Budapest University of Technology and Economics
Department of Automation and Applied Informatics

SEPARATION OF POLYPHONIC RECORDINGS USING INSTRUMENT PRINTS

PhD thesis booklet

Kristóf Aczél

Advisors:
Dr. István Vajk DSc (BME AAIT)
Dr. Szabolcs Iváncsy (BME AAIT)

Budapest, 2009

1. Overview and goals

Several unanswered questions are investigated today in the field of sound processing. One of the main research areas is the analysis, interpretation, modification and correction of polyphonic music. When a musical event is recorded there is often need for postprocessing steps due to the presence of misplayed notes in the recording, be they either out of tune or incorrectly intonated. There are several methods used to avoid that the recording that finally gets to the listeners contains such imperfections.

One of the most popular methods is cutting the musical material. It is done by performing and recording the same musical content (or parts of it) multiple times. Afterwards it is the task of the editor to choose one from the many takes of the same parts that sounds best. The chosen parts are then pieced together with the help of sound editing software. Sometimes only as short as one single note is cut out from the recording and substituted with another copy of the same note from another take. In this case the other notes in the same time segment also inevitably get replaced with their copies from the other take, which may not be the intention.

The other commonly used method, multitrack recording, records the sound of each instrument (but at least the most important ones) and the voice of the singer to a separate channel using dedicated microphones. This approach allows for the modification of the channels independently from the other ones. Multitrack recording is very popular in pop music where the voice of the singer is often improved by automatic tone correction methods or manipulated in other ways. This makes the recorded music in-tune even if the real performance of the artist was out of tune.

However, the most interesting cases of sound manipulation are those polyphonic recordings that for some reason cannot be retaken (e.g. old recordings). Neither are there any musical fragments available to be used in place of the incorrect notes in these recordings, nor is the multichannel representation available, only the original mono or stereo source signal. In these cases altering the musical structure in any way requires the decomposition of the signal. The multichannel representation of the signal needs to be generated algorithmically from the original signal, or at least the note to be fixed must be isolated to a separate channel. After this step we can choose from a vast set of methods to alter the note signal to fit our needs, after which the separated signal can be mixed back to the recording.

The goal of my research was to investigate the problem of sound source separation and propose methods in this field that are particularly applicable to polyphonic music signals. I considered the separation quality the most important property of the proposed separation method; therefore I set the following goals:

- separation of any user specified notes from the polyphonic recording,
- keeping the quality of the separated note signals as high as possible.

I considered the following issues less important:

- separation of each and every note in the recording, that is, generating the complete multitrack representation of the music signal - as it is not needed for fixing incorrect notes, but brings unnecessary complexity to the separation system or may even lower the separation quality;
- creating a fully automated separation system, where the multichannel representation is generated 'by one click' - as actively involving the user may allow for higher quality output;
- handling the noise - as there are many capable methods in that area;
- real-time operation.

My contribution involves the proposition for a global architecture of a polyphonic sound source separation system, the elaboration of its algorithms and modules, as well as the fine-tuning of the proposed methods.

2. Research methodology

As the first step I investigated the tools currently used for solving the sound separation problem. The most popular (and almost the only) technique used here is multitrack recording. In multitrack recording the voices originating from different sources are recorded separately to their own channel. This makes later modifications to the signals possible. However, multichannel recording is mainly used in pop music; realizing this technique for classical music and/or for greater orchestra is cumbersome, since microphones record sounds originating from the surrounding sources as well in addition to the target source. In the case of classical music the microphones are usually placed at distant locations, which makes it even harder to record each instrument in isolation. Studios very seldom use products for post separation, as the obtainable quality currently does not meet studio needs.

There are several ongoing researches in the field of sound source separation. These can usually be categorized into one of the following groups:

- Model based systems: In this category a parametric model of the input sources is established that serves as a set of constraints on the output signals. The model parameters are obtained from the mixture itself. The two main branches of this area are rule-based algorithms that use prior information to build the model, and Bayes estimation where prior information is explicitly given using probability density functions.
- Unsupervised learning methods usually operate on the basis of simple non-parametric models, and require less information on the original sources. They try to gather information on the source signal structures from the mixed data itself using information-theoretical principles, such as statistical independence between the sources. The most common approaches used to estimate the sources are based on independent component analysis (ICA), non-negative matrix factorization (NMF), and sparse coding. These algorithms usually factorize the spectrogram (or other short-time representation of the signal) into elementary components. This is followed by clusterization that builds the output separated channels from the elementary components.
- Other methods like azimuth discrimination that generates soundtracks using spatial information in the input signal

I did not find any approach that meets the goals of my research. Instead of separating arbitrary, user selected notes from the recording, the usual goal is automatic generation of multiple output tracks based on automatic recognition of the musical content. In addition to that, researches rarely investigate the case of notes in a harmonic relationship satisfactorily (e.g. none of them separated two notes that share the same fundamental frequency)

After the overview of current researches I decided to define my own sound source separation system. However, certain elements from other researches were used in the separation system introduced here.

Sound source separation is carried out in frequency domain in this work. Each audio signal is transferred from time domain to frequency domain using the conventional STFT transformation and windowing functions. In frequency domain the separation problem can be defined as follows. Let $\underline{\mathbf{c}}^*_{r\tau} = \left\{ c_{r\tau,k} \cdot e^{j \cdot \gamma_{r\tau,k}} \right\}$ denote the spectrum of the recording and $\underline{\mathbf{s}}^*_{i,r\tau} = \{ s_{i,r\tau,k} \cdot e^{j \cdot \sigma_{i,r\tau,k}} \}$ denote the spectrum of the instruments at time $r\tau$:

$$\underline{\mathbf{c}}^*_{r\tau} = \sum_{\forall i} \underline{\mathbf{s}}^*_{i,r\tau} , \qquad (1).$$

where the components of the spectra are expressed by using the polar coordinate form as:

$$\underline{\mathbf{s}}^*_{i,r\tau} = \begin{bmatrix} s_{i,r\tau,1} \cdot e^{j \cdot \sigma_{i,r\tau,1}} \\ s_{i,r\tau,2} \cdot e^{j \cdot \sigma_{i,r\tau,2}} \\ \ldots \\ s_{i,r\tau,K} \cdot e^{j \cdot \sigma_{i,r\tau,K}} \end{bmatrix}, \quad \underline{\mathbf{c}}^*_{r\tau} = \begin{bmatrix} c_{r\tau,1} \cdot e^{j \cdot \gamma_{r\tau,1}} \\ c_{r\tau,2} \cdot e^{j \cdot \gamma_{r\tau,2}} \\ \ldots \\ c_{r\tau,K} \cdot e^{j \cdot \gamma_{r\tau,K}} \end{bmatrix} \qquad (2),$$

where $s_{i,r\tau,k}, \sigma_{i,r\tau,k}, c_{r\tau,k}, \gamma_{r\tau,k} \in \mathbb{R}$. Carrying out the separation requires the solution of this equation system. However, the problem is underdetermined even if there are only two concurrent notes in the recording.

My dissertation consists of two parts: in the first part I investigate the main problem of sound source separation and provide an overview of related work. I elaborate the problem space, and the most commonly used representations of sound signals (time domain, frequency domain, STFT, Q-transform, cochleagram, wavelets, etc.). From the many methods I used a modified version of the Short Time Fourier Transform in my work.

The second part elaborates my contribution to the research area. It consists of three theses, and an evaluation section. Latter covers measurements of synthetic and natural separation cases that validate the methods described in the theses; a short introduction of the ReChord system which is a software implementation of the research results; and also cases are presented here where the software was used in the production of commercial music CDs. Beside looking at the measurement results I encourage the reader to listen to my set of separation samples that is available at my research site: http://avalon.aut.bme.hu/~aczelkri/separation.

3. New contribution

My work is divided into three theses. The first two of them provides a set of tools, on the grounds of which the third thesis proposes a complete separation system architecture.

The first thesis deals with the definition of sound source separation. It elaborates why sound source separation is an underdetermined problem and proposes a simplification in order to decrease the complexity of this problem. After that it studies the applicability of the simplified model along with the resulting separation error in different cases. It also presents a measure for evaluating different separation systems and comparing them to each other.

The second thesis provides a solution for decomposing audio signals to periodic and aperiodic components.

The third thesis introduces a system architecture for the separation of individual note signals. The separation system uses samples of instruments for its operation. A possible model for *instrument prints* is established. An algorithm is also proposed for dividing the energy of the input to multiple channels in a way that the output channels resemble the instrument prints as much as possible. In addition to that the thesis also offers a solution for handling the beating effect in the output channels.

Thesis 1: The separation problem and its simplification *([2], [3], [4], [6], [7], [8], [9], [10], [11], [14])*

In this thesis I present a modified version of the original sound source separation problem. The modification simplifies the original task to an energy redistribution problem in frequency domain.

Simplification of the separation problem

In my research sound source separation is carried out in frequency domain. Let $\underline{\mathbf{c}}^*_{r\tau} = \left\{ c_{r\tau,k} \cdot e^{j \cdot \gamma_{r\tau,k}} \right\}$ denote the spectrum of the polyphonic recording for a frame starting at time *rτ;* $\underline{\mathbf{s}}^*_{i,r\tau} = \{ s_{i,r\tau,k} \cdot e^{j \cdot \sigma_{i,r\tau,k}} \}$ represent the spectrum of original note *i*. The separation problem can be expressed as:

$$\underline{\mathbf{c}}^*_{r\tau} = \sum_{\forall i} \underline{\mathbf{s}}^*_{i,r\tau} \,, \tag{3}$$

where $s_{i,r\tau,k}, \sigma_{i,r\tau,k}, c_{r\tau,k}, \gamma_{r\tau,k} \in \mathbb{R}$. Equation (3) is an underdetermined system of equations which cannot be solved unambiguously without any further constraints. I have proposed a simplification for solving the separation problem that assumes that the phases of the individual bins in the Fourier spectrum of any instrument match the phases of the respective bins of the original recording:

$$\gamma_{r\tau,k} = \sigma_{i,r\tau,k} \tag{4}$$

This modification neglects the phases, thus we do not have to take their effect into account when solving the simplified equation system:

$$\underline{\mathbf{c}}^*_{r\tau} = \sum_{\forall i} \left| \underline{\mathbf{s}}^*_{i,r\tau} \right| \tag{5}$$

This is basically an energy redistribution problem, where we need to work out how the energy on the bins was generated. Although the solution for (5) is still not trivial, it is much simpler than for (3). Here in Thesis I. I present the simplification itself as a basic principle that can be used as the grounds of a concrete separation system. Such a solution for (5) is proposed later in Thesis III.

Effect of the proposed simplification on the separation quality

Due to its nature the simplified problem neglects the facts that the phases in the spectrum of the individual instruments are usually different. Thus we have to consider a certain level of distortion when building any separation algorithm on the top of the simplified version of the separation problem.

I have studied the theoretical level of distortion caused by the proposed simplification.

- For two components of the same amplitude and frequency the separation error is anywhere between 0% and 100%. Its typical value is 35%, that is, this is the fraction of the original amplitude that ‘disappears’ due to cancellation when they get mixed together. This fraction of the amplitude cannot be recovered by using the simplified equation system (5). However, in polyphonic music only in extremely rare cases are the base frequencies of any two notes precisely equal, and this hardly ever lasts longer than a few milliseconds.
- As the frequency difference between the base frequencies of the two notes increases, the energy loss quickly decreases towards 0%.
- Components on close (but not equal) base frequencies introduce beating which cannot be resolved by the simplified equation system.
- In the case of components of different amplitudes the level of distortion decreases as they cannot cancel each other completely.

I have analyzed the separation error caused by the simplification. Let *x(t)* and *y(t)* denote the time function of two sinusoidal signals:

$$\begin{aligned} x(t) &= A_x \cdot \sin(f_x \cdot t) \\ y_\varphi(t) &= A_y \cdot \sin(f_y \cdot t + \varphi) \end{aligned} \tag{6}$$

where $0 < \varphi < 2\pi$ is the phase difference between the signals, A_x and A_y denote the amplitude, and f_x and f_y denote the frequency of the two signals. When mixed together these signals will generate amplification and cancellation effects. In the best case the phases of the two signals are equal. In this case the two signals amplify each other and there is no loss of energy:

$$E_{lossBestCase,f_x,f_y} = 1 - \max_\varphi \frac{\left|X + Y_\varphi\right|}{\left|X\right| + \left|Y_\varphi\right|} = 1 - \frac{\left|\left|X\right| + \left|Y_\varphi\right|\right|}{\left|X\right| + \left|Y_\varphi\right|} = 0\,. \tag{7}$$

In the worst case the two signals have opposite phases, and the energy of the mixture is the difference of the energy of the individual signals:

$$E_{lossWorstCase,f_x,f_y} = 1 - \min_\varphi \frac{\left|X + Y_\varphi\right|}{\left|X\right| + \left|Y_\varphi\right|} = 1 - \frac{\left|\left|X\right| - \left|Y_\varphi\right|\right|}{\left|X\right| + \left|Y_\varphi\right|}\,. \tag{8}$$

In a typical case the energy loss can be calculated from the average of all the possible cases:

$$E_{lossAverage,f_x,f_y} = 1 - \frac{\int\limits_{\varphi=0}^{2\pi} \frac{\left|X + Y_\varphi\right|}{\left|X\right| + \left|Y_\varphi\right|} d\varphi}{2\pi}\,, \tag{9}$$

which varies with the frequency and amplitude relation of the two original sine waves: the farther the two components are the lower the energy loss is. Figure 1 depicts synthetic test results on the energy loss when using the simplified version of the sound separation equation.

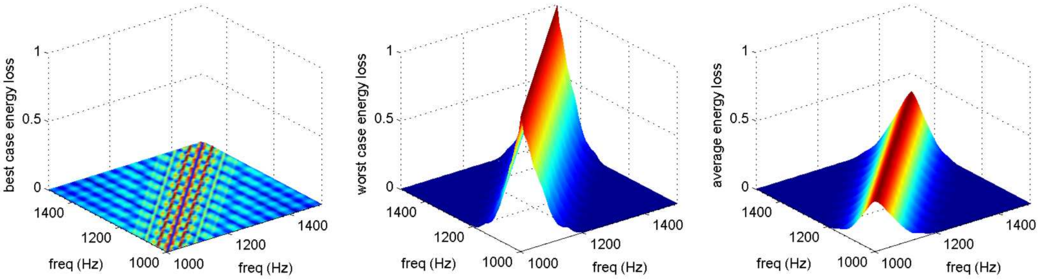


Figure 1: energy loss for two sine waves of equal amplitudes

From these results it follows that the simplification I have proposed is a reasonable compromise between quality and the complexity of the separation problem.

Quality measure for evaluating sound separation systems

The quantitative measurement of the quality of sound separation systems is a complex problem. I have proposed a new measure that calculates the error of an arbitrary separation algorithm in frequency domain. Exploiting the fact that the human ear is very insensitive to phase distortions the measure calculates the error from the energy of the separated note signals:

$$SDR_i^F \text{ [dB]} = 10\log_{10} \frac{\sum\limits_{\forall r\tau}\sum\limits_{k=0}^{K} s_{i,k,r\tau}{}^2}{\sum\limits_{\forall r\tau}\sum\limits_{k=0}^{K} \left[\hat{s}_{i,k,r\tau} - s_{i,k,r\tau}\right]^2} \tag{10}$$

Thesis 2: Periodic/aperiodic decomposition of sound signals ([12], [13], [18])

This thesis takes up the task of decomposing digital sound signals into an aperiodic and a periodic component. I have developed a method for solving the decomposition problem in a simple and fast way.

Periodic-aperiodic decomposition

I have proposed an algorithm that decomposes the periodic and aperiodic components of a sound signal. I have used the frequency estimation method proposed by Brown. This method observes the phase of the individual bins in the spectrogram and estimates a true frequency for each bin at each time frame. Thus, in addition to the $c_{r\tau,k}$ amplitude and $\varphi_{r\tau,k}$ phase values, an $f_{r\tau,k}^{true}$ true frequency is assigned to each bin. (This is later referred to as the *Frequency Estimated spectrogram*) This true frequency is exploited in my algorithm.

The method proposed in this thesis is based on the fact that the estimated true frequency values given by the frequency estimator change very slowly over time in the case of periodic components, while the estimation gives true frequency values of high variance over time for noise-like components. By observing the true frequency values each bin at each time frame can be labeled *periodic* or *aperiodic* based on its frequency history.

The periodic component of the original sound signal can be synthesized from the bins that are labeled *periodic*, while the noise-like component can be generated using the bins that are labeled *aperiodic*.

Let $d_{k,r\tau}$ denote the square deviation of the subsequent $f_{k,r\tau}^{true}$ true frequency values of bin *k*:

$$d_{r\tau,k} = \sum_{p=0}^{P-1} \left(f_{(r-p)\tau,k}^{true} - f_{(r-p-1)\tau,k}^{true} \right)^2 , \qquad (11)$$

where P is the number of frames included in the observation the bin's frequency history. The periodicity of a bin can be calculated using the respective $d_{r\tau,k}$ value. Under a certain ε threshold the bin will be considered periodic, otherwise aperiodic:

$$\vartheta_{r\tau,k} = \begin{cases} d_{r\tau,k} < \varepsilon & 1 \\ d_{r\tau,k} \geq \varepsilon & 0 \end{cases} \qquad (12)$$

Instead of the binary decision described above we can assign a *periodicity score* to each bin at each time frame based on the frequency history, showing how probable is that the bin is holding periodic energy. This case can be realized using a continuous $\vartheta_{r\tau,k}$ periodicity score function that provides a distribution of the amplitudes in non-trivial cases.

Finally, the amplitude on the bins is divided into two parts based on their periodicity status or score. The resulting periodic and aperiodic components can be expressed as:

$$\begin{aligned} c_{r\tau,k}^{per} &= \vartheta_{r\tau,k} \cdot c_{r\tau,k} \\ c_{r\tau,k}^{aper} &= (1 - \vartheta_{r\tau,k}) \cdot c_{r\tau,k} \end{aligned} \qquad (13)$$

In both cases (binary decision and periodicity score) the phase values of the bins of the periodic and aperiodic components equal the phase values of the bins of the original signal, while the sum of the amplitude values of the bins of the two components equal the amplitude of the bins of the original signal.

Thesis 3: Sound source separation system based on instrument prints ([2], [3], [4], [6], [7], [8], [9], [10], [14], [15], [16], [17], [18])

I have developed a sound source separation system that separates monoaural polyphonic signals to note signals. Besides the **basic architecture** I have also elaborated the three most important parts of the system:

- the **instrument print model** which is used to store an effective representation of instrument notes,
- the **Simplified Energy Splitter algorithm** that divides the energy of the input recording between the output note signals,
- the **beating correction algorithm** that regains energy that was lost during the separation due to cancellations in the original mixture

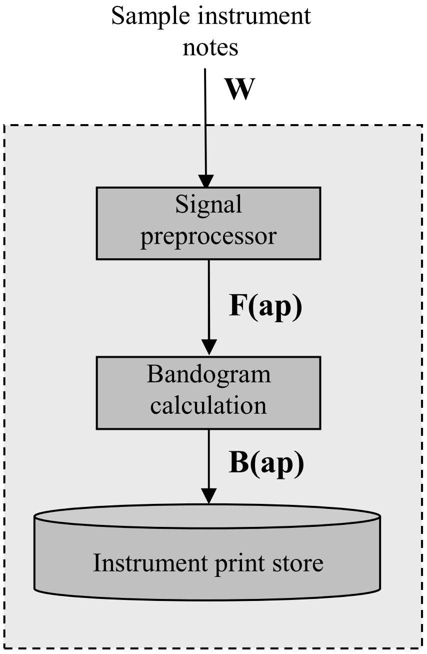
Basic architecture of the note-based sound source separation system

I have proposed a system architecture that carries out the task of separation of polyphonic music signals. The separation system generates individual output soundtracks for each of the musical notes[1] in the input recording[2]. The problem of lack of information about the input signal is solved by both requesting information from the user like the musical score and using samples of real instruments.

The system can operate in two modes. In the first mode, the *instrument print creation mode*, the system takes sample waveforms from real-life instruments and transforms them to a representation that will later be useful for separation purposes. The block diagram of this operation mode is shown in Figure 2.

The second operation mode of the system, the *separation mode*, is depicted in Figure 3. It extracts individual note signals from the source recording using three inputs: the original music, the musical score that is entered by the user, and the instrument prints that were created in the first operation mode. The most important blocks are as follows:

- **Playmode detector**: this component estimates some parameters of the notes that cannot be reliably entered by the user, such as note strength, warmth, sharpness etc. The term *playmode* addresses these properties of the various notes collectively.
- The **Simplified Energy Splitter** (SES): This component carries out the redistribution of the energy in the input signal to the several output signals.



W: *Waveform*: Simple waveform

F: *FE spectrogram*: Frequency estimated spectrogram. $c_{k,t}$ bin amplitudes and $\varphi_{k,t}$ phase values remain the same as those of the simple FFT, but an $f_{k,t}^{true}$ true frequency value is stored in addition for each bin.

B: *Bandogram*: A spectrogram split to subbands, in which the energy is summed. Only these sums are stored, no detailed information on bin amplitudes and no phase information either.

(a/p): *Aperiodic/periodic component:* the aperiodic (noise-like) or periodic (harmonic) component of a signal

Figure 2: Signal flow and block diagram of the instrument print creation process

[1] A musical note may be played by more than one artist, e.g. in the case of a string section

[2] In practice only the notes of interest are separated from the recording, but there is no theoretical limitation on the number of separable note signals

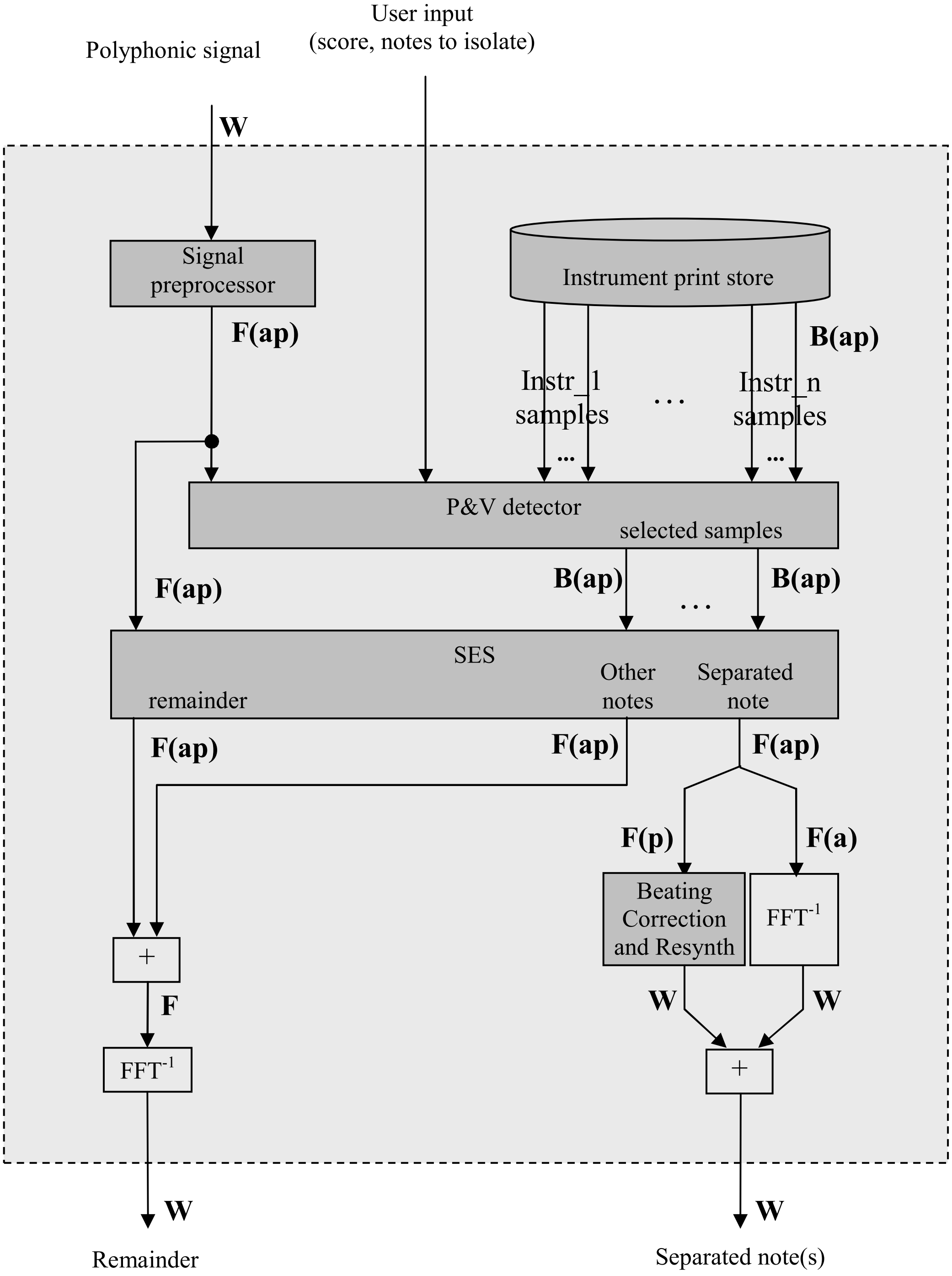


Figure 3: Signal flow and block diagram of the separation process

- **Beating-correction**: The SES output signals may suffer from beating, this step aims to eliminate this artifact by recognizing it and amplifying/resynthesizing the note signal where needed.

The instrument print model

I have established a model for storing important features of instruments in a way that it is useful later at the separation step. This model is called the *instrument print*.

An instrument print contains samples from an instrument on different frequencies and with different *playmodes*. For every sample it stores the amplitude characteristics of the notes on the different overtones as well as the amplitude characteristics of the noise component (which I called *bandogram*). In an ideal case it is a function that returns how the amplitude content changes over time

($r\tau$) at any specified frequency (*f*) for a note played on base frequency f^{base} if the specified playmode (**M̲**) is used:

$$\underline{\mathbf{A}}(\underline{\mathbf{M}}, f, f^{base}, t)$$

In reality this function is realized by interpolating the needed amplitude values from a number of presampled notes that are taken at different frequencies and playmodes. A typical instrument print contains 10-30 note samples. Each sample stores the sum of the amplitudes on the bins that are in a certain frequency range. This representation is called a *bandogram*. In a bandogram the periodic and aperiodic components are stored separately. The aperiodic component can be calculated as follows. The distance in subbands of two arbitrary frequencies can be calculated as:

$$b = \left\lfloor \log_{\sqrt[R]{2}} \frac{f^{base}}{f^{true}_{r\tau,k}} \right\rfloor \tag{14}$$

With the help of (14) let $\rho(f, f^{base}, b)$ denote a boolean function that is true if the frequency distance between *f* and f^{base} is exactly *b* subbands:

$$\rho\ (f, f^{base}, b) = \begin{cases} f^{base} \cdot 2^{\frac{b-0,5}{R}} < f < f^{base} \cdot 2^{\frac{b+0,5}{R}} : & 1 \\ otherwise: & 0 \end{cases}. \tag{15}$$

The bandogram of a note played on f^{base} base frequency is calculated as

$$A_{\mathbf{M}, f^{base}, b, r\tau} = \sum_{\rho^{aper}(f^{true}_{r\tau,k}, f^{base}, b)} c_{r\tau,k} \tag{16}$$

The periodic component can be expressed as:

$$A^{per}_{\mathbf{M}, f^{base}, o, r\tau} = \sum_{\rho^{per}(f, f^{base}, o)} c^{per}_{r\tau,k}\ , \tag{17}$$

where *o* identifies the overtone, and

$$\rho^{per}(f, f^{base}, o) = \begin{cases} f^{base} \cdot o - \Delta f < f < f^{base} \cdot o + \Delta f : & 1 \\ otherwise: & 0 \end{cases}, \tag{18}$$

is a function that is true if the frequency *f* is the o^{th} overtone of f^{base}. Minor deviations from f^{base} are allowed, and the maximum allowed deviation that is still considered an overtone is defined by

$$\Delta f = f_{base} \cdot \delta \tag{19}$$

with

$$\delta \approx \frac{\sqrt[12]{2} - 1}{2} \approx 0,029732 \tag{20}$$

where *δ* is an experimental value.

The Simplified Energy Split algorithm

I have proposed an algorithm for redistributing the energy in the source recording between the output channels, the separated note signals. The algorithm uses the simplified version of the separation problem that was proposed in Thesis I.

Assuming that the score and playmodes of the notes are known, I propose the following iterative algorithm to approximate the solution of (5) (to divide the amplitude between the target note signals). The energy of the recording is divided between the target note signals in the following way. We start out with the original Frequency Estimated Spectrogram of the recording. The separation is completed in *N* steps. In each step a fraction of the amplitude of the selected bandograms is transferred from the spectrogram of the recording to the spectrogram of the separated note signals. In an ideal case it is possible to transfer the required amplitude fraction in each step. However in a usual case the remaining amplitude in the original signal will decrease to zero before we could reach the last step due to the cancellations in the original recording. Nevertheless, this strategy ensures a fair division of the amplitude of the recording between the output note signals. Having *N* steps avoids cases when almost the full amplitude gets transferred to one output note, leaving no amplitude for the other.

Here I illustrate the proposed algorithm for the aperiodic part of the signal, however the periodic part can be handled in a very similar way. Figure 4 shows the skeleton of the pseudo-code that carries out the energy redistribution[3]. The inputs of the SES algorithm are:

- the (FE) spectrum of the recording (c),
- the chosen instrument prints (A_i),
- the step count *N* that determines the preciseness of the approximation.

```
//divide the amplitude in the original recording ($\underline{c}$) using prints ($\underline{A}$) in N steps
function SES($\underline{\mathbf{c}}, \underline{\mathbf{A}}_1, \underline{\mathbf{A}}_2, ..., \underline{\mathbf{A}}_I$, N)
    for each i (instrument) $\hat{\underline{\mathbf{s}}}_i = 0$
    for (n = 0 to N) TransferAmpToNotes($\underline{\mathbf{c}}, \hat{\underline{\mathbf{s}}}_1, \hat{\underline{\mathbf{s}}}_2, ..., \hat{\underline{\mathbf{s}}}_I, \underline{\mathbf{A}}_1, \underline{\mathbf{A}}_2, ..., \underline{\mathbf{A}}_I$, N)
    //residual is the amplitude remaining from the original recording
    $\hat{\underline{\mathbf{d}}} = \underline{\mathbf{c}}$
    //return the remaining amplitude and the note signals
    return $\hat{\underline{\mathbf{d}}}, \hat{\underline{\mathbf{s}}}_1, \hat{\underline{\mathbf{s}}}_2, ..., \hat{\underline{\mathbf{s}}}$

//transfer a fraction (N) of the amplitude from the recording (c) to the notes (s) using prints (A)
function TransferAmpToNotes($\underline{\mathbf{c}}, \hat{\underline{\mathbf{s}}}_1, \hat{\underline{\mathbf{s}}}_2, ..., \hat{\underline{\mathbf{s}}}_I, \underline{\mathbf{A}}_1, \underline{\mathbf{A}}_2, ..., \underline{\mathbf{A}}_I$, N)
    for each i (instrument)
        for each b (subband)
            //fraction of the amplitude indicated by the print
            neededAmpTotal = $\underline{\mathbf{A}}_{i,b}$ / N
            //total amplitude of bins related to subband b
            totalAmp = $\sum_{c_k \text{ is part of subband b}} c_k$
            //coefficient for calculating how much amplitude needs to be transferred.
            coeff = min (neededAmp / totalAmp , 1)
            for each k (bin) that is part of b (subband)
                transferredAmp = coeff * $c_k$
                $c_k$ = $c_k$ - transferredAmp
                $s_{i,k}$ = $s_{i,k}$ + transferredAmp
```

Figure 4.: Pseudo-code skeleton of the SES algorithm

The algorithm operates as follows.

1. In the beginning the output note signals ($\hat{\underline{\mathbf{s}}}_1$, $\hat{\underline{\mathbf{s}}}_1$,..., $\hat{\underline{\mathbf{s}}}_I$) contain no amplitude on any bin.
2. The amplitude content of the recording will be transferred to the output note signals in an iterative way, in *N* steps. After completing all the steps the original recording will contain only the residual, that is, amplitude content that could not be assigned to any of the output signals.

[3] As noted earlier the algorithm is presented here in a very basic form for better understanding. Thus, time, fundamental frequency and playmode indices are omitted.

(This may be perceived as very low level artificial noise by the listener.) The note signals (*child notes*) will contain amplitudes that together generate a note signal that is close to the instrument print used for the separation (*mother print*).

3. The actual amplitude splitting is done in function `TransferAmpToNotes`[4]. It is called *N* times during the operation of the SES. Its inputs are:
   - remaining amplitudes (FE spectrum) of the recording,
   - current spectrum of the output note signals,
   - the mother prints chosen for the separation,
   - the total number of steps.
4. Each run of `TransferAmpToNotes` iterates through the *i* outputs. For each output the algorithm tries to transfer some amplitude from the remaining part of the recording to the output. If the remaining part does not contain any more amplitude on the respective bins, the output is left unaltered in the current step (and all the forthcoming steps).
5. The actual amount of amplitude to transfer is available for each subband from the mother print. The print basically indicates the sum of how much amplitude its child note *should contain* in the different bands (summing the amplitudes of all the bins in the band).
6. The actual amount of amplitude currently present in the recording is calculated as a sum of amplitudes of bins that are part of the current (b) subband. A bin is part of a subband if its true frequency (not its nominal frequency!) falls in the subband.
7. In one step the algorithm tries to transfer amplitudes from bins of the recording to bins of the child note in a way that the sum of the individual transferred bin amplitudes equals a predefined fracture (1/N) of the sum amplitude indicated by the mother print. A coefficient is calculated to indicate how much of the amplitude of the bins (in the current subband) has to be transferred to the note in order that the sum of the transferred amplitude reaches the desired value. The coefficient is the same for all of the bins in a subband.
8. The right amount of amplitude is subtracted from the recording and added to the note signal. If there is not enough amplitude in the recording, then the maximum possible amount is transferred (leaving no amplitude on the bins of the recording).
9. The above procedure is repeated for all instruments, N times.
10. Finally the SES algorithm returns the residual (the leftover energy) and the note signals.

The beating correction and resynthesis algorithm

I have developed an algorithm for reducing the beating effect in the separated output channels. The algorithm exploits the fact that the amplitude envelope (as the amplitude changes over time) on the different overtones of a note signal is in an optimal case similar to the amplitude envelope stored in the respective instrument print. If this is not the case, is cancellation at the respective areas is to be suspected, where the note signal should be amplified back to the level indicated by the print.

Beating can be observed mostly in the periodic component. As the components are available separately (using the findings of Thesis II) it is possible to amplify this component at the respective areas independently from the aperiodic component.

$$\hat{s}^{per,beatCor}_{i,r\tau,k} = a_{i,o,r\tau} \cdot \hat{s}^{per}_{i,r\tau,k} \tag{21}$$

where $a_{i,o,r\tau}$ denotes the amplification factor, for which I proposed two strategies in my thesis:

[4] From the software developer's perspective it is important to note that the values of the inputs get changed by `TransferAmpToNotes`!

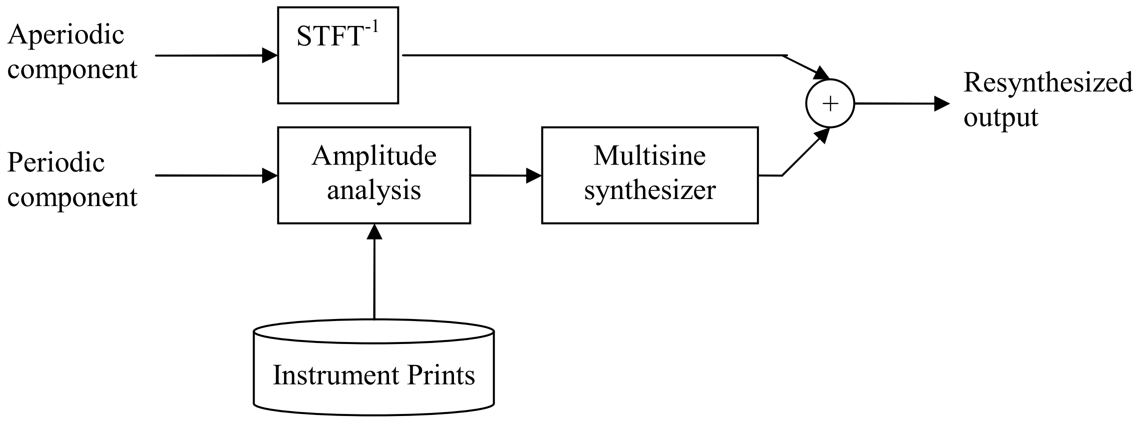


Figure 5: Block diagram of beating correction

$$a_{i,o,r\tau} = \frac{att^{print}}{att^{realNote}} \cdot a_{i,o,(r\tau-1)} \tag{22}$$

and

$$a_{i,o,r\tau} = \lambda \cdot \left(\frac{att^{print}}{att^{realNote}} \cdot a_{i,o,(r\tau-1)} \right) + (1-\lambda) \cdot \left(a_{i,o,(r\tau-1)} \right), \tag{23}$$

where att^{print} and $att^{realNote}$ stand for the attenuation of the print and the separated note, respectively.

In the case of total cancellation there is no energy at the respective areas of the periodic component. Thus, amplification of the existing energy content is not an option. However, the instrument prints hold enough information about the amplitudes on different overtones, thereby making a complete resynthesis of the periodic component possible. (This method also allows for many kinds of modifications like pitch shifting, formant adjusting etc.)

4. Application of the scientific results

As a possible application of the results I have implemented a sound source separation system for PC that made the verification of the results possible. In the dissertation the operation of certain theses are demonstrated by examples and comparisons to other methods. In this booklet I present an example for the operation of the separation system as a whole. This example can also be found in greater detail in the dissertation.

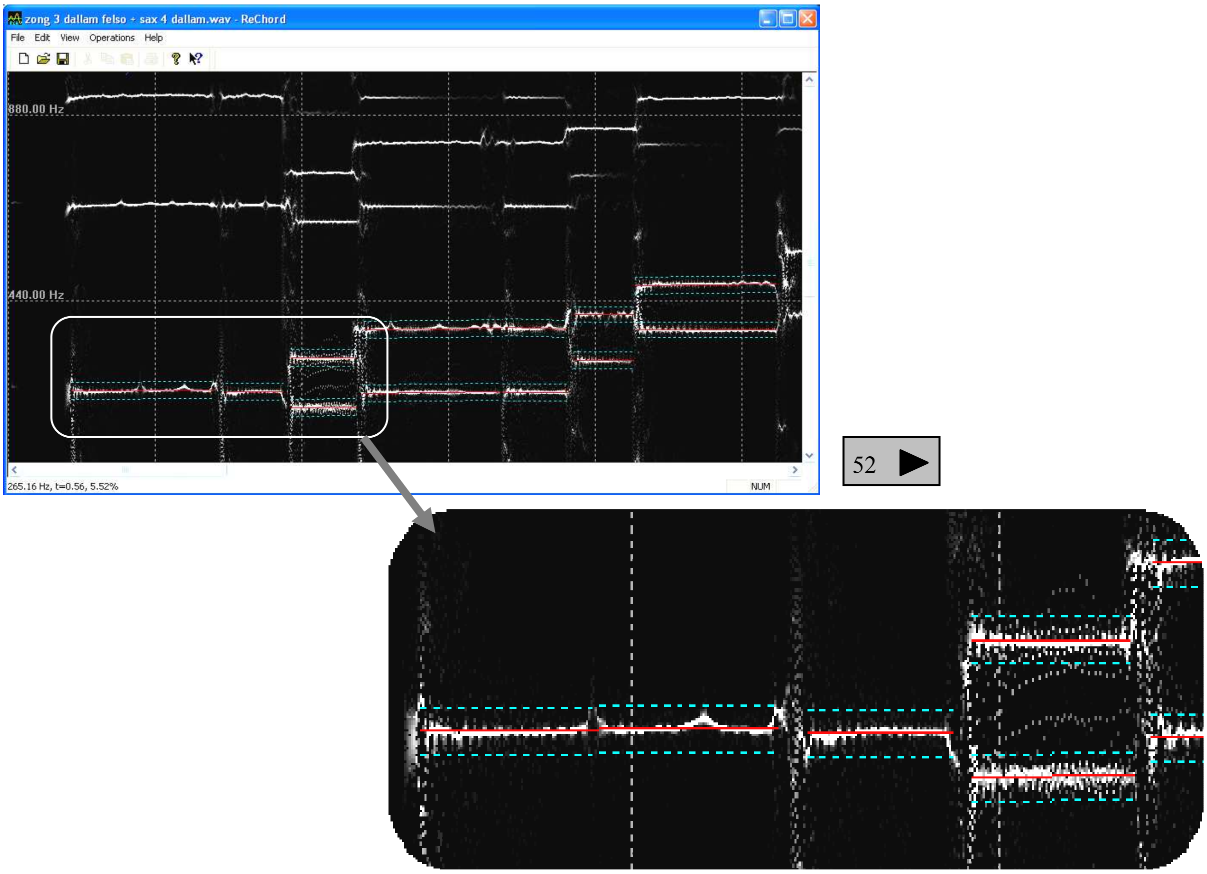

zong 3 dallam felso + sax 4 dallam.wav - ReChord
File Edit View Operations Help
880.00 Hz
440.00 Hz
265.16 Hz, t=0.56, 5.52%
NUM
52 ▶

Figure 6: The user interface of the ReChord system

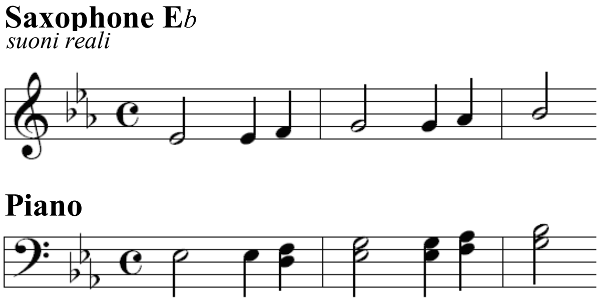


Figure 7: Sheet music of the musical piece opened in the application

Figure 6 shows the user interface of the ReChord system. A musical piece is loaded, the score of which is shown on Figure 7. The application draws the Frequency Estimated spectrogram of the loaded signal. This particular example contains two instruments. Their melodies were recorded separately and then mixed into one channel. The user can manually input the score of the loaded piece and the contained instruments.

After this step the application calculates the most probable playmodes for the instruments using the pre-stored instrument prints in its database. Figure 8 - 9 show a short part the original waveforms of the two instruments. Figure 10 depicts the waveform of the mixed signal while Figure 11 and 12 show the separated note signals.

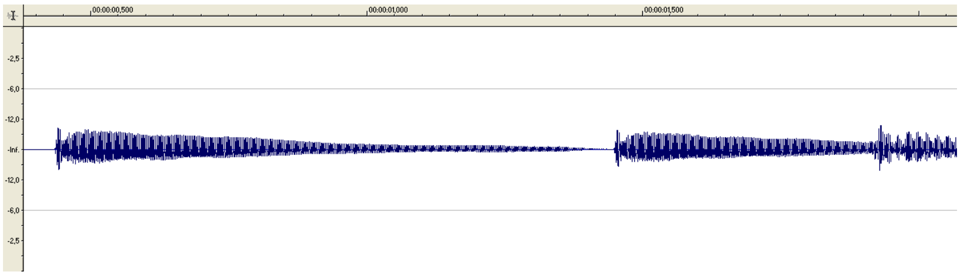


Figure 8: Waveform of the original piano soundtrack

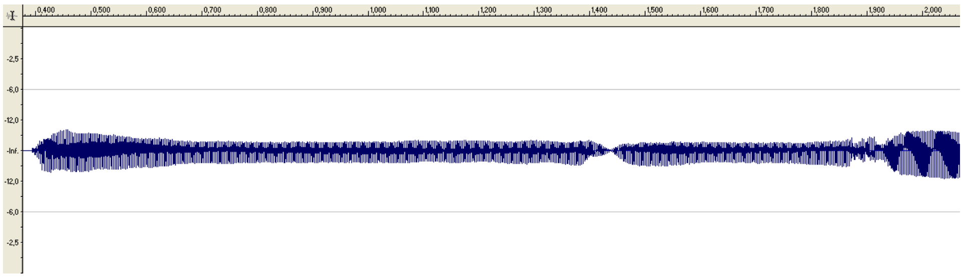


Figure 9: Waveform of the original saxophone soundtrack

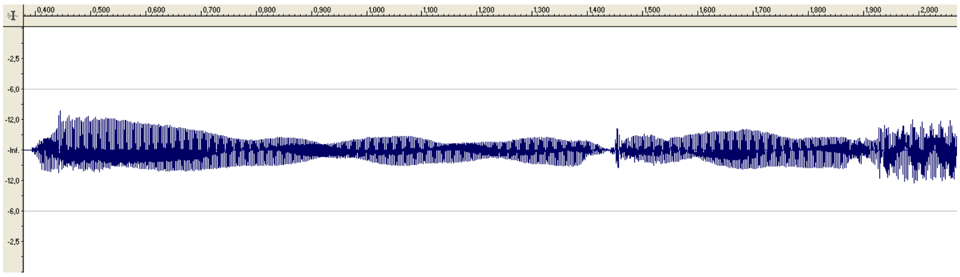


Figure 10: Waveform of the mixed soundtrack

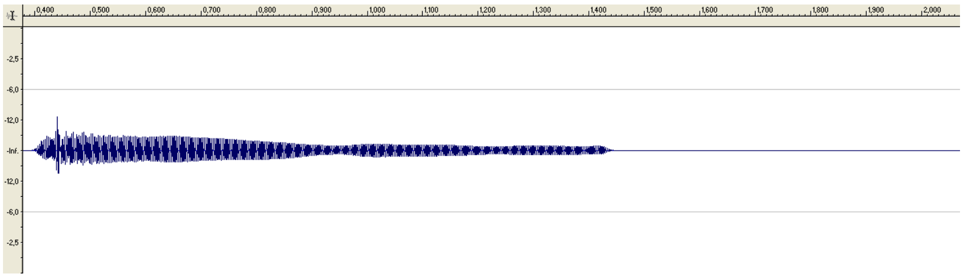


Figure 11: Waveform of the first separated piano note

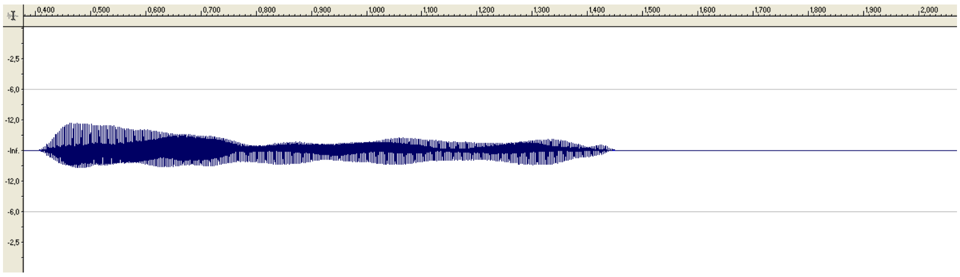


Figure 12: Waveform of the first separated saxophone note

These example waveforms as well as other waveforms of synthetic tests and real-life scenarios are included on the CD appendix and at http://avalon.aut.bme.hu/~aczelkri/separation. The ReChord system was also used for editing commercial recordings including '*Alexander Scriabin: Étrangeté' (played by Gábor Csalog, published by BMC Records, 2005)*' and '*Edison Denisov: Sonata (played by Gergely Ittzés (flute) and József Gábor (piano))*'.

5. Publications

[1] K. Aczél, Sz. Iváncsy, "Musical source analysis with DFT", *Proceedings of MicroCAD 2006*, Miskolc, Hungary, 2006, pp. I:1–6

[2] K. Aczél, Sz. Iváncsy, "Supporting pitch shifting: Sound separation using instrument prints", *Proceedings of MicroCAD 2006*, Miskolc, Hungary, 2006, pp. I:7–12

[3] K. Aczél, Sz. Iváncsy, "Instrument separation in polyphonic recordings using instrument prints", CSCS *2006*, Szeged, Hungary, 2006

[4] K. Aczél, Sz. Iváncsy, "Manipulation of musical recordings using instrument prints", *Proceedings of Automation and Applied Computer Science Workshop 2006*, Budapest, Hungary, 2006, pp. 143–152

[5] K. Aczél, Sz. Iváncsy, "Musical source analysis: spectrogram vs cochleagram", *Proceedings of MicroCAD 2007*, Miskolc, Hungary, 2007, pp. M:1–6

[6] K. Aczél, Sz. Iváncsy, "Semi-automatic sound separation of polyphonic music using instrument prints", *Proceedings of Automation and Applied Computer Science Workshop 2007*, Budapest, Hungary, 2007, pp. 217–228

[7] K. Aczél, Sz. Iváncsy, "Sound separation of polyphonic music using instrument prints", *Proceedings of* the *15th European Signal Processing Conference (EUSIPCO 2007)*, Poznan, Poland, 2007, pp. 931–935.

[8] K. Aczél, I. Vajk, "Note separation of polyphonic music by energy split", *Proceedings of WSEAS International Conference on Signal Processing, Robotics and Automation (ISPRA 2008)*, Cambridge, England, 2008, pp. 208–214.

[9] K. Aczél, I. Vajk, "Instrument prints in note separation of polyphonic music", *Proceedings of WSEAS International Conference on Signal Processing, Robotics and Automation (ISPRA 2008)*, Cambridge, England, 2008, pp. 215–220

[10] K. Aczél, I. Vajk, "The simplified energy split algorithm in the separation of polyphonic music", *Proceedings of MicroCAD 2008*, Miskolc, Hungary, 2008, pp. J:1–6

[11] K. Aczél, I. Vajk, "Polyphonic music separation: instrument prints detailed", *Proceedings of MicroCAD 2008*, Miskolc, Hungary, 2008, pp J:7-12

[12] K. Aczél, I. Vajk, "Separation of periodic and aperiodic sound components by employing frequency estimation", *Proceedings of* the *16th European Signal Processing Conference (EUSIPCO 2008)*, Lausanne, Switzerland, 2008, pp. P1-1-2

[13] K. Aczél, I. Vajk, "Beating correction and resynthesis in the SES-based polyphonic music separation system", *Proceedings of Automation and Applied Computer Science Workshop 2008*, Budapest, Hungary, 2008, pp. 75–88

[14] K. Aczél, I. Vajk, "Polyphonic music separation based on the Simplified Energy Splitter", *WSEAS Transactions on Signal Processing*, Vol. 4, No. 4, 2008, pp. 201–210

[15] K. Aczél, I. Vajk, "Polifonikus zenei felvételek hangjegy alapú szétválasztása", *Híradástechnika*, Dec. 2008, pp. 37-41

[16] K. Aczél, I. Vajk, "Note-based sound source separation of polyphonic recordings", *Híradástechnika*, Jan. 2009, pp. 36–40

[17] K. Aczél, I. Vajk, "Simple and powerful instrument model for the source separation of polyphonic music", *WSEAS Transactions on Acoustics and Music*, Vol. 5, No. 1, 2009, pp. 1–10

Submitted for publication

[18] K. Aczél, I. Vajk, "Instrument print aided sound source separation", *Periodica Polytechnica*