SOUND SEPARATION OF POLYPHONIC MUSIC USING INSTRUMENT PRINTS

Kristóf Aczél [1] *and Szabolcs Iváncsy* [2]

Department of Automation and Applied Informatics, Budapest University of Technology and Economics
3-9. Muegyetem rkp. , H-1111, Budapest, Hungary
aczelkri@aut.bme.hu[1], ivancsy@aut.bme.hu[2]

ABSTRACT

Decomposing a polyphonic musical recording to separate instrument tracks or notes has always been a challenge. Such a signal is the superposition of many separate tracks, and it is theoretically impossible to extract the component tracks without the information that was lost at the superposition. This paper introduces a new way of sound separation of mono-aural digital recordings. The proposed algorithm inputs the lost information by using a model of real instruments in order to make the separation of individual musical notes possible. The separation method mainly targets the processing and correction of musical recordings that cannot be re-recorded.

1. INTRODUCTION

Our long term interest in sound separation is motivated by the problem of correcting existing musical recordings, adjusting volumes of instruments separately, fixing misplayed notes etc. Although much work has been presented on audio source separation, restoring individual notes that are present in the recording is still unsolved. However, the human auditory system is very effective in differentiating between sound sources and individual notes. We can block out unwanted noise, voices of other speakers in a crowded environment, we can focus on certain instruments in a polyphonic musical piece. Even today we know very little about how the human brain exactly works.

However, the fact that we are able to imagine the sound of different instruments even in complete silence, or that we can recognize the voice of our relatives without seeing their face clearly shows that we store memories of properties of different sounds. This leads us to the assumption that the human brain uses this a priori information for real-time separation of the music we hear. This assumption is confirmed in situations when we hear new, unusual instruments. Until we get to know the features of the new sound source (which may only take a few seconds or minutes), our separation capability is rather limited without this a priori information.

This paper proposes a separation algorithm that is capable of separating individual notes even in mono recordings. The importance of requiring only one channel lies in the fact that the solution is suitable also for older recordings, or recordings which may have been recorded to two or more channels, but the original tracks are for some reason no more available. If the source recording has more than one channel, then other techniques are also available in addition to the solution described in this paper to get even better results.

2. RELATED WORK

[1] is a sound source separation algorithm that requires no prior knowledge, and performs the task of separation based purely on azimuth discrimination within the stereo field. The results are impressive. However, separating individual notes is not in the focus.

[4],[5],[6] describe a method which separates harmonic sounds by applying linear models for the overtone series of the sound. The method is based on a two-stage approach: after applying a multipitch estimator to find the initial sound parameters, more accurate sinusoidal parameters are estimated in an iterative procedure. Separating the spectra of concurrent musical sounds is based on the spectral smoothness principle [3].

Beamforming techniques [10] along with the Independent Component Analysis framework offer a different way of separation. However, these methods rely on certain preliminary conditions and studio setup to achieve good results.

In [7],[8] different transformation methods were studied in order to determine the best possible means for analysis and processing of recorded digitalized polyphonic music signals.

This paper deals with an approach that aims to separate single notes from the remaining part of the recording. The focus is on the quality of the output signals rather than the speed or automation level of the process.

3. CONCEPT

Figure 1 shows the block diagram of the algorithm we are going to discuss in detail. First the signal is transformed into frequency-domain using FFT transform. After the transformation special algorithms are applied to the resulting spectrogram in order to retrieve the details and precise information that cannot be extracted directly from the FFT results. Section 4 deals with the details of these algorithms, focusing mainly on Frequency Estimation (FE) and Phase Memory (PM).

Section 5 introduces the instrument model that is used in our environment to support the separation process.

Section 6 goes through the steps of the actual separation. After the separation the separated sound channels (the isolated notes and the remainder of the original recording) are transferred back to time domain.

Finally, Section 7 summarizes the results and the performance of the system, draws the conclusions and points out improvement possibilities.

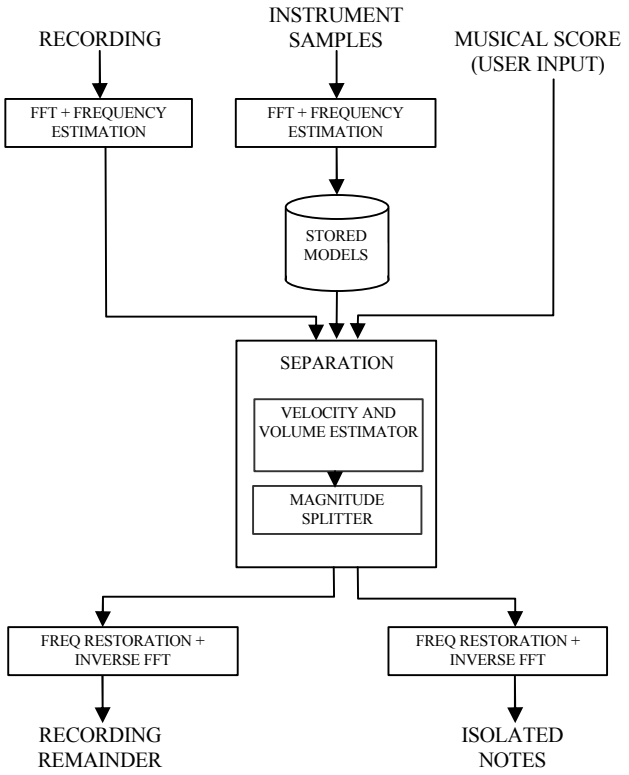

Figure 1 - Block diagram of sound separation

4. PRECISE SPECTROGRAM CALCULATION

Fast Fourier Transform is used for converting the sound data from time domain to frequency domain. We use overlapping signal fragments (frames) to analyze the sound signal. In contrast to Fourier Transform, which operates in continuous frequency space, FFT decomposes the signal to a sum of discreet frequency values. As it is known, this causes smearing in the spectrogram, which makes it hard to get the exact frequencies out of the original sound signal.

However, at polyphonic sound signal analysis, it is necessary to minimize the effect of the smearing and get the precise frequencies of sinusoidal components. In order to overcome this problem, several steps are taken. First, windowing is applied to each frame in time-domain [7]. This will reduce the smearing effect of the FFT to some extent. The resulting signal is then processed with FFT. The resolution of the resulting image in frequency domain is still not satisfactory. This paper introduces the Frequency Estimation (FE) method, which is an extension to [11].

FFT describes the signal in terms of sinusoids that have a well defined *bin frequency*, *phase* and *magnitude.* Any sinusoidal source signal with a frequency that matches one of the bin frequencies will produce magnitude only on one specific bin, while other frequencies will produce magnitudes on several neighbour bins, leaving no clue on the precise frequency that was originally present in the signal. Frequency Estimation method finds the true frequency for each bin by analysing the phase information on the same bin in successive frames.

The original method introduced in [11] compares two successive frames. The true frequency of a bin is calculated as follows.

$$f_k = k \cdot (S/K)$$

$$j^{\exp t}_{k,t_2} = j_{k,t_1} + (t_2 - t_1) \cdot 2\Pi f_k$$

$$j^{dev}_{k,t_2} = j_{k,t_2} - j^{\exp t}_{k,t_2} + l \cdot 2\Pi \qquad l \in Z$$

$$-\Pi < j^{dev}_{k,t_2} \le +\Pi$$

$$f^{true}_{k,t_2} = f_k + \frac{j^{dev}_{k,t_2}}{2\Pi \cdot (t_2 - t_1)},$$

where S is the sample rate of the signal; K is the frame size; f_k and $f_{k,t}$ represent, respectively, the bin frequency and phase of bin k in time t; $j^{\exp t}_{k,t}$ is the expected phase; $j^{dev}_{k,t}$ is the deviance between the expected and measured phase; $f^{true}_{k,t}$ is the estimated true frequency of bin k in time t. The greater the time difference between the start of frames the more precise the estimated value of $f^{true}_{k,t}$. On the other hand, big time differences limit the maximum detectable distance between $f^{true}_{k,t}$ and f_k.

To overcome this limitation and further improve the preciseness of the calculation, an extension to the original algorithm is proposed. The true frequencies can be found more precisely by taking the weightened average of the last m phase deviations ($c_{k,t}$ is the coefficient of bin k in time t). This extension will be referred to as Phase Memory (PM).

$$\hat{j}^{dev}_{k,t_x} = \frac{\sum_{t=t_{x-m}}^{t_x} j^{dev}_{k,t} \cdot c_{k,t}}{\sum_{t=t_{x-m}}^{t_x} c_{k,t}}$$

$$\hat{f}^{true}_{k,t_2} = f_k + \frac{\hat{j}^{dev}_{k,t_2}}{2\Pi \cdot (t_2 - t_1)}$$

Figure 2 shows the effect of the Frequency Estimation and Phase Memory on a spectrogram. (Spectrograms are plotted in two dimensions (frequency and time) with grayscale colors indicating the magnitude). Figure a) plots the raw spectrogram; b) the spectrogram with [11] applied and c) shows the effect of the Phase Memory method. If the recording is not very complex, the spectrogram in c) is understandable even to human eyes.

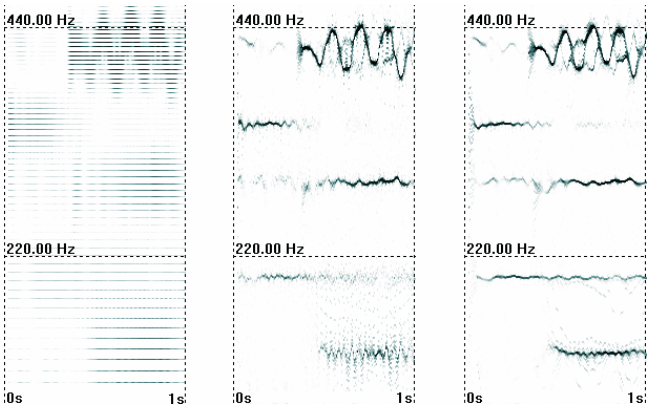

Figure 2 - spectrogram: a) FFT b) FFT+FE without PM c) FFT+FE

5. INSTRUMENT SAMPLES

To understand the features of a sound recording, we must discuss the features of separate musical instruments first. In general, the sound of a musical instrument in any given short moment in time can be decomposed into two main components: a periodic and an aperiodic sound component.

Periodic sounds are those emitted by a source that produces regular vibrations over time, resulting in a collection of frequencies called harmonics, partials, or overtones. Harmonic frequencies originating from the same source are related in a way that they occur in multiples of the lowest frequency, referred to as the fundamental frequency. Thus, a collection of harmonically related frequencies, of which the fundamental is 200 Hz, would occur with frequencies of 400 Hz, 600 Hz, 800 Hz, 1000 Hz, 1200 Hz, etc.

Aperiodic sounds are those which most often occur perceptually as noise (cymbal crash, drums, snare or the sound of the piano hammer at the beginning of each piano note – see Figure 3). Acoustically, noise is defined as a random collection of frequencies from a single source which are not harmonically related and whose waveform is therefore irregular.

Most instruments generate both harmonically related frequencies and noise-like transients. If we want to eliminate a note from the recording, we must first examine one single note of that instrument with the same *fundamental frequency*, *velocity* and *amplitude*. We must keep in mind that velocity and amplitude are not synonyms here. A key on the piano can be pressed harder and softer (velocity difference), resulting in different spectrograms even if normalized before comparing; while the same key-press can be recorded with different microphone gain settings (volume difference), the spectrograms of which – after normalizing – will resemble each other [9].

Figure 4 plots a piano note at 260Hz in frequency domain. The spectrogram shows its two main components. Horizontal lines represent the periodic component, which slowly decays in time in case of a piano note. The short vertical fuzzy area at the beginning that is caused by the piano hammer represents the aperiodic component.

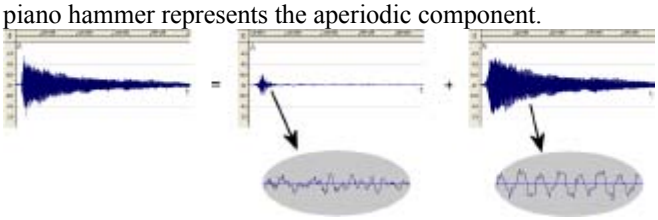

Figure 3 - waveform of a piano note (time domain), decomposed to aperiodic (hammer) and periodic (strings) parts

Human hearing is limited to about 15-20000Hz, depending mainly on age. Our model stores the magnitudes in the spectrogram of notes of the instrument in this range. Since it is practically impossible to store all possible notes an instrument is able to generate, only a few are stored at different frequencies and velocities. The number of needed samples is subject of future research, the current implementation works with 3-5 velocity levels per instrument and 6-8 sampled frequencies per octave. If the spectrogram of an unsampled note is needed later in the separation algorithm, the missing sample is interpolated from existing ones. If enough samples are stored, the difference in the output quality will not be noticeable. Finding the required number of sampled notes is out of the scope of this paper. From now on this model will be referred to as 'instrument print' or, simply, 'print'.

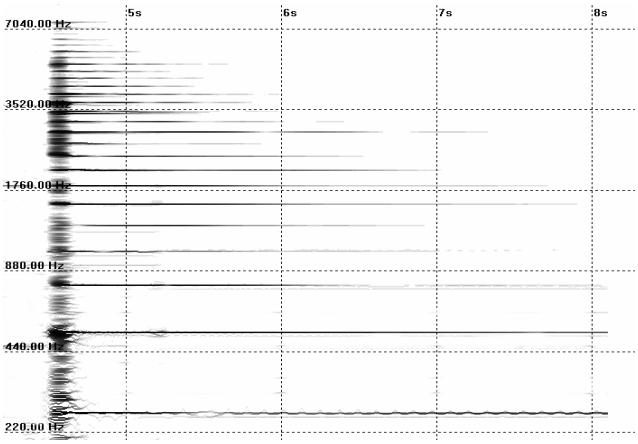

Figure 4 - Plot of a piano note at 260Hz

The model can shortly described as follows. If we take the spectrogram coefficients ($\mathbf{C} = \left[c_k \right]$) of a note on a certain base frequency f_{base_i}, with velocity M, then

$$A_{\mathbf{M}, f_{base}, r, t} = \sum_{f_{base} \cdot 2^{\frac{r-0,5}{R}} < \hat{f}_{k,t}^{true} < f_{base} \cdot 2^{\frac{r+0,5}{R}}} c_{k,t}$$

$$r = \left\lfloor \log_{\sqrt[R]{2}} \frac{f_{base}}{\hat{f}_{k,t}^{true}} \right\rfloor$$

$$R = 24$$

where A represents one instrument sample starting at t=0. $A_{\mathbf{M},f_{base},s,t}$ denotes the sum of the energy of a narrow frequency band in time t; r denotes the distance from the base frequency, and R is an experimental value which defines the size of the frequency band.

6. SEPARATION

After we have the instrument prints, and can produce the right print for any frequency and velocity by interpolation, we can move on to the source recording to be processed. The spectrogram of the right print will be separated from the remaining part of the recording using linear decomposition. Assuming that we know the exact frequency, volume and velocity of a certain note that we want to separate from the remaining part of the recording, the following algorithm can be proposed for the separation.

The phase and magnitude information of the spectrograms will be handled separately. The phases of the resulting spectrograms will be for all notes (i) the same as the original phases, while the magnitudes of the recording will be split between them.

$$j_{i,k,t} = j_{orig,k,t}$$
$$j_{remain,k,t} = j_{orig,k,t}$$
$$c_{orig,k,t} = c_{remain,k,t} + \sum_{i=1}^{I} c_{i,k,t}$$

The implementation of the actual note separation divides the energy between the target notes iteratively in D steps:

$$\underline{\mathbf{C}}_{[0],0,t} = \left[c_{[0],k,t} \right] = \left[c_{k,t} \right]$$
$$\underline{\mathbf{S}}_{[0],i,t} = \underline{\mathbf{0}}$$
$$a_{[d],r,i,t} = \frac{A_{i,\mathbf{M},f_{base},s,t-T_{i,start}}}{\sum_{f_{base}\cdot 2^{\frac{r-0,5}{12}} < \hat{f}_{k,t}^{true} < f_{base}\cdot 2^{\frac{r+0,5}{12}}} c_{[d],0,k,t}^{per}}$$
$$\underline{\mathbf{C}}_{[d],i,t} = [c_{[d],i,k,t}]$$
$$c_{[d],i,k,t} = \begin{cases} d\left(c_{[d],i-1,k,t}\left(1-\frac{a_{r,i,t}}{D}\right)\right) & \text{if } f_{base}\cdot 2^{\frac{r-0,5}{R}} < \hat{f}_{k,t}^{true} \\ & \hat{f}_{k,t}^{true} < f_{base}\cdot 2^{\frac{r+0,5}{R}} \\ c_{[d],i,k} & \text{otherwise} \end{cases}$$
$$\underline{\mathbf{S}}_{[d+1],i,t} = \underline{\mathbf{S}}_{[d],i,t} + (\underline{\mathbf{C}}_{[d],i-1,t} - \underline{\mathbf{C}}_{[d],i,t})$$
$$\ldots$$
$$\underline{\mathbf{C}}_{[d+1],0,t} = \underline{\mathbf{C}}_{[d],I,t}$$
$$\ldots$$
$$\underline{\mathbf{C}}_{[D],0,t} = \underline{\mathbf{C}}_{[D-1],I,t}$$

where $T_{start,i}$ is the attack time of note i, D is the number of steps, [d] is the current step, $\mathbf{S}_i$ is the spectrum of note i after the separation, $\mathbf{C}_{[D]}$ is the remaining energy in the recording after the separation, while $\mathbf{M}_i$ refers to the volume and velocity values which are assumed to be known.

After the separation, the spectrograms can be transformed back to time domain.

In the above paragraphs the starting time, frequency, volume and velocity of the note to be separated were assumed to be known. These starting parameters are needed by the separation algorithm. However, this does not resemble a real-life scenario at all. Generally, when given a good representation of the sound signal (spectrogram, musical score etc.), the user can interactively input the frequency and time of the note on a terminal quite precisely. On the other hand, specifying the velocity and volume is usually a much harder task, since the average user does not either recognize or understand the difference between these two expressions. Therefore it is safe to feed the starting time and frequency as parameters to the separation engine, but the velocity and volume must be calculated algorithmically.

For the above mentioned problem, we propose an iteration algorithm that is based on the gradient method. The desired volume-velocity pairs are approximated with initial values, and after a number of repetitions the optimal values can be approached. The algorithm is as follows:

1. The user interactively inputs all the concurrent notes existing in the musical section to be processed. We require information on all note starting and ending times and frequencies. However, no information is required on volume or velocity values this time. This is a reasonable compromise between convenience for the user and complexity in the algorithm.
2. An initial volume and velocity level is determined for all notes. (From now on, a (start, end, frequency, volume, velocity) couple will be referred to as a 'region'). This initial level can safely set at 100% the strength of our stored prints. We found that the initial volume level has no influence on the outcome of the algorithm; however, it may affect the overall speed.
3. The separation algorithm is run using the selected volume and velocity values.
4. The error of the separation is calculated:

$$Err_{rem} = \sum_{t=T_{start}}^{T_{end}} \sum_{k=0}^{K} c_{[D],k,t}$$
$$Err_{pr} = \sum_{\forall i} \sum_{t=T_{start}}^{T_{end}} \sum_{\forall r} \left(A_{i,\mathbf{M}_i,f_{base},r,t} - \sum_{\substack{f_{base}\cdot 2^{\frac{r-0,5}{R}} < \hat{f}_{k,t}^{true} \\ \hat{f}_{k,t}^{true} < f_{base}\cdot 2^{\frac{r+0,5}{R}}}} s_{[D],i,k,t} \right)$$
$$Err_{sum} = a \cdot Err_{rem} + (1-a) \cdot Err_{pr} \ ,$$

where Err_{rem} sums the error caused by remaining energy on the bins in the recording after the separation, Err_{pr} sums the error if there was less energy in the recording than the prints required, Err_{sum} is the global error of the separation step with the current region parameters, T_{start} and T_{end} are, respectively, the starting and ending times of the observed time. r is the region identifier in the observed time slice, and $0<a<1$ is the quality preference parameter of the separation. High a value means preference on the quality of the separated notes to the quality of the remaining part, while low a value provides better remainder quality, but poorer separated note quality.

5. Volume level of one of the regions is slightly increased (e.g. from 100% to 101%). Error is counted again. This procedure is repeated with all the regions in the observed time slice.
6. Same as 5 with velocity levels.
7. The gradient vector can be computed from the error values. This vector shows the direction in which we should change the current volume and velocity values to reduce the error values. Volume and velocity values are modified accordingly.
8. Steps 3-7 are repeated as many times as necessary to get precise enough volume and velocity values.
9. After finding the desired record level and velocity values giving the lowest possible error value, the separation algorithm is run with the calculated parameters.

The application of the gradient method is only possible if there is only one local minimum in the error-space, otherwise it could lead the algorithm towards the wrong direction. Proving that the volume-velocity level space meets this condition is out of the scope of this paper.

We must mention that if there are notes in the original recording that are located closely in frequency, it introduces the beating effect. This effect is not resolved by the current algorithm. This issue is the subject of future research.

7. SUMMARY

The paper showed a method for separating single instrument notes from a recording using pre-recorded instrument prints. The results are quite promising. An example recording and its separated notes can be downloaded from http://aczelkri.fw.hu/separation. However, experiments are needed for a mathematical validation. For recordings that only contain harmonically unrelated notes the algorithm provides very clear results, and even if some notes are located on each other's base or overtone frequencies, the separation provides reasonably good results. However, in these cases improvement is still required to deal with beating and get higher quality output.

The other area of future research is building more flexible instrument models. We cannot always have prints for all the possible notes of an instrument, in most cases we do not even have access to the original instrument the recording was taken with. Thus, as we do not use the right print, only a close approximate, we may experience some distortion in the separation output which might be audible also to less audiophile listeners.

REFERENCES

[1] Barry, D., Lawlor, R. and Coyle E., "Sound Source Separation: Azimuth Discrimination and Resynthesis", in *Proc. 7th International Conference on Digital Audio Effects*, DAFX 04, Naples, Italy, 2004.

[2] Samer A. Abdallah and Mark D. Plumbley, "Polyphonic transcription by non-negative sparse coding of power spectra", in *Proc ISMIR 2004*, Barcelona, Spain, October 10-14, 2004.

[3] Klapuri, A., "Multipitch estimation and sound source separation by the spectral smoothness principle", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Salt Lake City, USA, 2001.

[4] Virtanen, T. and Klapuri, A., "Separation of Harmonic Sounds Using Multipitch Analysis and Iterative Parameter Estimation", in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New York, USA, 2001.

[5] Virtanen, T. and Klapuri, A., "Separation of Harmonic Sound Sources Using Sinusoidal Modeling", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, 2000.

[6] T. Virtanen and A. Klapuri, "Separation of harmonic sounds using linear models for the overtone series", in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '02)*, Orlando, Fla, USA, May 2002.

[7] K. Aczél, Sz. Iváncsy, "Musical source analysis with DFT", in *Proc. MicroCAD 2006 International Scientific Conference*, Miskolc, Hungary, March 2006.

[8] K. Aczél, Sz. Iváncsy, "Musical source analysis: spectrogram versus cochleagram", in Press, *MicroCAD 2007 International Scientific Conference*, University of Miskolc (Miskolc, Hungary), March 2007.

[9] K Aczél, "Manipulation of Musical Recordings Using Instrument Prints", in *Proc. of Automation and Applied Computer Science Workshop 2006 (AACS'06)*, Budapest, Hungary, June 2006

[10] N. Mitianoudis, M. E. Davies, "Using Beamforming in the audio source separation problem", *7th Int Symp on Signal Processing and its Applications*, Paris, July 2003

[11] http://www.bernsee.com/dspdimension.com/html/pshiftstft.html (11-02-2007)