ALGORITHM FOR THE SEPARATION OF HARMONIC SOUNDS WITH TIME-FREQUENCY SMOOTHNESS CONSTRAINT

Tuomas Virtanen

Institute of Signal Processing, Tampere University of Technology
P.O.Box 553, FIN-33101 Tampere, Finland
tuomas.virtanen@tut.fi

ABSTRACT

A signal model is described which forces temporal and spectral smoothness of harmonic sounds. Smoothness refers to harmonic partials, the amplitudes of which are slowly-varying as a function of time and frequency. An algorithm is proposed for the estimation of the model parameters. The algorithm is utilized in a sound separation system, the robustness of which is increased by the smoothness constraints.

1. INTRODUCTION

Most audio analysis systems which operate in frequency domain are completely frame-based. It is, the parameters are estimated independently in each frame. To obtain temporal continuity, some post-processing can be done for the estimated parameters. This kind of bottom-up approach may work well in some applications, but it is clear that constraining dependencies between the frames already during the core estimation can increase the robustness of a system.

In this paper, a signal model is proposed which forces time-frequency smoothness of the parameters by representing them as a linear combination of pre-defined basis functions. Previously presented frequency-domain models [1] are extended into time-domain.

The signal model is used as a core of a sound separation system, allowing robust estimation of the amplitudes of harmonic components. Because there is dependency of parameters between frames, the least-squares solution has to be calculated for several frames at the same time. Basically all the parameters of one sound have to be estimated simultaneously. Since the usual least-squares solution is computationally too complex for all notes at time, an efficient algorithm is proposed for an approximation of the solution.

2. THE SIGNAL MODEL

The k^{th} frame $s^k(t)$ of a signal s is expressed as a sum of harmonic sounds and residual. The harmonic sounds are expressed as a sum of sinusoids:

$$s^k(t) = \sum_{n=1}^{N} \sum_{h=1}^{M_n} a_{n,h}^k \cos(2\pi t \omega_{n,h}^k + \theta_{n,h}^k) + r^k(t) \qquad (1)$$

where t is time, N is the total number of sounds, M_n is the number of harmonic components of n^{th} sound, and a_n^n, $\omega_n^n(k)$, and $\theta_n^h(k)$ are the amplitudes, frequencies and phases of the h^{th} component, respectively. $r^k(t)$ is the residual which is not representable with sinusoids. The onset and offset times ($t_n^{(0)}$ and $t_n^{(1)}$) of each sound are assumed to be known. The amplitudes of harmonic components of each sound are zero outside the interval $[t_n^{(0)}, t_n^{(1)}]$, and non-negative inside the interval.

For sinusoids, the frequencies of which are not close to each other, the parameters can be easily estimated. The problem is that for polyphonic music signals, the number of sounds may be large and several sinusoids are overlapping with each other. Also, harmonic relations are preferred in music, which further increases the number of components which coincide in frequency. The parameters of overlapping components can be estimated only by making some further assumptions concerning the parameters.

2.1. Linear models for overtone series

Spectral smoothness is one of the features that the human auditory system uses in grouping spectral components to sound sources [2]. Most natural sounds have a smooth spectrum. This was utilized by Virtanen & Klapuri [1] in a sound separation system, in which the overlapping harmonic components were resolved using linear models for the overtone series. The linear models force spectral smoothness and allow the estimation of overlapping components.

The fundamental idea of linear models is the following: instead of estimating the amplitudes, the estimation is done for a parameter vector $\boldsymbol{y}_n$, which is a lower-dimensional linear projection of the amplitudes:

$$\boldsymbol{a}_n^k = \boldsymbol{X}_n \boldsymbol{y}_n^k \qquad (2)$$

where $\boldsymbol{a}_n^k$ is the amplitude vector containing amplitudes $a_{n,1}^k..a_{n,M_n}^k$ of sound n in frame k, $\boldsymbol{X}_n$ is the transform matrix, and $\boldsymbol{y}_n^k$ is the parameter vector. Using this procedure, the amplitudes of harmonic partials are represented as a linear combination of columns of $\boldsymbol{X}_n$:

$$a_{n,h}^k = \sum_{i=1}^{I_n} (\boldsymbol{X}_n)_{h,i} y_n^{k,i} \qquad (3)$$

where I_n is the number of columns of X_n, which is smaller than M_n, the number of harmonic components. There are several possible structures for the matrix $\boldsymbol{X}$. Earlier simulations showed at a good choice for $\boldsymbol{X}$ is a structure which simulates a critical-band filter bank [1]. The structure of this kind of matrix is illustrated in Figure 1. Some other models are discussed in Section 4.

2.2. Linear models for temporal evolution

The robustness of parameter estimation can be increased by applying the linear models also in time domain. In addition to the within-frame constraint of Equation 3, similar restriction is placed for the temporal evolution of each harmonic component:

$$a_{n,h}^k = \sum_{j=1}^{J_n} z_{n,j}^k c_{n,h}^j \tag{4}$$

where $z_{n,j}^k$ is the j^{th} time-domain basis function and J_n is the number of functions.

Similarly to the frequency-domain model, triangular basis functions are used also for the time-domain model. These basis functions result in amplitude spectrum which is piece-wise linear as a function of time and frequency. Davy and Godsill [3] have earlier modelled the time-varying amplitudes of harmonic components using Hanning windows. They used Bayesian analysis of parameters, which allows the usage of a prior distributions of parameters. However, the system was reported to be computationally very slow.

3. APPLICATION TO SOUND SEPARATION

The described signal model is used in a separation system for harmonic sounds. The system is initialized using a multipitch estimator (MPE), which estimates the number of sounds and their fundamental frequencies in large frames [4].

The exact time-varying frequencies and amplitudes of the components are analysed using an iterative approach. Starting from the estimates given by the multipitch estimator, the accuracy of the parameters is improved in the least-squares sense, retaining the harmonic structure of the sounds. Each iteration consists of amplitude and frequency estimation steps. The frequency estimation is exactly similar to the one described in [5], independent of the amplitude estimation. The amplitude estimation algorithm is described in the following sections.

3.1. Least-squares solution

The phase of each frequency component is estimated directly from the phase spectrum of the original mixed signal. The model for each vector $\boldsymbol{s}^k$ of samples in the time frame k can be expressed as:

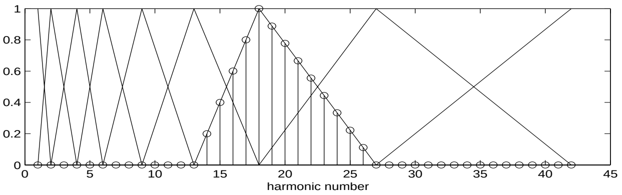

Figure 1: *Triangular basis functions of the frequency-band model. Each line corresponds to one column of the transform matrix X, the values outside the triangle being zero. The 7th column is plotted using stems and circles, each circle corresponding to one harmonic component.*

$$\boldsymbol{s}^k = \sum_{n=1}^{N} \boldsymbol{H}_n^k \boldsymbol{a}_n^k + \boldsymbol{r}^k \tag{5}$$

where each column of $\boldsymbol{H}_n^k$ corresponds to one sinusoid:

$$(\boldsymbol{H}_n^k)_{i,t} = \cos(2\pi\hat{\omega}_n^k t + \hat{\theta}_n^k), \tag{6}$$

$$t = 1 \ldots T \tag{7}$$

where $\omega_n^h(k)$, and $\theta_n^h(k)$ are the current frequency and phase estimates, respectively. By using the frequency-domain constraint of Equation 3, we get

$$\boldsymbol{H}_n^k \boldsymbol{a}_n^k = \boldsymbol{H}_n^k \boldsymbol{X}_n^k \boldsymbol{y}_n^k \tag{8}$$

$$= \boldsymbol{G}_n^k \boldsymbol{y}_n^k \tag{9}$$

where

$$\boldsymbol{G}_n^k = \boldsymbol{H}_n^k \boldsymbol{X}_n^k \tag{10}$$

Equation 5 can now be written as:

$$\boldsymbol{s}^k = \sum_{n=1}^{N} \boldsymbol{G}_n^k \boldsymbol{y}_n^k + \boldsymbol{r}^k \tag{11}$$

which can be written for all the frames 1..K by:

(12)

$$\boldsymbol{s} = \sum_{n=1}^{N} \boldsymbol{G}_n \boldsymbol{y}_n + \boldsymbol{r} \tag{13}$$

where $\boldsymbol{s}$, $\boldsymbol{r}$ and $\boldsymbol{y}_n$ are the frame-wise vectors concatenated into a longer vector:

$$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{s}^1 \\ \boldsymbol{s}^2 \\ : \\ \boldsymbol{s}^K \end{bmatrix}, \ \boldsymbol{y}_n = \begin{bmatrix} \boldsymbol{y}_n^1 \\ \boldsymbol{y}_n^2 \\ : \\ \boldsymbol{y}_n^K \end{bmatrix}, \text{ and } \boldsymbol{r} = \begin{bmatrix} \boldsymbol{r}^1 \\ \boldsymbol{r}^2 \\ : \\ \boldsymbol{r}^K \end{bmatrix} \tag{14}$$

and $\boldsymbol{G}$ is a matrix in which the frame-wise matrices $\boldsymbol{G}^k$ are on the diagonal:

$$\boldsymbol{G}_n = \begin{bmatrix} \boldsymbol{G}_n^1 & & 0 \\ & \boldsymbol{G}_n^2 & \\ & & \ldots \\ 0 & & \boldsymbol{G}_n^K \end{bmatrix} \tag{15}$$

From linear dependence of $\boldsymbol{a}$ and $\boldsymbol{y}$ it follows that the time-domain dependence similar to Equation 4 can also be written for the parameters $\boldsymbol{y}$:

$$y_{n,i}^k = \sum_{j=1}^{J_n} z_{k,n}^j d_n^{i,j} \tag{16}$$

where $d_n^{i,j}$ is the gain of time-frequency basis function $z_{k,n}^j$. This can be expressed for all frames by a single matrix operation $\boldsymbol{y}_n = \boldsymbol{T}_n \boldsymbol{d}_n$. The elements of $\boldsymbol{y_n}$ correspond to gain of j^{th} basis function of n^{th} sound in k^{th} frame through the following indexing:

$$(\boldsymbol{y}_n)_{(k-1)I_n+i} = y^k_{n,i} \qquad (17)$$

The I_nK by I_nJ_n matrix $\boldsymbol{T}$ has the following structure:

$$\boldsymbol{T}_n = \begin{bmatrix} \mathrm{diag}(\boldsymbol{z}_{1,n}) & \mathrm{diag}(\boldsymbol{z}_{1,n}) & .. & \mathrm{diag}(\boldsymbol{z}_{1,n}) \\ \mathrm{diag}(\boldsymbol{z}_{2,n}) & \mathrm{diag}(\boldsymbol{z}_{2,n}) & .. & \mathrm{diag}(\boldsymbol{z}_{2,n}) \\ : & : & & : \\ \mathrm{diag}(\boldsymbol{z}_{K,n}) & \mathrm{diag}(\boldsymbol{z}_{K,n}) & .. & \mathrm{diag}(\boldsymbol{z}_{K,n}) \end{bmatrix} \qquad (18)$$

where vector $z_{k,n}$ has the elements $z^j_{n,k}$, j=1..J_n. Vector $\boldsymbol{d}_n$ is:

$$(\boldsymbol{d}_n)_{jI_n+i} = v_n^{i,j} \qquad (19)$$

Let us denote

$$\boldsymbol{P}_n = \boldsymbol{G}_n\boldsymbol{T}_n \qquad (20)$$

so that

$$\boldsymbol{s} = \sum_{n=1}^{N} \boldsymbol{G}_n\boldsymbol{T}_n\boldsymbol{d}_n + \boldsymbol{r} \qquad (21)$$

$$= \sum_{n=1}^{N} \boldsymbol{P}_n\boldsymbol{d}_n + \boldsymbol{r} \qquad (22)$$

The summation can be avoided by expression

$$\boldsymbol{s} = \boldsymbol{r} + \boldsymbol{Pd} \qquad (23)$$

where

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{P}_1 & \boldsymbol{P}_2 & \dots & \boldsymbol{P}_N \end{bmatrix} \qquad (24)$$

and

$$\boldsymbol{d}^T = \begin{bmatrix} \boldsymbol{d}_1^T & \boldsymbol{d}_2^T & \dots & \boldsymbol{d}_N^T \end{bmatrix} \qquad (25)$$

The solution for $\boldsymbol{p}$ which minimizes the residual energy can be calculated from the Equation 23 as:

$$\hat{\boldsymbol{d}} = (\boldsymbol{P}^H\boldsymbol{P})^{-1}\boldsymbol{P}^H\boldsymbol{s} \qquad (26)$$

The inverse $(\boldsymbol{P}^H\boldsymbol{P})^{-1}$ exists, if the rows and columns of $\boldsymbol{P}$ are linearly independent. In practise, this is fulfilled if notes with the same fundamental frequency do not overlap each other. Perfect harmonicity may cause problems if overlapping notes are in exact octave relation. Linear independency can be ensured by preprocessing step in which overlapping notes with too similar harmonic structure are removed.

Frame-wise amplitudes can be solved by the following procedure: the elements of $\hat{\boldsymbol{d}}$ which correspond to sound n are selected to get $\hat{\boldsymbol{d}}_n$, from which the $\hat{\boldsymbol{y}}_n$ are obtained by $\hat{\boldsymbol{y}}_n = \boldsymbol{T}_n\hat{\boldsymbol{d}}_n$. The elements of $\hat{\boldsymbol{y}}_n$ which correspond to frame k are selected to obtain $\hat{\boldsymbol{y}}_n^k$, from which the amplitudes are solved by $\hat{\boldsymbol{a}}_n^k = \boldsymbol{X}_n^k\hat{\boldsymbol{y}}_n^k$.

3.2. Computationally efficient algorithm

If estimation is performed simultaneously over all frames as in Equation 13, the computational complexity and memory usage of the normal least-squares solution is huge. A computationally efficient algorithm is proposed for the estimation of parameters.

Equation 26 can be written as

$$\hat{\boldsymbol{d}} = \boldsymbol{R}^{-1}\boldsymbol{S} \qquad (27)$$

where

$$\boldsymbol{R} = \boldsymbol{P}^T\boldsymbol{P} \qquad (28)$$

$$\boldsymbol{S} = \boldsymbol{P}^T\boldsymbol{s} \qquad (29)$$

Matrix $\boldsymbol{R}$ has the following structure:

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_{1,1} & \boldsymbol{R}_{1,2} & .. & \boldsymbol{R}_{1,N} \\ \boldsymbol{R}_{2,1} & \boldsymbol{R}_{2,2} & .. & \boldsymbol{R}_{2,N} \\ : & : & & : \\ \boldsymbol{R}_{N,1} & \boldsymbol{R}_{N,2} & .. & \boldsymbol{R}_{N,N} \end{bmatrix} \qquad (30)$$

where

$$\boldsymbol{R}_{i,j} = \boldsymbol{P}_i^T\boldsymbol{P}_j \qquad (31)$$

Vector S has structure

$$\boldsymbol{S} = \begin{bmatrix} \boldsymbol{S}_1 \\ \boldsymbol{S}_2 \\ : \\ \boldsymbol{S}_N \end{bmatrix} \qquad (32)$$

where

$$\boldsymbol{S}_n = \boldsymbol{P}_n^T\boldsymbol{s} \qquad (33)$$

The algorithm is based on approximating $\boldsymbol{R}^{-1}$ by inverses of submatrixes $\boldsymbol{R}(Q)$ of $\boldsymbol{R}$. $\boldsymbol{R}(Q)$ contains submatrixes $\boldsymbol{R}_{n,m}$, the indices of which belong to set Q: $n \in Q$ and $m \in Q$. Similarly, subvector $\boldsymbol{S}(Q)$ of $\boldsymbol{S}$ contains subvectors $\boldsymbol{S}_i$, for which $n \in Q$

In the algorithm, each note is estimated simultaneously with overlapping notes by forming set Q of overlapping notes.This is illustrated in Figure 2 Additionally to the note the parameters which we are estimating, set Q contains notes which at least partially overlap with the note. Non-overlapping notes do not have to taken into account since $R_{n,m} = 0$ for $t_n^{(0)} > t_m^{(1)}$ or $t_m^{(0)} > t_n^{(1)}$.

Memory-consuming variables H, G, P, and R are stored dynamically. Their initialization and deletion are described in the algorithm by commands “calculate”, and “delete”, respectively.

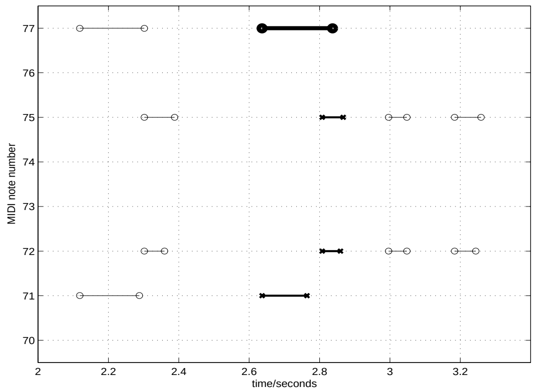

Figure 2: *Illustration of overlapping notes. Each line corresponds to one note, the onsets and offsets of which have been estimated by the MPE. In the estimation of the parameters of the note with bold line and circles the bold-line notes with x-marks are taken into account.*

Since most elements of the transform matrices *X* and *T* are zero, also matrices *G* and *P* are sparse. To further reduce memory consumption, the our implementation utilizes “sparse matrix” datatype of Matlab.

Input parameters of the algorithm:

-onset times $t_n^{(0)}$ and offset times $t_n^{(1)}$ of notes 1..*N*.

-frequency and phase estimates $\omega_n^h(k)$, and $\theta_n^h(k)$ of harmonic components of each sound in each frame

-mixture spectrum $\boldsymbol{s}_k$ in each frame *k*

Output parameters of the algorithm:

-time-frequency model parameters $\boldsymbol{d}_n$ of each sound

Initialization:

-during the processing, notes are divided into three sets:

-Let *Z*, the set of processed notes be $Z = \varnothing$

-Let *D*, the set of notes under processing be $D = \varnothing$

-Let *E*, the set of unestimated notes be [1,N]

-Sort notes to increasing offset times: $t_1^{(1)} < t_2^{(1)} < \ldots < t_N^{(1)}$

-Initialize $\boldsymbol{X}_n$, $\boldsymbol{T}_n$ for each sound *n* (frequency- and time matrices, described in Section 4).

The algorithm:

1. Let note index *n*:=1. Let $D{:=}D \cap n$, $E{:=}E\backslash n$
2. Calculate $\boldsymbol{H}_n^k$ (basis functions of each harmonic component) and $\boldsymbol{G}_n^k$ (basis functions of frequency models) for frames $k \in [t_n^{(0)}, t_n^{(1)}]$ as described in Equations 6 and 10.
3. Calculate $\boldsymbol{G}_n$ (Equation 15) using $\boldsymbol{G}_n^k$, $k \in [t_n^{(0)}, t_n^{(1)}]$. (all basis frequency-model functions of a sound *n*). $\boldsymbol{G}_n^k = 0$ for $k \notin [t_n^{(0)}, t_n^{(1)}]$. Delete all $\boldsymbol{G}_n^k$
4. Calculate $\boldsymbol{P}_n$ (time-frequency basis functions of a sound *n*) (Equation 20). Delete $\boldsymbol{G}_n$.
5. Calculate $\boldsymbol{S}_n$. (Equation 33)
6. Find a set *Q* of notes under processing which overlap with note *n*:
   $Q = \{m | m \in D \wedge min(t_n^{(1)}, t_m^{(1)}) - max((t_n^{(0)}, t_m^{(0)}) > 0)\}$
7. Calculate $\boldsymbol{R}_{m,n}$ for all $m \in Q_n$.
8. Find a set *V* of notes under processing which do not overlap with future notes: $V = \{m | (t_m^{(1)} < t_q^{(0)}) \forall q \in E \wedge (m \in D)\}$.
9. For each $m \in V$:

   -Construct $\boldsymbol{R}(Q)$ (use $\boldsymbol{R}_{n,m} = \boldsymbol{R}_{n,m}^T$ for $n > m$)

   -Construct $\boldsymbol{S}(Q)$.

   -Let $\boldsymbol{d}(Q) = \boldsymbol{R}(Q)^{-1}\boldsymbol{S}(Q)$

   -From elements of $\boldsymbol{d}(Q)$ which correspond to parameters of sound *m*, store $\hat{\boldsymbol{d}}_m$.

   -Delete $\boldsymbol{P}_m$.
10. Find a set of notes under processing which can be deleted: $W = \{m | m \in D \wedge t_m^{(1)} < t_1\}$, where $t_1 = min(t_m)$, $m \in E$
11. Delete all $\boldsymbol{R}_{m,i}$, $m \in W$, all *i*.
12. Let *D*:=*D*\W and $Z{:=}Z \cap W$
13. Let *n*:=*n*+1 (next note). If $n \le N$ goto step 2. Otherwise, repeat steps 8 to 12 once.

The frame-wise amplitudes can be solved using the procedure described in the end of Section 3.1. The equations and algorithm explained in this paper are formulated using time-domain signals. The computational efficiency is increased by transforming the signals into frequency domain, because this allows limiting of the frequency range on which the least-square solution is calculated. Basically this corresponds to decimating the input signals, but allows selection of sampling frequency according to the highest frequency component of the estimated notes.

4. LINEAR MODELS

The proposed method allows arbitrary linear models. As described in Equation 3, the amplitude vector $\boldsymbol{a}_n$ is represented as a linear combination of the columns of the transform matrix $\boldsymbol{X}_n$. When frequencies and phases are taken into account, one frame of a sound is represented as a linear combination of the columns of matrix ***G***. Instead of single overtones, the amplitudes are estimated for these basis functions. By using time-domain models, these basis functions are extended to several frames.

4.1. Frequency models

In [1], some practical structures for the frequency-model transform matrix were studied. For example, a *M*th-order polynomial fit is obtained with the matrix

$$(X)_{p,q} = (f_p)^q, q \in [0, M] \tag{34}$$

where f_p is the frequency of the *p*:th component. A more perceptually-oriented model is obtained by using a matrix which approximates a critical-band-scale filterbank:

$$(X)_{p,q} = max(min(p2^{1-q} - 1, 2 - p2^{-q}), 0). \tag{35}$$

This transform matrix is intuitively very applicable, because the model parameters correspond to short-time energies within octave frequency bands. The frequency bands were optimized using generated test signals. The resulted optimal bands were approximately 2/3-octave bands [1].

In the separation described in this paper the optimized frequency-band model was used. Additionally to it’s intuitiveness, the transform matrix is sparse, which reduces the computational complexity of the algorithm.

4.2. Time models

Triangular basis functions were used also in time models. The transform matrix *T* is formulated by

$$(T)_{t,q} = max\left(min\left(\frac{t - t_q}{t_{q+1} - t_q}, \frac{t - t_{q+2}}{t_{q+1} - t_{q+2}}\right), 0\right) \tag{36}$$

where the places of the triangles are determined by the time instants t_q. Two different methods for the placement of t_q were tested: In constant spacing the *t*0 is placed at the sound onset $t^{(0)}$ and the rest instants at constant intervals:

$$t_q = t_0 + (q-1)k, \qquad q \in [1, M] \tag{37}$$

Usually natural sounds tend to have fast changes in the beginning and slowly-varying decay. This was taken into account by trying also exponentially spaced instants, which have increasing intervals:

$$t_q = t_0 + p^{q-1} - 1, \quad q \in [1, M] \tag{38}$$

The constant-interval and exponentially spaced basis functions are illustrated in Figure 3.

Naturally, it is advantageous to select the models according to the amount of interfering notes: If there is only one sound present and no interfering sounds, an identity transform matrix can be used, since it is allows estimation of individual frequency components. If there is more than one interfering sound present, some of the components are probably overlapping and the rough spectral shape has to be utilized to estimate amplitudes.

5. SYNTHESIS

Once the parameters of the harmonic components have been estimated in each frame, the sounds can be synthesized separately. In synthesis, the frequencies, amplitudes and phases are interpolated from frame to frame, and time-domain signals are obtained by summing up all the harmonic components of each sound.

The parameters of the sounds can also be further analysed, or manipulation can be performed on the parametric data.

6. EXPERIMENTAL RESULTS

The proposed algorithm is intended for real-world music signals. For those a quantitative evaluation of the separation results is impossible because separation reference is not available. This problem was bypassed by generating test signals from MIDI using a software synthesizer. In this case evaluation is possible by synthesizing the individual notes separately and comparing to the separation results. Also the correct fundamental frequencies are known so that the performance of the multipitch estimation and sound separation can be studied separately.

One hundred ten-second excerpts were randomly selected from a collection of 359 MIDI songs, the styles of which ranged from classical to popular music. The excerpts were listened to exclude excerpts that were considered “too difficult for human to transcribe or separate”. These included for example fast glissandos.

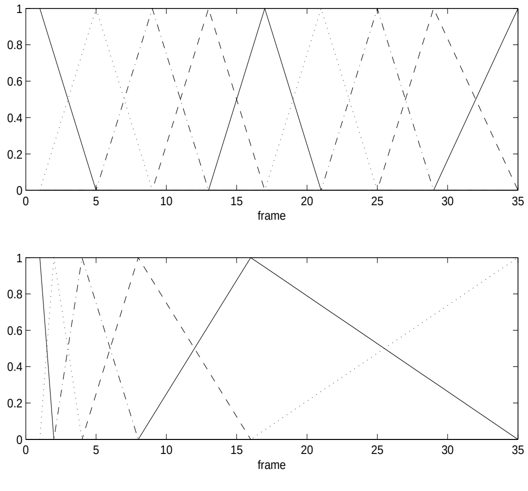

Figure 3: *Examples of the constant-interval (upper plot) and exponentially spaced (lower plot) basis functions. Each triangle corresponds to one basis function.*

The excerpts were synthesized using Timidity software synthesizer. Also individual notes in each excerpt were synthesized separately, to allow comparison to the separated notes. The excerpts were transcribed using the MPE system. Since the performance of the MPE is not perfect, the original MIDI data was used as another initialization for the separation algorithm.

The synthesized excerpts were separated using the algorithm described in Section 3. The separated notes were matched to the individual original notes by their pitch and time location. If several matches were possible, the notes were further analysed by an auditory model, using which the separated notes were matched to perceptually most similar original notes. Because of errors in MPE, several separated notes did not correspond to any original notes, and vice versa, for some original notes the corresponding separated notes were not found.

The Perceptual Audio Quality Measure (PAQM) [6] was calculated between each matched separated-original signal pair. This gives a rough quantitative estimate of the separation quality. One modification was made to the PAQM algorithm to make it more suitable for comparison of single notes: scaling in three frequency ranges was bypassed, because it gave too good results if the original note had a very little energy in some frequency band and the separated note had some interference in that band.

The total number of notes in the original MIDI excerpts was 22800, which does not include drum notes. The MPE analysed 10456 notes. These were used as an initialization for the separation algorithm. Depending on the parameters of the algorithm, about 6300 of the separated notes were matched to the original notes. The histogram of the PAQM between original and separated notes using the MPE system as initialization and constant-interval time models is illustrated in Figure 4. The histograms of other simulations are very similar in shape and are therefore not illustrated..

The simulations were carried out using three different time models: constant-interval model, exponentially spaced model and no time model so that each frame is independent of others. In the

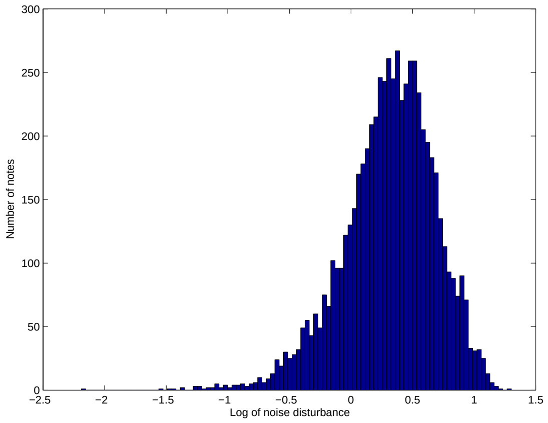

Figure 4: *Histograms of the Perceptual Audio Quality Measures between original and separated notes obtained using the MPE system as initialization and constantly-spaced time models.*

Table 1: Simulation results.

time model	mean of log(noise disturbance)		percentage of notes with log(noise disturbance) < 0	
	MPE	MIDI	MPE	MIDI
constant-interval	0.311	0.301	18.2	23.5
exponentially spaced	0.285	0.278	21.4	25.4
no time model	0.030	0.030	45.3	51.7

constant-interval model the triangles were spaced four frames from each other and the in the exponentially spaced model the powers of two were used in the placement of triangles, as illustrated in Figure 3.

Mean of logarithmic noise disturbances was calculated to measure the average quality of separation. Percentage of notes for which the logarithmic noise disturbance was below zero was calculated to measure the amount of separated notes, the quality of which was considered “acceptable”. The statistics are presented in Table 1. The results were further analysed by listening to the separated samples. For all three models, using the original MIDI as an initialization gives better results than the MPE. However, the difference is surprisingly small considering the difficulty of the multipitch estimation of polyphonic music. Exponentially spaced model performs slightly better than constant-interval model, but separation without time model is clearly better according to the PAQM. However, in listening comparisons it was noticed that several signals separated without time model had irritating modulation even though the PAQM indicated good quality. The modulation was supposedly caused by rapid changes of amplitudes between frames. In general, it was considered that from separated notes with the same PAQM, the notes separated using time models were perceptually better. This indicates that the PAQM is not very suitable in comparison of single harmonic sounds.

7. CONCLUSIONS

The proposed algorithm allows efficient estimation of amplitudes of harmonic partials with interframe dependency. The dependency is obtained by modeling the components as a sum of pre-defined basis functions. Simulations and quantitative measures do not directly show improvement in quality, but informal listening comparisons indicate that that the perceptual quality of separated sounds is increased by the time-frequency constraints.

8. REFERENCES

[1] T. Virtanen, A. Klapuri. "Separation of Harmonic Sounds Using Linear Models for the Overtone Series", IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2002.

[2] Bregman, “Auditory Scene Analysis,” MIT Press, 1990.

[3] M. Davy and S. Godsill, "Bayesian Harmonic Models for Musical Signal Analysis", Seventh Valencia International meeting (Bayesian Statistics 7), Oxford University Press, 2002.

[4] A. Klapuri, T. Virtanen, J.-M. Holm. “Robust multipitch estimation for the analysis and manipulation of polyphonic musical signals,” In Proc. COST-G6 Conference on Digital Audio Effects, Verona, Italy, 2000.

[5] T. Virtanen, A. Klapuri. “Separation of Harmonic Sounds Using Multipitch Analysis and Iterative Parameter Estimation,” IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New York, U.S.A. 2001.

[6] J. Beerends, J. Stemerdink. “A Perceptual Audio Quality Measure Based on a Psychoacoustic Sounds Presentation,” J. Audio Eng. Soc., Vol. 40, No. 12, 1992 December.