in Proc. of IV Jornadas de Engenharia Electrónica
e Telecomunicações e de Computadores, pp186-191
November 2008, Lisbon, Portugal

AUTOMATIC TRANSCRIPTION OF MUSICAL-WHISTLING:
Comparing Pitch Detection Methods

Bruno Dias, Rodrigo Ventura, José Gaspar
Instituto de Sistemas e Robótica, Instituto Superior Técnico,
Universidade Técnica de Lisboa, Portugal
bfsd@ist.utl.pt; {yoda,jag}@isr.ist.utl.pt

Keywords: Music transcription, pitch tracking, whistling.

Abstract: *In this paper we describe an automatic system for transforming (transcribing) man-made melodic whistles into MIDI-like symbolic representations. Given the monophonic nature of whistling, our system is mainly based in pitch detection and tracking methodologies. In particular, we compare four pitch detection techniques: Temporal Autocorrelation Function (tACF), Average Magnitude Difference Function (AMDF), Spectral Autocorrelation Function (fACF), and the Harmonic Product Spectrum (HPS). Results for both synthetic and real (man-made) whistling signals are presented in the paper, showing that the system can effectively do the transcription work. A comparative evaluation of the four pitch detection algorithms is also performed.*

1 Introduction

With the continuous advances in digital signal processing techniques, the automatic transcription from an acoustic waveform to a symbolic representation is now possible. Musical transcription of audio data is the process of taking a segment of digital audio data and extracting from it the symbolic information that can be represented, for instance, in a music score[1]. It can be viewed as reverse-engineering the "source code" of a music signal [3], that we do not expect the average human to be able to do that easily.

The reason why we have chosen whistling as our input instrument is due to its universality and accessibility to anyone regardless of the musical background. The most common mode of whistling, and the one considered in this work, is the '*sporgendo*' in which the lips are rounded (forming an 'o').

The human whistle is a close flue pipe instrument, high-pitched, that can be classified in term of its range as a 'sopranino' instrument. A typical compass range from a $C5$ to a $C8$, although exact compass and range varies with each individual[1].

The purpose of the present work is to develop an automatic score extraction system, using monophonic melodic whistles as input. Four different pitch detection techniques were tested and evaluated. The features used to identify each note are the pitch (*i.e.*, the fundamental frequency F_0), the amplitude, the onset, and the duration.

Pitch is the perceived quality of a sound that is chiefly a function of its fundamental frequency [7]. Whereas pitch is a perceptual attribute evoked in the auditory system, the fundamental frequency (F_0) is the corresponding physical attribute defined for periodic or nearly periodic sounds only, and corresponds to the inverse of the period. Humans are said to be interval-sensitive (the difference between two pitches is called an interval), perceiving two different melodies that have the same pattern of intervals (melodic contours) to be essentially equivalent[2], despite their absolute pitches.

To achieve the stated objective, the first two features (pitch and amplitude) were considered. The strategy consists in tracking the pitch of the signal and with information of loudness (rhythm), correct possible errors of the tracker (in particular, note onset and duration).

[1] `http://www.synthonia.com/artwhistling`

[2] One notable exception are the ones who are said to have absolute pitch, meaning that they are able to identify notes by hearing in absolute terms.

For the pitch detection task, four algorithms were tested. In general, these algorithms can be divided into two groups: those that look for frequency partial at harmonic *spectral locations* (in time domain), and those that observe *spectral intervals* between partials [3] (in frequency domain). The temporal Autocorrelation Function (tACF) and the Average Magnitude Difference Function (AMDF) that belongs to the first group, while the frequency Autocorrelation Function (fACF) and Harmonic Spectrum Product (HSP) to the second.

Notes can be specified in terms of octaves, semitones, or others units, however, and because the pitch is a continuous variable, assigning a musical note to a given frequency involves quantization. The Musical Instruments Digital Interface (MIDI), a standard for controlling and communicating with electronic musical instruments, employs the standard representation of Western music, assigning an integer to each note of the scale, in semitone intervals.

The identification of pitch has been object of research for several decades, and is practically a solved problem now in the case of single instruments [6]. However, transcribing sounds produced by humans, such as singing, humming and whistling, is a much more difficult task. Related work, transcribing singing monophonic music [4] and hummed tunes [1], have shown good results.

The structure of this paper is the following: sec.2 describes the main blocks of music transcription algorithm, followed by the description of the pitch tracker methods in Sec.3. Finally, a comparison of the results is presented in Sec.4, followed by some concluding remarks in Sec.5.

2 TRANSCRIPTION SYSTEM OVERVIEW

The main algorithm has the task of translating sound to a musical score format. In this format all notes have a starting time, a duration and a pitch, listed sequentially in time.

Our transcription system has three main steps: preprocessing, pitch detection, and notes segmentation (see Fig 1). These steps are detailed next.

The first step of the transcription system consists in segmenting the sound signal into frames (time windows) of constant length. The signal envelope is then calculated for each frame in order to skip the pitch detection when the energy falls below an audibility threshold. When a silence moment is detected, a null pitch value is assigned to that frame.

Pitch detection assigns a fundamental frequency, F_0 to each signal frame. It is based on the computation of a similarity measure between a frame of the signal and delayed versions of that frame, and then finding the pitch at the maximal peak of the similarity. In the next section we detail four alternative methods for pitch detection, including one detector based not on time but on products of decimated spectrograms.

Given an array of pitch values, one for each signal frame, our system quantizes the pitch values to a MIDI-like quantized-scale, and applies non-linear filtering to remove pitch outliers. The non-linear filtering reduces the number of pitch-outliers within silence according to a minimum note duration parameter ($100ms$). Pitches lasting less than the minimum duration are discarded. Pitch-outliers, spanning less time than the minimum duration and detected within groups of samples with constant pitch values, are also corrected: the middle (outlying) group of pitches is assigned the value of their neighbors. This allows correcting some variations inside a tone. In order to separate the notes accurately, our system comprises also an outliers removal procedure for pitches between tones.

The **notes segmentation** process groups sequences of equal pitch values into notes. The constant pitch value characterizing each group defines the note height while the number of grouped pitch values defines the duration. Hence, the termination of a note is determined by the beginning of a new note or by the detection of silence. An energy based onset detector is not directly applied, but is used to fix some notes fragmentation at the non-linear filtering block. Finally, the pitch value of each note is converted to a MIDI-key, $K(F_0)$:

$$K(F_0) = round\left(12 \times \log_2\left(\frac{F_0}{440}\right)\right) + 69 \qquad (1)$$

For example, the A4 pitch ($F_0 = 440Hz$) corresponds to the MIDI-key number $K(F_0) = 69$. A total of 10 complete octaves (128 notes) are possible to represent, ranging from $8.176Hz$ to $13344Hz$.

3 PITCH DETECTION METHODS

3.1 Temporal Autocorrelation

Time-domain autocorrelation function ($tACF$) based algorithms are among the most popular F_0 estimators [3]. The technique consist in picking peaks in the autocorrelation function:

$$r_{xx}(n) = \frac{1}{N} \sum\nolimits_{K=0}^{N-n-1} x(k)x(k+n) \qquad (2)$$

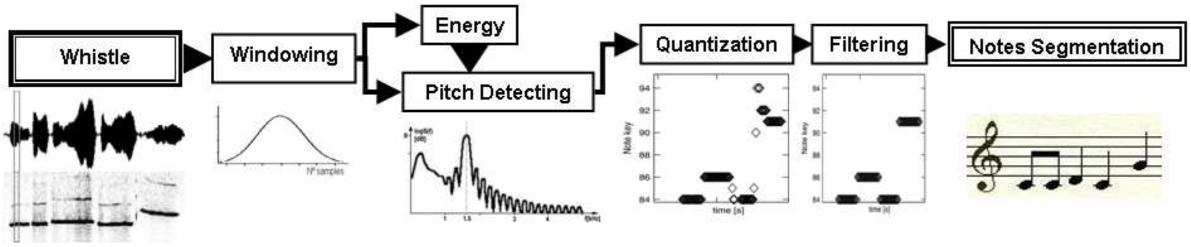

Figure 1: Transcription system.

Because a periodic signal will correlate strongly with itself when delayed by the fundamental period, the time offset (n) corresponding to the highest peak in the autocorrelation will give the period $(\frac{1}{F_0})$ of the waveform. In practice, a similar and more efficient ($NlogN$ instead of N^2) function is used via the fast Fourier transform (FFT). This expression, based on the *Wiener-Khinchin* theorem [2], is given by:

$$r(\tau) = IDFT(|DFT(x(n))|^2) \tag{3}$$

Another advantage of this method is its efficiency in identifying hidden periodicities, e.g. in situations with a weak fundamental. Its superior robustness to noise is another quality. Presenting a desired logarithmic resolution[3] even with a lower FFT order and window size, excellent results are possible. However, this effectiveness at mid to low frequencies could introduce errors at high fundamental frequencies, in which the range of possible fundamental frequencies is limited. The calculation of F_0 is performed directly from a shift of samples, and then a lower sampling rate implies a lower resolution in pitch. Another shortcoming of this method, besides this sensitiveness to the sampling rate, is its tendency to halving the correct F_0 in harmonic sounds. This happens because the periodicity of that signals provokes a periodicity in the *ACF*, with peaks at integer multiples of the period.

3.2 Average Magnitude Difference Function

Average Magnitude Difference Function (*AMDF*) looks for the difference of a signal with a time lag of itself, rather than the product.

$$\psi(\tau) = \frac{1}{N}\sum_{n=0}^{N-1}|x(n) - x(n+\tau)| \tag{4}$$

In opposition to *tACF*, there will be valleys at maximum similarity instead of peaks. Excluding the first null at time zero, the smallest minimum will correspond to the fundamental period (T_0). As in *tACF*, several candidates to T_0 are discarded, corresponding to multiples of T_0.

3.3 Spectrum Autocorrelation

The frequency-domain autocorrelation function (*fACF*), as the *time-domain ACF*, has been used with success over the years in some F_0 estimators. But in opposition with the *tACF*, the *spectral ACF* is a spectral-interval type F_0 estimator, observing frequency locations between partials. The basic principle is based on the observation that harmonic sounds possess a periodic magnitude spectrum. Thus, any two spectral components with a frequency interval m, multiplied by the sampling rate and the inverse of the FFT order (mF_s/K) is a F_0 candidate, as shown in Eq.5, where t represents the time delay:

$$\tilde{r}(m) = \frac{2}{K}\sum_{k=0}^{K/2-m-1}|X(k)||X(k+m)| \tag{5}$$

The maximum value of the autocorrelation correspond to the period of the signal waveform, discarding the obvious maximum value at $m = 0$ (that corresponds to the energy of the signal). Another local maximums appear at integer multiples of F_0, because the periodicity of the magnitude spectrum at multiples values of F_0 rate, and so, some erroneous doubled values of F_0 could appear. Another handicap results from its constant frequency resolution. This problem is more severe in low frequencies, where resolution could be not enough. That contradiction with the human perception of music, which is logarithmic, will damage the final detection. Thus, bigger FFT orders[4] are needed to increase the range of possible frequencies at low frequencies, which lead to a unnecessary wide range of possible frequencies at high frequencies, and a rise in computation time.

[3] Notice that in occidental representation of music the two consecutive semitones are separated by a $2^{\frac{1}{12}}$ ratio, copying the logarithmic sensibility of the human auditory system.

[4] Notice that increasing the FFT order the temporal resolution is degraded.

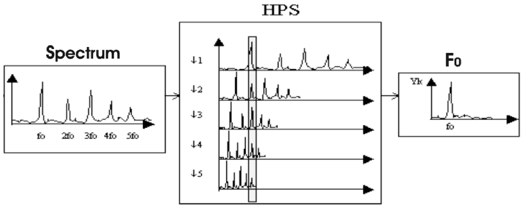

Figure 2: Overview of the HPS algorithm

3.4 Harmonic Product Spectrum

The Harmonic Product Spectrum *HPS* pitch-detection algorithm [5], measures the maximum coincidence for harmonics, for each spectral frame $X(\omega)$,

$$Y(\omega)=\prod_{d=1}^{D}|X(\omega r)| \qquad \hat{Y}=\max_{\omega_i}\{Y(\omega_i)\} \tag{6}$$

where D is the number of harmonics to be included in calculations, and ω_i are the range of candidates to F_0. The value of ω_i corresponding to the maximum value of the resulting periodic correlation array, $Y(\omega)$, will support the F_0. The idea follows from the fact that a musical input signal presents a spectrum consisting of a series of peaks, corresponding to fundamental frequency with harmonic components at integer multiples of the fundamental frequency. Hence when the spectrum is downsampled (compressed a number of times), and compared with the original spectrum, the strongest harmonic peaks line up. The first peak in the original spectrum coincides with the second peak in the spectrum compressed by a factor of two, which coincides with the third peak in the spectrum compressed by a factor of three. Hence, when the various spectrums are multiplied together, the result will form clear peak at the fundamental frequency. Figure 1 demonstrates the HPS algorithm graphically for a windowed signal waveform, where an FFT is executed to the window *(left)*, and several downsampled versions are made *(center)*. The product of the downsampled signals, with a most likely pitch for the analysis window very clear, is shown on the far right of the figure.

The method presents some nice features: it is computationally inexpensive, it is reasonable resistant to additive and multiplicative noise, and it is adjustable to different kind of inputs. However, its resolution is only as good as the length of the FFT used to calculate the spectrum. If a short and fast FFT is performed, a limitation in the number of discrete frequencies occurs. So, and as the *fACF*, in order to gain a higher resolution in the output, a larger FFT order is necessary[5].

4 RESULTS

The data sets used to test the transcription method comprised synthetical and man-made whistle sounds. The synthetical sound allows evaluating the influence of the various parameters on the accuracy of the transcription, without having to consider the usually large errors introduced by the performance of a human whistling.

Figure 3(a), shows the *Happy-Birthday* score used to create a *MIDI* file and subsequent synthetic whistle sound, using a common PC software synthesizer. The figure shows also the translations by the proposed system using each of the four pitch detection methods detailed in Sec.3. As expected, the transcription of the synthetical whistle did not produce significant errors: all the pitches correspond exactly to the played notes; only small time misalignments occur.

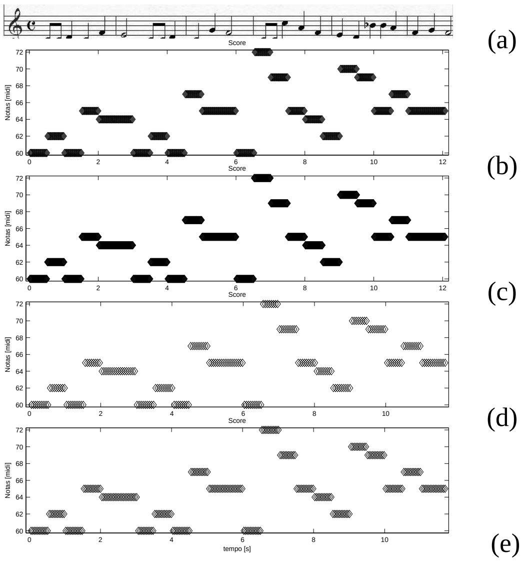

Figure 3: Artificial whistling generated from (a) and its transcription based on temporal *ACF* (b), *AMDF* (c), *spectral ACF* (d), and *HPS* (e).

The accuracy of the translation methodology was evaluated considering various transcription-parameters: minimum and maximum detectable frequencies, minimum note duration, window type, and window length.

The minimum frequency threshold is an important factor in the performance of temporal techniques: set-

[5] This requires more time and decreases the temporal resolution.

ting it too high implies losing low pitches, while if set too low implies processing very large samples. We chose 440Hz as a trade-off between the risk of not detecting low frequency notes and fast computation. The minimum note duration and maximum detectable frequencies, were set to 100ms and 4410Hz respectively. These allows for the detection of notes larger than 100ms (10 notes *per* second, or one semiquaver in 150 B.P.M. ≈ *Vivace*), with pitches between an A4 and a C8, while covering the range of a typical whistling (see Sec.1).

The window type in our experiments did not influence significantly the results. Hence, in this paper we document just the experiments done with the *Hanning* window. The windows size affects significantly however the results. Table 1 shows the ranges of window sizes for which there are no transcription errors. Using large windows prevents detecting fast notes, while using small windows implies processing a larger number of frames (better time resolution), but with a reduced frequency resolution.

In our experiments the *tACF* and the *HPS*, with 1024 and 2048 window sizes respectively, yielded the smallest alignment errors between the transcribed sound and the *MIDI* file format (ground truth) generated from the original score. The *tACF* was the fastest method, even in the case of the largest window size (note that this method actually works faster as the window size grows). In a 12 seconds signal, the *tACF* takes about 1 second of processing time, as compared to the 4 seconds taken by the *AMDF*.

Table 1: Tolerance to the choice of the frame window size, for all the pitch tracking techniques, for a 44.1kH sample rate. *Dec* indicates the temporal decimation order (when used).

Method	Dec	Tolerance	Dec	Tolerance
tACF	No	[512-8192]	5x	[128-1024]
AMDF	**No**	**[256-8196]**	5x	[64-1024]
HPS	No	[2048-8192]	5x	[512-2048]
fACF	No	[None]	5x	[512-1024]

The tests with man-made whistle sounds allow evaluating the effectiveness of the proposed solution. As noted in the introduction, the inaccuracy of whistling, as for instance with non-steady pitches, or with missing, inserted or translated notes, imply transcription errors. The extent to which these difficulties are resolved shows the robustness of the proposed system in the detection and amendment of unintentional pitch variations in the whistling sound.

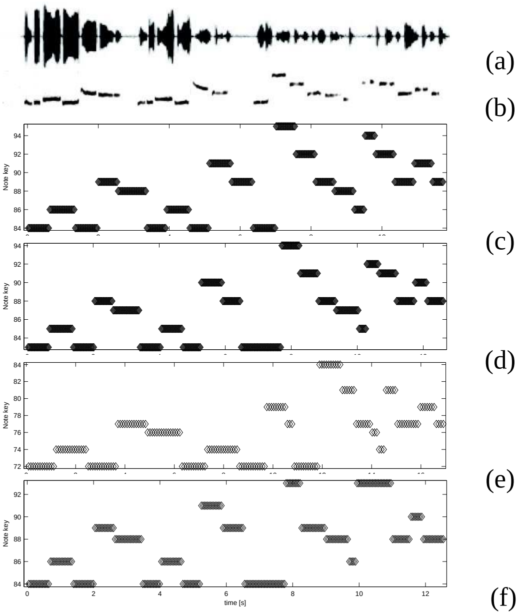

Figure 4: Transcription of a human whistle sound (a) with spectrogram (b). Results obtained with *tACF* (c), *AMDF* (d), *fACF* (e), and *HPS* (f).

Figure 4 shows the amplitude envelope and the spectrogram of a *Happy-Birthday* whistling, together with the resulting transcription for each pitch tracking method described in Sec.3. The three best results for each technique are presented in Tab.2, as well as the insertion, deletion, and substitution errors (as a percentage of the total number of notes-symbols).

In our experiments, the *HPS* method, despite making an almost correct transcription, showed less performance than the other methods. In particular, it was necessary to pre-filter the man-made whistle sound, with a 99th-order FIR bandpass filter tuned for typical whistling sounds, as otherwise the estimated pitch would be oscillating between a base range of values and the same range translated one octave upwards. This was expected as *HPS* is known to require signals with significant harmonic components (partials / overtones), which do not happen in whistle sounds, as noted in Sec.1. This characteristic of the whistling timbre, that reduces the efficiency of spectral-interval type F_0 estimators as the *HPS*, actually improves the performance of spectral-location type F_0 estimators as the *tACF* and the *AMDF*. The existence of weak F_0 multiples (overtones) have the positive effect of reducing octave errors.

Comparing the *tACF* method with the *fACF*, the former is more robust, accurate and faster, even in the case where the *fACF* decimated the input signal. Temporal methods are both accurate, but the use of the FFT in the *tACF* algorithm makes it faster. The best overall transcription performance was obtained using *tACF* with a window of 512 samples. No tracking errors occurred in a calculation time of 15% of the input signal time.

Table 2: Errors in the detected notes. Insertions error rate e_i, deletions error rate e_d, and substitutions error rate e_s. Methods tested with the specified window sizes (three best results in each method). The † means a previous decimation of the signal ($2x$ to a sample rate of $8kH$). Bold shows the best window size within each method.

Method	window	e_i	e_d	e_s	e_T
tACF	128	0%	0%	0%	0%
	256	0%	0%	4%	4%
	512	**0%**	**0%**	**0%**	**0%**
AMDF	256	4%	0%	0%	4%
	512	**0%**	**0%**	**0%**	**0%**
	1024	0%	0%	4%	4%
fACF	1024	4%	4%	0%	8%
	512	4%	0%	4%	8%
	512†	**4%**	**0%**	**0%**	**4%**
HPS	1024	0%	8%	4%	12%
	256†	0%	8%	4%	12%
	128†	0%	8%	4%	12%

5 CONCLUSIONS

In this paper we proposed a transcription system of musical-whistling to a MIDI-like representation. In particular, we have compared four options for pitch detection, a central component of the system. Our experiments showed that the system performs successfully the transcription, and the pitch detection methods proved to be functional in a wide range of the parameters.

As future work envisage applications of the transcription system as an input device for a music identification and database retrieval system.

6 ACKNOWLEDGEMENTS

Research partly funded by the Portuguese FCT Programa Operacional Sociedade de Informação (POSI) in the frame of QCA III, the Portuguese FCT/ISR/IST plurianual funding through the POS Conhecimento Program that includes FEDER funds and the European Project EU-FP6-NEST-5010 - CONTACT.

REFERENCES

[1] T. Brøndsted, S. Augustensen, B. Fisker, C. Hansen, J. Klitgaard, L. Nielsen, and T. Rasmussen. A system for recognition of hummed tunes. In *Proceedings of the COST G-6 Conference on Digital Audio Effects(DAFX-01)*, pages 6–8, Limerick, Ireland, Dec 2001.

[2] L. Cohen. The generalization of the wiener-khinchin theorem. *IEEE T-ASSP*, 3:1577–1580, 1998.

[3] A. Klapuri. Automatic music transcription as we know today. *Journal of New Music Research*, 33(3):269–282, 2004.

[4] W. Kuhn. A real-time pitch recognition algorithm for music applications. *Computer Music Journal*, 14(3):60–71, 1990.

[5] M. Noll. Pitch determination of human speech by the harmonic product spectrum, the harmonic sum spectrum, and a maximum likelihood estimate. *In Proceedings of the Symposium on Computer Processing Communications*, pages 779–797, 1969.

[6] M. Plumbley, S. Abdallah, J. Bello, M. Davies, G. Monti, and M. Sandler. Automatic music transcription and audio source separation. *Cybernetics and Systems: An International Journal*, 33(6):603–627, 2002.

[7] D. Randel. *The new Harvard dictionary of music*. Belknap Press, Cambridge, MA, 1st edition, 1986.