Fast labelling of notes in music signals

Paul M. Brossier, Juan P. Bello, Mark D. Plumbley
Queen Mary College, University of London
Centre for Digital Music

ABSTRACT

We present a new system for the estimation of note attributes from a live monophonic music source, within a short time delay and without any previous knowledge of the signal. The labelling is based on the temporal segmentation and the successive estimation of the fundamental frequency of the current note object. The setup, implemented around a small C library, is directed at the robust note segmentation of a variety of audio signals. A system for evaluation of performances is also presented. The further extension to polyphonic signals is considered, as well as design concerns such as portability and integration in other software environments.

1. INTRODUCTION

1.1. Motivation

The real-time segmentation and attribute estimation of musical objects are still novel fields of research with possible applications in audio coding, music-oriented tasks such as score following and live content-based processing of music data. In [1] a framework was presented for the real-time transmission of audio contents as objects, within the context of spectral modelling synthesis. In [2] we introduced a system for the object-based construction of a spectral-model of a musical instrument. This work was extended in [3], when we presented an algorithm for the real-time segmentation of note objects in music signals.

In the present paper we will concentrate on the real-time estimation of attributes of the segmented note objects, specifically their fundamental frequency f^0. This is an important step towards the understanding of higher-level structures in music – e.g. melody, harmony, etc.

We describe a new system for the low-latency characterisation of a temporal sonic object, based on the information provided by our note segmentation algorithm [3]. The aim is to obtain a robust note labelling on a large variety of musical signals and in various acoustic environments.

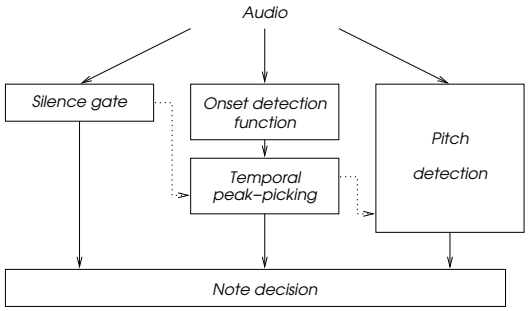

Figure 1. Overview of the different modules of the system

A method for evaluation of this labelling process is proposed and tested. Special attention will be paid to the delay introduced by the f^0 computation, and how the constraints imposed by low-latency environments affect the reliability of our estimation.

1.2. f^0 Estimation

The fundamental frequency f^0 is the lowest frequency of a harmonic sound: its partials appear at frequencies which are multiples of f^0. For most sounds the fundamental frequency is strongly related to the psychoacoustical concept of pitch – although in some cases they can be found to differ. This explains why, often in the literature, the terms pitch detection and f^0 estimation are used indistinctively.

There are a number of methods developed for f^0 estimation in monophonic audio mostly for speech signals. These include approaches such as time-domain and spectral autocorrelation [4], the two-way mismatch algorithm [5] and the use of perceptual models [6]. However, while some of these algorithms have already been successfully tested for real-time f^0 estimation on a frame-by-frame basis, little has been done to estimate the pitch of segmented note objects online.

The complexity of this task increases when dealing with polyphonic sounds. In this paper we will use a pitch algorithm [7] to focus on the analysis of note objects in monophonic music, using both onset and pitch detections together. Our choice of pitch detection algorithm is driven towards the extension to polyphonic signals. While other methods for selecting the note pitch must be further tested, we focus here on the behaviour of the system rather than on the f^0 estimation itself, so as to tackle the different issues of the note decision process.

1.3. Paper organisation

This paper is organised as follows: in Section 2 we explain the process implemented for onset detection and f^0 estimation, and describe a first approach for maximising performance based on object segmentation; in Section 3 we present an evaluation method for this segmentation process;the implementation in the form of a software library is detailed. Section 4 presents quantitative results of the integration of the different parts of the system. Finally, we present our conclusions in Section 5.

2. SYSTEM OVERVIEW

Figure 1 shows an overview of the main system components. The different elements composing the system will be described in the following order: the silence gate, the onset detection functions module, its associated peak picking module, the fundamental frequency estimation, and the final note decision module. In this last step pitch values and onset times are filtered into a list of note candidates.

We use two phase vocoders in parallel for both onset and pitch detections. For a signal x at time n, $X[n]$ defines its Short Time Fourier Transform (STFT), calculated using the phase vocoder. $X_k[n]$, the value of the k^{th} bin of $X[n]$, can be expressed in its polar form as $|X_k[n]|e^{j\phi_k[n]}$ where $|X_k[n]|$ is the spectral magnitude of this bin, and $\phi_k[n]$ its phase.

2.1. Silence gate

A silence gate first ensures the suppression of spurious candidates in background noise. When the signal drops under a certain level, the onsets are discarded. Because the noise level can dramatically change between different auditory scenes, this level can be adjusted to minimise the onsets detected during pauses and silences.

2.2. Onset detection

Our onset detection implementation has been previously described in [3], along with key references to the relevant literature. The process consists of the construction of a detection function derived from one or a few consecutive spectral frames of a phase vocoder. The detection function increases at the beginning of the note attacks. Peak-picking is required to select only the relevant onsets.

Different detection functions are available in our implementation, and we have measured how well they perform on a variety of signals: the average rates of correct detections and false positives have been evaluated at different values of the peak picking thresholding parameter. A simple and very efficient example is the *High Frequency Content* (HFC) [8], which can be derived from one spectral frame $X_k[n]$ as:

$$D_H[n] = \sum_{k=0}^{N} k|X_k[n]| \tag{1}$$

The HFC precisely identifies percussive onsets, but is less responsive to non- percussive components. In the *complex-domain* approach [9], to cope with harmonic changes of low transient timbres, a target STFT value is generated as follows:

$$\begin{cases} \hat{X}_k[n] &= |X_k[n]|e^{j\hat{\phi}_k[n]} \\ \hat{\phi}_k[n] &= \text{princarg}\left(2\phi_k[n-1] - \phi_k[n-2]\right) \end{cases} \tag{2}$$

where $\hat{\phi}_k[n]$ is the estimated phase deviation. The measure of the Euclidean distance, in the complex-domain, between the target STFT $\hat{X}_k$ and the observed frame X_k allow the construction of a detection function as:

$$D_C[n] = \frac{1}{N}\sum_{k=0}^{N}\left\|\hat{X}_k[n] - X_k[n]\right\|^2 \tag{3}$$

By looking for changes at both energy and phase, the complex-domain approach – and other methods such as the spectral difference approach – quantifies both percussive and tonal onsets.

The detection functions still contain spurious peaks and some pre-processing, such as low pass filtering, is required before peak picking. In order to select the onsets independently of the current context, a dynamic threshold is computed over a small number of $D[n]$ points. A median filter is applied first for smoothing and derivation of the detection function; a proportion of the mean over that same number of points is included in the threshold to reject the smaller peaks:

$$\begin{aligned} \delta_t[n] &= \text{median}(D[n-b]\ldots D[n+a]) \\ &+ \quad \alpha\langle D[n-b]\ldots D[n+a]\rangle \end{aligned} \tag{4}$$

The values a and b define the window of detection points considered, typically one frame in advance and five frames in the past. Increasing the proportion α prevents the selection of the smallest peaks in the detection function and decrease the number of false positives.

While the HFC detects percussive events successfully, the complex-domain approach – and other methods such as the spectral difference approach – reacts better on tonal sounds such as bowed strings but tend to over-detect percussive events. Experimental results have shown that the combined use of two detection functions, such as the multiplication of the complex domain and the HFC functions, increase the overall reliability of the results on the set of real recordings, suggesting the complementarity of the functions.

2.3. Pitch estimation

The following f^0 estimation algorithm is derived from [7] and the improvements described in [10]. Although we focus on its behaviour with monophonic signals, the method has been designed to tackle polyphonic music signals. The algorithm is based on the spectral frames $X_k[n]$ of a phase vocoder similar as that used in the onset detection functions. The input signal is first pre-processed through an

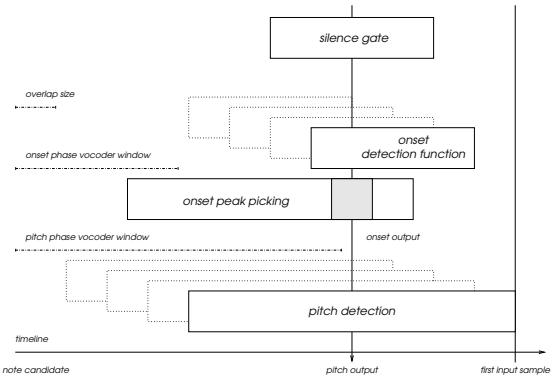

Figure 2. Detail of the synchronisation of the onset detection module and the pitch detection module

A-weighting IIR filter so as to enhance medium frequencies and reduce the high and low parts of the spectrum.

The filtered signal is sent to a phase vocoder using a window of typically 4096 samples for an audio samplerate of 44.1 kHz. The longer windowing implies a longer delay in the system but is required for accurate frequency estimation of the mid frequencies. The overlap rate is the same as the one used for the onset detection function. On each frame, the magnitude spectrum is low pass filtered and normalised.

After pre-processing, peaks are detected in the magnitude spectral frame and the list of peaks is passed to an harmonic comb. We assume that one of the P highest peaks corresponds to one of the partials of the present notes – for monophonic signals, we will limit to the case where $P = 1$. Each of these peaks generates a set of pitch hypotheses defined by the first Z sub-harmonics as:

$$\{f^0_{p,z} = \frac{f_p}{z} | z \in [1 \dots Z] | p \in [1 \dots P]\} \qquad (5)$$

and where f_p is the frequency associated to the bin of the p^{th} peak, computed using a quadratic interpolation method. For each of these $f^0_{p,z}$ hypotheses a harmonic grid is constructed over the spectral bins as:

$$C_{p,z}(k) = \begin{cases} 1 & \text{if } \exists\, m \text{ s. t. } \left| \frac{1}{m} \frac{k}{f^0_{p,z}} - \frac{N}{f_s} \right| < \omega_b \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

where f_s is the sampling frequency, m is an integer between 1 and M, the maximum number of harmonic considered. ω_b, typically a quarter tone, is set to allow for some uncertainty in the harmonic match of the the comb filter.

Different criteria are checked along the evaluation of each candidate combs. The two most important are the number of partials matching to the comb harmonic grid, and the comb energy, estimated as the total energy carried by the set of partials.

2.4. Note Decision

The data incoming from the different modules must be ordered carefully before the decision process. Different approaches can be taken to address this issue. In our system, we rely on the temporal onset detection, assuming it is correct, and look for note pitches over the frames past each onset.

While both pitch and onset vocoders operate at the same overlap rate every 512 samples, long windows are required for pitch estimation, and shorter windows of typically 1024 samples will feed the onset detection functions. Synchronisation of the pitch candidates to the onset time is required. The temporal peak picking module of the detection function takes one overlap period. When using windows 4 times longer for the pitches than for the onsets, the pitch candidates of the frames are typically delayed by about 2 overlap periods. The process is depicted in Figure 2.

In the attack of the note, just after the onset, pitch detection during strong transient noises will tend to be difficult, since the transient covers most of the harmonic components. These spurious pitch candidates need to be carefully discarded. Another source of error is when the amplitude of the different partial are changing within the note. Octave or fifth errors may then occur.

To evaluate a note pitch candidate in a limited number of frames after the onsets, a simple and efficient system has been built by choosing the median over the candidates that appear in the frames after the onset:

$$P_{\text{note}} = \text{median}(P_q, P_{q+1}, ..., P_{q+\delta}) \qquad (7)$$

where δ is the number of frames considered for the pitch decision and will determine the total delay of the system. The first q frames are not considered, so as to take into account the delay between both pitch and onset phase vocoders.

The median filter favors the selection of the most frequently detected pitch candidate, while δ provides a control of the trade-off between the system delay and its accuracy. This simple method proved to be successful at selecting the most predominant pitch candidate over a few frames after the onset. The note labelling is done by attaching the selected pitch candidate to the onset time previously detected.

3. EXPERIMENTAL SETUP

We describe here the evaluation technique used to estimate the performance of our note identification system. We will then describe our software implementation.

3.1. Performance estimation

A test bed for the evaluation of our system has been implemented. Audio waveforms are generated using MIDI files and analysed through our note labelling program. The evaluation consist of the comparison between the original MIDI score and the list of candidate event detections we obtain, both against pitch and start time.

For our evaluation purposes, the scores were chosen amongst various single voiced scores for piano, violin,

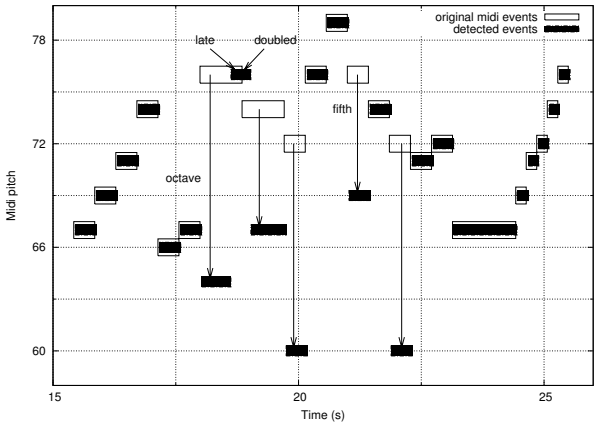

Figure 3. Example of typical note segmentation errors: detected events are shown in filled box, the original score is outlined. Extract from *Partita in A minor for Solo Flute, J. S. Bach, BWV* 1013*,* 1^{st} Movement: Allemande.

clarinet, trumpet, flute and oboe. All scores have been extracted from the Mutopia project [11]: some from separate tracks of Mozart and Bach concertos and symphonies, others from more recent compositions for solo instrument. The MIDI files contain nuances and expressiveness. The current database currently totalises 1946 notes.

The midi files are converted into raw waveforms using the Timidity++ MIDI rendering software [12]. Each of the scores can be rendered using any instrument, while a large amount of settings and effects are available. For instance, we used the Reverberation and Chorus settings, so that the results sound more natural.

NOTE-ON events are extracted as a pair of *MIDI pitch* and *start time*. Our main program is then called, in its offline mode, along with a set of parameters, to process the waveform and store the note candidates in a similar list. A Python script is then used to compare the original list to the list of detected events.

If the detected event corresponds to a real note within a tolerance window of length ϵ_t (ms) and with the correct MIDI pitch rounded to the nearest integer, only then the event is labelled as a correct detection. Incorrect detections can be easily characterised by their frequency error, octave or fifth jumps, and their temporal error, doubled or late detections.

An example of such a score comparison is given in Figure 3. Pitches are plotted versus times in this piano-roll like graph. The original notes are drawn in solid lines, the detected events in filled black squares. The plot illustrates various types of errors.

3.2. Software implementation

This implementation is a development of the software library presented in [3]. The f^0 estimation runs in parallel with the onset functions. The process runs within the Jack Audio Connection Kit (JACK) server for experiments on live signals. In this case, an audio input is created for listening to incoming data, and a MIDI output port is created to output the result of the transcription to another MIDI device.

The library has been kept small and its dependancies limited to a small number of widely used libraries.

We believe that the current setup provides a solid foundation for the development of test beds for various Music Information Retrieval techniques, such as blind instrument recognition methods or testing algorithm robustness against various compression formats.

4. INITIAL RESULTS

In [3], the different detection functions were evaluated on a set of real recordings, representing a wide range of music styles and sound mixture. All detection functions have been peak-picked using a window of size $a = 5$ and $b = 1$ in Eq. 4. The proportions of correct and misplaced onsets have been estimated by comparing the detections against the hand-labelled onsets.

We found the product of the HFC and complex domain onset functions gave best performance. With a fixed threshold of $\alpha = 0.300$, we obtained typical detection rates of 96% correct detections and 6% of false positives. In the following two experiments, we fixed α to a value of 0.300: a slight over-detection is allowed to ensure a high correct detection proportion, on which we rely in the note pitch decision.

We can observe the performance of the note decision process by varying δ the number of frame considered in Eq. 7. The results obtained with different instruments for values of δ between 2 and 25 are plotted in Figure 4.

The performance for the whole set of MIDI test files (plus signs in Figure 4) reaches 90% of correct note labelling at $\delta = 20$, which gives a decision delay of 200 ms with a total rate of 12.5% of false positives.

Some instruments tend to be successfully labelled within as few as $\delta = 3$ frames, as shown with the harpsichord results (open squares). Low δ values affect the performance of the flute, which may be explained by the soft and breathy attack of the flute. This is corrected using a longer value of δ.

Another problem occur when a large value of δ is used on long notes: the performance then tends to decrease. Changes in the relative amplitude of the partials may cause the pitch candidate to switch to another harmonic. This behaviour is observed on the second violin score (asterisks) which has a moderated tempo and numerous long notes.

The general irregular behaviour of the curves is probably due to the combined effect of the transients and the median filter, when only 2 to 9 frames are considered.

The time precision of our system is evaluated using a fixed value of $\delta = 15$ by changing the time tolerance ϵ_t in the list comparison function. The results are displayed in Figure 5. Within a 50 ms tolerance window, the system reaches an optimal score alignment. Overall results

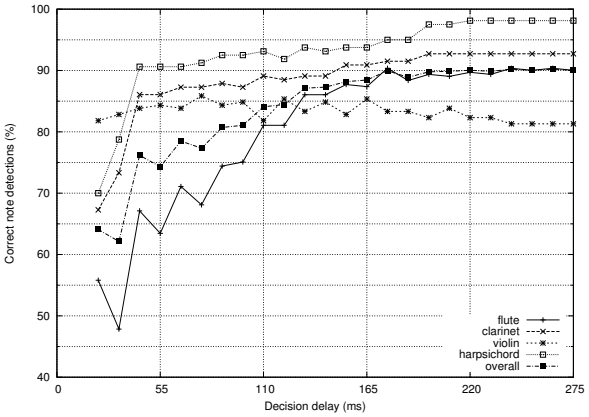

Figure 4. Correct note estimation results for different values of δ in (7), the number of pitch frames the decision is taken on, and for different instruments. α is fixed at 0.300.

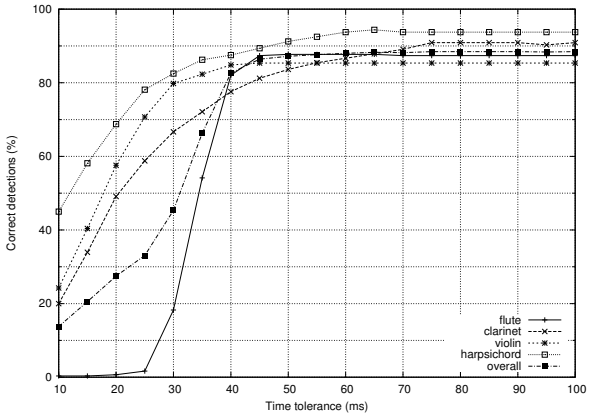

Figure 5. Correct note detections for different instruments plotted against time tolerance ϵ_t for values from $\epsilon_t = 10$ ms to $\epsilon_t = 100$.

are adversely affected by the flute performance, especially when the time tolerance is decreased to under 30 ms.

5. CONCLUSIONS

We have presented a complete system to perform note objects labelling of a wide range of monophonic music signals within a short delay after the onset detections. Our small library aims at being lightweight, portable, and used from other softwares, either real time or off-line. The library will be made available under the GNU General Public License (GPL). End user applications have started with various interfaces to other software environments such as CLAM and Audacity.

Using a varied set of MIDI files, the evaluation of this note object extraction system has shown useful performances can be obtained with different instruments. Results have enlightened some of the issues encountered: soft attacks tend to delay the detection of onsets; transient components affect the pitch decision in the few frames after the onset. The onset detection threshold (α) and the note decision delay (δ) are important parameters controlling under or over-detection on one hand, and the delay and accuracy of the system on the other hand.

Evaluation of the performance should also be tested on real recordings. Possible improvements include the use of a specific module for bass line detections and the elaboration of additional features to be added to the note labels.

6. ACKNOWLEDGEMENTS

PB is supported by a Studentship from the Department of Electronic Engineering at Queen Mary College, University of London. This research has been partially funded by the EU-FP6-IST-507142 project SIMAC (Semantic Interaction with Music Audio Contents) and by EPSRC grant $GR/54620$.

7. REFERENCES

[1] X. Amatrian and P. Herrera, "Transmitting audio content as sound objects," in *Proc. of AES22 Internat. Conference on Virtual Synthetic and Entertainment*, Audio Espoo, Finland, 2002, pp. 278–288.

[2] P. M. Brossier, M. Sandler, and M. D. Plumbley, "Real time object based coding," in *Proceedings of the Audio Engeeniring Society, 114th Convention*, Amsterdam, The Netherlands, 2003.

[3] P. M. Brossier, J. P. Bello, and M. D. Plumbley, "Real-time temporal segmentation of note objects in music signals," in *Proceedings of the ICMC*, Miami, Florida, 2004, ICMA, Conference submission.

[4] J. C. Brown and B. Zhang, "Musical Frequency Tracking using the Methods of Conventional and 'Narrowed' Autocorrelation," *J. Acoust. Soc. Am.*, vol. 89, pp. 2346–2354, 1991.

[5] P. Cano, "Fundamental frequency estimation in the SMS analysis," in *Proc. of COST G6 Conference on Digital Audio Effects*, Barcelona, 1998, pp. 99–102.

[6] M. Slaney and R. F. Lyon, "A Perceptual Pitch Detector," in *Proc. ICASSP*, 1990, pp. 357–360.

[7] P. Lepain, "Polyphonic pitch extraction from music signals," *Journal of New Music Research*, vol. 28, no. 4, pp. 296–309, 1999.

[8] P. Masri, *Computer modeling of Sound for Transformation and Synthesis of Musical Signal*, PhD dissertation, University of Bristol, UK, 1996.

[9] C. Duxbury, M. E. Davies, and M. B. Sandler, "Complex domain onset detection for musical signals," in *Proc. of the DAFx Conf.*, London, 2003.

[10] J. P. Bello, *Towards the Automated Analysis of Simple Polyphonic Music*, PhD dissertation, Queen

Mary, University of London, Centre for Digital Music, 2003.

[11] "Mutopia Project, a public domain collection of sheet music," http://www.mutopiaproject.org/.

[12] T. Toivonen and M. Izumo, *Timidity++, a MIDI to WAVE converter/player*, http://www.timidity.jp/, 1999, GNU/GPL.