced/unvoiced determination with an auditory model. J. Acoust. Soc. Am. 91, 3511-3526.

[27] Vorstermans A., Martens J.P., Van Coile B. (1996). Automatic segmentation and labeling of multi-lingual speech data. Speech Communications 19, 271-294.

[28] Widi music recognition system 2.7, Music Recognition Team. http://www.widisoft.com.

[29] Zwicker E. and Terhardt, E. (1974). Facts and Models in Hearing. Springer, Berlin/Heidelberg.