Towards Automatic Music Transcription: Extraction of MIDI-Data out of Polyphonic Piano Music

Jens WELLHAUSEN

RWTH Aachen University
Institute of Communications Engineering
52056 Aachen, Germany
E-mail: wellhausen@ient.rwth-aachen.de

ABSTRACT

Driven by the increasing amount of music available electronically the need of automatic search and retrieval systems for music becomes more and more important. In this paper an algorithm for automatic transcription of polyphonic piano music into MIDI data is presented, which is a very interesting basis for database applications and music analysis. The first part of the algorithm performs a note accurate temporal audio segmentation. The resulting segments are examined to extract the notes played in the second part. An algorithm for chord separation based on Independent Subspace Analysis is presented. Finally, the results are used to build a MIDI file.

Keywords: Music Transcription, Audio Segmentation, Independent Subspace Analysis

1. INTRODUCTION

Today's available audio database applications allow to retrieve music from a database on the basis of a few notes sung or hummed ("query by humming") as a very convenient human-machine-interface. To perform this task, a pice of music sung into a microphone is analyzed and transcribed into a set of notes. The well examined human vocal tract helps this step to be relatively easy. More difficult is the side of the database. As many publications in this field of research show, up to now there is no possibility to transcribe very different kinds of music into notes in an automatic way.

Concentrating on polyphonic music played by one instrument, i.e. one intrument playing several notes or chords at one time, is also an interesting task. For example, a musician who is composing by playing his instrument, could use an automatic transcription system to write down his work.

In this paper a technique of note-accurate temporal audio segmentation and MIDI-file generation is proposed, which is currently able to extract polyphonic piano sounds. First, the music is segmented into tone bricks. This segmentation process can also be the basis for many other applications, for example in the field of tempo analysis. After the segmentation, each segment is analyzed which notes are played. Using a priori knowledge, that a piano instrument is playing, polyphonic music can be transcribed. For the separation of chords and an easier note classification the Independent Subspace Analysis is used.

Related work can be found for the segmentation process [1], but also music transcription is an upcoming theme. An other piano music transcription system is presented in [2] and general music transcription is discussed in [3].

This paper is organized as follows. After the introduction the algorithm for note accurate audio segmentation is described in section 2. Both features in the time domain and features in the frequency domain are used. In section 3 the algorithm extracting notes played in each segment is introduced. At this time it is limited to polyphonic piano sounds. The generation of MIDI files is described in section 4. An approach to separate chords for a better note classification is presented in section 5. In section 6 results are discussed. Finally, a concluding summary is given in section 7.

2. SEGMENTATION INTO NOTE EVENTS

This part of the algorithm shall not be limited to piano music and is optimized for any audio sources, because it could be used in other applications, too, where mixed audio sources are examined. The segmentation into note events without knowing the kind of instruments playing has to be done using both features in the time domain and

features in the frequency domain of the audio signal. For example, if a piano is playing twice the same note consecutively, dealing with the properties in the frequency domain of the audio signal will give no result. The other side is a string instrument playing a set of different notes in one go. Here it is not possible to get significant results out of the course of the signal power. In the following, the usage of both frequency and signal power properties to extract segment boundaries are described.

2.1. Segmentation in the frequency domain

The basis for this processing step is a short time Fourier transform (STFT), which is performed for windowed and not overlapped analysis frames. For two consecutive analysis frames the cross-correlation coefficient of the spectrum is calculated. The resulting signal has values near 1 within a note segment and significant lower values marking segment boundaries, which are extracted by a search for relative minima. First, the spectrum of the signal is stored in an array v_i of vectors.

$$v_i = \begin{pmatrix} v_{i,1} \\ v_{i,2} \\ \vdots \\ v_{i,N} \end{pmatrix}; \qquad i = 1, \cdots, N \tag{1}$$

For two consecutive analysis frames the cross-correlation coefficient $\phi(i)$ is calculated.

$$\phi(i) = \sum_{k=1}^{N} \frac{v_i(k) \cdot v_{i+1}(k)}{\sqrt{p(i)p(i+1)}}; \qquad i = 1, \cdots, n-1 \tag{2}$$

To be independent of the overall signal power, the correlation is normed by the spectral energy $p(i)$ of each analysis frame.

$$p(i) = E_i = \sum_{k=1}^{N} v_i(k) \tag{3}$$

The threshold for the decision, whether a relative minimum marks a segment boundary or not, is set adaptive considering a neighborhood around the minimum. In Fig. 1 the signal $\phi(i)$, the varying threshold and the extracted segment boundaries are shown. At this step there may be some false alarms for segment boundaries when the signal power is too low or noisy, for example when one tone is decaying into silence. Because of that only these boundaries where the signal power is higher than a defined value are kept.

2.2. Segmentation in the time domain

The analysis frames for this step are the same as described in section 2.1. With Parseval's theorem the signal power

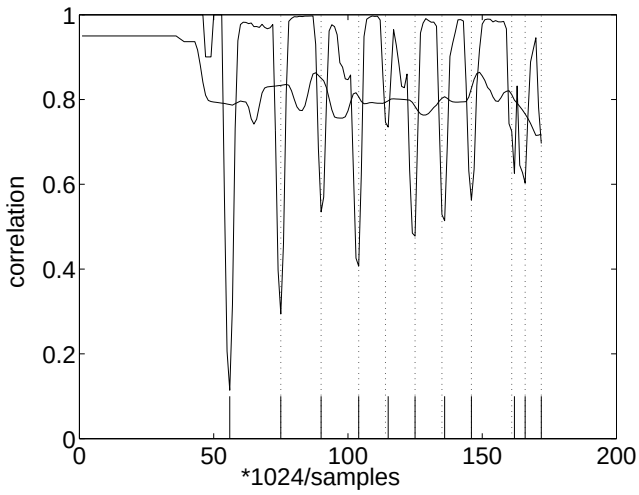

Fig. 1. Segmentation using an adaptive threshold

for an analysis frame is calculated by summing up all frequency coefficients. The waveform of the resulting power signal is smoothed by filters and to detect the beginnings of note segments, local maxima are searched. To be independent of the absolute signal power, first the derivative $d(i)$ of the power signal is calculated.

$$d(i) = E_i - E_{i+1} = \sum_{k=1}^{N} v_i(k) - \sum_{k=1}^{N} v_{i+1}(k); \quad i = 1, \cdots, n-1 \tag{4}$$

Local maxima within the power signal are now represented by zero crossings of $d(i)$ from positive to negative values, which can be easily extracted. A threshold, which compares the absolute values of the derivative before and after the zero crossing, is used to decide whether a local maximum of the power curve represents the beginning of a new note or not.

At this point it is necessary to take into account that the maximum of the signal power does not represent the onset time. The first part of the audio envelope, the attack time, has to be estimated. In the preceding surrounding of the maximum the local minimum is searched, which marks the actual beginning of the tone and thus a segment boundary.

2.3. Benchmarking the segmentation

The results achieved in the processing steps described in sections 2.1 and 2.2 are now merged together using a logical OR-operation. If one segment boundary is detected by both processes, there may be a little difference in the detected time. This can obviously be ascribed back to the imprecision of the attack time estimation. In these cases

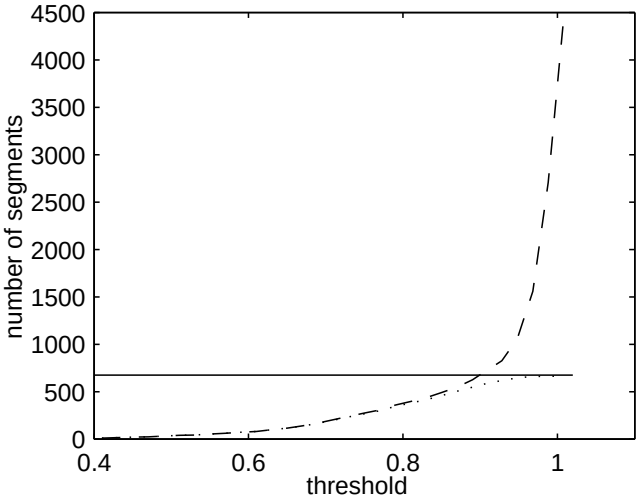

Fig. 2. Trade-off between finding all segments and over-segmentation

the boundary detected by the segmentation using the correlation analysis is used.

In the segmentation process described above are some parameters used, whose influence has to be discussed to achieve good segmentation results. As shown in Fig. 2, there is a trade off between finding all true segment boundaries and over segmentation, i.e. finding false positive results, depending on the application. The solid line shows the correct number of segments, the dashed line represents all found segments whereas the dotted line shows the true segments found. For the task of music transcription, over segmentation is less worse than ignoring segment boundaries, because the following note extraction algorithm is able to detect notes sounding over several segments. To achieve an optimal set of parameters for the segmentation, a cost function was defined using equation 5.

$$q = \frac{correctsegments}{totalsegments} - b \cdot \left(\frac{foundsegments}{correctsegments} - 1 \right)^2 \quad (5)$$

An other important property of a note segmentation algorithm is the time resolution. Perceptual tests have shown, that with up to 40*ms* between the onsets of two notes, they are heard synchronous at the same time. For more than two notes, this time is getting even longer up to 70*ms* [4].

3. NOTE EXTRACTION

In this section the note extraction process is described. Into note events segmented audio streams are input data for the now following note extraction algorithm.

3.1. Piano Sounds in the frequency domain

In this section, properties of piano sounds in the frequency domain are discussed, which are used to identify notes within a segment. First, the fundamental frequencies of notes played by a piano can be calculated relative to each other using equation (6).

$$f_{i+1} = 2^{\frac{1}{12}} f_i \quad (6)$$

Usually, the standard pitch *A* with its fundamental frequency of $f_A = 440Hz$ is used as starting point. The claviature of a piano has a range of max. 88 half-tones. This results in a range of possible fundamental frequencies from $27.5Hz$ to $4186Hz$.

The second important aspect is the intensity of the harmonics of a tone. The intensities of harmonics have a significant influence to the timbre of a sound. For a reliable detection of notes, the harmonics of a tone must be taken into account as well. The arrangement of the intensities of the harmonics depends on the absolute value of the fundamental frequency. For fundamental frequencies higher than around the standard pitch *A*, the power of each harmonic is lower than the power of the fundamental frequency and, going to higher frequencies, decreasing. On the other side, for lower fundamental frequencies, the power of the first harmonics may be higher than the power of the fundamental frequency itself. The reason for this may be found in the construction of a piano [5]. The length of the strings within a piano is decreasing from lower notes to higher notes, while the striking position of the sledge has a constant distance from one end of the strings. When the striking position relative to the string length is not constant, the strings are activated to oscillate in different modes.

3.2. Examining the spectrum

The properties of piano sounds in the frequency domain described above are used to extract notes. The basis for this examination is a Fourier transform to get a signal representation in the frequency domain. Because of the segmentation of the signal before, it can be assumed that audio signal is stationary except of the signal power. To get the highest possible frequency resolution, the Fourier transform is calculated for the whole segment as one block. Each five percent at the beginning and at the end of a segment are not used due to possible segmentation inaccuracies.

In the next step, the frequency bins resulting out of the Fourier transform are grouped into sub bands using the

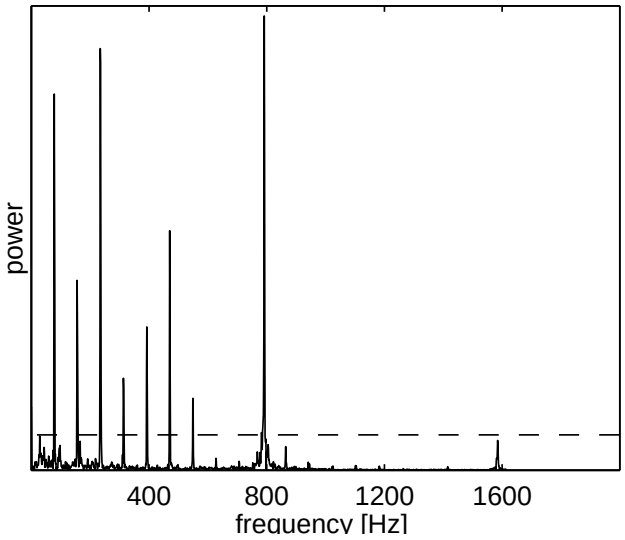

Fig. 3. Building a list of possibly played notes in the frequency domain

possible fundamental frequencies calculated by equation 6 as center frequency f_m. The boundaries for the sub bands are determined using equations 7.

$$f_1 = 2^{\frac{1}{24}} f_m, \qquad f_2 = 2^{-\frac{1}{24}} f_m \tag{7}$$

Within each sub band the frequency bin with a power maximum is searched. Because of the frequency resolution of the Fourier transform the power of a harmonic may be broadened over several frequency bins. Two additional frequency bins at each side are added in order to loose no power of a harmonic.

The result of this step is a set of power maxima, one maximum within each sub band. To be independent of the loudness of the audio source, a threshold is calculated using these power maxima. Each maximum, which is higher than this threshold, is stored in a set of possibly played notes. Each maximum below this threshold is not used any more. In Fig. 3 this threshold is represented by the dashed line.

The most important problem now is the decision, whether a maximum is the fundamental frequency of a note or one of its harmonics. To cope with this decision, first our system was trained with piano music to learn the arrangement of harmonics depending on the fundamental frequency. Taking the first position in the list of possibly played notes, the pattern of expected harmonics is compared with the next elements within the list. If an item in the list has equal or less power than expected, the item is declared as harmonic and deleted from the list of possibly played notes. This algorithm is done for all remaining items in the list. At the end, only list items representing fundamental frequencies of played notes should remain within this list.

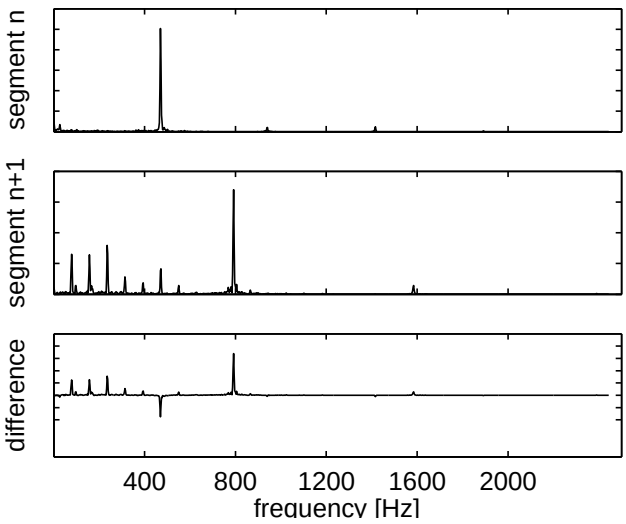

Fig. 4. Examination of consecutive analysis frames to detect notes sounding longer than one segment

The loudness of a note is determined by adding the power of the fundamental frequency and a defined number of harmonics.

4. GENERATING A MIDI FILE

The objective of this paper is to generate a MIDI representation out of an audio file. In this case, the instrument is known as a piano. The following items are necessary to build a MIDI file:

- Start time and end time
- MIDI note number
- Loudness

The start time and end time of a note are given by the segment boundaries found by the segmentation process described in section 2. Due to the temporal segmentation into single note events, it is possible that one tone sounds over several segments, whereas other notes start and stop. The detection of these long sounding notes is done by an examination of the difference between two analysis frames in the frequency domain. An example is given in Fig. 4. Negative values within the difference of two frames show that the corresponding note decays, even if the current sub band is a harmonic of an other fundamental frequency now.

The MIDI note value is determined by a look-up table, where each MIDI note number is assigned to a note on the claviature [6]. Finally, the loudness of a note is determined as described in section 3.2 by summing the power of all identified harmonics to the power of the fundamental frequency.

5. CHORD SEPARATION USING INDEPENDENT SUBSPACE ANALYSIS

An other approach to classify notes sounding in chords is to separate them using Independent Subspace Analysis (ISA) beforehand. The basis for this examination is the assumption, that all sounding notes within polyphonic music are statistically independent from each other and can be separated using this property. A well-known algorithm for signal separation is the Independent Component Analysis (ICA), which is the basis for the ISA.

5.1. Independent Component Analysis

Assuming that there are n statistically independent audio sources $s(t) = [s_1(t), \cdots, s_n(t)]^T$ and n microphones $m(t) = [m_1(t), \cdots, m_n(t)]^T$ recording the auditory scene, the relationship between $s(t)$ and $m(t)$ can be defined using a mixing matrix A as follows:

$$m(t) = A \cdot s(t) \tag{8}$$

In the case that there are the same number of microphones as signal sources, the separation of statistically independent sources performs well. An un-mixing matrix $W \approx A^{-1}$ is calculated iteratively to maximize statistic independence for the output signals $y(t) = [y_1(t), \cdots, y_n(t)]^T$.

$$y(t) = W \cdot m(t) = W \cdot A \cdot s(t) \tag{9}$$

The result of independent components is achieved, when

$$p(y) = \prod_{i=1}^{N} p(y_i) \tag{10}$$

where $p(y_i)$ is the probability density function (PDF) of y_i and $p(y)$ is the joint PDF of y. More detailed descriptions of the ICA can be found in [7] and especially in [8].

For many applications in the field of audio classification the ICA is inapplicable because of the number of required microphones and the unknown number of components, especially if already recorded data has to be examined. In this case, both the number of components and the mixing matrix are unknown and in most cases only one audio stream is available.

5.2. Independent Subspace Analysis

To exceed the limits of the ICA, the Independent Subspace Analysis (ISA) is a powerful tool, because it tries to segregate polyphonic audio sources recorded in one audio stream into their components. Related work based on Independent Subspace Analysis reports that this algorithm for separating sound into its components has still difficulties in the sound quality of the resulting components, but this does not influence the succeeding note classification presented in this paper.

To perform the ISA, the basis of the input space is changed before a canonical ICA algorithm is employed. A one-channel audio stream of size $1 \times N$, which is one segment after the temporal segmentation in our algorithm, is first projected onto a new basis using a windowed Short Time Fourier Transform (STFT) to yield a spectrogram S of dimension $n \times m$, where n is the number of frequency bins and m is the number of time slices. The dimension of this new multidimensional manifold is reduced by performing a Singular Value Decomposition (SVD) on the covariance matrix C of the input spectrogram.

$$C = U \cdot D \cdot V^T \tag{11}$$

The diagonal matrix D contains the eigenvalues σ_i of C in descending order, which are a measurement for the significance of the belonging spectral components. For dimension reduction, only the first r eigenvalues representing information up to a threshold Φ are used for the next steps.

$$\frac{\sum_{i=1}^{r} \sigma_i}{\sum_{i=1}^{m} \sigma_i} \geq \Phi \tag{12}$$

In matrix $\tilde{D}$, all eigenvalues σ_i with $i > r$ of matrix D are set to zero. By this operation some information is lost, but this does not affect the designation of this work, the classification of notes in audio sources. The input spectrogram is projected onto this new basis to get a dimension-reduced input space X of dimension $n \times r$.

$$X = \tilde{D} \cdot V^T \cdot S^T \tag{13}$$

This new input space X is used as input data for an ICA, the inverse W of the mixing matrix of the audio sources is estimated. This un-mixing matrix is multiplied against the dimension reduced basis vectors from the spectrogram projection to get the independent components of the audio source oriented in time.

$$F = W \cdot X; \qquad T = F^{-1} \cdot S \tag{14}$$

Finally, individual subspace spectrograms $V_i(n)$ are calculated out of the inverse of the mixing matrix and the basis vectors for each separated component.

$$V_i(n) = f_i \cdot t_i^T; \qquad i = 1, \cdots, r \tag{15}$$

These spectrograms may be transformed into the time domain using an inverse STFT to achieve the independent components in the time domain, but here the spectrograms itselves are examined in the next step.

5.3. Analyzing Segments

One tone played by an instrument consists of the fundamental frequency and its harmonics. Usually, the ISA results in more independent components than simultaneously played notes, because not all harmonics are separated to the component containing their fundamental frequencies. Some harmonics are separated in components of their own.

An grouping algorithm tries to estimate which components may be merged together to get a fundamental frequency and its harmonics into one component for further classification. First, for each component a probability function $p_i(n)$ out of the spectrum $f_i(n)$ of each component is calculated.

$$p_i(n) = \frac{f_i(n)}{\sum_{j=1}^{n} f_i(j)}; \qquad i = 1, \cdots, r \tag{16}$$

Now, the Kullback-Leibler divergence $KL(p_i(n), p_j(n))$ between all components $p_i(n)$ and $p_j(n)$ is calculated.

$$\begin{aligned} KL(p_i(n), p_j(n)) \quad &= \\ \frac{1}{2}\sum_n p_i(n) log \frac{p_i(n)}{p_j(n)} \quad &+ \quad \frac{1}{2}\sum_n p_j(n) log \frac{p_j(n)}{p_i(n)} \end{aligned} \tag{17}$$

If the divergence is smaller than a given threshold, components are added together. This is done for all components until the divergences between all components are bigger than the threshold. The result of this algorithm is a set of components containing single notes consisting of a fundamental frequency an its harmonics.

In the next step, for each component the frequency bins are grouped into subbands using possible fundamental frequencies calculated by equation (18) as center frequency f_m, using the standard pitch A with its fundamental frequency of $f_A = 440Hz$ as starting point.

$$f_{m+1} = 2^{\frac{1}{12}} f_m \tag{18}$$

The boundaries f_1 and f_2 for the sub bands are determined using equation (19), and within each sub band the frequency bin with maximum power is searched.

$$f_1 = 2^{\frac{1}{24}} f_m, \qquad f_2 = 2^{-\frac{1}{24}} f_m \tag{19}$$

a)

Title	right positive	false positive
Chopin-Nocturnes	71.3 %	36.9 %
Satie-Gymnopedie	71.2 %	10.5 %

b)

Title	right positive	false positive
Chopin-Nocturnes	67.6 %	19.4 %
Satie-Gymnopedie	70.8 %	10.3 %

Table 1. Results for detected notes
(a) using all segments (b) using segments >500ms

An example of a separation of a chord is given in Fig. 5, which shows a segment containing two notes played at the same time. In Fig. 5 a) the segment is presented in the time domain and in the frequency domain. In the frequency domain, the two notes consisting of fundamental frequencies and harmonics can be seen. The note with lower fundamental frequency has a characteristic set of harmonics, whereas the note with higher fundamental frequency has one small harmonic in the presented frequency range. In Fig. 5 b) and c) both separated notes are showed. In the frequency domain can be seen, that the assignment of harmonics to fundamental frequencies after the separation of all statistically independent components works well.

6. RESULTS

To evaluate the results of the algorithm, piano music from compact disc recordings was used. The results of the automatic transcription were compared with hand-crafted transcriptions. Table 1 (a) shows the results for two pieces of piano music, no matter how long the segments out of the segmentation process are. The first one - Chopin's Nocturnes - is characterized by a high tempo. Too short segments can not properly be examined, because the frequency resolution is too low within short segments. This leeds to a high number of false positive detected notes. The second one - Satie's Gymnopedie - is characterized by lower tempo resulting in longer segments, which can be better analyzed due to higher frequency resolution. In table 1 (b) the results for the same pieces of music are shown if segments shorter than $500ms$ are neglected. The number of right detected notes does not change, but the number of false positive detected notes decreases significantly for Chopin's Nocturnes, whereas for the slower piece of music the results are not changed. Keeping in mind that very short segments occur due to segmentation errors, it is suggestive to neglect segments shorter than a given length.

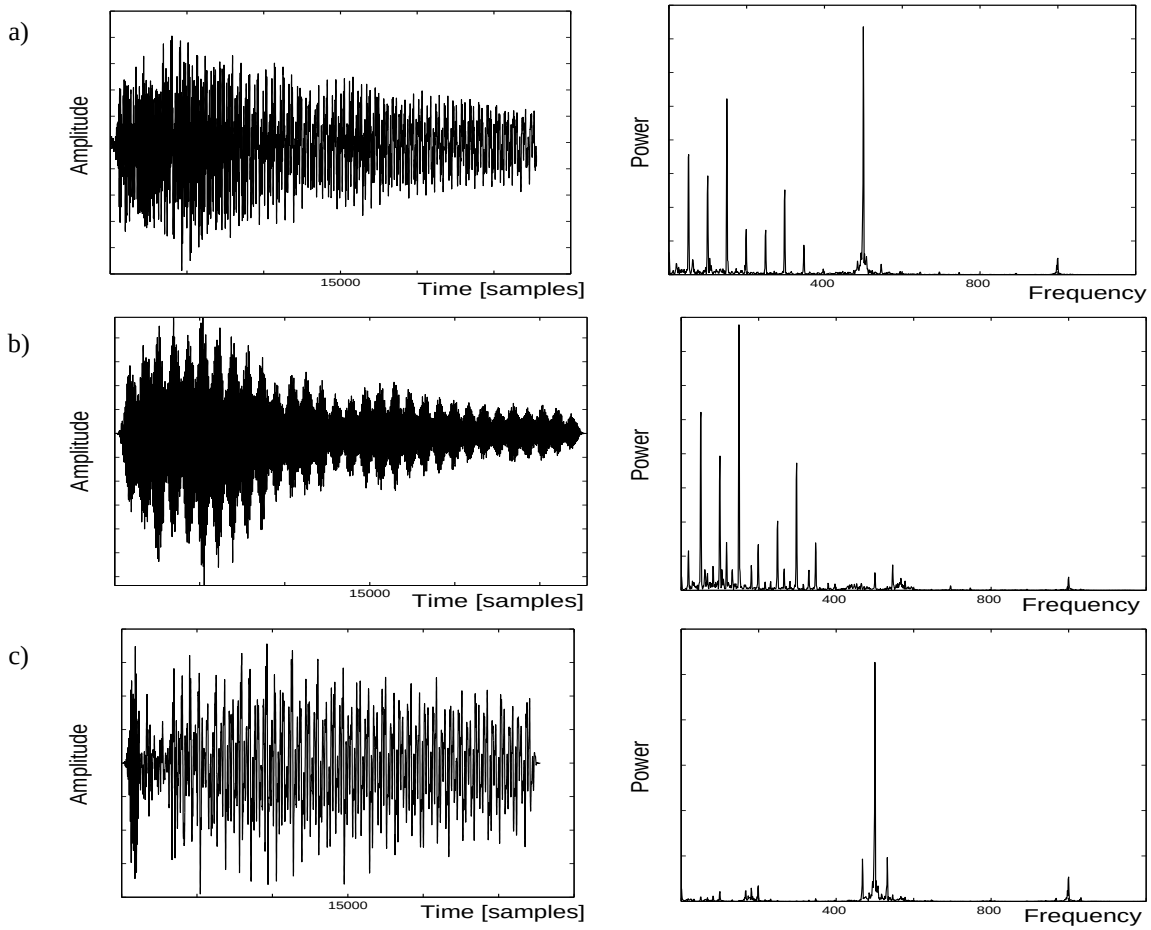

Fig. 5. a) Segment with two notes played simultaneously in the time domain and in the frequency domain.
b) and c) Separated notes after grouping of harmonics

7. CONCLUSION

In this paper an algorithm for automatic transcription of polyphonic piano music into MIDI data is discussed. The process is divided into two parts. First, the audio stream is segmented into note events in the time domain. The second step is the analysis of each segment in the frequency domain to extract which notes are played. Using the results of both processing steps sufficient data is available to build a MIDI file. For piano music played by one instrument this algorithm performs well. Problems arise when there is background noise or even if there is more than one instrument playing at one time. A very promising approach to cope with sound mixes is the *Independent Subspace Analysis* (ISA). The ISA could be used as a preprocessing step to separate different audio sources and chords into single notes.

8. REFERENCES

[1] S. Dixon, "Learning to detect onsets of acoustic piano tones," in *MOSART Workshop on Current Research Directions in Computer Music*, Barcelona, Spain, 2001.

[2] S. Dixon, "On the computer recognition of solo piano music," *Mikropolyphonie*, vol. 6, 2000.

[3] A. Klapuri, "Automatic transcription of music," M.S. thesis, Tampere University of Technology, 1998.

[4] S. Handel, *Listening: An Introduction to the Perception of Auditory Events*, Bradford, MIT Press, 1989.

[5] Anders Askenfelt, Ed., *The Acoustics of the Piano*, Royal Swedish Academy of Music, 1990.

[6] "Midi specifi cation 1.0," http://www.ibiblio.org/emusic-l/info-docs-FAQs/MIDI-doc/index.html.

[7] Lucas C. Parra, "An introduction to independent component analysis and blind so urce separation," Tech. Rep., Sarnoff Corporation, 1999.

[8] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja, *Independent Component Analysis*, John Wiley & Sons, 2001.