Predominant Vocal Pitch Detection in Polyphonic Music

Xi Shao[#], Changsheng Xu[#], Mohan S Kankanhalli**

[#]Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613
{shaoxi,xucs}@i2r.a-star.edu.sg
*School of Computing, National University of Singapore, Singapore 117543
mohan@comp.nus.edu.sg

ABSTRACT

We present a novel method for predominant vocal pitch detection in two-channel polyphonic music. The proposed method contains two stages. In the first stage, we apply the Frequency Domain Independent Component Analysis (FD-ICA) for the two-channel polyphonic music to separate the vocal content from the background music. Considering the vocal singing voice and background music are two heterogeneous signals, we employ a statistical learning based method to solve the permutation inconsistency problem in FD-ICA. In the second stage, a noise insensitive vocal pitch detection method is proposed, which is robust to noise and errors introduced by the separation process in the first stage. The proposed method has been tested on the two-channel polyphonic music signals, and experimental results show promising performance.

1. INTRODUCTION

A reliable algorithm for predominant vocal pitch tracking in polyphonic music is critical for many auditory processing tasks such as polyphonic music lyrics transcription and music retrieval. However, the design of such algorithm is difficult. The first difficulty is to separate the vocal content from the background music which interferes with vocal content both in time domain and frequency domain. Traditional Independent Component Analysis (ICA) [1] algorithm is not applicable as it assumes the independent sources are mixed instantaneously, while common polyphonic music has two channels (mixtures) which generally are convolutions of the two sources (singing voice and background music). In [2][3], time domain algorithms for convolved mixture separation were proposed using the maximum entropy cost function. These time domain algorithms work well for small length mixing filters, but when it comes to real time implementations with realistically long filters, they may be unrealizable because of lack of computational efficiency. In addition, updating one coefficient for a particular filter will account for the already updated preceding filter coefficients, which prevents the convergence to the optimal filter coefficients. Therefore, it is intuitive to move from the time domain solution to the frequency domain as the convolution in the time domain is multiplication in the frequency domain and apply ICA methods for instantaneous mixtures in each frequency bin. In this way, the unmixing matrix in each frequency bin is independent and will not interact with each other. However, since we obtain the unmixing matrix in each frequency bin independently and arbitrary permutation of unmixing matrix in certain frequency bin will lead to the same value for maximum entropy cost function. This seems to be a serious problem as only consistent permutations for every frequency bins will correctly reconstruct the sources. This problem is called *permutation inconsistency* problem [4] in FD-ICA. Some channel-based frequency coupling methods [4][5] were proposed to solve this problem by placing smoothness constraint across the frequency bin. However, such constraints reduce the available degrees of freedom to reconstruct the sources. On the other hand, alternative approaches called sources based frequency coupling were proposed in [6] and [7]. They tried to solve the problem by exploiting the relationship between the reconstructed sources at a frequency bin and the original sources in the time domain. However, the basic assumption that one source is louder at certain time slot may be valid for convolutive mixtures of speech signals, but may not always valid for the mixtures of singing voice and background music.

The second difficulty to deal with vocal pitch tracking in polyphonic music is to detect the predominant vocal pitch in the significant broadband interference noise introduced from the separation step. Most current pitch detection methods are limited to monophonic audio signals [8] or single pitch detection in modest noise [9]. Although some algorithms for predominant fundamental frequency tracking have been investigated, for example, Goto [10] employed a Maximum A Posteriori probability (MAP) estimation to estimate the relative dominance of every fundamental frequency and the shape of harmonic structure tone models, but the performance on tracking predominant vocal pitch mixed with significant broadband noise interference is not clear.

In this paper, we present a novel predominant vocal pitch detection method for polyphonic music. Compared to the existing work, our contribution includes: 1) proposing a new solution to solve the permutation inconsistency problem in FD-ICA using a statistical learning based method, and 2) proposing a noise insensitive pitch detection algorithm to detect the dominant pitch from the separated singing voice.

The rest of the paper is organized as follows. In section 2, we employ the FD-ICA algorithm to separate the vocal content from the background music and propose a statistical learning based approach to solve the permutation inconsistency problem in FD-ICA. Noise insensitive pitch detection algorithm for our specific problem is described in section 3. Experiment on testing the performance of our proposed method is presented in section 4. We conclude the paper with future work in section 5.

2. VOCAL CONTENT SEPARATION

According to [11], the music industries produce their music CDs in basically two stages. First, sound from each individual instrument is recorded in an acoustically inert studio on a single track of a multi-track tape recorder. Then, the signals from each track are manipulated by the sound engineer to add special audio effects and are combined in a mix-down system to finally generate the stereo recording on a two-track recorder. The audio effects are artificially generated using digital signal processing techniques and these digital signal processing techniques can be considered as direct filter and cross filter placed between the sources and output channels. Therefore, the generation of musical programs can be modeled as the Figure 1 (a) and the mixture process can be modeled by following equation:

$$x_1(n) = A_{11}(n) * s_1(n) + A_{21}(n) * s_2(n) \quad (1\text{-a})$$

$$x_2(n) = A_{12}(n) * s_1(n) + A_{22}(n) * s_2(n) \quad (1\text{-b})$$

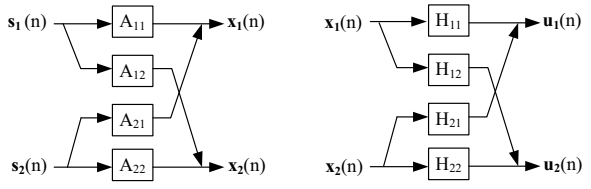
where $\boldsymbol{x}_1$ (n) and $\boldsymbol{x}_2$(n) represent two channels of the polyphonic music respectively, $\boldsymbol{s}_1$(n) and $\boldsymbol{s}_2$(n) represent two sources respectively. A_{11} and A_{22} denote the P points direct filter between sources and channels, and A_{12} and A_{21} denote the P points cross filter between sources and channels. Then the basic problem can be described as following:

Given the observed channels $\boldsymbol{x}_1$ (n) and $\boldsymbol{x}_2$(n), we expect to find a filter matrix $\boldsymbol{H}$ to separate the independent sources $\boldsymbol{s}_1$(n) and $\boldsymbol{s}_2$(n) from the observed mixtures $\boldsymbol{x}_1$ (n) and $\boldsymbol{x}_2$(n). The unmixing process can be modeled by following equation:

$$u_1(n) = H_{11}(n) * x_1(n) + H_{21}(n) * x_2(n) \quad (2\text{-a})$$

$$u_2(n) = H_{12}(n) * x_1(n) + H_{22}(n) * x_2(n) \quad (2\text{-b})$$

Our goal is to obtain the separated sources $\boldsymbol{u}_1$(n) and $\boldsymbol{u}_2$(n) to approximate the original sources $\boldsymbol{s}_1$(n) and $\boldsymbol{s}_2$(n) as close as possible. The unmixing process can be illustrated in Figure 1(b).



(a) Mixing process (b) unmixing process

Figure 1. The convolutive source separation problem

As mentioned in the previous section, the time domain algorithms for separating the vocal content lack of computational efficiency, and it is intuitive to move from time domain solution to the frequency domain, since the time-domain convolutive mixture can be transformed to an instantaneous mixture in the frequency domain by computing its Q-points Short-Time Fourier Transform (STFT):

$$\underline{\boldsymbol{X}}(f,t_s)=\boldsymbol{A}(f)\underline{\boldsymbol{S}}(f,t_s) \qquad f=1,2,\ldots,Q \quad (3)$$

where t_s is the block index, $\underline{\boldsymbol{X}}(f,t_s)=(\boldsymbol{X}_1(f,t_s), \boldsymbol{X}_2(f,t_s))^T$ represents the Short-Time Fourier Transform of two observed channels and $\underline{\boldsymbol{S}}(f,t_s)=(\boldsymbol{S}_1(f,t_s), \boldsymbol{S}_2(f,t_s))^T$ denotes the STFT of two independent sources. $\boldsymbol{A}(f)$ denotes a 2×2 instantaneous complex matrix at the frequency f. Then the problem can be defined as the estimation of an unmixing matrix $\boldsymbol{H}_f \approx \boldsymbol{A}^{-1}(f)$ for each frequency bin. This unmixing matrix $\boldsymbol{H}_f$ can be obtained by extending the real value blind source separation approach for instantaneous mixture [1] to the complex domain. The estimation process can be considered as to obtain a Maximum Likelihood solution separately for each frequency bin by maximizing the following criteria function [1]:

$$\log p(\underline{\boldsymbol{X}}(f,t)|\boldsymbol{H}_f)=E\{\log p(\underline{\boldsymbol{U}}(f,t))\}+\log\det \boldsymbol{H}_f \quad (4)$$

where $\underline{\boldsymbol{U}}(f,t)$ represents the separated sources, and $E(\cdot)$ represents the expectation. According to [1][4], to optimize the criteria function, the learning function for unmixing matrix $\boldsymbol{H}_f$ derived from Eq.(4) can be expressed as:

$$\triangle \boldsymbol{H}_f \propto (\mathrm{I} - \varphi(\underline{\boldsymbol{U}}(f,t))\cdot \underline{\boldsymbol{U}}(f,t)^{Herm})\boldsymbol{H}_f \quad (5)$$

where $(\cdot)^{Herm}$ denotes the Hermitian transposition and $\varphi(\cdot)$ is the activation function proposed in [4]:

$$\varphi(z)=\tanh(z_R)+i\cdot\tanh(z_I) \quad (6)$$

where z_R is the real part of z and z_I is its imaginary part. More details about the derivation can be found in [1][4].

Figure 2 illustrates the blind source separation process in frequency domain.

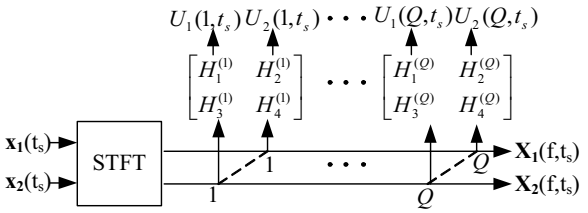


Figure 2.Frequecny Domain Blind Source Separation

We obtained the unmixing matrix independently in each frequency and the arbitrary permutation of any particular unmixing matrix in certain frequency bin will not change the value of the criteria function. As a result, the algorithm produces different permutations of separated sources along the frequency axis and the sources still remain mixed. This problem is called *permutation inconsistency* problem [4] in FD-ICA. To solve this problem, we propose the use of a statistical learning based approach to classify the output sources in each frequency bin and keep the output sources consistent along the frequency axis. The basic idea behind this approach is that the background music and vocal singing are two heterogeneous signals and the time series of them have different characteristics for each frequency bins. Figure 3 illustrates our approach to solve permutation inconsistency problem. For each frequency bin f, we have two T points complex time series output, denoted as $\underline{\boldsymbol{U}}_i(f)=\{U_i(f,1),\ldots,U_i(f,t_s),\ldots,U_i(f,T)\}$, i=1,2 , t_s denotes the time index and f is the frequency index.

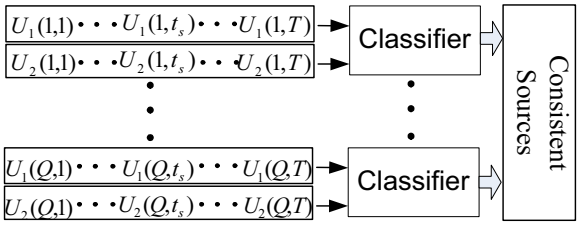


Figure 3. Statistical learning based approach to solve the permutation inconsistency problem

Figure 4 illustrates magnitude of the time series of two different source signals in the same frequency bin. The horizontal axis represents the time index and vertical axis represents the magnitude. As the figure shows, the curve of the singing voice has different behavior from that of the background music. The singing voice shows some tempo continuity, which

the background music does not have. It is probably because the vocal singing is produced by vocal organ, which always stays stable for a period of time once being activated in certain frequency, while the background music consists of many music instruments and the music instruments show less stable nature than singing voice in the particular frequency bin.

We first employ 13-dimentional linear prediction coefficients (LPC) to characterize two output time series of each frequency bin with the fixed-length (i.e., 1000 time points), and followed by a Support Vector Machine (SVM) classier to classify these two outputs time series of that particular frequency.

After classification, the classification results can be denoted as $\underline{\boldsymbol{U}}'_i(f)=\{\ U'_i(f,1),\ldots,U'_i(f,t_s),\ldots,U'_i(f,T)\}$, i=1,2. Along the frequency bin, $\underline{\boldsymbol{U}}'_1(f)$ always belongs to one particular source and $\underline{\boldsymbol{U}}'_2(f)$ always belongs to other particular source. In this way, the permutation inconsistency problem can be solved.

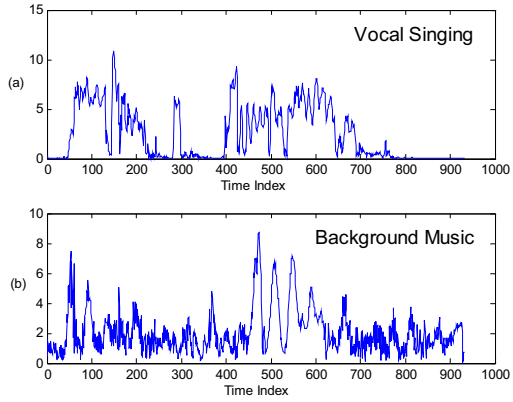


Figure 4. Two different output signals for a certain frequency

3. SINGING VOICE PITCH DETECTION

The separated vocal singing spectrum contains interference noise and errors introduced from the separation step. Among them, some were caused by imperfect separation, and some were caused by background music time series misclassified as singing voice time series. To correctly extract the singing pitch from the separated vocal spectrum, we have to handle these noise and errors. In our proposed approach, smoothing function was employed to correct the misclassification errors, followed by an algorithm to robustly locate the pitch in the vocal spectrum.

Figure 5(a) shows the spectrum of the vocal singing in a particular time index after the separation process. The horizontal axis represents the frequency and the vertical axis represents the magnitude. The circles denote the errors introduced by misclassification. Since the misclassification occurs only occasionally, the misclassification errors are characterized as isolated, short-term discontinuous points, which can be corrected by the smoothing function. We employ a 5-points median smoother function followed by a 5-points Han window linear smoothing function to correct these isolated errors. In Figure 5(b), we can easily see that the misclassification errors have been corrected after smoothing process.

After we correct the misclassification errors in the spectrum, the pitch value can be determined by the position of peaks in the spectrum. Considering the effect from local jitters and ripples introduced from separation, we propose to employ following algorithm to robustly locate the pitch in each frame of vocal singing spectrum:

1) Identify the first 10 peaks in the spectrum with the highest magnitude, and substitute each of them with a single point in frequency, and the magnitude of each point is the height of the corresponding peak. We employ 10 peaks because in most cases, the first 10 peaks contain more than 95% of total energy of the spectrum and are enough for pitch detection.

2) Since pitch can be measured as the greatest common divisor of the harmonics, we can estimate the pitch by compressing the 10 peak spectrums along the frequency axis with the compression factors of 2, 3, 4, etc., subsequent adding of the original and compressed spectrums, and picking up the distinct maximum. To avoid the effect of vocal formants (formants created by vocal tract of human beings often predominates the spectrum), these 10 peaks are all normalized into the unit value before spectrum compression.

3) The extracted pitch of the current frame cannot be too distant from that of adjacent frames, since the correct pitch should be stable for a period of time while the incorrect pitch does not have tempo continuity.

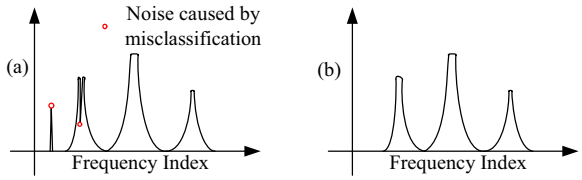


Figure 5. Misclassification errors correction

4. EXPERIMENT

We conducted the following 3 experiments to evaluate our proposed method. In the first experiment, we evaluate the performance of the SVM classifier. In the second experiment, we evaluate the performance of our proposed separation approach to the convolutive mixtures. In the third experiment, we compare the vocal pitch detected from the polyphonic recording using our proposed method to the pitch detected from the corresponding pure singing voice version.

4.1 Performance of the Classifier

In our experiment, the training set contains 20 pure instrumental/vocal songs, 10 are pure instrumental music and 10 are pure vocal singing (5 from female voice and 5 from male voice). All are Pop music sampled at 16K Hz. The training data are segmented into fixed-length and overlapping window frames (in our experiment we used 1024 samples with 50% overlapping) and the number of STFT is 2048 points after zero padding to each segmentation window.1000 consecutive window frames are grouped as one super block and after STFT, there is one 1000-points complex time series (one time frame) in each frequency bin corresponding to one super block. Considering the frequency range of the vocal singing, which ranges from 0 to 4K Hz, we only collect the complex time frames from the frequency index 1 to 512 (512 corresponds to 4K Hz in our experiment setup), and exclude the low energy time frames which can be defined as the time frames whose energy is less than the predefined threshold. Totally 40000 time frames are collected to train the classifier, half are pure vocal time frames and half are pure instrumental time frames. We employ the radial basic function (RBF) with Gaussian kernel as the kernel function in SVM training.

After training SVM, we test it using polyphonic music. The test dataset contains 40 polyphonic music excerpts of 20-30 seconds length from the 8 songs collected from the Internet. Among these 8 songs, 4 songs sung by female singers and 4 songs sung by male singers. All are sampled at 16k Hz sampling rates. The classification result on the pre-labeled test set is 95.3%.

4.2 Performance of the Vocal Separation

The success of the subsequent pitch detection for the vocal singing is dependent on the performance of the proposed vocal separation approach. In our experiment, we have to use artificial mixtures other than real world recordings because the ground truth of source signals used to create corresponding polyphonic recordings is not available (i.e. the singing version of the polyphonic music collected from Internet may not be sung by the same singer). We created 10 synthetic convolutive mixtures of one singing voice source and one corresponding background music source, each lasting 30 seconds. The sources are selected from 20 pure music/vocal songs in training set of the experiment 1 and four mixing filters A_{11}, A_{22}, A_{12}, A_{21} used in each mixing process are filters learnt in the polyphonic recordings. The separation results can be measured by Signal-to-Noise Ratio (SNR), which can be defined as:

$$SNR = 10 \cdot \log_{10} \frac{\sum_t \sum_f S^2(f,t)}{\sum_t \sum_f (S'(f,t) - S(f,t))^2} \quad (7)$$

where $S(f,t)$ denotes the discrete spectrum representation of pure singing voice, and $S'(f,t)$ denotes the vocal content spectrum obtained using our proposed method. To make comparison, we also employ the method proposed in [4] and [6] using the same dataset. The average SNR for each method to separate the vocal content from these 10 mixed songs is reported in Table 1.In addition, the standard deviation for each method is also included in the table. As the table shows, the high SNR and low standard deviation represents the effective of our proposed separation scheme. In order to highlight the contribution of SVM classifier, we also compare with the performance of FD-ICA algorithm without employing SVM classifier to align the permutation inconsistency. From the result, we can see that, without any permutation alignment, the vocal source separation result is much worse than any methods with permutation alignment.

Table 1. Vocal Content Separation Performance of Different Approaches

	Average SNR(dB)	Standard deviation
Proposed method	10.57	1.4896
Smaragdis's method	7.96	1.6153
Mitianoudis's method	8.74	1.8332
FD-ICA without SVM	2.51	3.6358

4.3 Pitch Detection Results

In this experiment, the test dataset contains 40 polyphonic music excerpts in test set of experiment 1. We also collected the pure singing version of these songs from the Internet as the ground truth due to the fact that although the polyphonic version and pure singing version are sung by different singers, the melody contours of singing in these two versions are similar.

In order to measure the similarity of these two pitch contours, we first convert the pitch value into music cents according to its frequency value, and the fact that the smallest interval in western music is 100 cents (one semitone) is used to group a sequence of samples into one note. The two note contours are aligned in time manually and we denote a character "U" at the current note if the note is higher than previous one and a "D" if the note is lower than previous one. The matching accuracy can be defined as the number of matching notes divided by total number of notes in comparison.

In our experiments, for the 40 music excerpts, the average matching accuracy is 81.4%.

5. CONCLUSIONS AND FUTURE WORK

We have presented a new approach to track the vocal pitch in the polyphonic music. The experiment on two channel polyphonic music signals shows promising performance.

There are two directions that need to be investigated in the future. First, the proposed vocal content separation approach can be improved by more accurate classification results for time frame of vocal singing and background music in each frequency. In the future, we will explore more features other than LPC to better characterize the time frame of two classes. The second direction is to refine the pitch tracking algorithm in our specific problem and make it more robust to the noise.

6. REFERENCES

[1]. A.J.Bell and T.J.Sejnowski. "An information-maximization approach to blind separation and blind deconvolution", Neural Computation, Vol.7: 1129-1159, 1995.

[2]. K.Torkkola. "Blind Separation of convolved sources based on information maximization", In *IEEE Workshop on Neural Networks for Signal Processing*, pp.423–432, 1996.

[3]. T.W. Lee and A.J. Bell "Blind separation of delayed and convolved sources". *Advances in Neural Information Processing Systems*, vol.9, pp. 758-764, 1996.

[4]. P.Smaragdis, " Information Theoretic approaches to source separation", M. Sc. Thesis, MIT Media Lab, June 1997.

[5]. L.Parra and C.Spence, "Convolutive blind source separation of non-stationary sources", *IEEE Trans. Speech Audio Processing*, pp.320-327, May 2000.

[6]. N.Mitianoudis and M.E.Davies, "Audio source separation of convolutive mixtures", *IEEE Trans. Speech Audio Processing,* pp.489-497,Sep 2003.

[7]. A.Dapena and C.Serviere, "A simplified frequency-domain approach for blind separation of convolutive mixtures", *Proc. ICA* 2001, pp. 569--574, San Diego, USA

[8]. N. Kunieda, T. Shimamura, and J. Suzuki "Robust Method of Measurement of Fundamental Frequency by ACOLS-autocorrelation of log Spectrum". *Proc. IEEE Int. Conf.on Acoustics, Speech, and Signal Processing*, vol. 1, Atlanta, GA, pp. 232-235, May 1996.

[9]. J. Rouat, Y. C. Liu, and D. Morissette, "A pitch determination and voiced/unvoiced decision algorithm for noisy speech" *Speech Communication*, vol. 21, pp. 191-207,1997.

[10]. M.Goto, " A predominant- F0 estimation method for real-time detection of melody and bass lines in CD recordings", Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,*pp.II-757-760,2000

[11]. J.M.Eagle, "Handbook of Recording Engineering", 4th ed., *Springer-Verlag New York*, 2002