Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-00), Verona, Italy, December 7-9, 2000

MONOPHONIC TRANSCRIPTION WITH AUTOCORRELATION

Giuliano Monti, Mark Sandler

Department of Electronic Engineering, King's College London,
Strand, London WC2R 2LS, UK
giuliano.monti@kcl.ac.uk

ABSTRACT

This paper describes an algorithm, which performs monophonic music transcription. A pitch tracker calculates the fundamental frequency of the signal from the autocorrelation function. A *continuity-restoration* block takes the extracted pitch and determines the score corresponding to the original performance. The signal envelope analysis completes the transcription system, calculating attack-sustain-decay-release times, which improves the synthesis process. Attention is also paid to the extraction of timbre and wavetable synthesis.

1. INTRODUCTION

Musical transcription of audio data is the process of taking a sequence of digital data corresponding to the sound waveform and extracting from it the symbolic information related to the high-level musical structures that might be seen on a score[1].
In a very simplistic way, all the sounds employed in the music to be analyzed may be described by four physical parameters, which have corresponding physiological correlates[2]:

1. Repetition rate or fundamental frequency of the sound wave, correlating with pitch.
2. Sound wave amplitude, correlating with loudness.
3. Sound wave shape, correlating with timbre.
4. Sound source location with respect to the listener, correlating with the listener's spatial perception.

The latter is not considered determinant for music transcription, and will be discarded for this investigation. The other three generate the difference between the parts that can be defined in a musical track [3]: the orchestra and the score. The orchestra is the sound of the instrument itself, the specific characteristics of the instruments (timbre, envelope), which make it sound unique; the score consists of the general control parameters (pitch, onsets, etc), which define the music played by the instrument. In an academic music representation, just the latter can be described, *i.e. which notes to play and when to play them*. The purpose of the present work is to automatically extract score "features" from monophonic music tracks, using an autocorrelation pitch tracker.

1.1. Monophonic transcription with autocorrelation

If the fundamental frequency of a harmonic signal is calculated, and the resulting track is visualized, it can be noticed that, for most of the duration of the notes, the pitch maintains approximately constant. This relation, so clear to the eyes, requires some comments. In order to implement some grouping criteria and rules for sounds, emphasis should be given to the similarity in human perception between image and sound[4]. Important clues can be obtained by observing carefully the plot of the pitch track. The current system doesn't use a conventional (energy based) onset detector, instead, it implements a pitch based onset detector, which is more robust when confronted to slight note changes (glissando, legato).
Monophonic music means that the performer is playing one note at a time. More than one instrument can be played, but their sounds must not overlap. In this case the sound is characterized by only one pitch.

1.2. Autocorrelation pitch tracking

In order to estimate the pitch in the musical signal, autocorrelation pitch tracking has been chosen, showing good detection and smooth values during the steady part of a note. The steady part of a note is just after the attack, where all the harmonics become stable and clearly marked in the spectrum.
An estimate of the Autocorrelation of an N-length sequence x (k) is given by:

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

Where n is the lag, or the period length, and x (n) is a time domain signal. This function is particularly useful in identifying hidden periodicities in a signal, for instance, when the fundamental is weak. Peaks in the autocorrelation function correspond to the lags where periodicity is stronger. The zero lag autocorrelation $r_{xx}(0)$ is the energy of the signal. The autocorrelation function shows peaks for any periodicity present in the signal, therefore it is necessary to discard the maxima, which correspond to the multiple periodicities. If the signal has high autocorrelation for a lag value, say K, it will have maximum for n*K as well, where n is a positive integer. Consequently, the first peak in the autocorrelation function, after the zero lag value, is considered as the inverse of the fundamental frequency, while the other peak values are discarded. The implementation takes advantage of some algorithms implemented by Malcolm Slaney in the 'Auditory toolbox'[5], a Matlab toolbox, freely available, implementing auditory models and functions to calculate the correlation coefficients.
Why autocorrelation?
Autocorrelation is simple, fast and reliable. The equation (1) represents a very simple relation between the time waveform and

the periodicities of the signal expressed by the autocorrelation coefficients.
The calculation of the autocorrelation is computed through the FFT, which has a computational complexity of N·log(N), where N is the length of the windowed signal. The calculation process is therefore very fast. The simulations performed confirm the reliability of this method. In 1990, Brown published results of a study where the pitch of instrumental sounds was determined using autocorrelation[6]; she suggested this method to be a good frequency tracker for musical sounds.

1.3. Transcription

The scheme of the monophonic transcription system implemented here is illustrated in figure 1.
The outputs of the blocks are explained in the next figures. The pitch tracker is based on the autocorrelation method described in section 2.1. Its output is the instantaneous pitch of the signal.
Beside the pitch tracker, a block calculates the envelope of the signal. This information is used by the pitch tracker, in order to skip the calculation of the pitch when the energy of the signal falls below the audibility threshold. This procedure avoids ineffective elaborations.

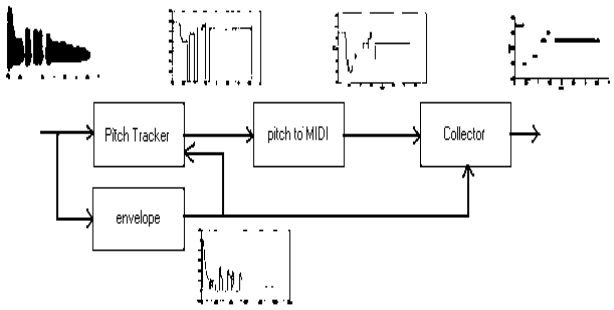

Figure 1. Scheme of the transcription system.

Figure 2 portrays the output of the pitch tracker. The pitch is set to 0 in the silence parts.

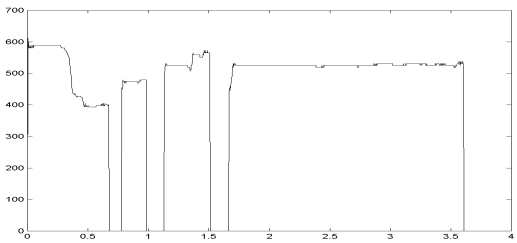

Figure 2. Pitch from autocorrelation

The conversion of the pitch (Hertz), to MIDI number is the result of a rounding up to the nearest musical frequency. Unless the pitch, the MIDI numbers keep the same value during the steady part of a note. The relation is given as:

$$MIDIn = 49 + \left(12 \times \frac{\log(f / 440)}{\log(2)} \right) \qquad (2)$$

Where the $(\ )$ operator calculates the nearest integer value.

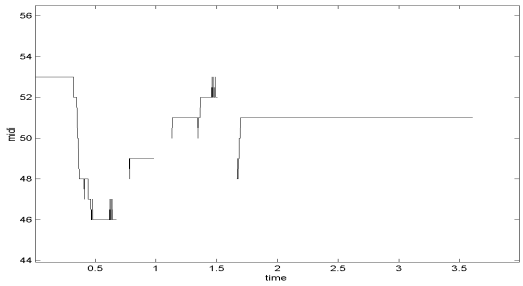

Figure 3. Pitch2MIDI conversion

If we consider a violin vibrato, in the rounding up process all the information regarding the frequency modulation is lost. However, the absence of frequency modulation in the synthesized sound has little effect on the perceptual response to violin vibrato, while the absence of amplitude modulation causes marked changes in sound quality [7]. Moreover, it is known one algorithm can extract the vibrato information from the signal envelope, after the sound has been segmented note by note [8].

1.3.1. The Collector

The collector extracts the score, considering the pre-elaborated track and the signal amplitude, sorting: *onsets, pitch and offsets*.
If the short-time autocorrelation is calculated on a monophonic music signal and the results are plotted, the pitch information is almost constant during the steady parts of the notes. The attack part of a note is usually noisy, however periodicity in the signal is still present. The transient part can last a few tenths of msec and varies depending on the instrument family [9]. In the attack part of the signal, the pitch tracker cannot provide useful information for the transcription system.
The collector recognizes when the pitch maintains the same value, and proposes a note onset in the first value of the constant sequence. The onset is confirmed when the pitch lasts for the minimum note duration accepted. When a note is recognized, the system is able to write in the score file: the onset and the pitch of the note.
The *minimum note duration* is the main parameter in the collector. By modifying its value, the system adapts to the speed of the music, improving the performance of the transcription. If the minimum note duration is set for instance to 40 msec, all the pitch sequences, with constant values, lasting less than 40 msec are discarded. Hence, errors concerning spurious notes are eliminated.
The minimum duration parameter controls also the memory of the system: when a note is detected, the pitch may vary inside the 40 msec window before taking the same value again, and still be considered part of the same note. This is very similar to the consideration taken in sound restoration [10][13]: the human brain takes information from the cochlea, and interprets them with the knowledge of the previous samples; this behavior is called streaming or integration process in psychoacoustics [4].
The termination of a note is determined by the start of a new note or by the recognition of silence. After an onset, the offset detector checks if the signal energy falls below the audibility threshold.

The duration of the note in the score is calculated by the difference between its onset and the next onset/offset.
During the decaying part of a note, the pitch can slightly change. The collector allows the pitch to have different values, until a new note is predicted. However, if the conditions for a new note aren't met, the system keeps the last note.

1.4. Synthesis

Csound [11] synthesizes the transcribed music, providing the score and the orchestra file. From the collector comes the score in terms of pitch, onset and duration of the notes, however, in order to recreate the original melody, the envelope and the timbre are essential.

1.4.1. The envelope

The Csound function "linseg" traces linear segments between defined points as shown in figure 4. The note's envelope is divided in 4 parts: attack, decay, sustain, release. The information about the onset and the duration of a note is already known from the collector. The segmentation of the signal in notes makes the calculation of the envelope's parameters a trivial task. Although different shapes should be considered for different instrument. In figure 5, for instance, the thresholds for the envelope segmentation are 50% of the maximum amplitude for the beginning of the sustain part and 20% for the beginning of the release. These values fit particularly well to the piano, which typical note's shape is shown below, however they should be optimized for different instruments.

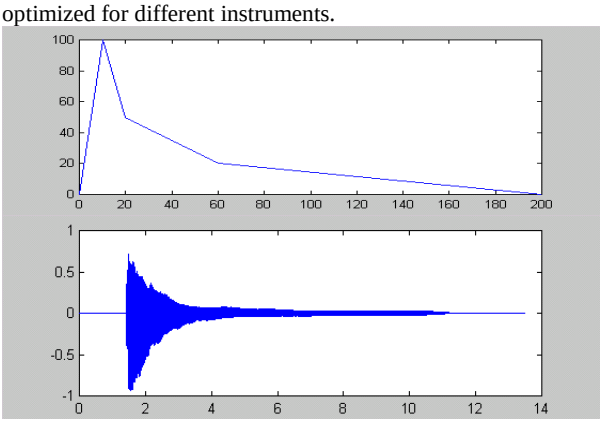

Figure 5. Piano's Note : Attack, Decay, Sustain, Release.

The amplitude information can be very important to detect missed notes. Since the onset time is determined from the pitch information, a sequence of two same notes could lead to only one lasting for the duration of the two. Therefore, if the amplitude rises twice in the same note, we can assume that another note has been played.

1.4.2. The Timbre

The timbre of the note is characteristic of the instrument. After the attack time the note is supposed to be in its steady part. The wavetable synthesis is considered taking advantage of the typical structure of the signal. Figure 6 shows this structure, which is the basic waveform repeated in time at the pitch frequency. The wavetable synthesis exploits this property repeating the basic waveform in time and multiplying its amplitude by the envelope.
For each note the basic waveform is extracted from the original signal and stored in a table. In this way more than one instrument can play in the same song since the synthesis follows the timbre changing among octaves.

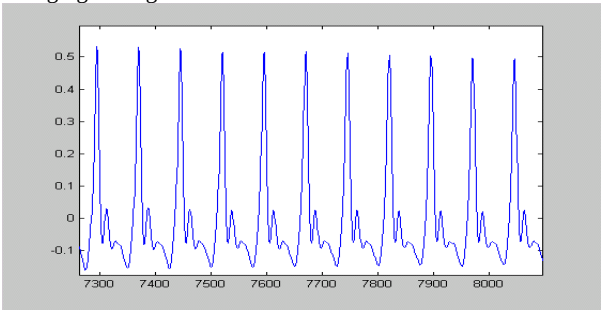

Figure 6. Periodic waveform of a brass instrument.

Considering figure 6, the table is completed with the original samples between 2 consecutive maxima in the sustain part of the note. However, the tests reveal poor synthesis due to the attack part of the signal, which is not considered in this paper. One development of the system will focus on the attack synthesis. A similar table will be extracted from the beginning of the note and cross-faded with the table of the sustain part.

1.5. Results

The number of lags considered in the autocorrelation determines the pitch range of the transcription system. The following table gives an idea of the relation between the autocorrelation coefficients considered and the pitch range covered (notes).

No. coeff.	From	To
256	32	76
512	26 (~120Hz)	80(~2700Hz)

Table 1. Relation between the number of autocorrelation coefficients and pitch range in the transcription system.

The configuration with 512 coefficients was chosen in the transcription. The wider pitch range covered was preferred to the faster computational time with 256 coefficients.
To verify that the pitch has been correctly tracked and the melody of the original file has not been modified, the system writes a Csound [11] score file. By providing an orchestra file, the score can be converted into wav format. The orchestra file contains the description of the instrument. Hence, from the same score, the same melody can be re-synthesized with different instruments specifying different orchestra files.
The test samples were obtained from a CD collection of brass instruments riffs. Comparative listening between the synthesized score and the original riffs reveal the transparency of the transcription. By transparency, we mean that the tempo and the pitch are correctly extracted.
As shown in Figure 7, the matlab script also plots the original signal (top) and its score(bottom) in a "piano roll" form.
It was interesting to compare this system with a commercial program downloaded from Internet, performing WAV2MIDI

[12]. Even if no specification about the transcription system was given, the two systems seem to work in a very similar way from the transcription point of view, but the second one doesn't consider timbre analysis. The minimum note duration can be modified in both the system. Finally, the simulations results are both fairly successful.

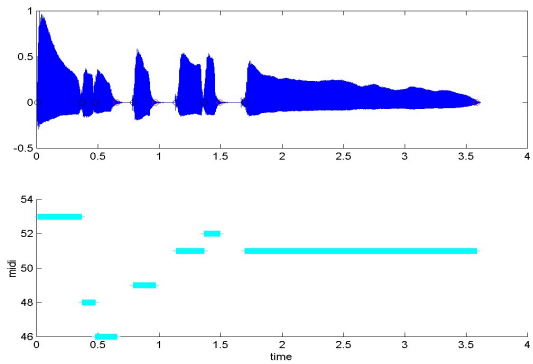

Figure 7. Original music (top) and score file (below).

2. CONCLUSIONS

This paper has reviewed a traditional method of performing pitch tracking, widely used in speech processing and it has demonstrated to be also good for musical instruments. Furthermore, the implementation of a successful monophonic transcription system has been illustrated.

The transcription system described doesn't have an onset detector based on the signal waveform. The onset is only recognized at the beginning of a constant pitch value sequence. As a result, the onset time can be delayed of a few tens of msec. The great advantage of this approach is that in glissando or legato passages the onset is easily detected: the new note is recognized by analyzing the pitch, instead of looking at the energy of the signal, which is usually ambiguous.

The implementation of the pitch tracker with 512 fft points makes real time implementation possible.

Amplitude and timbre are crucial parameters of the synthesis process. Once extracted, the monophonic signal could be coded with a compression factor of 100-1000. Development of the system under the MPEG-4 Structured Audio standard [3] will be also considered in the future.

3. REFERENCES

[1] Eric Scheirer. "Extracting expressive performance information from recorded music". Master's thesis, MIT, 1995.

[2] R.F Moore. "Elements of Computer Music". Prentice Hall, Englewood Cliffs, New Jersey, 1990.

[3] Eric D. Scheirer, "*The MPEG-4 Structured Audio Standard*", IEEE ICASSP Proc., 1998.

[4] Bregman A., "*Auditory Scene Analysis*", MIT Press, 1990.

[5] Slaney M. "*Auditory Toolbox for Matlab*" available at URL http://www.interval.com/papers/1998-010/

[6] Brown , "*Musical frequency tracking using the methods of Conventional and Narrowed Autocorrelation*" J.A.S.A. , 1991.

[7] Wakefield G.H., "*Time-frequency characteristic of violin vibrato: modal distribution analysis and synthesis*", JASA, Jan-Feb 2000.

[8] Bendor D, Sandler M., "*Time domain extraction of Vibrato from monophonic instruments*", to be published in Music IR 2000 Conference, October 2000.

[9] Martin K., "*Sound-Source recognition*", PhD thesis, MIT, ftp://sound.media.mit.edu/pub/Papers/kdm-phdthesis.pdf, 1999.

[10] Ellis D, "*Hierarchic models of sound for separation and restoration*", Proc. IEEE Mohonk Workshop, 1993.

[11] Csound web page URL: http://music.dartmouth.edu/ /~dupras /wCsound/ csoundpage.html.

[12] WAV2MIDI, URL: http://www.audiowork.com.

[13] Daniel Ellis. "Prediction-driven computational auditory scene analysis". PhD Thesis, MIT, June 1996.