Source separation using the spectral flatness measure

Rolf Bardeli[1]

[1]Fraunhofer IAIS, Sankt Augustin, Germany
rolf.bardeli@iais.fraunhofer.de

Abstract

Complex audio scenes with a large number of sound sources pose one of the most difficult problems for audio pattern recognition. Therefore, methods for source separation are very important in this context. Many source separation methods try to exactly recover every source in an audio scene. In this paper, however, we propose an algorithm for the extraction of simpler components from complex audio scenes based on an optimisation approach using a sound complexity measure derived from the spectral flatness measure. We yield good separation for artificial mixtures of three signals with time dependent mixing conditions.

Index Terms: source separation, spectral flatness measure

1. Introduction

Pattern recognition in real-world audio scenes is very difficult because of the high complexity of these scenes. They are built up from a multitude of single sources in an additive as well as in more complex manners. Blind source separation [1] has been suggested as a pre-processing step allowing to analyse each source by itself. With rising complexity of the scene, achieving blind separation becomes more and more difficult. Currently, for complex audio scenes, it does not seem to be feasible to reconstruct all of the constituting source signals exactly. In this work, we therefore develop an algorithm that extracts combinations of the input signals that constitute less complex components of an audio scene. These components will still be simpler to analyse by pattern recognition methods than the complex mixture.

One of the predominant methods for blind source separation is independent component analysis [2, 3]. It assumes a fixed linear mixture of signals and recovers the sources exactly by assuming their statistical independence. Different solutions to this problem exist, based on either algebraic [4] or statistical/information theoretic optimisation methods, and covering either only overdetermined cases or also tackling the underdetermined case [5], i.e., when there are fewer microphones than sources. Other than these methods, we do not assume fixed mixing parameters over time. Moreover, we choose a different measure for judging candidate unmixing parameters. We design a measure for the complexity of the unmixed signals based on the spectral flatness measure. This measure indicates whether the energy in the spectrum is concentrated or spread out. We use a particle based optimisation algorithm to extract low complexity components.

The rest of the paper is organised as follows. In Section 2 we derive the complexity measure for extracted components based on the spectral flatness measure and give a top level overview of the source separation algorithm. In Section 3, we describe the details of the algorithm. We compute the necessary derivatives for the optimisation algorithm, discuss the optimisation algorithm and its initialisation, as well as component extraction. In Section 4, we give evaluation results on artificial mixtures with time-varying mixing conditions.

2. Spectral Flatness Components

The central step in devising our source separation algorithm is the choice of a measure describing the complexity of an audio scene. Given such a measure, it is possible to evaluate it for several combinations of input sounds and choose the combination that gives the lowest complexity score.

The measure we use in our approach is the spectral flatness measure. It measures how much the energy at a given time is spread in the spectrum, giving a high value when the energy is equally distributed and a low value when the energy is concentrated in a small number of narrow frequency bands. The spectral flatness measure is computed from the spectrum as the geometric mean of the Fourier coefficients divided by the arithmetic mean. If $S(\omega, t)$ is the windowed power spectrum of a signal s at time t and frequency ω, its *spectral flatness measure* is given by

$$\mathrm{SFM}[S](t) = \frac{(\prod_{\omega=0}^{\Omega-1} S(\omega,t))^{\frac{1}{\Omega}}}{\frac{1}{\Omega}\sum_{\omega=0}^{\Omega-1} S(\omega,t)}.$$

The spectral flatness measure is also known as *Wiener entropy*.

Given a sequence of signals $f := (f_1, \ldots, f_n)$ from a microphone array we assume that a lower complexity source can be derived by choosing a linear combination of the signals. In order to apply the spectral flatness measure, we are interested in the windowed power spectrum U_f of such a linear combination:

$$U_f(\omega, t; a_1, \ldots, a_n) := |\sum_{i=1}^{n} a_i \hat{f}_i(\omega, t)|^2.$$

Scaling of the input signals does not affect the complexity measure of a component U_f. Hence, we assume $\|(a_1, \ldots, a_n)\|_2 = 1$.

The mixing coefficients of the source signals should be estimated from segments of constant mixing conditions. We assume that mixing conditions are locally constant and form a windowed spectral flatness measure by convolving short signal windows with a Hann window. If $h(n) := \frac{1}{2}(1 - \cos(\frac{2\pi(n-W)}{2W}))$ denotes a discrete Hann window of length $2W + 1$ centered at 0, the measure of complexity Φ for the mixture U_f is given by:

$$\Phi[U_f(\cdot,\cdot;a_1,\ldots,a_n)](x) := \sum_{t=-W}^{W} h(t)\mathrm{SFM}[U_f(\cdot,\cdot;a_1,\ldots,a_n)](x+t).$$

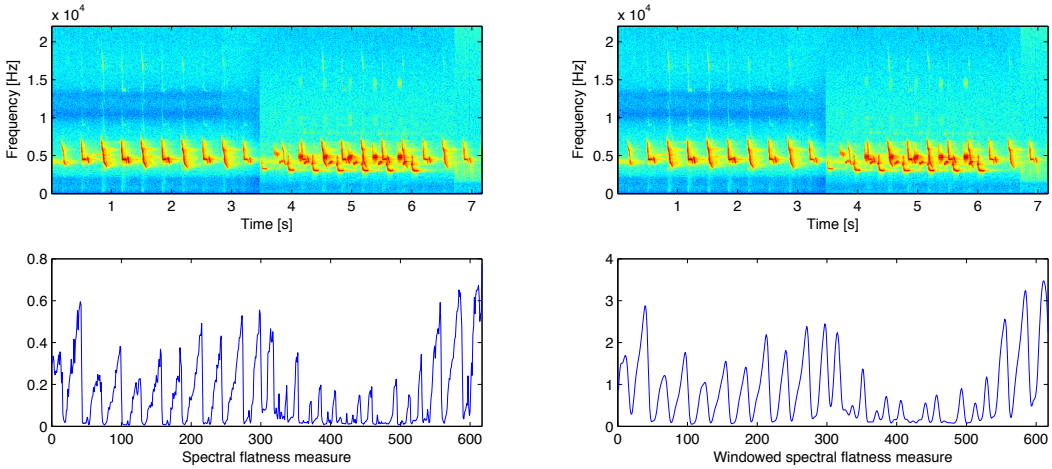

Figure 1: Left: The spectral flatness measure of a mixture of bird songs. Right: Windowed version of the measure which is used as an objective function for source separation. Note that the second half of the signal is more complex than the first.

Figure 1 gives an example of the spectral flatness measure of an audio signal and the windowed measure. In the following we will first give an overview of how to extract lower complexity components based on this complexity measure. Afterwards, we will give a more detailed description of the algorithm, followed by experimental results for artificial as well as natural audio scenes.

Starting from p initial hypotheses $A \in \mathbb{R}^{p\times n}$ for the n unmixing coefficients at the first analysis window of the signal, starting with time position τ_0, we use an optimisation algorithm to find local minimisers $(a_{k,1},\dots,a_{k,n})$ of $\Phi[U_f(\cdot,\cdot;a_1,\dots,a_n)](\tau_0)$. Because of our assumption that $\|(a_1,\dots,a_n)\|_2 = 1$, we represent each hypothesis k — given by row k of A — by polar coordinates $H_{k,1},\dots,H_{k,n-1}$. Now, we examine the audio scene at T equally spaced time steps and estimate p hypotheses for each of them by applying a local optimisation algorithm starting from the hypotheses of the previous time step. In this way, we enable our algorithm to start from good initial hypotheses whenever the unmixing parameters vary slowly. Therefore, our algorithm performs source tracking where applicable. Finally, we find hypotheses $H \in \mathbb{R}^{p\times n-1}$ for each time step. These can be combined in a matrix $\tilde{H} \in \mathbb{R}^{p(n-1)\times T}$ with one row for each mixing coefficient of each hypothesis. In order to extract meaningful components, we apply a dynamic programming algorithm to the matrix $\tilde{H}$ which extracts up to p components by selecting one hypothesis for each time step. In each component, the algorithm minimises the ℓ_2-distance between mixture hypotheses of neighbouring time steps. Moreover, we ensure, that no single mixing hypothesis is used in more than one extracted component. Mixing coefficients for time positions between two time steps are found by linear interpolation.

3. The Algorithm

In order to complete the algorithm sketched above, three parts have to be detailed. First, we need a good initialisation of the hypotheses for the n mixing coefficients of the first time step regarded by the algorithm. Second, we use gradient-based optimisation to find good mixing coefficients and therefore need to compute the partial derivatives of the complexity measure w.r.t. the mixing coefficients. Finally, we need to choose components from the hypotheses generated by the optimisation procedure.

3.1. Initial Hypotheses

We start by generating h initial hypotheses for the mixing parameters. Each hypothesis, given by polar coordinates $H_{k,1},\dots,H_{k,n-1}$, describes a point on the sphere S^{n-1}. Therefore, a good covering of the parameter space is obtained by choosing h points on the sphere which are distributed as regularly as possible. Finding a good measure for the regularity of the distribution is far from trivial. A number of measures have been proposed emphasising different aspects of the point distribution (see, e.g., [6]). A very simple approach is to find approximate solutions to Thomson's problem [7]. This problem asks for the distribution of a fixed number h of electrons on a sphere, when the electrons repel each other following Coulomb's law. Two equal electric charges repel each other with a force inversely proportional to the square of their distance.

An approximate solution is found by iteratively moving each point according to the sum of the forces exerted by the other points. This simple approach gives a good distribution of the points for the initial hypotheses used in our algorithm after a small number of iterations. For the numbers of hypotheses used in our algorithm, the computational cost for this initialisation process is negligible.

3.2. Optimisation

In order to use the complexity measure Φ defined above in an optimisation algorithm, we need to compute the partial derivatives with respect to the mixture weights. The mixture weights $(a_1,\dots,a_n)$ are given as functions of their polar coordinates $(\varphi_1,\dots,\varphi_{n-1})$ as follows:

$$a_s = \begin{cases} \cos(\varphi_s)\prod_{i=1}^{s-1}\sin(\varphi_i) & s < n \\ \prod_{i=1}^{n-1}\sin(\varphi_i) & s = n \end{cases}.$$

In the following, we will need the derivatives of the weights with respect to the polar coordinates. For a_n they are given as

$$\frac{\partial a_n}{\partial \varphi_j} = \cos(\varphi_j) \prod_{i \neq j} \sin(\varphi_i).$$

For a_s, $s < n$, the derivatives are

$$\frac{\partial a_s}{\partial \varphi_j} = \begin{cases} \cos(\varphi_s)\cos(\varphi_j)\prod_{i=1, i\neq j}^{s-1} \sin(\varphi_i) & j < s \\ -\prod_{i=1}^{s-1} \sin(\varphi_i) & j = s \\ 0 & j > s \end{cases}.$$

In order to simplify the notation, in the following, we will abbreviate

$$(a_1(\varphi_1, \ldots, \varphi_{n-1}), \ldots, a_n(\varphi_1, \ldots, \varphi_{n-1})) =: a(\varphi).$$

Now, we can proceed to calculating the partial derivatives of $\Phi[U_f(\cdot, \cdot; a(\varphi))]$ with respect to the polar coordinates $\varphi := (\varphi_1, \ldots, \varphi_{n-1})$:

$$\frac{\partial \Phi}{\partial \varphi_j}(x; \varphi) = \sum_{t=-W}^{W} h(t) \frac{\partial}{\partial \varphi_j} SFM[U_f(\cdot, \cdot; a(\varphi))](x + t).$$

Thus, in order to compute the partial derivatives of Φ, we need to know the partial derivatives of SFM. For this purpose, we abbreviate numerator and denominator of SFM by $p(x; \varphi) := \prod_{\omega=0}^{\Omega-1} (U_f(\omega, x; a(\varphi)))^{\frac{1}{\Omega}}$ and $q(x; \varphi) := \frac{1}{\Omega} \sum_{\omega=0}^{\Omega-1} U_f(\omega, x; a(\varphi))$, respectively. Now, the derivative is

$$\frac{\partial SFM[U_f(\cdot, \cdot; a(\varphi))]}{\partial \varphi_j}(x; \varphi) = \frac{\frac{\partial p}{\partial \varphi_j}(x; \varphi) q(x; \varphi) - p(x; \varphi) \frac{\partial q}{\partial \varphi_j}(x; \varphi)}{q^2(x; \varphi)}.$$

Here, the corresponding derivatives of p and q are given by

$$\frac{\partial q}{\partial \varphi_j}(x; \varphi) = \frac{1}{\Omega} \sum_{\omega=1}^{\Omega} \frac{\partial}{\partial \varphi_j} U_f(\omega, x; a(\varphi))$$

and

$$\frac{\partial p}{\partial \varphi_j}(x; \varphi) = \frac{1}{\Omega} p(t) \sum_{\omega=1}^{\Omega} \frac{\frac{\partial}{\partial \varphi_j} U_f(\omega, x; a(\varphi))}{U_f(\omega, x; a(\varphi))}.$$

Finally, the derivative of U_f can be computed from the data as

$$\frac{\partial U_f(\omega, x; a(\varphi))}{\partial \varphi_j} = \sum_{i=1}^{n} \hat{f}_i(\omega, x) \frac{\partial}{\partial \varphi_j} a_i(\varphi_1, \ldots, \varphi_{n-1}).$$

Using this derivative, locally optimal unmixing parameters can be found by a variant of a line-search algorithm. We found the simple Algorithm 1, adjusting the step size by a factor of two according to whether or not the new step size gives a better increase in the objective function than the last one, to be sufficient for our purposes.

Algorithm 1 A simple line-search step for the optimisation of U_f.

Require: step size α from previous optimisation step, objective function U_f, position x and derivative dx.
  $state := 0$
  $done :=$ **false**
  while not done **do**
    $v := U_f(x + \alpha \cdot dx)$
    if $v > U_f(x)$ **then**
      if $state == 2$ **then**
        $done :=$ **true**
      else
        $state := 1$
      end if
      $\alpha := 2\alpha$
    else
      if $state == 1$ **then**
        $done :=$ **true**
      else
        $state := 2$
      end if
      $\alpha := \alpha/2$
    end if
  end while
  return $x + \alpha \cdot dx$

3.3. Component Extraction

Finally, after transforming back angular representations into unmixing parameters, we have computed a matrix $H \in \mathbb{R}^{hn \times T}$ giving, at each time step, h hypotheses for the mixing parameters. A component is described by a sequence $((h_1, 1), (h_2, 2), \ldots, (h_T, T))$. To each element (i, j) of such a sequence, we associate the corresponding vector $v(i, j) := (H(i, j), \ldots, H(i + n - 1, j))$ of unmixing parameters. In order to extract components without abrupt changes in the unmixing parameters, where possible, we want to minimise the sum S of differences between the unmixing parameters for each component:

$$S((h_1, 1), \ldots, (h_T, T)) := \sum_{i=1}^{T-1} \|v(h_{i+1}, i + 1) - v(h_i, i)\|$$

Moreover, we want to extract disjoint components, i.e., each parameter vector $v(i, j)$ should appear in at most one component.

The first component can be extracted by computing the values of matrices D and P defined as follows. Entry D_{ij} gives the minimal costs for a component of length j, ending with the parameter vector $v(i, j)$. Entry P_{ij} gives the predecessor of $v(i, j)$ on a minimal cost component. D can be computed by standard dynamic programming, observing that $D(i, 1) = 0$ for all i and

$$D(i, j) = \min_k D(k, j - 1) \text{ for } j > 1. \quad (1)$$

The entries of P are easily found by storing the minimisers in (1).

In order to guarantee disjoint components, this process has to be slightly modified after the extraction of the first component by setting $D_{ij} = \infty$ whenever $v(i, j)$ has already been used in a component.

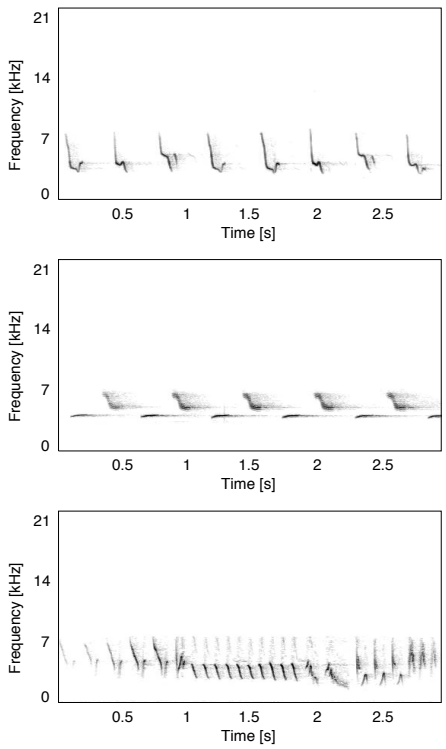

Figure 2: Three signals used for artificial mixing of test signals: Chiff-chaff (left), great tit (middle) and chaffinch (right).

For the analysis of audio scenes, it is usually sufficient to reconstruct the spectrum of a component by applying the selected unmixing parameters. It is, however, sometimes desirable to reconstruct audio signals from the components. In this case, it is possible to use the estimated power spectrum as a mask for weighting the complex spectrum of the mixed signals. Then, a signal can be reconstructed from this weighted spectrum which includes phase information in addition to the magnitude information contained in the power spectrum.

4. Results

The algorithm described above has been tested on two datasets. First, in order to test basic functionality, artificial mixtures of natural sound sources have been created. Then, in order to get closer to real-world applications, we present an example of two speakers recorded by two microphones in an office room.

Artificial mixtures of sound sources were created using recordings from three species of birds, the chiff-chaff, the great tit and the chaffinch as depicted in Figure 2.

In the first test, the songs of chiff-chaff and great tit were mixed with constant mixing parameters to obtain the signals shown in the top of Figure 3. The first mixture is obtained by weighting each signal by a factor of $\sqrt{2}$, the second signal is obtained by weighting the great tit song by a factor of $\frac{\sqrt{3}}{2}$ and the chiff-chaff song with a factor of $\frac{1}{2}$. As can be seen in the figure, the chiff-chaff song can be recovered almost perfectly from these mixtures.

One of the main features distinguishing our method of source separation from previous solutions is that it is able to track varying mixing conditions. Figure 4 shows an example

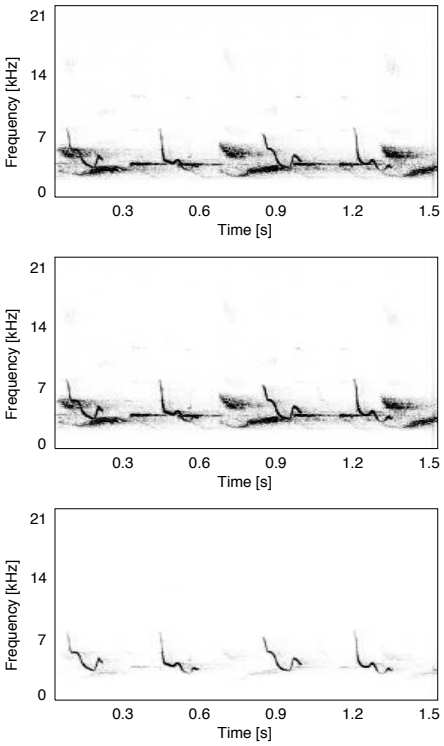

Figure 3: Source separation for two artificially mixed sources. The upper part shows the spectrograms of two different mixed signals of chiff-chaff and great tit song. The lower part shows the spectrogram of the simplest component extracted from these mixtures.

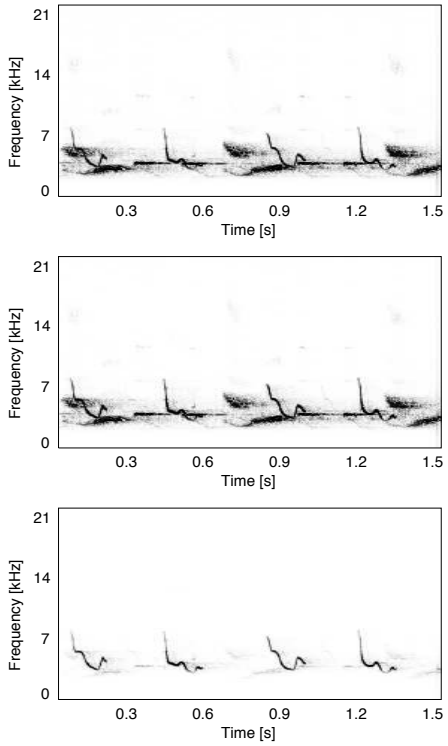

Figure 4: Source separation for two artificially mixed sources with varying mixing parameters.

of the same two signals as in the previous example, this time mixed by varying mixing parameters. For a second mixture, the two source signals were mixed by weighting the first one by the factor $\sin\alpha$ and the second one by the factor $\cos\alpha$ for an angle α which changes gradually over the playback time of the signals. It starts with $\alpha = 0$ in the first fourth of the signal, then changes to $\frac{\pi}{8}$ in the second fourth, to $\frac{\pi}{4}$ in the third, and to $\frac{3\pi}{8}$ in the final fourth. A second mixture signal was obtained by using the weight $\cos\alpha$ on the first source signal and the weight $\sin\alpha$ on the second. Mixing parameters are interpolated linearly between the different parts. Again, the song of the chiff-chaff can be extracted from the mixtures with the same quality as in the case of constant mixture parameters.

As a final test with artificial mixtures, we present a result for a mixture of three sources with varying mixing parameters. The first three spectrograms in Figure 5 show three mixtures of the three signals shown in Figure 2. The mixing parameters have been created similar to those in the previous example. Using the same angle α as a parameter, the signals have been mixed according to Table 1. Here, the song of the chaffinch is extracted almost perfectly, except for some remnants of the song of the great tit to be found at the beginning and the end of the extracted component.

A result which is closer to real world applications is the separation of a mixture of two speakers talking simultaneously in a room recorded by two microphones. The data used for this test is an example from an ICA-based source separation method which incorporated estimating time delay between the two microphones [8]. Note, that no estimation of time delay is necessary for our method. Figure 6 shows spectrograms of the signals recorded by the microphones. Each of the two speak-

Table 1: Mixture parameters for the three signals shown in Figure 5. The mixtures were created using an angle α varying from 0 over $\frac{\pi}{8}$ and $\frac{\pi}{4}$ to $\frac{3\pi}{8}$.

Mixture	Weight for signal 1	Weight for signal 2	Weight for signal 3
Mixture 1	$\cos(\alpha)$	$\sin(\alpha)$	$\cos(\alpha + \frac{\pi}{2})$
Mixture 2	$\sin(\alpha + \frac{\pi}{2})$	$\cos(\alpha)$	$\cos(\alpha)$
Mixture 3	$\cos(\alpha + \frac{\pi}{2})$	$\cos(\alpha)$	$\sin(\alpha)$

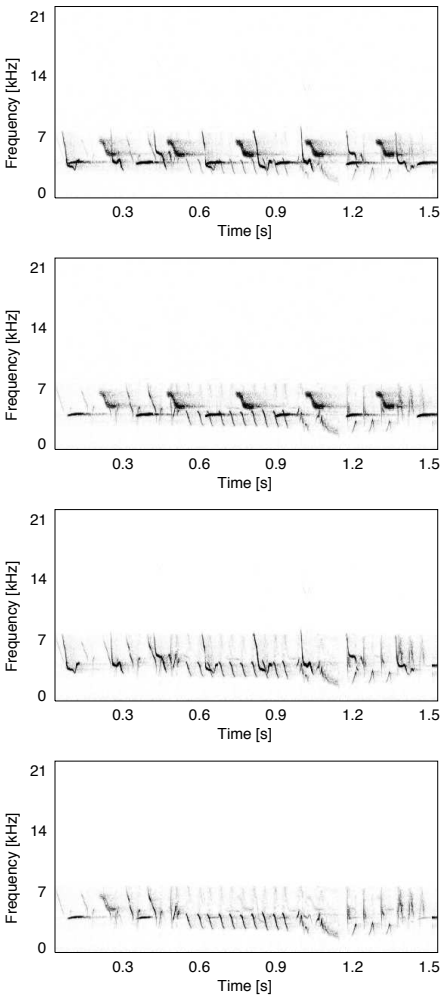

Figure 5: Source separation for three artificially mixed sources with varying mixing parameters.

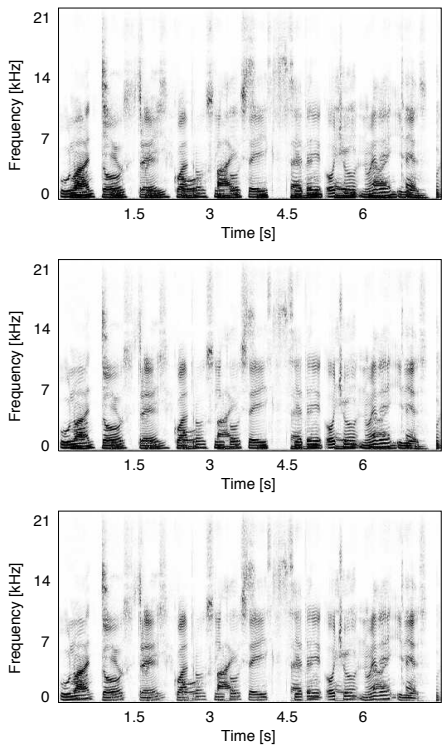

Figure 6: Source separation for two speakers in a room recorded by two microphones. The upper spectrograms show spectrograms of the two signals picked up by the microphones. The lower spectrogram shows the best component reconstructed by the algorithm.

ers is counting from one to ten, one speaker in English, one in Spanish. Their utterings strongly overlap in time and frequency. Looking at the spectrograms shows that the extracted component is indeed much simpler than the mixtures. Listening to a reconstructed signal reveals that the utterings of one of the speakers are strongly attenuated.

5. Conclusion

In this work, the spectral flatness measure is proposed as a measure of component complexity for blind source separation. Together with particle-based gradient descent optimisation of this measure and a dynamic programming approach for component extraction, it is possible to extract low complexity components from signals with time-varying mixing conditions.

Unfortunately, usually only the best component extracted from the mixtures gives sensible results. The other components extracted are very similar to the best component most of the time. Thus, in order to extract more than one component with our methods, a different way of extracting these additional components from the optimisation hypotheses has to be found. A simple method to do so would be to subtract the best extracted component from the mixtures with a suitable weight and then repeat the optimisation process with these simplified mixtures. Another idea would be to include a measure of similarity to already extracted components into the component extraction process.

Additional experiments show that complex real-world signals such as real mixtures of bird songs are yet out of the scope of our method. This problem is probably due to non-linear mixing. Thus, a non-linear unmixing approach would be necessary.

6. Acknowledgements

During the preparation of this work, the author has been a member of the Multimedia Signal Processing Group of the University of Bonn headed by M. Clausen. The author wishes to express his gratitude for all the support he has received.

7. References

[1] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, 1st ed. Academic Press, 2010.

[2] P. Comon, "Independent component analysis, A new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[3] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications." *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[4] J. F. Cardoso and A. Souloumiac, "Blind beamforming for non-gaussian signals," *IEE Proceedings F Radar and Signal Processing*, vol. 140, no. 6, p. 362, 1993.

[5] T. W. Lee, M. S. Lewicki, M. Girolami, and T. J. Sejnowski, "Blind source separation of more sources than mixtures usingovercomplete representations," *IEEE Signal Processing Letters*, vol. 6, no. 4, pp. 87–90, 1999.

[6] J. H. Conway, N. J. A. Sloane, and E. Bannai, *Sphere-packings, lattices, and groups*. New York, NY, USA: Springer-Verlag New York, Inc., 1987.

[7] J. J. Thomson, "On the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure," *Philosophical Magazine*, vol. 7, no. 39, pp. 237–265, March 1904.

[8] T.-W. Lee, A. Ziehe, R. Orglmeister, and T. Sejnowski, "Combining time-delayed decorrelation and ICA: towards solving the cocktail party problem," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, May 1998, pp. 1249–1252.