Separation of Sound Sources by Convolutive Sparse Coding

Tuomas Virtanen

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

ABSTRACT

An algorithm for the separation of sound sources is presented. Each source is parametrized as a convolution between a time-frequency magnitude spectrogam and an onset vector. The source model is able to represent several types of sounds, for example repetitive drum sounds and harmonic sounds with modulations. An iterative algorithm is proposed for the estimation the parameters. The algorithm is based on minimizing the reconstruction error and the number of onsets. The number of onsets is minimized by applying the sparse coding scheme for onset vectors. A way of modeling the loudness perception of the human auditory system is proposed. The method compresses high-energy sources, and enables the separation of low-energy sources which are perceptually significant. The algorithm is able to separate meaningful sources from real-world signals. Simulation experiments were carried out using mixtures of harmonic instruments. Demonstration signals are available at http://www.cs.tut.fi/~tuomasv/demopage.html.

1. INTRODUCTION

In real-world audio signals several sound sources are usually mixed. The process in which individual sources are estimated from the mixture signal is called sound separation. Humans are extremely skilful in "hearing out" individual sound sources from complex mixtures even in noisy conditions. Computational modeling of this ability is very difficult. All the existing separation systems are limited in either polyphony or quality. The most successful ones are those which try to extract only the most prominent source [1, 2].

Without any prior knowledge of the sources, the problem of estimating several overlapping sources from one input signal is ill-defined. By making some assumptions of the underlying sounds, it is possible to analyze and synthesize signals which are perceptually close to the originals before mixing. Harmonicity of sources has been assumed in most sound separation systems which separate musical sounds. In this paper the assumption of recurrence of sources is used.

Independent Component Analysis (ICA) has been successfully used to solve blind source separation problems in several application areas. A related technique called Independent Subspace Analysis has been used for sound separation by Casey and Westner [3] and FitzGerald et al. [4]. The method tries to find sound spectra which are statistically independent from each other. However, it can be claimed that in the case of music signals independency is usually not a valid assumption. In frequency domain the fundamental frequencies of the sources are often in a harmonic relationship, and in the time domain the sources have dependencies because of rhytmic concordance.

A data-adaptive technique very similar to ICA called sparse coding has been successfully used for example to model the functioning of the early stages of vision [5]. The term *sparse* is used to refer to a signal model, in which the data is represented in terms of a small number of active elements chosen out of a larger set. The basic idea is shortly described in Section 1.1. Sparse coding has been used for audio signal separation e.g. by Plumbley et al. [6], Smaragdis & Brown [7], and Virtanen [8]. These studies show that by using the sparseness assumption, it is possible to some degree estimate sound sources without any other knowledge of sources, while it is clear that for robust high-quality separation more assumptions have to be used.

1.1. Sparse coding

The basic signal model in sparse coding is the same as in ICA: each observation vector $\boldsymbol{x}_i$ is a linear mixture of source vectors $\boldsymbol{s}_j$:

$$\boldsymbol{x}_i = \sum_{j=1}^{J} a_{i,j}\boldsymbol{s}_j \qquad i = 1 \ldots I, \tag{1}$$

where $a_{i,j}$ is the weight of j^{th} source in i^{th} observation. Both the source vectors and weights are unknown. In a matrix form the model can be expressed as $\boldsymbol{X} = \boldsymbol{AS}$.

In ICA, the estimation is done by assuming statistical independence of sources. The sources are obtained by multiplying the observation matrix by the estimate of the unmixing matrix $\boldsymbol{W} \approx \boldsymbol{A}^{-1}$. If the mixing weights and sources are restricted to non-negative values, this estimation method can not be used.

In sparse coding the sources are assumed to be non-active most of the time which means that the mixing matrix has to be sparse. The estimation can be done using a cost function which minimizes the reconstruction error and maximizes the sparseness of the mixing matrix. In a propabilistic framework, minimizing the cost function corresponds to maximizing the log-likelihood of sources [5].

An iterative algorithm for the estimation of non-negative parameters was proposed by Hoyer [9]. The algorithm is based on non-negative matrix factorization (NMF) [10], which was utilized in sound separation by Smaragdis and Brown in [7], and by Virtanen in [8]. The algorithm enables the use of further restrictions such as temporal continuity of the sources, as was proposed by Virtanen [8].

1.2. Improvements in the proposed method

In the case of audio signals the most obvious choice for the observation matrix is a time-frequency magnitude spectrogram, so that basis functions are magnitude spectra of sources. If the basic signal model (1) is used, the sources will have a fixed spectrum over time and only the gain is time-varying. This basic approach has already been used in some ICA, ISA, and sparse coding systems [3, 4, 6, 7, 8].

The basic approach has at least two major shortcomings. First, the assumption of fixed spectra over time is unrealistic for natural sound sources. Secondly, the model fitting criterion, e.g. the sum of squared elements of the reconstruction error ($\boldsymbol{E} = \boldsymbol{X} - \boldsymbol{AS}$), emphasizes very different things than the human sound perception.

In this paper some methods are proposed for solving these problems at least partially. Firstly, in the proposed source model the temporal characteristics can be modeled more accurately by using a time-frequency spectrogram for each source. As explained in Section 2, the source model can be formulated as a convolution between an onset vector and a source spectrogram, thus the name convolutive sparse coding. Secondly, the loudness perception of human auditory system is modeled by using compression, as explained in Section 3.

In multi-channel blind source separation, algorithms which separate delayed and convolved sources have been presented in several papers, e.g. by Lee et al. [11] and by Smaragdis [12]. Despite of the convolutive source model, the multi-channel separation algorithms do not have much common with the proposed algorithm.

2. SOURCE MODEL

The input signal is represented using the magnitude spectrogram, which is calculated as follows: at first, the time-domain input signal is divided into frames and windowed. A fixed 40 ms Hamming window is used with 50% overlap between frames. Second, each frame is transformed into frequency domain by taking the discrete Fourier transform (DFT). The length of the DFT is equal to the window size. Only positive frequencies are retained. Phases are discarded by taking the magnitude of the DFT spectra to result spectrogram $x_{f,t}$, where f is the discrete frequency index and t is the frame index. Vector $\boldsymbol{x}_f = \left[x_{f,1} \ \ldots \ x_{f,K}\right]^T$ is used to denote all the magnitudes of channel f.

The proposed source model was originally designed for repetitive transient sources. However, it turns out that the model is also capable for representing sustained sounds. The model is explained from the transient point of view, and suitability for other sounds is explained in Section 2.1.

Let us assume that sources are short in duration and that they occur repeatedly so that the magnitude spectrogram is similar for each occurrence of a source, only the gain is varying. Two-dimensional magnitude spectrogram $s_{n,f}(\tau)$ is used to characterize one event of source n at discrete frequency f, τ frames after the onset. τ varies between 0 and D. Fixed duration D determines the maximum duration of a single event. In our simulations durations between 100 and 600 ms were tried, which should be long enough for e.g. most drum sounds. With 20 ms hop size between frames this corresponds to delays from $D = 5$ to $D = 30$.

Onsets and gains of each event are characterized by parameters $a_n(t)$. When a sound sets on, the value is positive, while before and after onset it should be zero. $a_n(t)$ also describes the gain of each event. With this parametrization, the model for a spectrogram $\boldsymbol{M}$ in which a single event occurs at frame t_0 can be written as

$$(\boldsymbol{M})_{t,f} = a_n(t_0)s_{n,f}(t - t_0) \qquad t = t_0 \ldots (D + t_0) \qquad (2)$$

The model can be written for multiple onsets described by the non-zero elements of $a_n(t)$ and all frames $t = 1...K$:

$$(\boldsymbol{M}_n)_{t,f} = \left[\sum_{\tau=0}^{D} a_n(t - \tau)s_{n,f}(\tau)\right] \qquad (3)$$

Each onset corresponds to an impulse in $a_n(t)$. The summation inside the brackets is the definition of the finite-length convolution. The whole spectrogram, in which N sounds are overlapping, can be written as:

$$(\boldsymbol{M}_n)_{t,f} = \left(\sum_{n=1}^{N} [\boldsymbol{a}_n \otimes \boldsymbol{s}_{n,f}]\right)_t \qquad (4)$$

where $\otimes$ denotes the convolution of vectors $\boldsymbol{a}_n = \left[a_n(1) \ \ldots \ a_n(K)\right]^T$ and $\boldsymbol{s}_{n,f} = \left[s_{n,f}(0) \ \ldots \ s_{n,f}(D)\right]^T$.

The linear summation of the magnitude spectra has been assumed in the model. This has been assumed in almost all data-adaptive source separation systems which operate on discrete Fourier transform (DFT) spectra [3, 4, 7, 8]. Theoretically the assumption is not exactly valid for magnitude spectra, since only the time-domain signals and complex DFT spectra sum exactly linearly. However, phase information has to be discarded since the phase spectrum of only a small minority of sounds remains fixed. In some cases it is better justified to assume that the power spectra of sounds sum linearly, but in practise linear summation of magnitude spectra seems to work better.

2.1. Non-transient sources

It turns out that the proposed model suits well also for some other sound types than transients. For example, a harmonic sound with fixed spectrum $H(f)$ can be represented by $s_{n,f}(\tau)$ which equals $H(f)$ at all delays $\tau = 0...D$, and by $a_n(t)$ which has impulses at interval of $D+1$. In practise the impulses may have also smaller interval. A sound with vibrato is presented with $a_n(t)$ which has impulses at the rate of the vibrato and $s_{n,f}(\tau)$ which contains one period of the vibrato. When the parameters are estimated by fitting the model to the observed signal the impulses may not be exact.

The convolutive model is also a way of adding dependency between frames, which should increase the robustness of estimation. It can be noted that by setting the maximum delay D to zero the model is equal to (1).

3. LOUDNESS PERCEPTION

Computationally it is convenient to estimate the parameters

by minimizing the squared error between the observed magnitude spectrum and the model. However, the use of the squared error pays attention to very different things than the human auditory system, which is able to perceive very low-amplitude sounds. The large dynamic range of the human auditory system is mainly caused by the non-linear response of the auditory cells, which can be modeled as a compression of the input signal separately at each auditory channel.

In this system the compression is modeled by calculating a weight for each frequency bin in each frame. The weights are selected so that the sum of squared magnitudes is equal to the estimated loudness, since the separation algorithm uses the squared error criterion to fit the model to the data. This way the "quantitative significance" corresponds to the "perceptual significance."

Usually the loudness of one frame is modeled by compressing the excitation using perceptually motivated critical bandwidth [13] and frequency scale 1/Bark [14], and integrating over frequency. Thus, the loudness can be estimated individually for each critical band. The loudness model of the system is adopted from the loudness models of Moore et al. [15] and Zwicker and Fastl [14]. In our system, 24 separate bands are spaced uniformly on Bark scale, and denoted by disjoint sets F_b, $b = 1...24$. The fixed response of the outer and middle ear is taken into account by multiplying each bin of spectrum by corresponding response.

In this paper term *loudness inde*x is used for the loudness estimate in a frame within a critical band. The loudness index in frame t in critical band b is denoted by $L_{b,t}$, which is given as

$$L_{b,t} = \left[\sum_{f \in F_b} (h_b x_{f,t})^2 + \varepsilon_b^2 \right]^{\nu} - \varepsilon_b^{2\nu} \quad (5)$$

where h_b is the fixed response of the outer and middle ear within band b, ν is a fixed scalar with value 0.23 and ε_b is the threshold of hearing on band b, also fixed. In practise ε_b is not known, so it can be estimated from the input signal e.g. by calculating the average level of the signal, and scaling down 30 dB. The separation algorithm is noncausal so this is not a problem.

For each critical band in each frame, coefficient $g_{b,t}$ is assigned, which mimics the loudness perception: the coefficients are selected so that the error criterion, the scaled sum of squared magnitudes equals the estimated loudness:

$$g_{b,t} \sum_{f \in F_b} x_{f,t}^2 = L_{b,t} \quad (6)$$

from which $g_{b,t}$ can be solved as

$$g_{b,t} = \frac{L_{b,t}}{\sum_{f \in F_b} x_{f,t}^2} \quad (7)$$

To simplify the notation, let us use weight $w_{f,t}$ for each frequency index: $w_{f,t} = \sqrt{g_{b,t}}$, $f \in F_b$. Further, let us denote the weights by diagonal matrices $\boldsymbol{W}_f$:

$$\boldsymbol{W}_f = diag\left(\begin{bmatrix} w_{f,1} & \dots & w_{f,K} \end{bmatrix} \right), f = 1 \dots F \quad (8)$$

The compressed spectrogram can be expressed as

$$\boldsymbol{x}_f' = \boldsymbol{W}_f \boldsymbol{x}_f \quad (9)$$

In the separation it is simpler to use the original spectra and to store the weights. Therefore, the compressed spectra do not have to be calculated.

It was found out that very small threshold of hearing will cause problems in the optimization algorithm if there are low-amplitude sections in the input signal. Choosing thresholds which are larger than the actual threshold of hearing will increase the robustness of the estimation algorithm.

4. PARAMETER ESTIMATION

The estimation of the parameters is done by fitting the source model to the observed spectrogram $x_{f,t}$. Squared error criterion weighted by $\boldsymbol{W}_f$ is used to measure the goodness of the fit. The cost function for the reconstruction error c_{rec} is the sum across all frequencies and can be written as

$$c_{rec}(\lambda) = \sum_{f=1}^{F} \left\| \boldsymbol{W}_f \left(\boldsymbol{x}_f - \sum_{n=1}^{N} [\boldsymbol{a}_n \otimes \boldsymbol{s}_{n,f}] \right) \right\|^2 \quad (10)$$

where λ is used to refer to the parameter set $\{a_n, s_{n,f}\}$, $n \in [1, N]$, $f \in [1, F]$.

The number of events is minimized by applying the sparse coding scheme for vectors $\boldsymbol{a}_n$. The sparseness of $\boldsymbol{a}_n$ is measured as a sum of cost function q over all the elements:

$$sparse(\boldsymbol{a}_n) = \sum_{t=0}^{K} q(a_n(t)) \quad (11)$$

Olshausen and Field used $q(x) = \log(1 + x^2)$ in [5]. In this paper cost term $q(x) = |x|$ is used, which has been earlier used by e.g. Hoyer [9] and Virtanen [8]. The numerical range of $\boldsymbol{a}_n$ has to be fixed e.g. to unity norm, because otherwise the measure is minimized by decreasing $\boldsymbol{a}_n$ and increasing $s_{n,f}$. Instead of fixing the norm of $\boldsymbol{a}_n$, the cost function is modified to take into account the scale, so that the cost for N sources becomes:

$$c_{sparse}(\lambda) = \sum_{n=1}^{N} \frac{||\boldsymbol{a}_n||_1}{||\boldsymbol{a}_n||_2}, \quad (12)$$

where $||\boldsymbol{a}_n||_1 = \sum_{t=0}^{K} |a_n(t)|$ is the L_1-norm and $||\boldsymbol{a}_n||_2 = \sqrt{\sum_{t=0}^{K} a_n(t)^2}$ is the standard L_2-norm.

The overall cost function is the weighted sum of the reconstruction error cost and the sparseness cost:

$$c_{tot}(\lambda) = \alpha_{rec} c_{rec}(\lambda) + \alpha_{sparse} c_{sparse}(\lambda) \quad (13)$$

where α_{rec} and α_{sparse} are the weights, respectively. In our simulation experiments it turned out that the sparseness cost should have a very low cost compared to the reconstruction error term. Depending on the input signal, the optimal range of the sparseness term varies approximately between zero and one fourth of the reconstruction error term.

The optimization algorithm is an iterative algorithm, in which the magnitude spectrograms are updated using the multiplicative step [10], and the onset vectors using the steepest descent.

4.1. Magnitude spectrograms

For given onset vectors $\boldsymbol{a}_1 ... \boldsymbol{a}_N$, $\boldsymbol{s}_{n,f}$ can be updated using

the following procedure. At first, the convolution is expressed as a multiplication by matrix. Convolution matrix $\boldsymbol{A}_n$ is a matrix formed from vector $\boldsymbol{a}_n$, so that the inner product with vector $\boldsymbol{s}_{n,f}$ is the convolution of the vectors:

$$\boldsymbol{A}_n \boldsymbol{s}_{n,\,f} = \boldsymbol{a}_n \otimes \boldsymbol{s}_{n,\,f} \tag{14}$$

$\boldsymbol{A}_n$ is a Toepliz matrix which has the elements of $\boldsymbol{a}_n$ on diagonals. Matrices $\boldsymbol{A}_n$ are formed from $\boldsymbol{a}_n$, and the cost function for the reconstruction error of channel f is written as:

$$c_{rec(f)}(\lambda) = \left\| \boldsymbol{W}_f \left(\boldsymbol{x}_f - \sum_{n=1}^{N} \boldsymbol{A}_n \boldsymbol{s}_{n,\,f} \right) \right\|^2 \tag{15}$$

and further

$$c_{rec(f)}(\lambda) = \left\| \boldsymbol{W}_f \boldsymbol{x}_f - \boldsymbol{W}_f \boldsymbol{A} \boldsymbol{s}_f \right\|^2 \tag{16}$$

by using $\boldsymbol{A} = [\boldsymbol{A}_1 \ldots \boldsymbol{A}_N]$, $\boldsymbol{s}_f = \left[\boldsymbol{s}_{1,\,f}^T \ \ldots \ \boldsymbol{s}_{N,\,f}^T \right]^T$.

From this form there are several alternative possibilities for the update of non-negative $\mathbf{s}_f$. For example, one can solve globally optimal non-negative $\mathbf{s}_f$ by the active-set method [16], or use iterative methods, such as projected steepest descent.

In our simulations the fastest convergence with relatively low computational cost was achieved with the multiplicative step proposed by Lee and Seung [10]. The at p^{th} iteration, updated $\boldsymbol{s}^{\{p+1\}}$ for given $\boldsymbol{s}^{\{p\}}$, $\boldsymbol{A}$ and $\boldsymbol{W}_f$ is given by

$$\boldsymbol{s}^{\{p+1\}} = \boldsymbol{s}^{\{p\}} .* (\boldsymbol{A}^T \boldsymbol{W}_f^T \boldsymbol{W}_f \boldsymbol{x}_f) ./ (\boldsymbol{A}^T \boldsymbol{W}_f^T \boldsymbol{W}_f \boldsymbol{A} \boldsymbol{s}^{\{p\}}) \tag{17}$$

where .* and ./ are element-wise multiplication and division, respectively.

4.2. Onset vectors

The gradient of the cost function with respect to $\boldsymbol{a}_n$ is needed in the optimization algorithm. Let us start by writing the cost of the reconstruction error at frequency f in frame t by:

$$c_{rec(f,\,t)}(\lambda) = w_{f,\,t}^2 \left[x_{f,\,t} - \sum_{n=1}^{N} \sum_{\tau=0}^{D} a_n(t-\tau) s_{n,\,f}(\tau) \right]^2 \tag{18}$$

Let us denote the weighted error in frame t by

$$e_f(t) = w_{f,\,t}^2 \left[x_{f,\,t} - \sum_{n=1}^{N} \sum_{\tau=0}^{D} a_n(t-\tau) s_{n,\,f}(\tau) \right] \tag{19}$$

The derivative of the cost (18) with respect to $a_n(t_i)$ is given by:

$$\frac{dc_{rec(f,\,t)}(\lambda)}{da_n(t_i)} = \begin{cases} -2 e_f(t) s_{n,\,f}(t-t_i) & t-D \le t_i \le t \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

The derivative of the sum of costs in frames $t = 0 \ldots T$ is given by:

$$\frac{dc_{rec(f)}(\lambda)}{da_n(t_i)} = -2 \sum_{\tau=0}^{min\{D,\,K-t_i\}} e_f(t_i+\tau) s_{n,\,f}(\tau) \tag{21}$$

and the sum across all frequencies:

$$\frac{dc_{rec}(\lambda)}{da_n(t_i)} = -2 \sum_{f=1}^{F} \sum_{\tau=0}^{min\{D,\,K-t_i\}} e_f(t_i+\tau) s_{n,\,f}(\tau) \tag{22}$$

The gradient of the sparseness cost (12) is given as

$$\frac{dc_{sparse}(\lambda)}{d\boldsymbol{a}_n} = \frac{1}{\|\boldsymbol{a}_n\|_2} - \frac{\|\boldsymbol{a}_n\|_1 \boldsymbol{a}_n}{\|\boldsymbol{a}_n\|_2^3} \tag{23}$$

The total gradient is the weighted sum of the reconstruction cost gradient and the sparseness cost gradient:

$$\frac{c_{tot}(\lambda)}{d\boldsymbol{a}_n} = \alpha_{rec} \frac{c_{rec}(\lambda)}{d\boldsymbol{a}_n} + \alpha_{sparse} \frac{c_{sparse}(\lambda)}{d\boldsymbol{a}_n} \tag{24}$$

4.3. Iterative algorithm

Input and pre-processing:

The magnitudes $x_{f,t}$ and weights $w_{f,t}$ are calculated. The number of sources N is set by hand. N should be equal to the number of clearly distinguishable instruments. For a drum sequence, for example, one might use $N = 3$ for a pattern which consists of bass, snare and hi-hat. If the spectrum of one source varies lot, for example because of accentuation, one may have to use more than one component per source. It has to be noted that the model consideres the different fundamental frequencies of each instrument as separate sources.

Initialization:

Initialize $\boldsymbol{a}_1 \ldots \boldsymbol{a}_N$ and $\boldsymbol{s}_{n,\,f}$ with the absolute values of Gaussian noise.

Iteration:

1. Update $\boldsymbol{s}_{n,\,f}$ using the multiplicative step (17).

2. Calculate $\nabla \boldsymbol{a}_n = \frac{dc_{tot}(\lambda)}{d\boldsymbol{a}_n}$

3. Update $\boldsymbol{a}_n \leftarrow \boldsymbol{a}_n - \mu_k \nabla \boldsymbol{a}_n$. Set the negative elements of $\boldsymbol{a}_n$ to zero. μ_k is the step size which is adaptively varied.

4. Evaluate the cost function.

5. Repeat the steps 1..4 until the value of the cost function does not change. In practise this is done by keeping track of iterations steps for which the decrease of the cost function has been smaller than a small threshold. Iteration is stopped when the decrease has been smaller during a certain number of iterations.

For a 10-second input signal the algorithm takes a couple of hundred iterations to converge, which takes a couple of minutes on a regular PC when implemented in Matlab.

5. SYNTHESIS

In synthesis, the convolutions are evaluated to get frame-wise magnitudes $(\boldsymbol{M}_n)_{t,\,f} = \sum_{\tau=0}^{D} a_n(t-\tau) s_{n,\,f}(\tau)$ for each source. To get complex spectrum, phases are obtained from the phase spectrogram of the original mixture signal. Time-domain signal is obtained by inverse discrete Fourier transform and overlap-add. This procedure was found to produce best quality especially for drum signals, for example compared to phase generation method proposed by Griffin and Lim [17]. The use of original phases allows the synthesis of sharp attacks at accuracy which would otherwise be impossible with large window sizes.

6. SIMULATION EXPERIMENTS

The objective of the algorithm is to extract sound sources which are perceptually close to the originals before mixing. Quantitative evaluation of the perceptual separation quality is difficult. It can be measured either by listening tests or compu-

tational procedures which compare the ideal source signals to separated signals. Basically in both cases the source signals before mixing are required. In practise this will limit us to synthesized test signals.

6.1. Mixture of two harmonic sounds

Systematic evaluation and comparison to other separation algorithms was performed with mixtures of harmonic sounds. 200 two-note mixtures of harmonic sounds were used in the systematic evaluation. To generate a test signal, two samples were randomly drawn from a database which consists of 26 harmonic instruments of different fundamental frequencies, 850 samples in total. The longer one of the samples was truncated so that the lengths of the samples were equal. To get mixing conditions which correspond more to real situations, the second sample was scaled to 0 dB, and uniformly distributed random power between -10 and 0 dB was used for the first sample.

Mixture signal was generated so that the first sample sets on at the beginning of the mixture signal and the second one sets on at the half of the duration of the first sample. Using this procedure, the first third of the mixture signal contains only the beginning of the first sample, the second third contains the end of the first sample and the beginning of the second sample overlapping, and the last third contains the end of the second sample. An example of a mixture signal is illustrated in Fig. 1.

Each mixture signal was separated into two sources using the proposed algorithm. For comparison, also earlier published separation algorithms based on ISA [3], NMF [7, 10] and sparse coding with temporal continuity [8] were tested. The algorithms were implented using the references. In ISA, FastICA algorithm [18] was used to obtain the independent components. Casey and Westner [3] proposed the usage of several components per source to overcome the limitation of the basic source model of ISA. Therefore, the ISA algorithm was tested also with more than two components. For NMF the Euclidean distance objective was used, since it is similar to that used in the proposed method and sparse coding. Since the proposed weighting method can be used also for NMF and sparse coding, they were tested with and without weighting.

Clustering the components to sources is a difficult task. In our simulations the objective was to compare sepation algorithms, so the clustering was avoided by using the original sources as a reference, comparing to which an ideal association could be obtained. Also with other algorithms the separated signals were associated to original sources by using grouping which minimized the residual energy of a source.

The perceptual audio quality measure (PAQM, [19]) was calculated between the separated and original signals. The measure was calculated for three different signal sections: term 'whole' is used to refer to the whole duration of the mixture signal, 'clean' is used to refer to signal part in which the target source is active and the interfering sound inactive, and 'overlap' refers to section where both sources are active. The sections are illustrated in Fig. 1.

The obtained measures are illustrated in Table 1. For the overlapping section, the proposed method produces the best quality. However, for the clean section the weighted sparse coding and weighted NMF perform better. For the whole signal the weighted sparse coding produces the best results, even

Table 1: Perceptual audio quality measures (log of average noise disturbance) of separated components on four signal sections. The smaller the measure, the better the quality.

algorithm	clean	overlap	whole
ISA (2 components)	-0.93	-1.07	-1.40
ISA (6 components)	-1.07	-1.29	-1.57
ISA (40 components)	-1.15	-1.45	-1.69
NMF	-1.09	-1.29	-1.58
weighted NMF	-1.15	-1.31	-1.62
sparse coding	-1.09	-1.29	-1.59
weighted sparse coding	-1.19	-1.41	-1.69
proposed method	-1.12	-1.47	-1.68

though the difference to the proposed method is very small. Without weighting the performance of NMF is almost identical to that of sparse coding, which suggests that the use of sparseness and temporal continuity objectives as proposed by Virtanen [8] are not effective at least in this simple task. It is interesting to see that the performance of unweighted NMF and sparse coding are almost identical, even though they are based on very different optimization algorithms.

The quality of ISA is increased by adding more components per source, so that with 40 components the performance is comparable with that of the proposed method. However, the original sources were used as a reference for the clustering. In practise it will be difficult to get as good results with multiple components per source.

Informal listening test showed that the perceptual quality of synthesized sources is relatively high, and correlate well with the PAQM.

6.2. Discussion

In the proposed method the number of sources has to be set by hand. Currently there is no way for the reliable estimation of the number of sources. In ICA the number of components can be selected e.g. by using a threshold for the singular values of the observation matrix. This approach has been used by Casey and Westner [3]. However, the source model presented in this paper is more complex and there is no straightforward

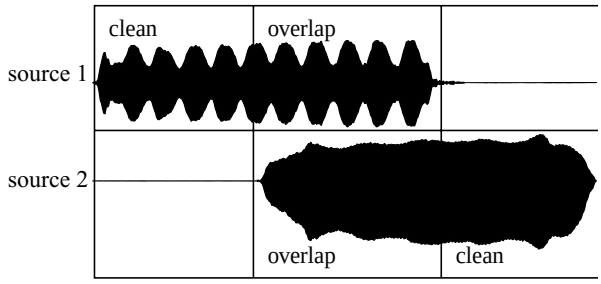

Figure 1: *An example of a mixture signal used in the simulations, flute with vibrato and trombone. The quality of the separated signals are estimated on signal sections 'clean', 'overlap', and 'silence' which are illustrated for both sources in the figure.*

way for the estimation of the number of sources.

It is possible that the optimization algorithm does not reach the global optimum, but is stuck in a local minimum. The result also depends somewhat on the initialization. This is typical for steepest-descent algorithms in general. For optimization problems which are as complex as the one presented in this paper, it is in practise not possible to design an algorithm which is guaranteed to reach the global optimum.

If one source is dominating, it is possible that the algorithm assigns more than one component for that source. To some degree it is possible to avoid this by using more components than sources, and clustering the components to sources, as has been suggested e.g. by Casey and Westner [3] and Virtanen [8].

Additionally to the separation of harmonic instruments, the proposed method works reasonably well e.g. with drum patterns. If the algorithm is able to converge to real sources, the perceptual quality of synthesized signals is high. If the complexity of the input signal is increased, the quality of the separation decreases gradually.

Most of the pitched instrument samples that were used in the simulation experiments can be represented rather well with the standard linear model used in sparse coding. Only the samples with heavy vibrato actually gain from using the convolutive model. The proposed perceptual weighting was found to be very effective, also with other models and algorithms e.g. non-negative matrix factorization and non-negative sparse coding.

The separation algorithm was also tested using polyphonic music signals. Some demonstration signals are available at http://www.cs.tut.fi/~tuomasv/demopage.html.

7. CONCLUSIONS

A data-driven algorithm for the separation sound sources has been proposed. The proposed source model allows separation of time-varying sources, for example repetitive short-duration transients and harmonic sounds with vibrato. The proposed separation algorithm is able separate sources from real-world signals, for example mixtures of harmonic sounds and drum patterns. Simulation experiments indicate that the proposed methods enables higher perceptual quality of the separated components than existing algorithms. Especially the use of perceptually motivated weights increases the quality.

8. REFERENCES

[1] Goto, M. “A robust predominant-F0 estimation method for real-time detection of melody and bass lines in CD recordings,” Proc. IEEE International Conf. on Acoust., Speech, and Signal Processing, Istanbul, Turkey, 2000.

[2] D.L. Wang, and G.J.Brown, “Separation of speech from interfering sounds based on oscillatory correlation,” IEEE Trans. on Neural Networks, vol 10. pp. 684-697, 2001.

[3] Casey, M.A., and Westner, A. “Separation of Mixed Audio Sources by Independent Subspace Analysis,” Proceedings of the International Computer Music Conference, Berlin, 2000.

[4] FitzGerald, D., Coyle, E., Lawlor, B., “Sub-band Independent Subspace Analysis for Drum Transcription,” Proc. of the 5th Int. Conference on Digital Audio Effects (DAFX-02), Hamburg, Germany, 2002.

[5] Olshausen, B. A., and Field, D. F. “Sparse coding with an overcomplete basis set: A strategy employed by V1?,” Vision Research, 37:3311-3325, 1997.

[6] Plumbley, M. D., Abdallah, S. A., Bello, J. P., Davies, M. E., Monti, G., and Sandler, M. B. “Automatic Music Transcription and Audio Source Separation,” Cybernetics and Systems, 33(6), pp. 603-627, 2002.

[7] Smaragdis, P., and Brown, J. C. “Non-Negative Matrix Factorization for Polyphonic Music Transcription,” IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY, 2003.

[8] Virtanen, T. “Sound Source Separation Using Sparse Coding with Temporal Continuity Objective,” International Computer Music Conference, ICMC 2003.

[9] Hoyer, P. “Non-negative sparse coding,” In Neural Networks for Signal Processing XII, Martigny, Switzerland, 2002.

[10] Lee, D. D., and Seung, H. S. “Algorithms for Non-negative Matrix Factorization,” in Advances in Neural Information Processing, vol. 13. MIT Press, 2001.

[11] Lee, T.-W., Bell, A. J., and Lambert, R. “Blind Separation of Delayed and Convolved Sources,” in Mozer, M. C., Jordan, M. I., and Petsche, T. (eds.), Advances in Nearal Information Processing Systems 9, 758-764. MIT Press, 1997

[12] Smaragdis, P., “Blind Separation of Convolved Mixtures in the Frequency Domain,” International Workshop on Independence & Artificial Neural Networks University of La Laguna, Tenerife, Spain, 1998.

[13] Moore, B. C. J. “Frequency Analysis and Masking,” in Moore, B.C.J. (Ed.), Hearing. New York: Academic Press, 1995.

[14] Zwicker, E., and Fastl, H. “Psychoacoustics: Facts and Models,” Springer, Berlin, 1999. Second Edition.

[15] Moore, B. C. J., Glasberg, B., and Baer, T. “A Model for the Prediction of Thresholds, Loudness, and Partial Loudness,” J. Audio Eng. Soc., Vol. 45, No. 4, 1997, pp. 224--240.

[16] Lawson, C. L., and Hanson, R. J. “Solving Least-Squares Problems,” Prentice-Hall, Chapter 23, p. 161, 1974

[17] Griffin, D., and Lim, J. “Signal Estimation from Modified Short-Time Fourier Transform,” IEEE Trans. on Acoustics, Speech and Signal Processing, 32, 236-242, 1984

[18] FastICA package for MATLAB http://www.cis.hut.fi/projects/ica/fastica/

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