Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2009, Article ID 729494, 11 pages
doi:10.1155/2009/729494

Research Article

A Computationally Efficient Method for Polyphonic Pitch Estimation

Ruohua Zhou,[1] Joshua D. Reiss,[1] Marco Mattavelli,[2] and Giorgio Zoia[2,3]

[1] *Centre for Digital Music, School of Electronic Engineering and Computer Science, Queen Mary University of London, Engineering Building, Mile End Road, London E14NS, UK*
[2] *Signal Processing Institute, Swiss Federal Institute of Technology, ELB-116, 1015 Lausanne, Switzerland*
[3] *Systems Department, Creative Electronic Systems SA, 1212 Gd-Lancy-Geneva, Switzerland*

Correspondence should be addressed to Joshua D. Reiss, josh.reiss@elec.qmul.ac.uk

Received 27 August 2008; Revised 2 February 2009; Accepted 27 May 2009

Recommended by Gregor Rozinaj

This paper presents a computationally efficient method for polyphonic pitch estimation. The method employs the Fast Resonator Time-Frequency Image (RTFI) as the basic time-frequency analysis tool. The approach is composed of two main stages. First, a preliminary pitch estimation is obtained by means of a simple peak-picking procedure in the pitch energy spectrum. Such spectrum is calculated from the original RTFI energy spectrum according to harmonic grouping principles. Then the incorrect estimations are removed according to spectral irregularity and knowledge of the harmonic structures of the music notes played on commonly used music instruments. The new approach is compared with a variety of other frame-based polyphonic pitch estimation methods, and results demonstrate the high performance and computational efficiency of the approach.

1. Introduction

Polyphonic pitch estimation plays an important role in music signal analysis. It can be essentially used for the detection of musically relevant features such as melody and harmony [1]. In the case of content-based music retrieval, the "automatic" extraction of melody information is a crucial element for any music retrieval system [2]. Another potential application is assisting the structured audio coding [3, 4].

A number of approaches have been proposed in literature. Klapuri proposed a polyphonic pitch estimation algorithm based on an iterative method [5], which was further explored for music transcription [6]. In such method, first the predominant pitch of concurrent musical sound is estimated. Then the spectrum of the sound with the predominant pitch is estimated and subtracted from the mixture. The estimation and subtraction is repeated iteratively on the residual signal.

Recognizing a note in note-mixtures is a typical pattern recognition problem. Therefore, some approaches transform the polyphonic pitch estimation into a pattern recognition problem, which is then solved by employing machine learning methods such as neural networks [7, 8] and support vector machines [9, 10]. Other methods such as Bayesian inference [11–13], sparse coding [14], and nonnegative matrix factorization [15] have also been investigated. More detailed reviews on the state of the art of polyphonic pitch estimation can also be found in [16].

The aim of this article is to describe a computationally efficient method for polyphonic pitch estimation. The method consists of time-frequency analysis and postprocess phases. For both phases, novel techniques are used to increase computational efficiency. In the postprocess phase, neither iterative processing nor machine learning is needed. First, a preliminary estimation is used to find all possible pitch candidates, which may include extra estimations. Then the incorrect estimations are removed according to the spectral irregularity and knowledge of the harmonic structures. The postprocess phase mainly involves pick-peaking, addition, and subtraction operations, and the computational overload is negligible. Accordingly, the computational cost of the method chiefly depends on the time-frequency analysis part. The constant-Q Fast Resonator Time-Frequency Image (RTFI) has been selected as the basic time-frequency analysis

tool. RTFI is employed here mainly because it can be implemented by the simplest filter banks. In addition, fast implementations of such filter banks can also further improve the computational efficiency.

As a result, the overall approach is 3 times faster than real time on a standard PC equipped with a 2.0 GHz Pentium processer. The method was also evaluated in the multiple fundamental frequency frame level estimation task of MIREX 2007 [17]. The achieved results demonstrate the high performance and computational efficiency of the new approach. The method was the fastest and ranks third place in overall performance of the 16 submitted systems. Compared to the state-of-the-art approaches, it is more than 13 times faster and has only slightly worse performance (the accuracy of state-of-the-art method is 60.5%, whereas our method's accuracy is 58.2%).

The paper is organized as follows. Section II briefly introduces a new time-frequency analysis tool called Resonator Time-Frequency Image (RTFI) and the motivation to select Fast RTFI constant-Q analysis. Section 3 describes a new polyphonic pitch estimation method. Notably, Section 3.3 explains the novelty of the proposed method. Section 4 describes the experimental setup and reports the performance evaluation, and Section 4.6 compares the method with other state-of-the-art methods evaluated in MIREX 2007. Finally, Section 5 summarizes the main results and discusses possible extensions and future work.

2. Time-Frequency Processing

2.1. Frequency-Dependent Time-Frequency Analysis. A Frequency-Dependent Time-Frequency (FDTF) analysis may be defined as follows:

$$\mathrm{FDTF}(t,\omega) = \int_{-\infty}^{\infty} s(\tau) w(\tau - t, \omega) e^{-j\omega(\tau - t)} d\tau. \tag{1}$$

Unlike the STFT, the window function w of an FDTF may depend on the analytical frequency ω. This means that time and frequency resolutions can be tuned according to the analytical frequency. Equation (1) can also be expressed as

$$\mathrm{FDTF}(t,\omega) = s(t) * I(t,\omega), \tag{2}$$

where

$$I(t,\omega) = w(-t,\omega) e^{j\omega t}. \tag{3}$$

Equation (1) is more suitable to express a transform-based implementation, whereas (2) leads to a straightforward implementation of a filter bank with impulse response functions expressed by (3).

A novel time-frequency representation, known as the Resonator Time-Frequency Image (RTFI), has been developed. Its main feature is that it selects a first-order complex resonator filter bank to implement a frequency-dependent time-frequency analysis. This was chosen due to the flexibility with regards to time and frequency resolution and the simplicity and computational efficiency of an implementation based on first-order filters.

2.2. Resonator Time-Frequency Image. The Resonator Time-Frequency Image (RTFI) can be described as follows:

$$\begin{aligned} \mathrm{RTFI}(t,\omega) &= s(t) * I_R(t,\omega) \\ &= r(\omega) \int_0^t s(\tau) e^{r(\omega)(\tau - t)} e^{-j\omega(\tau - t)} d\tau, \end{aligned} \tag{4}$$

where

$$I_R(t,\omega) = r(\omega) e^{(-r(\omega) + j\omega)t}, \quad t > 0. \tag{5}$$

In the above equations, I_R denotes the impulse response of the first-order complex resonator filter with oscillation frequency ω and the factor $r(\omega)$ before the integral in (4) is used to normalize the gain of the frequency response when the resonator filter's input frequency is the oscillation frequency. The decay factor r is dependent on the frequency ω and determines the exponent window length and the time resolution. It also determines the bandwidth (i.e., the frequency resolution).

Since the RTFI has a complex spectrum, it may be expressed as follows:

$$\mathrm{RTFI}(t,\omega) = A(t,\omega) e^{j\varphi(t,\omega)}, \tag{6}$$

where $A(t,\omega)$ and $\varphi(t,\omega)$ are real functions. The energy of the signal may then be given by

$$\mathrm{RTFI}_{\mathrm{Energy}}(t,\omega) = |A(t,\omega)|^2. \tag{7}$$

In this work, it is proposed to use the first-order complex resonator digital filter bank to implement a discrete RTFI. To reduce the memory requirements needed to store the RTFI values, the RTFI is separated into different time frames, and the average RTFI values are calculated in each frame. Finally the average RTFI energy is used to track the time-frequency characteristics of the music signal. The average RTFI energy spectrum can be expressed as follows:

$$\mathrm{ARTFI}(g, f_k) = \mathrm{dB}\left(\frac{1}{M} \sum_{n=J_g}^{J_g+M-1} |\,\mathrm{RTFI}(n, f_k)\,|^2 \right), \tag{8}$$

where M is the number of sample in the time frame, g is the index of frame, dB() converts the value to decibels, and the ratio of M to sampling rate is the duration time of the frame in the averaging process. $\mathrm{RTFI}(n, f_k)$ denotes the value of the discrete RTFI at sampling point n and frequency f_k, and J_g denotes the frame which begins at the J_gth sample of the analyzed signal.

2.3. Multiresolution Fast RTFI. The Fast RTFI is used to reduce the redundancy in computation. In some cases it is not necessary to keep the same sampling frequency of the input for every filter in the filter bank. For the filters with lower center frequencies, the sampling rate can be decreased. In the fast implementation, the filter bank is separated into different octave frequency bands. The inputs of the filter banks in the same frequency band maintain the same sampling rate. The input signal is recursively low-pass

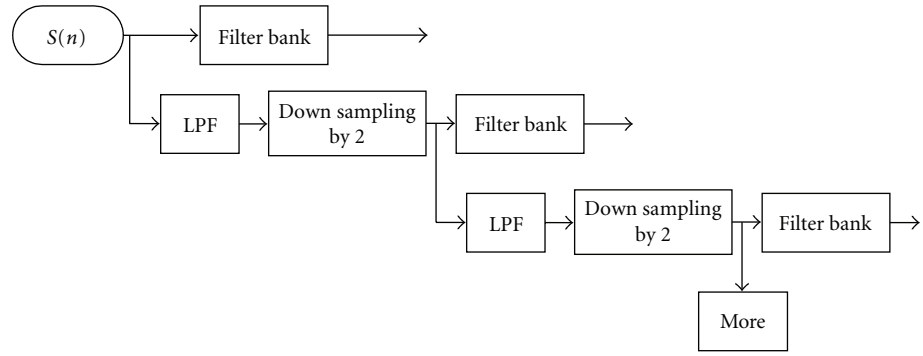


Figure 1: Block diagram of the multiresolution implementation.

filtered and down sampled by a factor of 2 from the highest to the lowest frequency band according to the scheme depicted in Figure 1.

This section has briefly introduced the basic idea behind RTFI analysis. A more detailed description of the discrete RTFI and its fast implementation can be found in [18, 19].

2.4. Motivation for Selecting Constant-Q Time-Frequency Analysis. Resolution is a key factor of any time-frequency analysis. In the following, it is explained how it may be reasonable to select a nearly constant-Q resolution for a general-purpose music analysis system. Using the Music Instrument Digital Interface (MIDI) note numbers, the fundamental frequency and corresponding partials of a music note k' can be described as

$$f_{k'}^0 = 440 \cdot \left(2^{(k'-69)/12}\right), \quad f_{k'}^m = m \cdot f_{k'}^0, \quad k' \geq 1. \tag{9}$$

Supposing that the energy of every music note is mainly distributed over the first 10 partials, thus $\text{Energy}(f_{k'}^m) \approx 0$ for $m \geq 11$, the frequency ratio between the partials of one note and the fundamental frequencies of other notes can be expressed as follows:

$$\begin{aligned}
2f_{k'}^0 &= f_{k'+12}^0, \qquad \frac{3f_{k'}^0}{f_{k'+19}^0} = 0.9989,\\
4f_{k'}^0 &= f_{k'+24}^0, \qquad \frac{5f_{k'}^0}{f_{k'+28}^0} = 1.0079,\\
&\qquad \frac{6f_{k'}^0}{f_{k'+31}^0} = 0.9989,\\
\frac{7f_{k'}^0}{f_{k'+34}^0} &= 1.018, \qquad 8f_{k'}^0 = f_{k'+36}^0,\\
\frac{9f_{k'}^0}{f_{k'+38}^0} &= 0.9977, \qquad \frac{10f_{k'}^0}{f_{k'+40}^0} = 1.008.
\end{aligned} \tag{10}$$

This means that the first 10 partials always overlap with another fundamental frequency. Since the fundamental frequencies follow an exponential law (9), most of the energy is concentrated in frequency bins that are evenly spaced on a logarithmic axis. This is the reason for which the required resolution is constant-Q.

2.5. Motivation for Selecting Fast RTFI to Implement Constant-Q Time-Frequency Analysis. The proposed method is mainly used for polyphonic pitch tracking, where a joint time-frequency analysis is first needed. Either filter bank or constant-Q transform can be used to compute constant-Q time-frequency spectrum. As RTFI is implemented by the simplest filter bank, it is faster than any other filter-bank-based implementation. The Fast RTFI is also compared with transform-based implementations as follows.

So as to use a constant-Q transform for a joint time-frequency analysis, the time signal needs to be cut into different frames, and then a constant-Q transform is performed in each frame [20]. It is assumed that the pitch tracking can report pitches every 10 milliseconds, so the time interval between two successive frames is set as 10 milliseconds. To perform a constant-Q time-frequency analysis for a 1-second signal, the constant-Q transform needs to be calculated 100 times, and the required number of complex multiplies can be expressed as

$$N_{\text{cq}} = 100 \cdot \frac{Qf_s}{f_{\min}} \frac{1 - 0.5^{N_1}}{1 - 0.5^{1/N_2}}, \tag{11}$$

where Q is the constant ratio of frequency to resolution, f_s is the sampling rate, $f_{\min}$ is the lowest analytical frequency, N_1 is the number of octave bands, and N_2 is the number of frequency components in one octave band. A fast constant-Q transform has been proposed in [21]. It employs an FFT to calculate constant-Q transform. When the fast constant-Q transform is used for time-frequency analysis of a 1-second signal, the required number of complex multiplies can be roughly expressed as

$$N_{\text{fcq}} = 100 \cdot N_{\text{fft}} \cdot \log(N_{\text{fft}}), \quad N_{\text{fft}} = \frac{Qf_s}{f_{\min}}. \tag{12}$$

For the Fast RTFI analysis of a 1-second signal, the required number of complex multiplies can be roughly obtained as

$$N_{fr} = 2f_s N_2 \left(1 - 0.5^{N_1}\right). \tag{13}$$

In the proposed method, the constant-Q factor Q is set as 17, the lowest analysis frequency $f_{\min}$ is 26 Hz, the number

of octave bands N_1 is 9, and the number of frequency components in one octave band is equal to 120. Accordingly, for constant-Q analysis of a 1-second signal, Fast RTFI implementation needs approximately $240 * f_s$ complex multiplies, constant-Q transform implementation needs approximately $24900 * f_s$, and fast constant-Q transform implementation needs approximately $2000 * f_s$. The comparison clearly suggests that Fast RTFI implementation is also much faster than transform-based implementation for a constant-Q time-frequency analysis.

3. Description of the Polyphonic Pitch Estimation Method

3.1. System Overview. Figure 2 provides an overview of the new polyphonic pitch estimation method. It can be conceptually partitioned into five different steps. First, a time-frequency processing based on the fast multiresolution RTFI analysis is performed. Harmonic components are then extracted by transforming the RTFI average energy spectrum into a relative energy spectrum (RES) according to the following (14):

$$\mathrm{RES}(f_k) = \mathrm{ARTFI}(f_k) - \frac{1}{M_1 + 1} \sum_{i=k-M_1/2}^{k+M_1/2} \mathrm{ARTFI}(f_i). \quad (14)$$

ARTFI denotes the input RTFI average energy spectrum, $k = 1, 2, 3, \ldots$ is the frequency index on the logarithmic scale, the second term in the right hand part of the equation denotes the moving average of ARTFI , and M_1 is the length of the window for calculating the moving average.

Similarly, preliminary estimates of the possible multiple pitches are found by a simple peak-picking procedure in a relative pitch energy spectrum, which is obtained from the RTFI average energy spectrum. Then a confidence measure is employed to remove pitch candidates whose harmonic components are not strongly represented. Finally, the pitches are found by investigating the spectral irregularity of the remaining candidates. These five steps are described in detail in the following subsections.

3.2. Detailed Description

3.2.1. Time-Frequency Processing Based on the RTFI Analysis. In the first step, the Fast RTFI is used to analyze the input music signal and to produce a time-frequency energy spectrum. The input sample is a monaural music signal frame at a sampling rate of 44.1 kHz. All 1080 filters are used. The center frequencies are set on a logarithmic scale. The center frequency difference between two neighboring filters is equal to 0.1 semitone, and the analyzed frequency range is from 26 Hz up to 13 kHz. Then, the time-frequency energy spectrum of the input frame is used to obtain an RTFI average energy spectrum according to (8). This RTFI average energy spectrum is used as the only input vector for later processing. An integer k is used to denote the frequency index

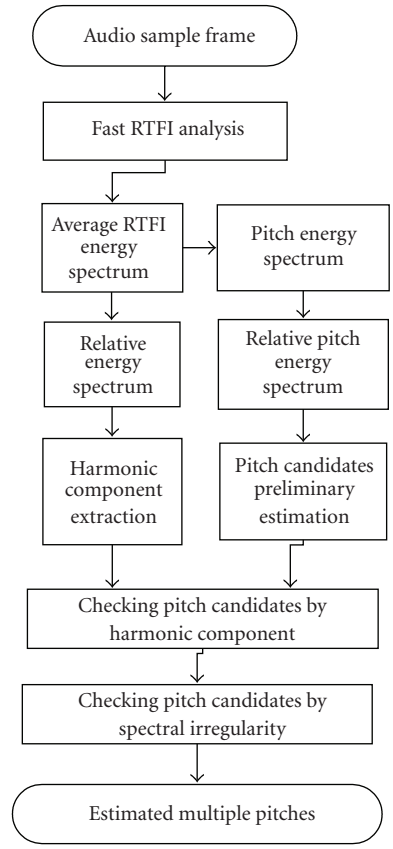


Figure 2: System overview of new polyphonic pitch estimation method.

on a logarithmic scale, and f_k denotes the corresponding frequency value expressed in Hz in the equation:

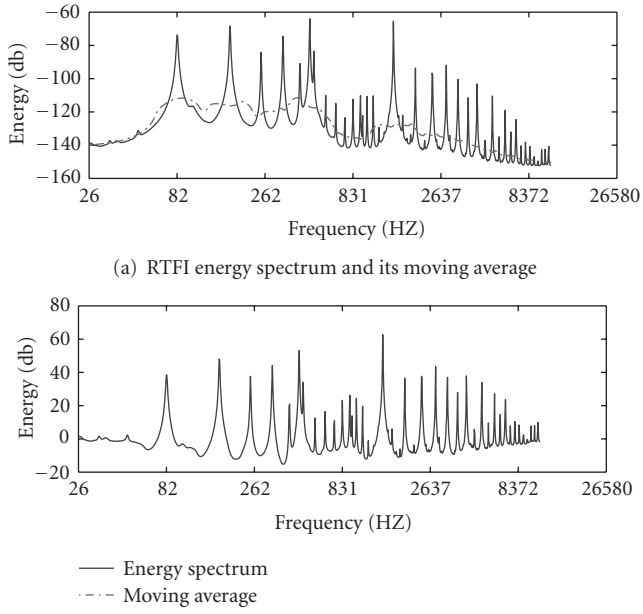
$$f_k = 440 \cdot 2^{(k-690)/120}. \quad (15)$$

Equation (15) has been derived from the fundamental frequencies of musical notes on the western music scale. One example for the input RTFI average energy spectrum of a piano note is provided in Figure 3.

3.2.2. Extraction of Harmonic Components. In the second step, the input RTFI average energy spectrum is first transformed into the relative energy spectrum according to the expression (14).

Figure 3 shows the RTFI energy spectrum and its moving average. The relative energy spectrum $\mathrm{RES}(f_k)$ is a measure of the energy spectrum for the kth frequency bin, relative to the energy spectrum over a frequency range near the kth frequency bin.

If there is a peak in the relative energy spectrum at the kth frequency index and the value $\mathrm{RES}(f_k)$ is larger than a threshold A_1, it is likely that there is a harmonic component at the frequency index k. The corresponding value $\mathrm{RES}(f_k)$ is assumed to be a measure of confidence in the existence of the harmonic component.



(a) RTFI energy spectrum and its moving average

(b) RTFI relative energy spectrum

Figure 3: The input RTFI energy spectrum, moving average and the corresponding relative energy spectrum of a piano polyphonic note consisting of two concurrent notes with fundamental frequencies 82 Hz and 466 Hz.

3.2.3. Preliminary Estimations of Pitch Candidates. In the third step, based on the harmonic grouping principle, the input RTFI average energy spectrum is first transformed into the pitch energy spectrum (PES) and the relative pitch energy spectrum (RPES) as follows:

$$\mathrm{PES}(f_k) = \frac{1}{L}\sum_{i=1}^{L}\mathrm{ARTFI}(i\cdot f_k), \quad k = 1,2,3,\ldots, \tag{16}$$

$$\mathrm{RPES}(f_k) = \mathrm{PES}(f_k) - \frac{1}{M_2+1}\sum_{i=k-M_2/2}^{k+M_2/2}\mathrm{PES}(f_i), \quad k = 1,2,3,\ldots, \tag{17}$$

where M_2 is the length of the window for calculating the moving average, and L is a parameter that denotes how many low harmonic components are together considered as important evidence for determining the existence of a possible pitch. Similar techniques have been proposed for pitch estimations by some researchers. In [22], the authors propose a polyphonic pitch estimation approach by summing harmonic amplitudes. There are two main differences between the method described in this paper and the approach introduced in reference [22]. First, the reference approach is based on the STFT spectrum, whereas the proposed method employs an RTFI constant-Q spectrum. Secondly, the reference approach directly sums harmonic amplitudes and does not use a decibel scale, whereas the new method produces a pitch energy spectrum by summing the harmonic energies on a decibel scale. Our experiments demonstrate that directly summing the harmonic energies yields lower estimation performances.

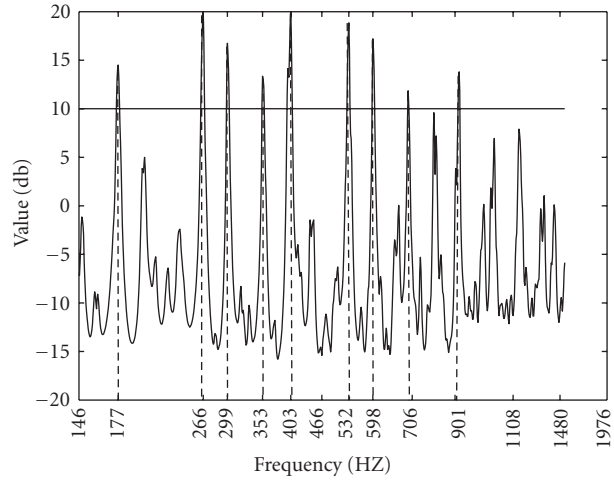


Figure 4: Relative Pitch Energy Spectrum of a violin example consisting of four concurrent notes with the fundamental frequencies 266 Hz, 299 Hz, 353 Hz, and 403 Hz.

In practical implementations, instead of using (16), the pitch energy spectrum on a logarithmic scale can easily be approximated by the following expression (here L is less than 10):

$$\mathrm{PES}(f_k) = \frac{1}{L}\sum_{i=1}^{L}\mathrm{ARTFI}(f_{k+A[i]}). \tag{18}$$

As shown in Table 1, the deviation between the approximate and ideal values of the pitch energy spectrum can be considered negligible for practical purposes.

There are two assumptions made when determining a preliminary estimate of the possible pitches from the relative pitch energy spectrum. If there is a pitch with fundamental frequency f_k, in the input signal, there should be a peak centred around the frequency f_k in the relative pitch energy spectrum, and the peak value should exceed a threshold A_2. Both assumptions are consistent with real music examples when a suitable threshold A_2 is selected.

Figure 4 illustrates the relative pitch energy spectrum of a violin example, which consists of four concurrent notes with fundamental frequencies of 266 Hz, 299 Hz, 353 Hz, and 403 Hz, respectively. As shown, there are 9 pitch candidates that can be preliminarily estimated when selecting the threshold $A_2 = 10$ dB. The fundamental frequencies of the 9 pitch candidates are 177 Hz, 266 Hz, 299 Hz, 353 Hz, 403 Hz, 532 Hz, 598 Hz, 796 Hz, and 901 Hz. Such preliminary estimation includes 4 correct pitch candidates and 5 incorrect ones. The incorrect pitch estimations usually share many harmonic components with the true pitches. In this example for instance, the false pitch of 177 Hz is positioned at a frequency that is nearly half that of the true pitch of 353 Hz.

3.2.4. Removal of Extra Pitches by Checking Harmonic Components. By means of a large number of experiments it has

Table 1: Deviation between approximate and ideal values of the pitch energy spectrum. $A[10] = [0, 120, 190, 240, 279, 310, 337, 360, 380, 399]$.

i	1	2	3	4	5	6	7	8	9	10
$\frac{f_{k+A[i]}}{i \cdot f_k}$	0%	0%	−0.1%	0%	0.2%	−0.1%	0.07%	0%	−0.2%	0.2%

been observed that the lowest harmonic components of the music notes are relatively strong and can be reliably extracted by applying the second step of the developed method. Only the low-pitch notes may have very faint first harmonic components that cannot be reliably extracted. Based on these observations, some assumptions concerning the extracted harmonic components can be made for determination of whether an extracted pitch is correct. For example, if there is a pitch with a fundamental frequency higher than 82 Hz, either the lowest three harmonic components or the lowest three odd harmonic components of this pitch should all be present in the extracted harmonic components. If there is a pitch with a fundamental frequency lower than 82 Hz, four of the lowest six harmonic components should be present in the extracted harmonic components.

In two typical cases, the extra estimated pitches can be removed based on the above assumptions. In the first case, the extra pitch estimation is caused by a noise peak in the preliminary pitch estimation. In the second case, the harmonic components of an extra estimated pitch are partly overlapped by the harmonic components of the true pitches. In such a case, the nonoverlapped harmonic components become important clues to check the existence of the extra estimated pitch. If a polyphonic set of notes contains two concurrent music notes C5 and G5, for example, the fundamental frequency ratio of the two notes is nearly 2/3. Then, it is probable that there is an extra pitch estimation on the C4 note, because its even harmonics are overlapped by the odd harmonics of C5, and the C4 note's third, sixth, ninth, and so forth, harmonic components are nearly overlapped by the G5 note's odd harmonics. However, the C4's first, fifth, and seventh harmonic components are not overlapped, so the extra C4 estimation can be easily identified by checking the existence of the first harmonic component based on the above assumption.

3.2.5. Determining the Existence of the Pitch Candidate by the Spectral Irregularity. By means of the previous steps, the extra incorrect estimations centered around the pitches whose note intervals are 12, 19, or 24 semitones higher than the identified true pitches. In such a case, the fundamental frequencies of the extra estimated pitches are placed 2, 3, or 4 times the frequency of a true pitch, and the harmonic components of each extra pitch are completely overlapped by the true pitch. For example, consider when two of the estimated pitch candidates are the notes with fundamental frequencies F_0 and $3F_0$. Here the difficulty is to determine if the note with the fundamental frequency $3F_0$ is an incorrect extra estimation caused by the overlapped frequency components of the lower frequency music note. This is the most difficult case in the polyphonic pitch estimation problem. However, such a problem can be solved by investigating spectral irregularity.

The spectral value difference between two neighboring harmonic components is small and random in most cases. But when a music note with the fundamental frequency F_0 is mixed with another note with the higher integer ratio fundamental frequency nF_0, then the corresponding spectral value of every nth harmonic component will become clearly larger than the neighboring harmonic components.

Figure 5 illustrates the RTFI average energy spectrum of the first 30 harmonic components of two piano music samples. The top image presents the analysis results for a piano sample that contains only one music note with a fundamental frequency of 147 Hz. The bottom image shows the result of analysis for a piano sample that has two concurrent music notes with a fundamental frequency of 147 Hz and 440 Hz ($\approx 3 * 147$ Hz). It is clear that, in comparison to the top image, the 3rd, 6th, 9th, and so forth, harmonic components are reinforced, and their spectral values are significantly larger than the neighboring harmonic components.

If there are two estimated pitch candidates that have fundamental frequencies of F_0 and F_0' ($F_0' \approx nF_0$) and a frequency ratio that is approximately an integer n, then the proposed method employs the following two steps to determine if the higher pitch with the fundamental F_0' occurs. First, the energy spectrum of the first $10n$ corresponding harmonic components with the fundamental frequency F_0 is calculated by an RTFI analysis with uniform resolution. The RTFI average energy spectrum of the harmonic components can be expressed as $\mathrm{ARTFI_H}(k)$, $k = 1, 2, 3, \ldots, (10n)$, where k denotes the harmonic component index.

The second step is composed of the following operations. The Spectral Irregularity (SI) is calculated using the expression:

$$\mathrm{SI}(n) = \sum_{i=1}^{9} \Big(\mathrm{ARTFI_H}(i \cdot n) - \Big(\frac{\mathrm{ARTFI_H}(i \cdot n - 1) + \mathrm{ARTFI_H}(i \cdot n + 1)}{2} \Big) \Big). \tag{19}$$

According to our observations, if two of the estimated pitch candidates have the fundamental frequencies, F_0 and F_0' for which ($F_0' \approx nF_0$) and if the higher pitch does not occur, then $\mathrm{SI}(n)$ is usually small. On the other hand, if the higher pitch does occur, then the overlapped harmonic components are often strengthened so that $\mathrm{SI}(n)$ results in a larger value. When $\mathrm{SI}(n)$ is smaller than a given threshold, the

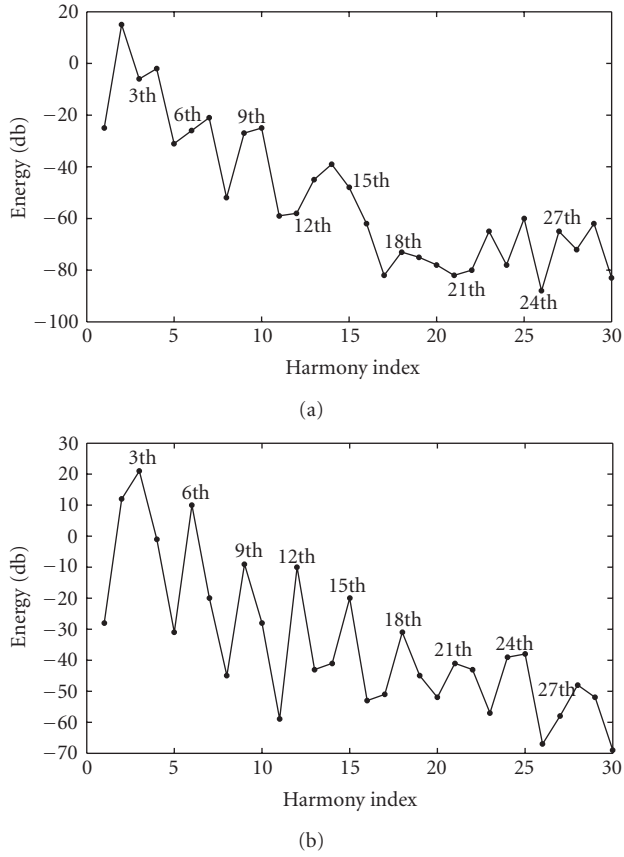


Figure 5: Harmonic component energy spectrum of a piano sample including a single note with fundamental frequency at 147 Hz (a) and a piano sample including two concurrent notes with fundamental frequencies at 147 Hz and 440 Hz (b).

overlapped higher pitch candidate is removed. The threshold is determined by experiments on a training database. In practical examples, most incorrect extra estimates caused by the overlapping of harmonic components are placed at a low integer multiple of the frequency of the true pitch. Consequently, the new method proposed in this paper only consider cases for which the fundamental frequency ratio of two pitch candidates is equal to 2, 3, or 4.

3.3. Novelty of the Proposed Method. In this subsection, the novelty and promising features of the proposed method is outlined. In the time-frequency processing part, the Fast RTFI constant-Q time-frequency analysis is first employed for polyphonic pitch tracking. As explained in Section 2.5, it is much more computationally efficient than other implementations.

In the postprocess phase, the developed method first estimates pitch candidates by peak-picking from the relative pitch energy spectrum. Since the sounds with integer fundamental frequency ratio can produce very similar peak patterns in a pitch energy spectrum, usually an extra incorrect estimation has an integer ratio to the fundamental frequencies of an identified pitch. This problem mainly arises from the coinciding frequency partials between Western polyphonic music notes.

The state-of-the-art method solves the problem by employing iterative estimation and cancelation schema [5]. The basic idea is to first find a predominant pitch and estimate the spectrum of the predominant pitch. Then the estimated spectrum is cancelled from the mixture and produces residual signals before the next estimation. The estimation and cancellation is repeated iteratively on the residual signal. It may also involve the process of estimating the polyphonic number of the analyzed sound.

So as to solve the problem of coinciding frequency partials, the basic idea of the new proposed method is completely different from the state-of-the-art approach introduced above. The proposed method provides a much simpler solution to the problem and does not require to implement an iterative procedure or to estimate the polyphonic number. In the new method, the preliminary estimation finds all possible pitch candidates. Then some pitch candidates are removed if their harmonic components are not enough represented in the energy spectrum. Finally, if fundamental frequencies between any two pitch candidates have an integer ratio, the spectral irregularity is calculated to remove the pitch candidate, which is considered to be an error estimation caused by coinciding frequency partials from a lower pitch.

By employing these new techniques, the proposed method is more computationally efficient, but presenting comparable performance with the other state-of-the-art methods.

4. Experiments and Results

4.1. Performance Evaluation Criteria. Three criteria were used to evaluate the performance of the polyphonic pitch estimation methods; "Precision", "Recall", and "F-measure". Given a reference fundamental frequency, if there is an estimation that is equal to or presents an error of no more than 3% deviation from the reference fundamental frequency, it is considered to be a correct detection. Otherwise, it is considered as a false negative (FN). Any estimation that deviates by more than 3% from all reference fundamental frequencies is considered to be a false positive (FP). Precision, Recall, and F-measure can be defined according to the following expressions:

$$P = \frac{N_{\mathrm{CD}}}{N_{\mathrm{CD}} + N_{\mathrm{FP}}},$$

$$R = \frac{N_{\mathrm{CD}}}{N_{\mathrm{CD}} + N_{\mathrm{FN}}}, \tag{20}$$

$$F - \mathrm{measure} = \frac{2PR}{P + R},$$

where N_{CD}, N_{FP}, and N_{FN} denote the total number of correct detections, false positives, and false negatives, and P and R denote the values of precision and recall, respectively. In addition, the Overall Accuracy, as defined in [9], is also used for the performance comparison with other state-of-the-art methods.

Table 2: Size of Training Set and Test Set.

Polyphony number	Training dataset	Test dataset
2	1000	1000
3	2000	2000
4	2000	2000
5	3000	3000
6	3000	3000
Total	11000	11000

4.2. Setting the Method Parameters. The real performance of an estimation method may be overestimated when parameters have been optimally selected to fit the test data. So as to prevent such occurrence, separate training and test datasets have been constructed.

It is quite difficult to record a large number of polyphonic samples from different musical instruments and label their polyphony content. A preferred method is to produce the polyphonic samples by mixing real recorded monophonic samples of different music instruments.

In these experiments, two different monophonic sample sets were used to create the training and test dataset. The monophonic sample set I consisted of a total of 755 monophonic samples from 19 different instruments, such as piano, guitar, winds, strings, and brass. To obtain fairer evaluation results of practical cases, the monophonic sample set II was used to generate the test dataset. Compared to set I, the monophonic samples in Set II, for the same type of instrumentation as samples in Set I, were played by different performers and instruments from different instrument manufacturers. Set II included 23 different instrument types, a total of 690 monophonic samples in the five octave pitch range from 48 Hz to 1500 Hz.

All the monophonic samples in Set I and Set II were selected from the RWC instrument sound database [23]. Every instrument sample was recorded at three levels of dynamics (forte, mezzo, piano) across the total range of that instrument. Generally speaking, different instruments play with different strengths. Accordingly, instead of being normalized, the natural amplitudes of the monophonic samples were kept in order to construct polyphonies by different energy ratios. The high number of polyphonic samples was generated by randomly mixing these different monophonic samples. These polyphonic samples were generated by first selecting an instrument and then a random note from the instrument's playing range. Based on the monophonic sample set I, a total of 11 000 polyphonic samples with the polyphony from two to six note mixtures were generated for the training dataset. Similarly, monophonic set II was used to generate 11 000 polyphonies for the test dataset. The size of every polyphonic subset in the training and test datasets is described in Table 2. All the following test experiments were performed on the whole test dataset, which was classified into five different subsets according to the polyphony number of the mixed polyphonic samples.

The described method has eight different parameters: L, M_1, M_2A_1, A_2, and the thresholds of spectral irregularity.

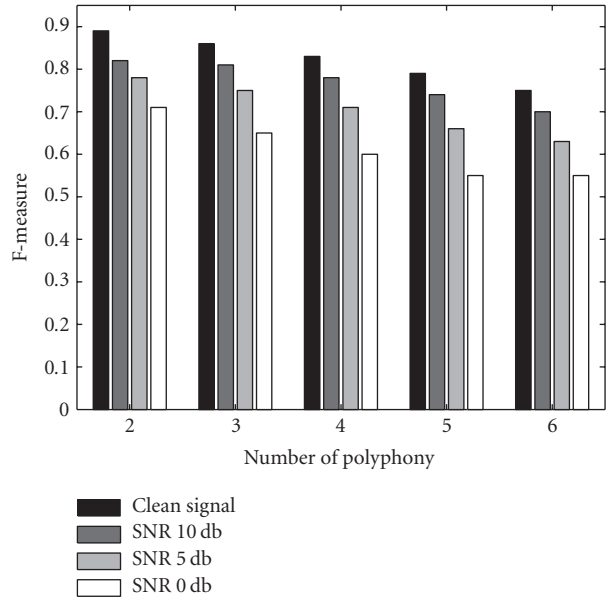


Figure 6: F-Measure of test results of the proposed method with a clean signal or various levels of added noise.

These parameters were tuned on the training dataset. The different parameter values were selected by a heuristic method. Table 3 reports the values, which were tried for different parameters. About 15 000 parameter combinations were tried. Values that yielded the best average F-Measure on the training dataset were selected, and parameters were fixed when the method was evaluated on the test dataset.

4.3. Performance and Robustness. The method was tested on the test dataset and achieved F-measures of 89%, 87%, 84%, 81%, and 78%, respectively, on polyphonic mixtures ranging from two to six simultaneous sounds. In order to test the robustness, pink noise was added into the polyphonic mixtures with different Signal-to-Noise ratios. The pink noise was generated in the frequency range of 50 Hz to 10 KHz. The Signal-to-Noise refers to the ratio between the clean input signal power and the added pink noise power.

Figure 6 shows the F-measure of the new method with different levels of added pink noise, where a value of 1 for the F-measure indicates optimal performance. In general, the method is robust, even in cases of severe noise levels. The tested samples were classified into five different sample subsets according to the polyphony number of the mixed polyphonic samples. For example, in Figure 6, the F-measure corresponding to the polyphony number 2 denotes the F-measure value estimated on the sample subset, in which every polyphonic sample consists of a two-note mixture.

4.4. Comparison Experiments with/without Applying Relative Spectra. In the described method, the relative spectra (relative energy spectra and relative pitch energy spectrum) have been used. A comparison experiment has been made to evaluate how the application of relative spectra improves the method's performance. The method was tested for every

Table 3: Values for different parameters.

Parameter	L	M_1	M_2	A_1	A_2	SI(1)	SI(2)	SI(3)
Values	3, 4, 5, 6	50, 300, 600	50, 300, 600	4, 8, 12, 16	4, 8, 12, 16	5, 10, 15	5, 10, 15	5, 10, 15

Table 4: F-Measures of proposed method with/without applying relative spectra.

Polyphony number	Using relative spectrum	Not using relative spectrum
2	89%	85%
3	87%	83%
4	84%	81%
5	81%	79%
6	78%	76%

polyphony sample subset of the test dataset. The test results of the method with or without applying the relative spectrum are reported in Table 4. The results demonstrate that the application of the relative spectrum improves the method's performance.

4.5. Tradeoff between Recall and Precision. Precision is the percentage of the transcribed notes that are correct, and Recall is the percentage of all the notes that are found. There is inherent tradeoff between Precision and Recall. Depending on applications, better Precision or better Recall is preferred. For example, in some music transcription systems, the extra incorrect estimations in the result are very harmful, so better Precision is preferred. However, if the output result will be used for further improvement with the combination of some higher level knowledge, better Recall is preferred.

The tradeoff between Precision and Recall can be controlled by adjusting the thresholds A_1, A_2 and the thresholds of spectral irregularity. In this method, harmonic components need to be extracted from the relative energy spectrum by peak-picking. Although the peaks with larger values have higher probability to represent harmonic components, there may still be some large peaks which represent noise. Thus, only the peaks with values larger than the threshold A_1 are considered to represent harmonic components. When A_1 is set to a small value, more true harmonic components may be extracted, but more noise peaks are also incorrectly assumed to be harmonic components. As a result, more true notes may be found, but the incorrect estimation are also increased. Therefore, when A_1 is set low, the method will get better Recall at the cost of lower Precision. Similarly, if thresholds A_2 and the thresholds of spectral irregularity are set low, estimation performance will probably have better Recall. Otherwise, the estimation performance will have better Precision. Figure 7 shows the estimation performance (F-measure, Recall, Precision) of this method with two different parameter sets. Compared with Figure 7(a) (small parameter values), the Precision shown in Figure 7(b) (large parameter values) increases at the price of a lower Recall.

Table 5: Results of multiple fundamental frequency frame Level estimation task of MIREX 2007.

Team ID	Accuracy	Running time (sec)
ZR	58.2%	271
RK	60.5%	3540
CY	58.9%	132300
PI1	58.0%	364
EV2	54.3%	2233
CC1	51.0%	2513
SR	48.4%	41160
EV1	46.6%	2366

4.6. MIREX 2007 Results—Performance Comparison to Other State-of-the-Art Methods. In order to compare our technique with other state-of-the-art approaches, the new method was submitted to the multiple fundamental frequency frame level estimation task of MIREX 2007 [17]. In this evaluation task, there were 28 test files, each of which had a 30-second duration. These files consisted of 20 real recordings, 8 synthesized from RWC samples. The summary results of the first 8 methods in the rank are reported in Table 5. In the evaluation, our method (labeled as team "ZR") was ranked the third in the 16 submitted approaches. However the difference of results between our method and the best method (team "RK") was really minor, whereas our method was approximately 13 times faster than the best method (team "RK"). The algorithm has been implemented as Matlab M-files and MEX-files. The execution time on a 2 GHz Pentium processor is about one third of the time duration of a monaural audio recording.

5. Conclusion and Future Work

In this article, a computationally efficient and robust method has been proposed to estimate pitches in real polyphonic music. Compared to the state-of-the-art approach, the proposed method is conceptually simple and much faster and presents comparable performance. In the method, the pitch estimation process can be separated into three consecutive stages. In order to show how each stage improves the performance, the method was run on the test dataset, and the result in each stage is reported. First, the preliminary estimation aims to find all possible pitch candidates. About 95% of true notes were successfully found. Then the method removes the pitch candidates, which do not have sufficient harmonic components in the energy spectrum. In this stage, the total performance F-measure is improved from 33% to 63%. Finally, possible remaining ambiguities (such as an integer ratio between fundamental frequencies) are partially solved by investigating the spectral irregularity. The final stage increases the F-measure from 63% to 83%.

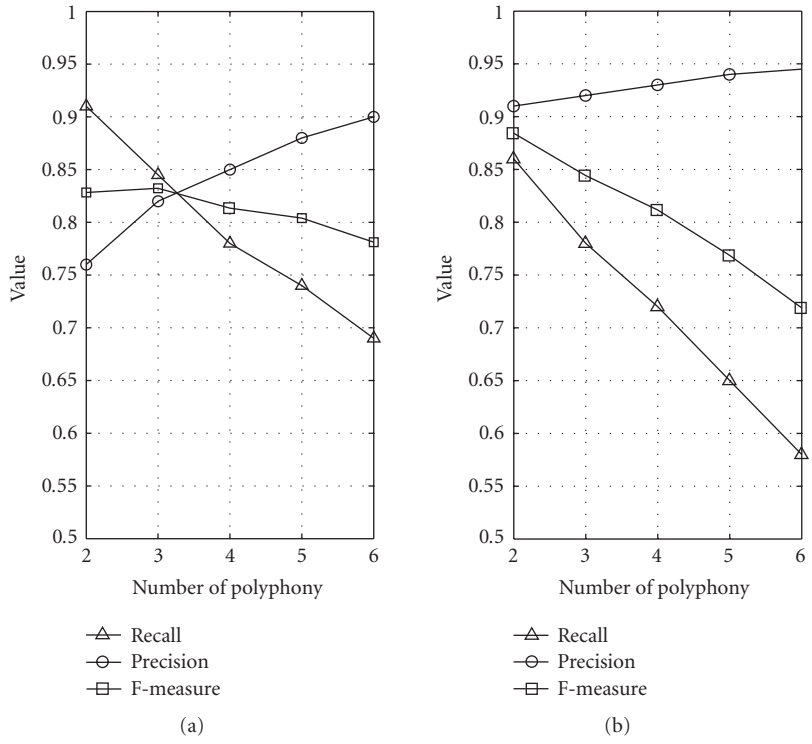


Figure 7: F-Measure, Recall, and Precision results for the proposed method with different parameters. (a) Better Recall, when parameters: $A_1 = 8$, $A_2 = 8$, SI(2) = 10, SI(3) = 10, SI(4) = 5. (b) Better Precision, when parameters: $A_1 = 12$, $A_2 = 12$, SI(2) = 15, SI(3) = 15, SI(4) = 10.

Approximately 30% of all errors are octave errors, and about 18% of all errors are due to confusion between notes with the fundamental frequency ratio of 1/3. These errors are mainly caused by coinciding frequency partials from a lower pitch. This result demonstrates that the coinciding frequency issue is only partially solved by identifying the spectral irregularity. In future work, this issue can be further investigated by combining the method with analysis of temporal features which were not yet exploited. The harmonic components from the same instrument sound source often present similar temporal features, such as a common onset time, amplitude modulation, and frequency modulation. The harmonic relative frequency components with similar temporal features should have a higher probability of representing the same note than those with different temporal features.

For example, a polyphonic note combination may consist of two notes, A3 and A4, where A3 is played by piano and the note A4 is played by violin. It is very difficult to make a polyphonic estimation for this case, because the harmonic components of A4 are completely overlapped by the even harmonic components of A3. However, such a difficult case may be resolved by using temporal features. As shown in Figure 8, the blue lines denote the first four odd harmonic components of A3, and the red/magenta lines denote the first four even harmonic components of the note A3. It can be clearly seen that the energy spectra of the first four even harmonic components have different temporal features than the first four odd harmonic components. This difference

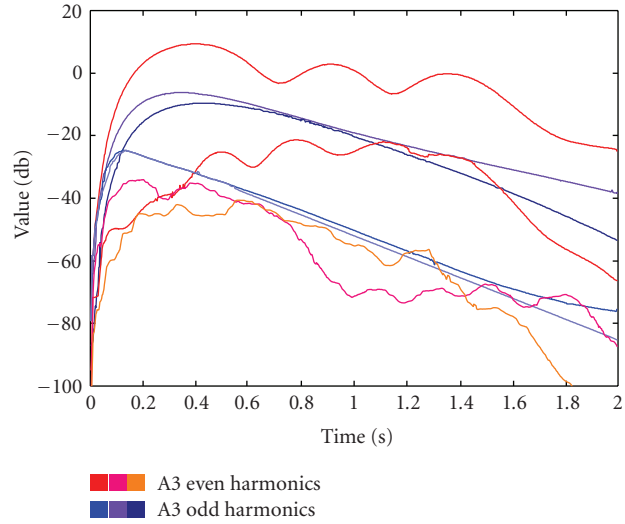


Figure 8: Energy changes of harmonic components of a polyphonic note with two polyphonies A3 and A4.

indicates that the even harmonic components are probably shared with another musical note.

The remaining errors are mainly related to the fact that the timbres of the instruments may differ greatly from each other. Therefore, assumptions concerning the spectral harmonic characteristics are unlikely to be suitable for all instruments. Further improvements to the approach

could be achieved by developing efficient instrument recognition algorithms. The recognition algorithms could first automatically estimate the dominant instrument type for the music signal being analyzed; then the method can use the known spectral harmonic characteristics of that instrument type. In this situation, instrument recognition needs only to be sufficiently accurate as to place the musical signal in the class of an instrument with similar spectral harmonic characteristics.

Acknowledgments

This work was supported in part by the Swiss Commission and Innovation (CTI), under Project 6893.2 (STILE) and by the European Commission Project IST-033902 (EASAIER).

References

[1] H. Thornburg, R. J. Leistikow, and J. Berger, "Melody extraction and musical onset detection via probabilistic models of framewise STFT peak data," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1257–1272, 2007.

[2] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: current directions and future challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668–696, 2008.

[3] P. Bellini, P. Nesi, and G. Zoia, "Symbolic music representation in MPEG," *IEEE Multimedia*, vol. 12, no. 4, pp. 42–49, 2005.

[4] P. Brossier, M. Sandler, and M. Plumbley, "Real time object based coding," in *Proceedings of the 114th AES Convention*, Amsterdam, The Netherlands, 2003, Paper no. 5809.

[5] A. P. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 804–816, 2003.

[6] M. P. Ryynänen and A. Klapuri, "Polyphonic music transcription using note event modeling," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 319–322, October 2005.

[7] M. Marolt, "A connectionist approach to automatic transcription of polyphonic piano music," *IEEE Transactions on Multimedia*, vol. 6, no. 3, pp. 439–449, 2004.

[8] A. Pertusa and J. M. Inesta, "Polyphonic music transcription through dynamic networks and spectral pattern identification," in *Proceedings of the International Conference on Artificial Neural Networks in Pattern Recognition Acoustics*, pp. 19–25, Florence, Italy, 2003.

[9] G. E. Poliner and D. P. W. Ellis, "A discriminative model for polyphonic piano transcription," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, Article ID 48317, 9 pages, 2007.

[10] R. Zhou and G. Zoia, "Polyphonic music analysis by signal processing and support vector machines," in *Proceedings of the 8th International Conference on Digital Audio Effects (DAFX '05)*, pp. 1–9, Madrid, Spain, September 2005.

[11] H. Kameoka, T. Nishimoto, and S. Sagayama, "Separation of harmonic structures based on tied Gaussian mixture model and information criterion for concurrent sounds," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '04)*, vol. 4, Montreal, Canada, 2004.

[12] A. T. Cemgil, H. J. Kappen, and D. Barber, "A generative model for music transcription," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 2, pp. 679–694, 2006.

[13] M. Davy and S. J. Godsill, "Bayesian harmonic models for musical signal analysis," *Bayesian Statistics*, vol. 7, pp. 105–124, 2003.

[14] S. A. Abdallah and M. D. Plumbley, "Unsupervised analysis of polyphonic music by sparse coding," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 179–196, 2006.

[15] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA '03)*, pp. 177–180, New Paltz, NY, USA, 2003.

[16] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*, Springer, New York, NY, USA, 2006.

[17] R. Zhou and J. D. Reiss, "A real-time frame-based multiple pitch estimation method using the resonator time-frequency image," in *Proceedings of the 3rd Music Information Retrieval Evaluation eXchange (MIREX '07)*, Vienna, Austria, September 2007.

[18] R. Zhou, *Feature extraction of musical content for automatic music transcription*, Ph.D. thesis, Swiss Federal Institute of Technology, Lausanne, Switzerland, October 2006, http://library.epfl.ch/en/theses/?nr=3638.

[19] R. Zhou and M. Mattavelli, "A new time-frequency representation for music signal analysis: resonator time-frequency image," in *Proceedings of the 9th International Symposium on Signal Processing and Its Applications (ISSPA '07)*, Sharijah, UAE, February 2007.

[20] J. C. Brown, "Calculation of a constant Q spectral transform," *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[21] J. C. Brown and M. S. Puckette, "An efficient algorithm for the calculation of a constant Q transform," *Journal of the Acoustical Society of America*, vol. 92, no. 5, pp. 2698–2701, 1992.

[22] A. Klaupri, "Multiple fundamental frequency estimation by summing harmonic amplitudes," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR '06)*, pp. 216–221, Victoria, Canada, October 2006.

[23] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: music genre database and musical instrument sound database," in *Proceedings of the International Conference on Music Information Retrieval (ISMIR '03)*, Washington, DC, USA, October 2003.