SOUND SOURCE SEPARATION: PREPROCESSING FOR HEARING AIDS AND STRUCTURED AUDIO CODING

Harald Viste, Gianpaolo Evangelista

Communication Systems Department
Swiss Federal Institute of Technology Lausanne (EPFL)
harald.viste@epfl.ch gianpaolo.evangelista@epfl.ch

ABSTRACT

In this paper we consider the problem of separating different sound sources in multichannel audio signals. Different approaches to the problem of Blind Source Separation (BSS), e.g. the Independent Component Analysis (ICA) originally proposed by Herault and Jutten, and extensions to this including delays, work fine for artificially mixed signals. However the quality of the separated signals is severely degraded for real sound recordings when there is reverberation. We consider the system with 2 sources and 2 sensors, and show how we can improve the quality of the separation by a simple model of the audio scene. More specifically we estimate the delays between the sensor signals, and put constraints on the deconvolution coefficients.

1. INTRODUCTION

The separation of sound sources is of great interest to many applications, in particular hearing aids and structured audio coding.

A hearing-impaired person will often have problems in perceiving a signal in the presence of noise. The "Cocktail party" problem, i.e. separating speech from background noise (unwanted speech, music, noise) is well known, and is of great nuisance for the hearing impaired person. By separating the different sound sources, one can easily suppress what is considered unimportant/ disturbing, and this would undoubtedly be a major achievement.

High quality audio compression techniques have previously been based on psychoacoustically motivated spectral models. Such techniques, often called perceptual coders, only seek to eliminate redundancy at a particular level, i.e. that of the audio waveform and its perception. In contrast to this, structured coding methods can exploit structure and redundancy at many different levels of a sound scene, and in many cases result in representations which are several orders of magnitude more compressed than the equivalent perceptually coded representation. A structured representation of sound is one which makes model-based assumptions about the origin and nature of its content. One can then transmit, first, a description of the model to be used for the content, and then the parameters for using this model. The structured audio feature included in the MPEG-4 standard allows the efficient description and transmission of synthetic sound. At present, structured coding is only feasible if the audio stream is generated by using synthesis algorithms and computer programs to define the score. For this reason, the structured audio feature of the MPEG-4 standard is not widely used yet. Only when one is able to separate the sound sources, and model and code these individually, will the standard become widely used.

Human perception of sounds is much more advanced than any technical system devised so far. A human listener is able to distinguish different tones, instruments, voices and noise in a complex auditory scene. One of the most important skills is the ability to decide what is considered relevant in the audio signal and what is not, and then focus on the relevant part while disregarding the rest. In general, the problem of separating different sound sources is a denoising problem, where the "signal" is the important part of the audio stream, and the noise is everything else.

The approaches that have been used to try to solve the separation problem can be grouped into two categories. The first one is Blind Source Separation (BSS), where no assumption is made about the sources except that they are statistically independent. The second one is model based, where the models can be at any level, ranging from low level models, e.g. harmonicity of sounds, common onset/offset, common FM/AM, to higher level models, e.g. note transition probability, rythm modelling, music theory, etc. These two approaches have traditionally been treated separately. By combining them, one expects to achieve improvements over both.

BSS techniques will be briefly reviewed in section 2. Section 3 illustrates the models and extensions we have introduced. The results of some experiments are presented in section 4. Finally, section 5 contains the conclusion.

2. BACKGROUND

Since BSS was introduced in the late 80's, the topic has become increasingly popular, and there is a vast number of articles on the subject. We will first present the method originally proposed by Herault and Jutten [4] for separation of

instantaneous mixtures of signals. Then we will look at one of the approaches that has been proposed to deal with mixtures of delayed and convolved sources, [6, 2, 8].

2.1. Instantaneous mixtures

In 1990, Herault and Jutten [4] proposed an adaptive method for blind separation of sources from an observed instantaneous mixture of these. The mixing model is given by:

$$X(t) = AS(t) \tag{1}$$

where $X(t)$ is the vector containing the N observed signals at time t, A is the $N \times N$ mixing matrix, and $S(t)$ is the vector containing the N original source signals. To separate the signals we want to find a demixing matrix W, yielding the vector of separated signals Y, up to a scaling and permutation uncertainty:

$$Y(t) = W^{-1}X(t) \tag{2}$$

Due to the scaling uncertainty, all elements on the diagonal of W are set to 1. By applying the gradient method to the LMSE, one arrives at the following iterative algorithm to update the coefficients of the demixing matrix:

$$\Delta W(i,k) \propto f(Y_i)g(Y_k), \qquad i \neq k \tag{3}$$

where f and g are two distinct odd functions that are applied in order to achieve independence rather than zero-covariance only [4].

The method is computationally inexpensive for small systems (small number of sources and sensors), and works surprisingly well for instantaneous mixtures. However, mixtures of audio signals are typically not instantaneous. Rather the sources are delayed and convolved (e.g. room impulse response). In such cases the method fails to properly separate the sources.

2.2. Convolved mixture

The original method has been extended to include delayed and convolved sources [6]. In the general case, the mixing system becomes:

$$X(t) = \sum_{d=0}^{m} A_d S(t-d) \tag{4}$$

where we have a sequence of matrixes A_d for the different delays d. The vector $A_d(i,k), d = 0,..,D$ for fixed i,k is a FIR filter describing the impulse response between source k and sensor i. This can be seen as a matrix A where each element is an FIR filter, called the convolution matrix. To separate the sources, we want to find the demixing matrices W_d, also called the deconvolution matrix W. The separation equation is:

$$Y(t) = W_0^{-1}\left[X(t) - \sum_{d=1}^{D} W_d Y(t-d)\right] \tag{5}$$

and the coefficients of the demixing matrices are updated as follows:

$$\Delta W_d(i,k) \propto Y_i(t)Y_k(t-d), \qquad i \neq k \tag{6}$$

This method works well for synthetically mixed signals, i.e., systems where each source signal is only scaled and delayed before arriving at the sensors. The method works well even for music, where the different instruments play together and are therefore not necessarily independent. However, when using real life recordings, some assumptions in the model are no longer valid, and the quality of the separation is severely degraded.

3. EXTENSIONS

In this section we will discuss the extensions we have introduced to improve the quality of the separated signals. First of all we will see why the method degrades for real recordings, and how this can be remedied. Then we will see how some knowledge about the system can improve the results.

3.1. Model change

The separation method discussed in section 2.2, is based on the assumption that in the mixing process, the elements on the diagonals of $A_d, d > 0$ are all equal 0. This means that sensor i records the original source i (unfiltered) plus delayed/filtered versions of the other sources. In the real world, this assumption does, of course, not hold, since all source signals are filtered (room impulse response). There are methods that take this into account, and try to equalize each separated signal, [3, 9, 5, 7]. However, these methods are computationally much more expensive. There is also a robustness issue, in that the transfer functions (room impulse responses) are modeled as FIR filters that are not necessarily minimum phase, and accordingly the inverse IIR filter/FIR filter approximation may not be stable or do not converge.

In order to devise a method that works with real recordings, while keeping the simple complexity, we assume that the transfer functions between the sources and sensors are mutually similar. When the sensors are closely spaced, like in hearing aids, this assumptions seems to hold. In this case, the method is still given by equations 5 and 6, but for each pair (i,k), we consider just one delay, namely δ_{ik} which is the difference in delay between sensor i and k for the signal from source i. In reality, since this delay does not have to be an integer, we consider a short range of integer delays around this value. In addition to improving the separated signals, this also decreases the complexity of the algorithm. Also, if zero delays are not considered, the matrix inversion term in equation 5 disappears.

This new assumption on the impulse responses also makes it possible to include knowledge about the system into the separation model to improve the separation.

3.2. Sensor geometry

In most cases, the sensor geometry will be fixed. This is true for the microphones in hearing aids, and most microphone setups in general recording situations. When the geometry of the sensors is known, this can be used to predict maximum delays between sensors for the sound wave that comes directly from the sources. In environments with moderate reverberation, this can be used as an upper limit for the ranges of delays considered. This is useful to restrict the number of coefficients in the deconvolution algorithm and obtain a more robust system. If the environment is highly reverberant, it may be necessary to include knowledge about the environment to increase these upper limits on the delays.

If the source positions are fixed and known, this may be used to give better estimates of the actual delays. However, in many situations, the environment will change, so it is more useful to estimate the delays in real time.

3.3. Delay estimation.

Not all sources emit all the time. We consider the problem with 2 sources and 2 sensors. The crosscorrelations between the sensor signals are evaluated on the fly. The maximum around lag 0 will give an indication of the delay, and in periods where one of the sources is silent, this maximum will be a good estimate of the real delay. Figure 1 shows the in-

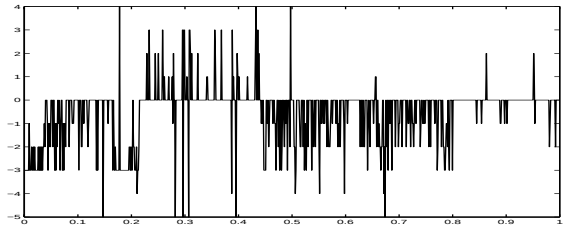

Figure 1: *Index of maximum of crosscorrelation between sensor 1 and 2, as a function of time*

dex of this maximum plotted versus time (x-axis, 1 second). One can see that the delay difference between the sensors is around 3-4 samples for source 1 (positive y-axis), and about 4-5 samples for source 2 (negative y-axis).

In most cases the sensor configuration is fixed, and one could assume that the sources are fixed as well, although this may not be true in general. When the sources move, the delay differences between the sensors change, and since we only consider short ranges of delays, the real delay may move out of this range. Anyway, since the speed of movement of the sources is normally limited, it is possible to track this movement by looking at the coefficients in this range of delays. When the maximum in this range moves away from the center, the range may be adjusted. Figure 2 shows the magnitude of the demixing coefficients for delays in the range $0 - 20$. We notice that the main peaks are

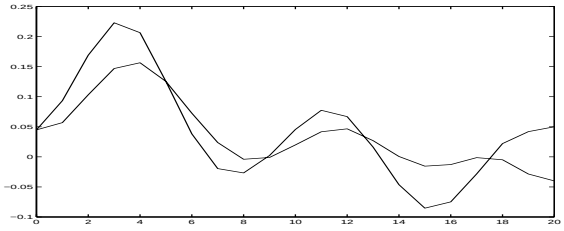

Figure 2: *Magnitude of coefficients in demixing matrices as a function of delay. Upper curve: source 1 to sensor 2. Lower curve: source 2 to sensor 1.*

found for delays about 3 and 4 samples, which corresponds to what we found in figure 1.

3.4. Constraints on the deconvolution coefficients.

When the sources move, not only the delay between the sensors may change, but also the amplitudes, due to the changing physical conditions. In general, the mixing matrices in equations 1 and 4 may vary over time, i.e. $A_d = A_d(t)$. If the convergence of the demixing matrices in equations (3) and (6) is fast enough, the algorithms may be able to keep up with these changes. This depends on the proportionality factor used in these equations. If this factor is large, the algorithm will converge faster, and also be better able to cope with changing mixing conditions.

The drawback of using a large proportionality factor, is that the demixing coefficients gets more noisy. Figure 3 shows this. A too agressive convergence speed can lead to annoying artifacts, thus decreasing the quality of the separated signals.

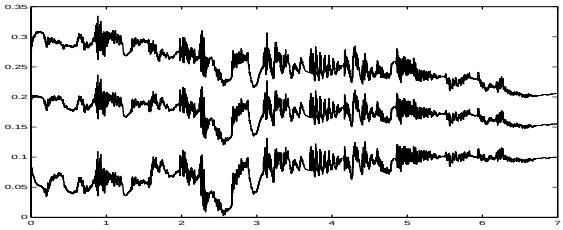

Figure 3: *Demixing coefficients as function of time. From top to bottom: delays* $3, 4, 2$

By smoothing these coefficients in time, one can avoid this problem, but of course this leads to a slower convergence. Here, one must be aware of the problem of masking, i.e. that some object suddenly crosses the sound beams. Depending on the sensor geometry this may affect only one of the sensors, thus giving rise to abrupt changes in the deconvolution coefficients.

4. RESULTS

We have tested our method on separation of real-life mixtures.

Environment	Computer lab room
Sources	2
Sensors	2 omnidirectional microphones
Sensor spacing	approx 3 cm (4 samples)
Sampling rate	44100 (CD-quality)

Figure 4 shows an excerpt of a piece of music for two guitars. It was recorded as two separate parts with the same guitar, first the rythm/chord part, and then the solo part with a different source/sensor arrangement. Rather than looking at the figure, you can listen to the sound samples at http://lcavwww.epfl.ch/∼viste/present.html. The separation is not perfect, but one can clearly hear that in each of the separated signals, one of the sources is attenuated significantly. Moreover, the noise artifacts heard with the original methods, have almost completely disappeared.

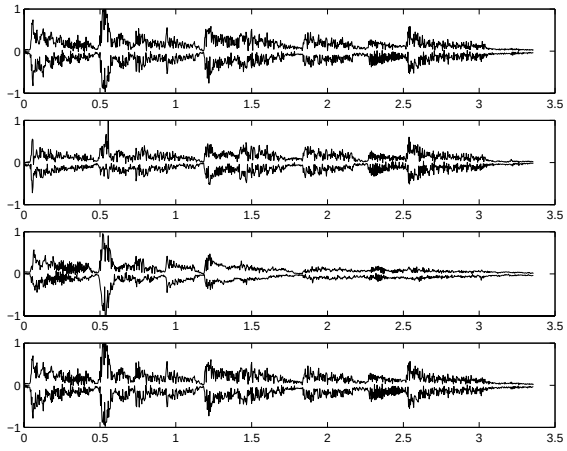

Figure 4: *Sound source separation in a piece for two guitars (Top to bottom): A: Recorded signal, left channel. B: Separated signal, rythm part. C: Separated signal, solo part. D: Recorded signal, right channel.*

Also, due to computer noise in the recording environment, the sources were placed relatively close to the sensors. This makes the problem of separating the sources even harder, since the sources can no longer be considered as point sources. This also makes our assumption more questionable, namely that the transfer functions are similar for all the sensors.

The separation algorithm was implemented in Java. For a 15 seconds excerpt, ran on a Pentium III 500MHz, the computation times are approximately:

Range of delays	Computation time
3	0.6 s
5	0.95 s
10	1.65 s

5. CONCLUSION

We have introduced some model extensions to improve the quality of the separated signals from real-life recordings. In addition, the extensions also decrease the computation power needed to perform the separation. The separation is not perfect, and the method suffers the restriction that the number of sources must be smaller than or equal to the number of sensors.

More sophisticated algorithms that may achieve better separation have been proposed, which also include equalization of the sources [5, 9, 7]. Even though they use the natural gradient [1], or can be performed in the frequency domain [7], these methods are computationally much more intensive.

6. REFERENCES

[1] Amari,S., Cichocki, A., and Yang, H.H., "A new learning algorithm for blind signal separation", In Advances in Neural Information Processing Systems, MIT Press, 1996.

[2] Bamford, P. and Canagarajah, N., "Separation of Multiple Signals in Hearing Aids by Output Decorrelation and Time-Delay Estimation", IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics, pp. 7–10, 1995.

[3] Bell, A.J. and Sejnowski, T.J., "An information-maximisation approach to blind separation and blind deconvolution", Neural Computations 7, pp. 1004–1034, 1995.

[4] Jutten, C. and Herault, J., "Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture", Signal Processing, vol. 24, pp. 1–10, 1991.

[5] Lee, T-W., Bell, A.J., Lambert, R.H., "Blind separation of delayed and convolved sources", In Advances in Neural Information Processing Systems, MIT Press, 1997.

[6] Nomura, T., Eguchi, M., Niwamoto, H., Kokubo, H. and Miyamoto, M., "An Extension of The Herault-Jutten Network to Signals Including Delays for Blind Separation", IEEE Neural Networks for Signal Processing VI, pp. 443–452, 1996.

[7] Smaragdis, P., "Information theoretic approaches to sound separation", Masters Thesis, MIT Media Arts and Sciences Dept., 1997

[8] Torkkola, K., "Blind separation of delayed sources based on information maximization", IEEE ICASSP, pp. 3509–3512, 1996.

[9] Torkkola, K., "Blind separation of convolved sources based on information maximization", IEEE Signal Processing Society Workshop, pp. 423–432, 1996.