Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2012, Article ID 302958, 13 pages
doi:10.1155/2012/302958

Research Article

A Combined Mathematical Treatment for a Special Automatic Music Transcription System

Yi Guo and Jiyong Tang

College of Automation, University of Electronic Science and Technology of China (UESTC), Chengdu 611731, China

Correspondence should be addressed to Yi Guo, mathguoyi@sohu.com

Received 5 September 2012; Accepted 21 October 2012

Academic Editor: Xinguang Zhang

This paper presents a combined mathematical treatment for a special automatic music transcription system. This system is specially made for computer-synthesized music. The combined mathematical treatment includes harmonic selection, matrix analysis, and probability analysis method. The algorithm reduces dimension by PCA and selects candidates first by human auditory model and harmonic structures of notes. It changes the multiple-F0 estimation question into a mathematical problem and solves it in a mathematical way. It can be shown in this paper that the experimental results indicate that this method has very good recognition results.

1. Introduction

Music transcription means an act of listening to a piece of music and writing down music score for the piece. The traditional way of making music is just like that: a performer reading a score, playing an instrument, and thus producing music. Transcription of polyphonic music is a reverse process: an acoustical waveform is converted into a parametric representation (such as MIDI), where notes, their pitches, starting times, and durations are extracted from the signal [1]. Automatic music transcription means converting acoustic music signals to musical scores automatically by computer analysis technology. Automatic music transcription greatly reduces the manual labor and time and it becomes a key technology in music signal processing [2, 3]. Automatic music transcription can be widely used for content-based music retrieval, low-rate coding, automatic musical accompaniment system, and so on.

Fundamental frequency (F0) is an essential descriptor of harmonic sound signals such as speech and music, and it determines the pitch of a music note. Single-F0 estimation algorithms assume that there is at most one musical of which the F0 is to be

extracted. Although single-F0 estimation algorithms have been considerably developed, their applications to music signals are somehow limited because most music signals contain more than one concurrent harmonic source. Multiple-F0 estimation algorithms are thus required for the general case and it needs to estimate each of the sources which play together, which is one of the most important parts of the automatic music transcription [4, 5].

The most important contribution of this paper is solving the multiple-F0 estimation in a mathematical analysis method. The signals we need to process are wave forms in time domain $x[m]$. At first we need to change it into frequency domain by

$$X(n,k) = \sum_{m=-\infty}^{\infty} x[m]w[n-m] \cdot e^{-j(2\pi/N)km} \tag{1.1}$$

or cepstrum domain by

$$\hat{x}_c[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} \log X(\omega) e^{j\omega n} d\omega. \tag{1.2}$$

And then we change the multiple-F0 estimation question into matrix linear equations. It is an ill-posed equation, so we can solve it using a truncation completely least squares method. Based on this, we can use a priori probability to filer the result to improve the correct rate.

This paper is organized as follows. At first it introduces the research background and situation of study of automatic music transcription and multiple-F0 estimation. And then it describs our algorithm and experimental stages in detail. We focused on the effect of harmonic structure in primary selection progress and the math model of multiple-F0 estimation. And it is followed by an experimental results display, and finally there are the conclusion and perspective and acknowledgment.

2. Automatic Music Transcription System

The system architecture of automatic music transcription system proposed in this paper can be found in another paper by the same author in [6]. In this system, input music signal passes through a spectral analysis module first and then performs a multiple-F0 estimation progress after preprocess and windowed DFT. The following multiple-F0 estimation algorithm is performed for each music frame to predict one or more notes that appear in this time slice. In the end, this algorithm will detect the starting time and the ending time of each note. Through these steps, a music signal can be transcribed into note event sequence, which can be described in MIDI form or in music score directly.

The key process of this system is multiple-F0 estimation and it is also the main contribution of this paper. The multiple-F0 estimation algorithm has two main stages: training and identify. It contains three main steps in training process: preprocess, PCA, and signal normalization, and six main steps: enframe, preprocess, PCA, signal normalization, frequency primary election, and multiple-F0 estimation. The steps which have the same name did the same things.

(i) *Preprocessing*: it is the first step in training process. It removes the silent part in the front and back of the notes section first and then through a windowed DFT transformation.

(ii) *Normalization*: signal normalization means converting the signal after preprocessing into equivalent one with mean 0 and variance 1, and it is used to facilitate the follow-up treatment. The signal is to be separated into two parts after normalization. One will go into PCA step, and the other will be used to analyze the harmonic structure.

(iii) *PCA*: PCA means principal component analysis. It makes all the notes together by a common preextraction of principal components. That is to map all the notes from a higher-dimensional space to a relatively low-dimensional space coordinate system. While PCA is used to train date here, we can define a PCA score (I_{pca}), which means that we only retain the components whose sum of ratios is greater than the score.

(iv) *Calculate* H_n: in this step we need to decide the number of harmonics for each note and the amplitude of spectrum for each harmonic. In fact, each note has clear harmonic characteristics. It means that there will be clear impulse near integer multiples of the fundamental frequency. The same notes played by different instruments have harmonics at almost the same location but the margins of each note may be different. Using this feature, we can roughly estimate a certain audio whether contains a note. This feature can be used to select candidates for the fundamental frequency to facilitate follow-up treatment.

(v) *Enframe*: the length of the music to be detected is variable; thus, an enframe processing is needed in order to obtain a more accurate detection and estimation of the notes. Each frame will have a detect result, which shows the notes that the frame included.

(vi) *Frequency primary selection*: in order to reduce the computation and improve the estimation accuracy of data, before the multiple fundamental frequency estimation step, a frequency primary selection step needs to be produced. In this step, all candidate notes are estimated. After this, it only needs to select the final notes from these candidate ones instead of from all the notes, which can greatly benefit the speed and estimation accuracy. The detail information will be described in the next section.

(vii) *Multiple-F0 estimation*: the multiple fundamental frequency estimation here means to estimate the consisting of notes from candidates based on the data after the above processing. This is a core part of the whole system and we can use some mathematical methods to solve this problem. The detailed information will be described in the next section.

3. Using Harmonic Structure

3.1. Essential Knowledge

The timbre refers to the feeling features of the sound, such as frequency, loudness, and the duration. The timbre has an identity function, and we can distinguish two kinds of sound with the same pitch and intensity but coming from different musical instruments according to

differences in sound timbre. The timbre is a multidimensional object. In addition to frequency, loudness, and duration, it also contains amplitude envelope and spectral envelope.

Spectral envelope is an important parameter to describe the sound in frequency domain and it is constituted with the amplitude of all harmonics. We can describe the spectral envelope by *harmonic structure,* labeled as H_n:

$$H_n = \{a_1, a_2, \ldots, a_n\}, \tag{3.1}$$

where a_k is the amplitude of the kth harmonic and n is the number of harmonics.

In accordance with western musical convention, note events are ordered using a logarithmic scale [7]. For linear frequency to MIDI conversion, the following expression can be used:

$$n = 69 + 12\log_2\left(\frac{f_0}{440}\right). \tag{3.2}$$

In music notation, each note is named with one of the following symbols:

Do Re Mi Fa Sol La Ci.

A sequence of notes fromDo to Ci is called an *octave*. In a given octave, the fundamental frequency of each note is an integer multiple of fundamental frequency of namesake from previous octaves. Since the harmonics of each note are also integer multiples of fundamental frequency, these harmonics represent namesake notes of it in next octaves [8]. For example, the fundamental frequency of La in octave 4 (i.e., La4) is 220 (Hz). So, the frequency of its second harmonic is 440 (Hz) that is equal to the fundamental frequency of La in octave 5 (i.e., La5). This is the frequency overlapping problem which we mentioned above and it is also a key problem when performing the iterative deletion.

Spectral envelopes of different instruments have obvious differences, while the same kind of instrument has similar spectral envelope. Music played by the same instrument has a high degree of similarity and a stable harmonic structure. However, the computer-synthesized music is established by the same soft wavetable. So we can believe that the harmonic structure of computer music synthesized by the similar instrument is almost the same. This paper assumes that the harmonic structure is unchanged when the F0 of one note is changing weakly in a semitone.

3.2. Usage of Harmonic Structure

Based on the above characteristics, the information of harmonic structure can be used to improve the multiple fundamental frequency estimation algorithms. In our algorithm, the information of harmonic structure is mainly used in two places: in the training stage, obtaining the information of harmonic structure of each note played by different instrument, and in the identification stage, using the harmonic structure matching rate to determine the candidate fundamental frequency, to increase the accuracy of following multiple-F0 estimation.

In the training stage, the parameters of each note can be extracted from the training materials, each training material contains only one note, and each note can be trained by 100 materials. We analyze the spectral envelope of each material and calculate the

harmonic structure H_n. We set a training threshold *th1*. If the normalized spectral envelope is larger than *th1*, the corresponding element in harmonic structure was set as the harmonic amplitude, otherwise was set to 0.

For an in-harmonic instrument, partial frequencies H_n can further deviate from this frequency range. As a consequence, these partials are missed or assigned erroneously to other partials by our system [9]. However, this situation only occurs for strongly in-harmonic instruments and at high values of parameter. Partial frequencies for an in-harmonic instrument can be expressed by $kf_{0_n}\sqrt{1+\beta k^2}$, where β is the inharmonicity coefficient. Typical values for coefficient β range from 10^{-4} to 10^{-3} in piano bass notes [10]. Partial frequencies exceed the selected frequency range from $k = 25$ for $\beta = 10^{-4}$, $k = 11$ for $\beta = 5 * 10^{-4}$, or $k = 8$ for $\beta = 10^{-3}$. The fundamental frequency $f0_n$ of the MIDI note n is calculated as

$$f_{0_n} = 440 \cdot 2^{(n-69)/12}. \tag{3.3}$$

The analysis above indicates a problem that harmonics of one note may not just be at the integer multiple position of the fundamental frequency because of the inharmonicity. According to this, when we calculate H_n, we use a'_k instead of a_k, where a'_k is the maximum partial amplitude found in the frequency range $[kf_0 \cdot 2^{-(1/24)}, kf_0 \cdot 2^{1/24}]$. When there are no partials in the frequency range, a'_k is set to 0.

In the identification stage, the harmonic structure matching rate can be used to determine the candidate fundamental frequency.

When playing a note, the spectrum will contain its entire harmonics. If the $f0$ is located at the same semitone as the note n, it will have the same harmonic structure as H_n, in which the harmonic matching ratio of the kth harmonic can be defined as the following:

$$r(f_0, k) = \frac{\left|Y(f_k)\right|}{a_k}, \tag{3.4}$$

where $Y(f_k)$ is the STFT of music and a_k is the k-th element in H_n.

In order to remove the influence of harmonic overlap in the polyphony music, the minimal harmonic matching ratio of all harmonic components is chosen as the harmonic matching ratio of this note, which just as shown in

$$r(f_0) = \min\left\{r(f_0, k)|_{k=1}^{n}\right\}. \tag{3.5}$$

And, then, in order to better describe the situation of playing the notes, the strength of notes can be defined as follows:

$$S(f_0) = \sum_{k=1}^{n} r(f_0)a_k. \tag{3.6}$$

The larger S shows the larger probability of including the note which has a fundamental frequency of f_0. If S is larger than the threshold *th2*, the note whose fundamental frequency is f_0 will be chosen as a candidate fundamental frequency. When all the candidates are chosen, the frequency primary selection step is completed.

4. Other Mathematical Treatments in This Algorithm

4.1. Principal Component Analysis

In training stage and identification stage, PCA is an important step to reduce dimensions, so we introduce the calculate process in detail here.

Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has as high a variance as possible (i.e., accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (uncorrelated with) the preceding components. Principal components are guaranteed to be independent only if the data set is jointly normally distributed. PCA is sensitive to the relative scaling of the original variables.

PCA is mathematically defined as an orthogonal linear transformation that transforms the data to a new coordinate system such that the greatest variance by any projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on. PCA is a mature tool which has classic calculation process. The implementation method of PCA can be found in [10].

The calculation steps of PCA are as follows.

(1) Calculate the correlation coefficient matrix:

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1p} \\ r_{21} & r_{22} & \cdots & r_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ r_{p1} & r_{p2} & \cdots & r_{pp} \end{bmatrix}, \tag{4.1}$$

where r_{ij} $(i, j = 1, 2, \ldots, p)$ is correlation coefficient of original variables x_i and x_j. The calculation formula is

$$r_{ij} = \frac{\sum_{k=1}^{n} (x_{ki} - \overline{x}_i)(x_{kj} - \overline{x}_j)}{\sqrt{\sum_{k=1}^{n} (x_{ki} - \overline{x}_i)^2 \sum_{k=1}^{n} (x_{kj} - \overline{x}_j)^2}}. \tag{4.2}$$

R is a real symmetric matrix, that is, $r_{ij} = r_{ji}$, so we only need to calculate the upper triangular elements or lower triangular elements.

(2) Calculate the eigenvalues and eigenvectors: first, solve the characteristic equation $|\lambda I - R| = 0$. Usually the Jacobi method is used to find out the eigenvalues λ_i $(i = 1, 2, \ldots, p)$, and let them be arranged in order of size, that is, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$, and find out the eigenvectors e_i $(i = 1, 2, \ldots, p)$ corresponding to the eigenvalues λ_i, respectively. Here $\|e_i\| = 1$, that is, $\sum_{j=1}^{p} e_{ij}^2 = 1$, where e_{ij} represent the j component of vector e_i.

(3) Calculate the contribution of the main components and cumulative contribution.

The contribution of main components z_i is

$$\frac{\lambda_i}{\sum_{k=1}^{p}\lambda_k} \quad (i = 1, 2, \ldots, p). \tag{4.3}$$

The cumulative contribution is

$$\frac{\sum_{k=1}^{i}\lambda_k}{\sum_{k=1}^{p}\lambda_k} \quad (i = 1, 2, \ldots, p). \tag{4.4}$$

(4) Calculate the weight of the main components:

$$l_{ij} = p(z_i, x_j) = \sqrt{\lambda_i} e_{ij} \quad (i, j = 1, 2, \ldots, p). \tag{4.5}$$

After it, we can calculate the score of each main components:

$$Z = \begin{bmatrix} z_{11} & z_{12} & \cdots & z_{1m} \\ z_{21} & z_{22} & \cdots & z_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ z_{n1} & z_{n2} & \cdots & z_{nm} \end{bmatrix}, \tag{4.6}$$

where

$$Z_m = l_{m1}x_1 + l_{m2}x_2 + \cdots + l_{mp}x_p. \tag{4.7}$$

4.2. Mathematical Representation in Multiple-F0 Estimation

Multiple fundamental frequency (multiple-F0) estimation is used to estimate multiple notes which are sounded at the same time in music. It is the core technology and the main difficulties in automatic music transcription system. Most actual music are polyphony ones, which means there are more than one notes at the same time. The situation is much more complex when the spectrums of the notes are added and it makes the multiple-F0 estimation more difficult. To improve the performance of the system, a good multiple-F0 estimation algorithm needs to consider many factors, such as inharmonicity factor, frequency missing, harmonic missing, frequency overlap, and frequency error [6]. The algorithm proposed in this paper establishes a mathematical model for multiple fundamental frequency estimation. To obtain the result of multiple-F0 estimation is equivalent to solve this math problem. In this part, we introduce the main idea of this algorithm.

In the music division rules, there is sufficient frequency separation between notes, and the masking effect is relatively weak among pure tones. Thus, the masking effect can be ignored while the requirement of computational accuracy is not very highly. Ignoring the masking effect, the loudness can be added with a linear characteristic. It means that the loudness of the sound mixing multiple notes is the sum of the loudness of each note.

From the above analysis we can see that the audio to be estimated is the linear combination of standard notes. Suppose that the number of notes is n, each note includes m features after PCA process, and then the training set S is an $n \times m$ matrix. The audio to be estimated after preprocessing, PCA, and normalization is recorded as Y, then Y is a $1 \times m$ vector, and m is the characteristic dimension. If the energy lost in PCA is recorded as Sr, we get the following:

$$X \cdot S + Sr = Y, \tag{4.8}$$

where S is the result set of the training process and it is an $n \times m$ matrix. X is a $1 \times n$ vector, $X = [x_1, x_1, \ldots, x_n]$, and it presents the combination coefficient corresponding to each note. If we regard the Sr as an error part, we can ignore it and get the next formula:

$$X \cdot S \approx Y. \tag{4.9}$$

The main task of multiple-F0 estimation is to estimate the best X to make XS as more close as Y. This is a problem of computing extreme and it can be resolved by the knowledge of linear algebra.

Let

$$f(x) = XS - Y = \begin{pmatrix} x_1 s_{11} + x_2 s_{21} + \cdots + x_n s_{n1} - y_1 \\ \vdots \\ x_1 s_{1m} + x_2 s_{2m} + \cdots + x_n s_{nm} - y_m \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{pmatrix},$$
$$g(x) = f^T(x) f(x) = f_1^2 + f_2^2 + \cdots + f_m^2. \tag{4.10}$$

The problem is to obtain a best X to make $f(x)$ get the maximum value. It can be proved that $g(x)$ and $f(x)$ get the maximum value at the same place. So the next problem is to calculate X to make $g(x)$ gets the maximum value. According to the knowledge of higher mathematics we can get

$$\begin{aligned} \frac{\partial g(x)}{\partial x_1} &= 2f_1 s_{11} + 2f_2 s_{12} + \cdots + 2f_m s_{1m} = 0 \\ \frac{\partial g(x)}{\partial x_2} &= 2f_1 s_{21} + 2f_2 s_{22} + \cdots + 2f_m s_{2m} = 0 \\ &\vdots \\ \frac{\partial g(x)}{\partial x_n} &= 2f_1 s_{n1} + 2f_2 s_{n2} + \cdots + 2f_m s_{nm} = 0. \end{aligned} \tag{4.11}$$

Equations (4.12) can be obtained after the decomposition (4.11):

$$\begin{aligned}
(x_1, x_2, \ldots, x_n) \cdot \begin{pmatrix} S_1 & S_2 & \cdots & S_n \end{pmatrix}^T \cdot S_1^T - Y \cdot S_1^T &= 0 \\
(x_1, x_2, \ldots, x_n) \cdot \begin{pmatrix} S_1 & S_2 & \cdots & S_n \end{pmatrix}^T \cdot S_2^T - Y \cdot S_2^T &= 0 \\
&\vdots \\
(x_1, x_2, \ldots, x_n) \cdot \begin{pmatrix} S_1 & S_2 & \cdots & S_n \end{pmatrix}^T \cdot S_n^T - Y \cdot S_n^T &= 0.
\end{aligned} \tag{4.12}$$

Simplification of (4.12) can get

$$\begin{aligned}
X \cdot S \cdot S_1^T - Y \cdot S_1^T &= 0 \\
X \cdot S \cdot S_2^T - Y \cdot S_2^T &= 0 \\
&\vdots \\
X \cdot S \cdot S_n^T - Y \cdot S_n^T &= 0.
\end{aligned} \tag{4.13}$$

Write (4.13) into the matrix form:

$$X \cdot S \cdot S^T = Y \cdot S^T, \tag{4.14}$$

where S and Y are known, so the X can be obtained by knowledge of linear algebra.

It should be noted that if there are some negative coefficients in X, it means that the audio will not contain the corresponding note and we should calculate this again without this note. It is repeated until all components of X are positive or repeated t times (to avoid an infinite loop, you need to manually set up a number of cycles t). A threshold x_0 can be set. If $x_i > x_0$, it indicats that the note i is included in this frame. In this way, we can estimate all the notes contained in this frame.

We can solve the function (4.9) in another way. In function (4.9), S is an $n \times m$ matrix. In general case, $m >> n$, so function (4.9) is an ill-posed matrix equation. A truncation completely least squares method can be used to get the solution of this function because this method is not sensitive to error. The step-by-step process is given as follows.

At first, the singular value decomposition of augmented matrix $[S, Y]$ needs to calculated:

$$[S, Y] = P^{\lambda} Q^T = \sum_{i=1}^{n+1} p_i \lambda_i q_i, \quad (\lambda_1 > \lambda_2 > \cdots > \lambda_{n+1}). \tag{4.15}$$

And then, select a truncation strategy $k \leq \min\{n, \operatorname{rank}(S, Y)\}$ and cut off the smaller singular value and let

$$Q' = \left(q_{n+1,k+1}, q_{n+1,k+2}, \ldots, q_{n+1,n+1}\right) \neq 0. \tag{4.16}$$

Third, let $n' = n - k + 1$, and make Q become a partitioned matrix:

$$Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix}, \tag{4.17}$$

where $Q_{11} \in C^{n\times k}, Q_{12} \in C^{n\times n'}, Q_{21} \in C^{1\times k}, Q_{22} \in C^{1\times n'}$.

At last we can get the solution:

$$X = -Q_{12}Q_{22}^{*} = -Q_{12}Q_{22}^{T}\|Q_{22}\|_2^{-2}, \tag{4.18}$$

where $Q_{22}^{*} = Q_{22}^{T}\|Q_{22}\|_2^{-2}$ is a generalized inverse matrix.

After we get X, the negative and smaller value components were deleted and the rest ones are the mixing coefficients of the corresponding notes.

We can use both the ways to solve function (4.9) and make them mutual authentication to make an improvement.

When calculating H_n in training stage we consider with the information of frequency missing, harmonic missing, and while in frequency primary selection stage we consider with inharmonicity factor and frequency error. Function (4.9) can indicate the information of frequency overlap. As a result, our algorithm has a better result than others.

4.3. Filer Based on the Music Category and Priori Probability

This content is one of the most contribution of this paper.

Just as we know, there are kinds of music actually, such as new age, rhythm and blues, rap, rock, and jazz. Different music has different rhythm and melody and many other things. From the audio features, they have different low short time energy ratio and nonzero pitch ratio and other characteristics. The probability of occurrence of each note is different for different types of music. The probability distribution of the same note in different types of music is also different. We can use this to perform a filer based on the music category for the previous experimental results.

It is needed to point out that to classify the music is not the research contents of this paper. We assume that all the category of each music belongs to are known.

We only consider the most popular notes (about 72) in actual songs. Let event A_{ix} = {note x is included in the ith frame}, $A_{(i+1)y}$ = {note y is included in the $(i+1)$ th frame}. It is obvious that the number of x and y is limited, and the number of event A_{ix} or $A_{(i+1)y}$ is also limited. Let event C_{xy} = {while current frame contains note x, the next frame contains note y}. According to the definition of conditional probability we can get that $P(C_{xy})$=$P(\mathrm{A}_{(i+1)y} \mid A_{ix})$. For different kind of music, the conditional probability $P(\mathrm{A}_{(i+1)y} \mid A_{ix})$ is not the same for the same note x or y. If we know the kind of the music, we can filter the following frames by the corresponding conditional probability. In this way, we can remove the notes with small probability and validate the notes in current frame through the relevant posterior probability.

In the training process, we can get the $P(A_{ix})$ through experiment by $P(A_{ix})$ = {the number of frames including note x}/{the number of total frames}. We can also get $P(C_{xy})$ = $\mathrm{P}(A_{(i+1)y} \mid A_{ix})$ = {the number of the next frame contains note y while current frame contains note x}/{the number of total frames}.

In the identification process we can calculate the following data.

(i) Get the $P(A_{(i+1)y})$ by querying training result set directly.

(ii) Get the $P(A_{(i+1)y} \mid A_{ix})$ by querying training result set directly.

(iii) Calculate the $P(A_{(i+1)y})$ by the whole probability formula:

$$P\left(A_{(i+\,1)y}\right) = \sum_{x=\,1}^{72} P(A_{ix}) P\left(A_{\,(i+\,1)y} \mid A_{ix}\right). \tag{4.19}$$

(iv) Calculate the $P(A_{ix} \mid A_{(i+1)y})$ by the Bayesian formula:

$$P\left(A_{ix} \mid A_{\,(i+\,1\,)y}\right) = \frac{P(A_{ix}) P\left(A_{(i+\,1)y} \mid A_{ix}\right)}{\sum_{x=1}^{72} P(A_{ix}) P\left(A_{(i+\,1)y} \mid A_{ix}\right)}. \tag{4.20}$$

Analysis of the above four probabilities can be seen. The first probability is a priori probability without any restrictions. The second one has been considered with the relation of the adjacent frames. The third one is calculated with the whole probability formula and the forth is a posterior probability calculated by the Bayesian formula. The first three probabilities are priori probabilities used to directly determine the probability of the current frame that contained a note. The fourth one is a posterior probability used to determine the probability of the previous frame that contained one note while the current frame contained another one. Each probability can be used to get the result alone but we put the four probabilities together to make the result more accurate.

Setting a probability threshold P, we consider the event $A_{(i+1)y}$ as true only when all of the probabilities mentioned above are larger than P.

5. Experimental Methods and Results

According to the algorithm described in the previous section, we did simulation in Matlab. The experiment parameters in each step are listed as follows.

(i) All audio signals are decomposed at a sampling frequency of $fs = 22.5$ kHz and 16 bit.

(ii) Hamming windowing.

(iii) FFT length $L = 1024$.

(iv) FIR High Pass Filter $H(z) = 1 - \alpha z^{-1}$, $\alpha = 0.98$, used to pre-emphasis.

(v) Frame length is 100 ms.

(vi) 72 notes played by piano are used to train, which from C3 (note 36) to B8 (note 107), and each note we get 100 wave-form data to train.

(vii) The number of songs used to training is 100 for each category, and do statistics and get $P(A_{(i+1)y})$ and $P(A_{(i+1)y} \mid A_{ix})$ base on them.

(viii) The PCA score $I_{\text{pca}} = 80\%$.

(ix) The threshold of spectral envelope in frequency primary selection step $th1 = 0.1$.

(x) The threshold of intensity coefficient: $th2 = 1.5$.

(xi) The probability threshold of filter step is $P = 0.6$.

(xii) We obtain note error rate (NRE) as follows:

$$E = \frac{\text{FP} + \text{FN}}{\text{Notes}}, \tag{5.1}$$

where FP (false positives) is the number of inactive note frames transcribed as active and FN (false negatives) is the number of active note frames transcribed as inactive.

In order to fully test the performance of the algorithm presented in this paper, we made a lot of experiments with different types and styles of music, including classic music, jazz, nursery rhyme, soothing music, fast-paced music, and the music which has a lot of fundamental frequency overlapping and the music which has just a little fundamental frequency overlapping. The size of a frame has a little influence on the recognition results of different types of music. Analysing of the results made us found that our algorithm was not sensitive about changes in the rhythm of music. Both fast rhythm music and slow rhythm have similar identification results, but it is sensitive with the change rules of rhythm. The music with more regular rhythm changes can get better recognition results.

A number of experiments show that our algorithm has an average NER about 13%. That is a very high accuracy rate compared with some state-of-the-art algorithms described in [11, 12]. The transcription result has no obvious impact on understanding this music for us.

6. Conclusion and Perspective

This paper presents a compositional pattern recognition and machine learning methods for computer-synthesized music specifically to multiple-F0 estimation and builds an efficient automatic music transcription system. This method also considers the human auditory model and the harmonic structure of music notes and improves the algorithm based on this. Although using the harmonic match and iterative delete alone can finish the multiple-F0 estimation, or deleting the frequency primary selection step in identification stage and establishing the mathematical model and solving it can also finish the task, combining the two methods can improve the performance of the algorithm and the recognition result. The usage of Bayesian estimation and a priori probability improved the performance a lot. Furthermore, for all this, something which needs to be improved must be pointed out. First, we can improve the definition of harmonic matching to get a more accurate result. And then, for FFT, we can consider using changeable FFT length, because the notes in the low-frequency region are rich in harmonic components while the notes in high-frequency region have relatively simple harmonic components. In low-frequency region, a higher frequency resolution is needed while in high-frequency region the situation is just the opposite. Third, the truncation threshold *th1* or *th2* in our algorithm can be defined in a changeable way to get a more accurate result. In addition, because ensemble music includes many instruments, we need to train with many kinds of notes and it has a large amount of computation. If we can develop an unsupervised method and the prior information about the instrument is not obtained from files of the same instrument in different music databases but is directly obtained from the music file to be analyzed, that will be improve efficiency a lot.

Acknowledgments

Thanks are due to Professor Hou Xiaorong for meticulous guidance all the time and to Professor Tang Jiyong for financial support of the authors' research.

References

[1] 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.

[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. Brossiser, M. Sandler, and M. Plumbley, "Real time object based coding," in *Proceedings of the AES 114th Convention*, 2003.

[4] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*, Springer, New York, NY, USA, 2nd edition, 1998.

[5] J. Harold and H. A. Conklin, "Generation of partials due to nonlinear mixing in a stringed instrument," *Journal of the Acoustical Society of America*, vol. 105, no. 1, pp. 536–545, 1999.

[6] Y. I. Guo, H. B. Xu, and J. Y. Tang, "Review of automatic music transcription system," *Application Research of Computers*, vol. 28, no. 4, pp. 1201–1205, 2011.

[7] J. J. Carabias-Orti, P. Vera-Candeas, F. J. Cañadas-Quesada, and N. Ruiz-Reyes, "Music scene-adaptive harmonic dictionary for unsupervised note-event detection," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 3, pp. 473–486, 2010.

[8] F. Hekland, *Automatic music transcription using autoregressive frequency estimation [M.S. dissertation]*, Norwegian University of Science and Technology, Trondheim, Norway, 2001.

[9] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*, Springer, New York, NY, USA, 2nd edition, 1998.

[10] H. C. Kim, D. Kim, and S. Y. Bang, "A PCA mixture model with an efficient model selection method," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '01)*, pp. 430–435, July 2001.

[11] J. Yin, T. Sim, Y. Wang, and A. Shenoy, "Music transcription using an instrument model," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '10)*, pp. III217–III220, Philadelphia, Pa, USA, 2010.

[12] C. Yeh, A. Roebel, and X. Rodet, "Multiple fundamental frequency estimation and polyphony inference of polyphonic music signals," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1116–1126, 2010.