MULTI-PITCH ESTIMATION BASED ON PARTIAL EVENT AND SUPPORT TRANSFER

Zhiyao Duan, Dan Zhang, Changshui Zhang and Zhenwei Shi

State Key Lab of Intelligent Technologies and Systems
Department of Automation, Tsinghua University, Beijing 100084,P.R.China
{duanzhiyao00, dan-zhang05}@mails.tsinghua.edu.cn, {zcs, shizhenwei}@mail.tsinghua.edu.cn

ABSTRACT

This paper proposes a method for the multi-pitch estimation of polyphonic music signals. Instead of on the frame level, the estimation is based on the Partial Event, which is defined like the note event in MIDI. All partial events in a piece of music are extracted dynamically in the process of the frame by frame Short Time Fourier Transform (STFT). For each event, Net Support degree received from other events is calculated and the events with the highest support degrees are selected to be the fundamental frequency (F0) events. From another point of view, the support is transferred from higher frequency partial events to lower ones and finally concentrated on the F0 events. This method can estimate the number of concurrent sounds, the onset and offset times of the notes. Experiments on both randomly mixed chord signals and synthesized ensemble music signals in “wav” format are conducted and the results are promising.

1. INTRODUCTION

Muti-pitch estimation (MPE) of several concurrent sounds in polyphonic music signals has been generally considered as one of the central problem in many music signal processing applications, including automatic transcription, music information retrieval and music content analysis.

Contrary to its importance, however, numerous conventional methods have fallen clearly behind human’s ability in both accuracy and flexibility. In recent years, several new methods have been proposed. Kashino and Murase [1] applied a Bayesian probability network to integrate musical context to address this problem. Goto [2], Davy and Godsill [3], and Kameoka [4] employed parametric signal models and statistical methods. Klapuri [5, 6] proposed methods based on human auditory system. Saito and Kameoka [7] proposed the specmurt analysis which is similar to a deconvolution process to estimate fundamental frequencies (F0s). Poliner and Ellis [8] viewed the polyphonic piano transcription as a classification problem.

All these methods share a common characteristic: F0s are estimated at the frame level. However, this is not the case when human perception is concerned. To some extent, MPE can be viewed as a process of grouping partials to notes. Bregman [9] pointed out that synchronous changes in the parameters of the components was one of the perceptual cues for grouping of time-frequency components. Note that this cue cannot be embedded in the frame level estimation, which may be also influenced severely by noise, therefore, several methods employed postprocesses to alleviate this problem [2, 8].

In this paper, we propose a method for estimating F0s on partial events rather than on the frame level. The concept of the partial event is borrowed from that of the note event in MIDI. A partial event e_i is defined as

$$e_i = (f_i, A_i, t_{ia}, t_{ib}) \tag{1}$$

where f_i is its average frequency, A_i is its average logarithm amplitude, t_{ia} is its onset time and t_{ib} is its offset time.

All the partial events in a piece of music are extracted and each of them is a F0 event candidate. They compete with each other to receive enough net support from other events to become to a F0 event. Finally several partial events with the highest degrees are selected to be F0 events. Experiments are conducted on both randomly mixed chord data and synthesized ensemble music data. The results are promising.

The rest of the paper is organized as follows. The proposed method is described in Section 2 and experimental results are presented in Section 3. Section 4 is the conclusion.

2. THE PROPOSED METHOD

2.1. Partial Event Extraction

For a particular piece of music, partial events are extracted from the Short Time Fourier Transform (STFT) spectrums of a series of frames. In each spectrum, significant peaks are detected. Peaks with approximately the same frequency of consecutive frames form a partial event.

Figure 1 illustrates how the peaks in the $(k+1)$th frame affects the generation process of partial events, with the events in the first k frames having been already extracted. For each extracted event that still exists in the kth frame, we try to find

The work was supported by the National Science Foundation of China (60605002) and by the Chinese Postdoctoral Science Foundation (2005038075).

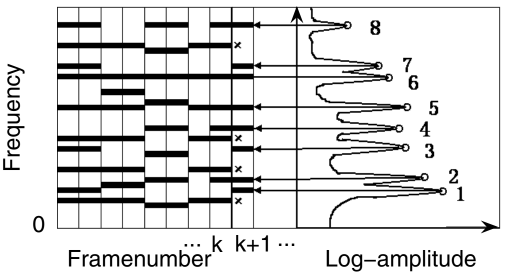

Fig. 1. Illustration of the partial events extraction process. The left part is the time-frequency plane up to the kth frame, each event is depicted as a horizontal line. The right part is the spectrum of the $(k+1)$th frame, with peaks labeled.

a corresponding peak in the spectrum of the $(k+1)$th frame, according to the minimal frequency difference principle. This correspondence is established when the frequency difference is within half of the seminote range. Note that an event can correspond to at most one peak in this frame, and vice versa.

If a partial event has found a corresponding peak, then the parameters of this event are updated as follows:

$$f_i(k+1) = \frac{l_k \cdot f_i(k) + f_{pi}}{l_k + 1}, A_i(k+1) = \frac{l_k \cdot A_i(k) + A_{pi}}{l_k + 1} \tag{2}$$

where l_k is the current length of e_i, f_{pi} and A_{pi} are the frequency and amplitude of the peak, respectively. In Figure 1, peaks labeled 2, 4, 5, 6 and 8 hold this situation.

For the partial event that has not found a corresponding peak, it is terminated, with the offset time being set to the time of the kth frame. In Figure 1, events marked with '×' are terminated in the kth frame.

For the peak that has not found the corresponding partial event, it is used to generate a new event. The onset time of this event is set to the time of the $(k+1)$th frame, and the initial average frequency and average amplitude is set to that of the peak. In Figure 1, peak 1, 3 and 7 are in this case.

After all frames having been processed, the generation of partial events on the time-frequency plane is completed. On this plane, some noise events exist, including short events caused by the detection of fake peaks, and some fragmentized events caused by the frequency fluctuation of notes. We employ two morphological operations (a close operation and an open operation) on the plane to get a clearer one, see Figure 2. On this plane, each horizontal line refers to a partial event (the average amplitude of each event is not depicted). All the events compose the whole partial events set

$$\mathscr{E} = \{e_i | i = 1, 2, ..., N\} \tag{3}$$

where N is the number of partial events. The following analysis will be based on this set.

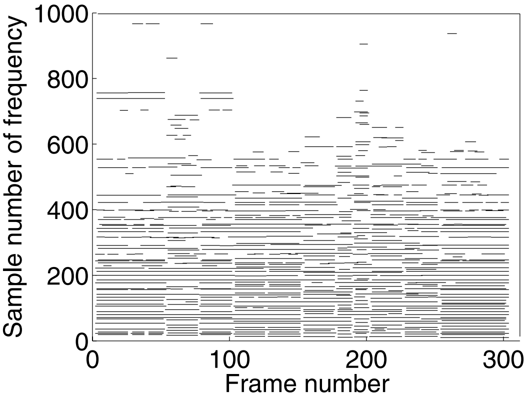

Fig. 2. Time-frequency plane after all frames have been processed , where each horizontal line refers to a partial event.

2.2. Support Degree Calculation

Each partial event in the events set is treated as a F0 event candidate and the following work is to determine which candidates are the true F0 events. Just like in the voting process, a man who wants to be a leader must receive enough support from others, each F0 event candidate must receive enough support from the other events to be selected as a F0 event. The support degree is defined as follows.

Firstly, for each partial event e_i in $\mathscr{E}$, consider a subset $\mathscr{E}_i$ whose elements have the onset times near that of e_i, and the average frequencies larger than f_i, i.e. the frequency of e_i.

$$\mathscr{E}_i = \{e_k | f_k > f_i, |t_{ka} - t_{ia}| < \theta, k = 1, 2, ..., N\} \tag{4}$$

where θ is set to 150ms typically.

Then suppose the exact frequency of e_i is f_i', since its average frequency f_i may not be precise due to the resolution in the frequency domain. The support degree that e_i receives from another event e_j is defined as

$$s_{ij}(f_i') = \begin{cases} R_{ij} \cdot P_{f_i'} \cdot Q_{f_i'f_j} \cdot A_i \cdot A_j & e_j \in \mathscr{E}_i \\ 0 & e_j \notin \mathscr{E}_i \end{cases} \tag{5}$$

where

$$R_{ij} = \frac{\min(t_{ib}, t_{jb}) - \max(t_{ia}, t_{ja})}{t_{ib} - t_{ia}} \tag{6}$$

$$P_{f_i'} = \exp(-\frac{(\frac{f_i'}{f_i} - 1)^2}{\sigma^2}) \tag{7}$$

$$Q_{f_i'f_j} = \exp(-\frac{(\frac{f_j}{f_i'} - [\frac{f_j}{f_i'}])^2}{\sigma^2}) \tag{8}$$

where $[\cdot]$ denotes rounding to the nearest integer. R_{ij} represents the overlap ratio between e_i and e_j. $P_{f_i'}$ defines the proximity between the average frequency f_i and the supposed exact frequency f_i'. $Q_{f_i'f_j}$ represents the weight caused by the harmonic relationship between f_i' and f_j. σ is set to 0.015 to prevent the difference between f_i and f_i' being larger than a seminote range in Eq.(7) and to ensure the harmonic relationship between e_i and e_j in Eq.(8).

For the supposed frequency f_i', the support degree of e_i received from all the other partial events is

$$S_i(f_i') = \sum\nolimits_{j=1}^{N} s_{ij}(f_i') \tag{9}$$

Then we search for f_i' in the semitone interval of f_i to get the maximum of Eq.(9).

$$\hat{S}_i = \max(S_i(f_i')) \tag{10}$$

Suppose the maximum is achieved when $f_i' = f_i^0$, then the exact average frequency of e_i is set to f_i^0. For the partial event e_i, the actual support it receives from e_j is $\hat{s}_{ij} = s_{ij}(f_i^0)$.

It is noticed that when a partial event receives support from other events, it usually also gives out support. So we should consider the Net Support NS_i that e_i receives:

$$NS_i = \sum\nolimits_{j=1}^{N} \hat{s}_{ij} - \alpha \sum\nolimits_{k=1}^{N} \hat{s}_{ki} \tag{11}$$

where α is the tradeoff between received and given support.

In monophonic case, α tends to be set large enough to ensure F0 events to be selected, since partial events of a note give support to the corresponding F0 event while the F0 event never give support to any others. However, in polyphonic case, the F0 event of a note may give support to partial events of other concurrent notes octaves lower. Therefore, α should be set properly to prevent this kind of F0 events being neglected. In our experiment, we find that $\alpha = 2$ is proper for the polyphony number ranging from 1 to 6.

Finally, the net support NS_i is normalized to [0,1]. All the events whose NS_i are larger than τ are selected to be F0 events. τ is set as

$$\tau = mean(NS_i) + \beta \cdot std(NS_i) \tag{12}$$

where β is set to 1.2 typically. It's better to adjust the value between [0.85,1.7] for polyphony number from 6 to 2, to make the precision and recall close (Eq.(13)).

Let us reconsider the calculation of the support degree from another point of view. Each partial event can only receive support from the events whose frequencies are higher and give support to the events whose frequencies are lower. Therefore support is transferred from higher partial events to lower ones. This transfer ends at the F0 events.

3. EXPERIMENTAL RESULTS

We applied the proposed method on both randomly mixed chord signals and synthesized ensemble music signals in "wav" format. The STFT frame was 100ms long with 30 ms step. The structure elements of close operation and open operation were 90 ms and 210 ms long horizontal lines, respectively.

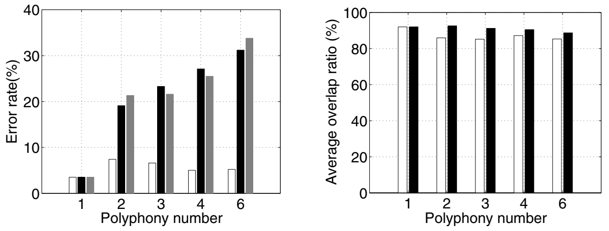

Fig. 3. MPE results for randomly mixed chord signals. Predominant-F0 (white) and Multiple-F0 (black and gray).

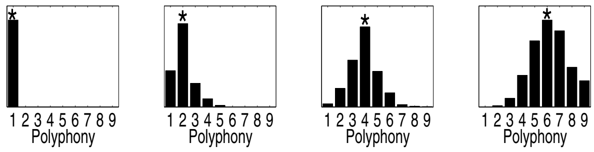

Fig. 4. Histograms of polyphony estimates. The asterisks indicate the true polyphony(1,2,4 and 6, from left to right)

3.1. Randomly Mixed Chords

The acoustic material consisted of note samples from the University of Iowa website [10]. There were altogether 369 note samples of dynamic "ff" from 14 wind instruments with pitch ranging from C3 (131Hz) to B6 (1976Hz). Randomly mixed chords were generated by mixing the samples with equal mean-square levels and no duplication in pitch. 100 mixtures of one, 500 mixtures of two, three, four, and six sounds were generated, totalling 2600 test cases.

The results are illustrated in Figure 3 and Figure 4. The left panel of Figure 3 is the error rate of the F0 estimation. In *Predominant-F0 estimation* (white bars), it is defined to be correct if the event with the highest support degree matches the F0 of any of the component sounds [2]. *Multiple-F0 estimation* is presented with two indices, *Recall* (black bars) and *Precision* (gray bars):

$$\textit{Recall} = \frac{c(cor)}{c(ref)}, \textit{Precision} = \frac{c(cor)}{c(trans)} \tag{13}$$

where $c(ref)$ is the number of reference notes, $c(trans)$ is the number of transcribed notes, and $c(cor)$ is the number of correctly transcribed notes [11].

For correctly transcribed notes (both *Predominant-F0* and *Multiple-F0*), in the right panel of Figure 3, *Average Overlap Ratio(AOR)* is calculated as

$$\textit{AOR} = mean(\frac{\min(offsets) - \max(onsets)}{\max(offsets) - \min(onsets)}) \tag{14}$$

where"onsets" refers to the onset times of both the reference and the corresponding transcribed note, and"offsets" accordingly to the offset times [11].

In Figure 3, the Predominant-F0 estimation is robust, because the error rates are all around 5% and do not increase

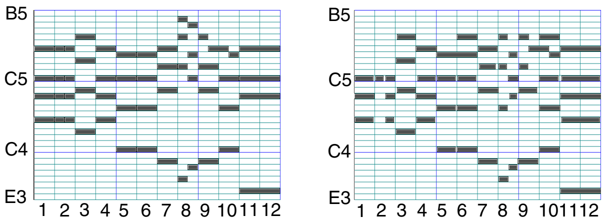

Fig. 5. Pianorolls of synthesized music (left) and the transcribed result (right). The horizontal axis refer to beat.

with the polyphony number. The error rates of Multiple-F0 are much higher, and increase with the polyphony number significantly, however, this result is promising as well, because all the error rates are estimated without the polyphony information. Moreover, from the right panel, it can be seen that AORs are around 90%. This indicates that the estimation of onset and offset times are accurate.

Figure 4 shows the histograms of polyphony estimation. The asterisk indicates the true polyphony in each panel. We can see that this method can estimate the polyphony number approximately while the result becomes worse when the polyphony number is large. This indicates the adjustment of the threshold τ in Eq. (12) should be investigated further.

3.2. Synthesized Music Piece

A piece of synthesized ensemble music data was also tested for the proposed algorithm. It was a four part chamber music played by flute, oboe, clarinet and bassoon respectively.

Figure 5 are the pianorolls of the original music and the transcribed result. Note that the transcribed pianoroll is exactly the same as the original one in the 3rd, 4th, 5th and 7th beats. In the 1st and 2nd beats, the note F5 is missing, since it probably gives overmuch support to the note F4 and is regarded as the second partial of F4. In the 6th, 10th, 11th and 12th beats, fake notes are detected which are partials of the true notes. Note that these octave mistakes are common in the MPE problem and do not affect the quality of the resynthesize audio severely. Re-synthesized audio examples are available at *http://mperesult.googlepages.com*.

3.3. Discussions

Compared with other algorithms, our method has the following advantages. (1) The estimation of F0s uses time information since it is operated on the events rather than a single frame. The process is more like the perceptual grouping process of human. (2) The output of this algorithm is note events, including not only fundamental frequencies but also onset and offset times of them (Figure 3). This information can be directly used in the Automatic Transcription task (Figure 5). (3) Our method does not need to be fed the number of concurrent sound. It is noticed that many other methods can not handle the MPE problem well without the polyphony information.

Like other MPE algorithms, the proposed method also has some limitations. (1) Significant changes of frequencies caused by vibrato and glissando may deteriorate the performance of the partial events extraction phase. (2) Eq.(8) limits the algorithm in harmonic instruments. Fortunately, most instrument sounds are harmonic. (3) The case of "missing F0" can not be handled, since the missing F0 events are not contained in the partial events set where F0s are chosen from. Note that all these limitations pose great challenges to all the existing algorithms.

4. CONCLUSIONS

In this paper, we propose a new MPE method which is based on the partial events rather than the frame level. This method can also estimate the number of concurrent sounds, the onset and offset times of the notes. It shows good performance on both randomly mixed chord signals and synthesized ensemble music. There are still some questions remaining unanswered such as the adjustment of the threshold τ and the risks in the extraction of partial events.

5. REFERENCES

[1] K. Kashino and H. Murase, "A Sound Source Identification System for Ensemble Music Based on Template Adaptation and Music Stream Extraction," *Speech Communication*, vol. 27, pp. 337-349, 1999.

[2] M. Goto, "A Real-Time Music Scene Description System: Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-World Audio Signals," *Speech Communication*, vol. 43, no. 4, pp. 311-329, 2004.

[3] M. Davy, S. J. Godsill and J. Idier, "Bayesian Analysis of Western Tonal Music," *Journal of the Acoustical Society of America (JASA)*, Vol. 119, No. 4, pp. 2498-2517, Apr. 2006.

[4] H. Kameoka, T. Nishimoto and S. Sagayama, "Harmonic-Temporal-Structured Clustering via Deterministic Annealing EM Algorithm for Audio Feature Extraction," *ISMIR 2005*.

[5] A. Klapuri, "Multiple Fundamental Frequency Estimation Based on Harmonicity and Spectral Smoothness," *IEEE Trans. Speech and Audio Processing*, vol. 11, no. 6, pp. 804-815, 2003.

[6] A. Klapuri, "Multiple Fundamental Frequency Estimation by Summing Harmonic Amplitudes," *ISMIR 2006*.

[7] S. Saito, H. Kameoka, T. Nishimoto and S. Sagayama, "Specmurt Analysis of Multi-Pitch Music Signals with Adaptive Estimation of Common Harmonic Structure," *ISMIR 2005*.

[8] G. Poliner and D. Ellis, "A Discriminative Model for Polyphonic Piano Transcription," *Eurasip Journal on Applied Signal Processing*, to appear, 2007.

[9] A. S. Bregman, *Auditory Scene Analysis*. The MIT Press, Cambridge, Massachusetts, 1990.

[10] The University of Iowa Musical Instrument Samples. *http://theremin.music.uiowa.edu/* [Online]

[11] M. Ryynänen and A. Klapuri, "Polyphonic Music Transcription Using Note Event Modeling," *WASPAA2005*