Applying Blind Source Separation and Deconvolution to Real-World Acoustic Environments

Alex Westner and V. Michael Bove, Jr.

MIT Media Lab
20 Ames Street
Cambridge, MA, USA
westner@media.mit.edu, vmb@media.mit.edu

A subset of signal processing researchers strive to enable digital systems to isolate individual sound sources from an environment containing several. As part of this effort, we experiment with an approach known as "blind source separation and deconvolution" (BSSD). The main contribution of this paper is the application of current BSSD algorithms to unconstrained real-world applications.

1 Introduction

Sound engineers commonly use digital systems to record and analyze audio in music and film studios. Often, they need to cleanly access a single sound source such as an instrument or voice. While humans can focus their attention on any one sound source out of a mixture of many (a phenomenon termed in 1953 by E. Collin Cherry [1] as the "cocktail-party effect"), current digital audio systems lack this ability.

This paper describes one way in which researchers in this field are approaching the goal of enabling a digital system to accomplish this task, which is commonly known as *source separation*. The main contribution of the work discussed in this paper is the application of current source separation algorithms to unconstrained real-world environments.

2 Blind Source Separation and Deconvolution

The objective of source separation is to extract certain desired signals from a mixture of many signals. When we know the transformation used to mix the sounds, the separation is straightforward. In practice, however, we often don't know *how* the sources were mixed together, which makes the problem much more difficult. The process of extracting the desired sources given *only* the mixtures is commonly referred to as *blind source separation* [2].

As we apply blind source separation techniques to real-world mixtures, we must first consider the way in which the sounds have been mixed. Consider two sounds recorded in a room using two microphones. Each microphone receives a direct copy of both sound sources (at different propagation delays between each source and microphone) as well as several reflected and modified copies of each source. These acoustic effects can be modeled in a linear system [3]:

$$x_1(t) = \boldsymbol{a}_{11} * s_1(t) + \boldsymbol{a}_{12} * s_2(t) \tag{1}$$

$$x_2(t) = \boldsymbol{a}_{21} * s_1(t) + \boldsymbol{a}_{22} * s_2(t) \tag{2}$$

where x_i are the microphone signals, a_{ij} are the room impulse responses between each source and microphone, and s_j are the sound sources. Consolidating the two above equations into matrix form yields

$$\boldsymbol{x}(t) = \underline{\boldsymbol{A}} * \boldsymbol{s}(t) \tag{3}$$

where we refer to $\underline{\boldsymbol{A}}$ as the *mixing matrix*.

To find the original sound sources recorded by microphones in a room, we must *deconvolve* the room impulse responses from the original sound sources. Since we have no prior knowledge of what these room impulse responses are, we call this process *blind deconvolution* [4].

We can combine blind source separation with blind deconvolution to find the *inverse* of $\underline{\boldsymbol{A}}$, which is the *unmixing matrix*, $\underline{\boldsymbol{W}}$. By convolving $\underline{\boldsymbol{W}}$ with $\boldsymbol{x}(t)$, we obtain estimates, $\boldsymbol{u}(t)$, of the original sources [5].

$$\boldsymbol{u}(t) = \underline{\boldsymbol{W}} * \boldsymbol{x}(t) \tag{4}$$

3 The Information Maximization Approach

After reviewing several blind source separation and deconvolution (BSSD) algorithms [6, 7, 8], we concluded that information maximization (infomax) methods [5, 8, 9, 10] best suited our experiments with acoustically-mixed sounds [11].

From the information theoretic viewpoint, the goal of BSSD is to find an unmixing matrix, $\boldsymbol{W}$, such that when multiplied with the mixtures, $\boldsymbol{x}(t)$, will yield outputs, $\boldsymbol{u}(t)$, that do not contain any mutual information. In other words, we are trying to decorrelate the outputs from one another in an optimal way such that one-and-only-one sound source resides in each output. For further details about the approach, please see Bell and Sejnowski's work [8, 12].

They use stochastic gradient ascent to adapt the unmixing filter weights:

$$\Delta \boldsymbol{W} \propto \left[\boldsymbol{W}^T\right]^{-1} + g(\boldsymbol{u}(t))\, \boldsymbol{x}(t)^T \tag{5}$$

where $g(\cdot)$ is a *sigmoid* function that produces the higher-order statistics needed to decorrelate the sources and make them independent of one another.

For our experiments, we used a slightly modified version of Paris Smaragdis's [10, 13] frequency domain implementation. The algorithm runs off-line and proceeds as follows [11]:

1. Pre-process the time-domain microphone signals, $\boldsymbol{x}(t)$, by filtering out as much noise as possible; subtract the mean from each signal.
2. Initialize the frequency domain unmixing filters, $\underline{\boldsymbol{W}}$.
3. Take a block of input data and convert it into the frequency domain, using the Fast Fourier Transform (FFT).
4. Filter the frequency domain input block, $\underline{\boldsymbol{X}}$, through $\underline{\boldsymbol{W}}$ to get the estimated sources, $\underline{\boldsymbol{U}}$.
5. Pass $\underline{\boldsymbol{U}}$ through the frequency domain nonlinearity: $g(\underline{\boldsymbol{U}}) = \tanh(\text{real}(\underline{\boldsymbol{U}})) + \tanh(\text{imag}(\underline{\boldsymbol{U}}))$
6. Use the weight update equation to compute $\Delta\underline{\boldsymbol{W}}$.
7. Take the next block of input data, convert it into the frequency domain, and proceed from step 4. Repeat this process until the unmixing filters have converged upon a solution, passing several times through the data, if necessary.
8. Normalize $\underline{\boldsymbol{W}}$ and convert it back into the time domain, using the Inverse Fast Fourier Transform (IFFT).
9. Convolve the time domain unmixing filters, $\boldsymbol{W}(t)$, with $\boldsymbol{x}(t)$ to get estimates of the original sources.

4 Initial Experiments

To gain a better sense of the filter configurations needed to achieve sound separation, we ran a *non*-blind experiment: we mixed two sound sources using known, pre-computed room impulse responses, and then unmixed them using the appropriate inverses. By varying the filter configurations, we obtained quantitative results about how the number of microphones and the unmixing filter lengths might affect the performance of a BSSD algorithm [14]. As expected, more microphones and longer unmixing filter lengths yielded better separation. When implementing *blind* algorithms, however, additional microphones did not seem to improve separation [11].

We then experimented with data from a controlled environment: Lucent Technologies' Varechoic Chamber, a $6.7 \times 6.1 \times 2.9$ meter room with computer-controlled reverberation [15, 16]. For our experiment, we used two sound sources (five seconds of male and female speech, played through two loudspeakers) and two microphones, mounted on orthogonal walls of the room. See Figure 1 for the physical layout. We used data from four different reverberations settings of the room: $0.1, 0.25, 0.5$ and 0.9 seconds.

After listening to the data, we observed that the male speech sample was louder than the female speech sample. As a result, the female speech was more difficult to extract. Table 1 shows the signal-to-noise ratios (SNR's) of the four data sets used in this experiment, where each value represents how much louder (in dB) one source is from the other.

With the room set at a short reverberation time of 0.1 seconds, our BSSD algorithm performed very well, isolating the male voice by 7.8dB and the female voice by 6.7dB. Comparing these output signals with the corresponding inputs, our algorithm was able to increase the separation of the male voice by 3.3dB and the female voice by 7.4dB.

Table 2 shows the overall separation SNR's, along with the *average separation improvement* (ASI) SNR's for our experiment. The ASI is the average difference between output and input SNR's, and is a better reflection of how the algorithm is performing. It is clearer in the ASI measurement that, as the room becomes more reverberant, the performance of our BSSD algorithm suffers significantly. These results led us to study how real-world conditions affect BSSD.

reverb time	male/female SNR mic 1	female/male SNR mic 2
0.1s	4.5dB	-0.7dB
0.25s	3.5dB	-1.5dB
0.5s	6.1dB	-6.1dB
0.9s	5.1dB	-3.5dB

Table 1. The relative signal power of one source to the other. For example, for the 0.1 second data set, the male source is 4.5dB louder than the female source in microphone 1, while the female source is 0.7dB quieter than the male source in microphone 2.

reverb time	male/female SNR	female/male SNR	ASI
0.1s	7.8dB	6.7dB	5.4dB
0.25s	2.2dB	3.7dB	2.0dB
0.5s	6.3dB	-1.2dB	2.6dB
0.9s	5.1dB	0.6dB	2.1dB

Table 2. After processing the microphone signals through a BSSD algorithm, we calculated the SNR's (in dB) of each separated source, relative to the other source. At about 6dB of separation, the other sound source is barely audible. The ASI shows how well the algorithm performed overall.

5 Real-World Experiments

BSSD algorithms are quite sensitive to the less-than-ideal conditions of the real world. Microphone choice, sound source-microphone distances, the sources' directivity and room noise all significantly affect BSSD performance.

5.1 Microphones

The performance of current BSSD algorithms depends greatly upon the choice and configuration of the microphones used to record the sounds. In our early experiments with real-world conditions, we hung omni-directional condenser microphones (Audio Technica ATM-10a's) from the ceiling of a conference room. These sensitive microphones recorded the audio in the room accurately and completely — which was not entirely to our benefit. The microphone signals contained too many reflections and were too rich in energy for our BSSD algorithms, which were unable to focus on the relevant sound sources.

We then chose to work with cardioid dynamic microphones (Shure SM-58's), hoping that this would shorten the effective distance range of the sound pickup, and reduce the amount of reflections in the signals. Later on, however, we learned that these microphones, as well, may not be the right choice for our BSSD application (see Section 6).

5.2 Sound Source-Microphone Geometry

Using an array of cardioid dynamic microphones, we began to collect our own data to experiment with. We recorded two simultaneous sounds in a conference room, varying the types of sounds: two people speaking; one person speaking and a radio playing; and two people playing musical instruments.

We did not have much success, however, getting our BSSD algorithm to find the original sound sources in these recordings. Russ Lambert and Te-Won Lee offered some advice on how to set up the experiment to yield better results. In their experience, BSSD algorithms tend to fix on the sound with the most energy in each microphone signal. For example, when separating two sound sources with a pair of microphones, source A should be closer to microphone 1 and source B should be closer to microphone 2.

This observation raises two important concerns about the acoustics of BSSD. First, when one of the sources being separated is significantly louder than the others, the loud source tends to dominate both microphone signals, leaving the algorithm unable to find and extract the quieter sounds from the mix.

The second concern is about how sounds propagate through a room. When a person speaks, she sends her speech in a particular direction. Her head, in fact, acts as a baffle, partially impeding the soundwaves from traveling in other directions [17]. If an array of microphones is being used to record the speaker, the microphones that lie in the path of the speaker's voice will acquire the direct soundwaves, while the other microphones will acquire only the reflected soundwaves at a lesser amplitude.

Now consider a musical instrument like a marimba. When someone strikes a bar of the marimba, the sound resonates and projects outward in all directions. This non-directive sound source permeates a space and creates many reflections, establishing a very strong presence in *every* microphone signal of the array. As a result of this signal domination — according to Lambert, Lee and our own experiments — a BSSD algorithm would not be able to find any other sound source than the marimba.

5.3 Room Noise

During one of our experiments with real-world data (two males speaking simultaneously in a noisy, cluttered conference room, one a little louder than the other), we observed that our BSSD algorithm was converging on noise from a computer disk drive, rather than on the quieter speaker. There are two reasons why this happened. First, the disk drive noise is a non-directive source, as described in the previous section — it disperses in all directions, filling the room. Therefore, the algorithm sees this room noise in every microphone signal it processes. Secondly, the persistence and invariance of the disk-drive noise leads the algorithm to fixate upon it, neglecting the more dynamic speech sounds.

Figure 2 shows two spectrograms: on the left is one of the two microphone signals that we fed into our BSSD algorithm; on the right is the corresponding output from the algorithm. Notice the thin, horizontal energy bands at about 7.1kHz and 6.5kHz, in the output signal. These bands are not present in the microphone signal on the left. The BSSD algorithm has, unfortunately, strengthened these undesired frequency bands, which are likely to have come from the computer disk drive.

To get a better "look" at the noise in the room, we recorded five seconds of data with no speakers present. In the spectrogram of this recording, shown in Figure 3, the horizontal noise

bands are much more apparent. Since our BSSD algorithm processed the microphone data in small chunks (about 0.25 seconds long), the persistent noises appeared in almost every chunk. The algorithm sees more data from the noise than the desired speech signals, and therefore considers the noise to be more important, erroneously converging toward it.

5.4 Real-World BSSD

Finally, we made a recording of a male and a female speaking simultaneously, hanging two cardioid dynamic microphones from the ceiling of that same noisy and cluttered conference room (with a reverberation time of about 200ms). After processing the mixture data through our BSSD algorithm, we observed that, while not perfect, the algorithm does an adequate job of separating the two voices, in spite of the noise and reverberation. In the output containing the female voice, the male voice isn't attenuated by much, but it is perceptually "blurred," to the point of making it much less intelligible in comparison to the female voice. In the output containing the male voice, the female voice is significantly attenuated, but the signal itself is more reverberant. Please see Alex Westner's WWW page for an audio demonstration: http://www.media.mit.edu/~westner/sepdemo.html

6 Conclusions and Future Directions

Several conclusions about BSSD result from this work, particularly regarding microphones, acoustics, noise, and filter initialization. After completing the experiment discussed in the previous section of this paper, we listened again to the data that we had been using and discovered one drawback to using directional microphones for BSSD: if a sound source is not positioned inside the pick-up pattern of the microphone, then the microphone *only* records the reflections of that sound source, and *none* of the direct sound. Since BSSD algorithms have a great deal of difficulty dealing with reflections, perhaps directional microphones are not the right choice for this application. We might be better off ensuring that we pick up the direct sound than attempting to suppress the reflections. More experimentation using different types of microphones needs to be done. We are currently looking into boundary microphones, which we hope will help us solve this dilemma.

As shown in the experiments we've discussed here, reverberation and room noise considerably degrade the performance of a BSSD algorithm. Since current BSSD algorithms are so sensitive to the environments in which they are used, they will only perform reliably in acoustically-treated spaces devoid of persistant noises. To deal with room noise, we should be able to *automatically* detect, and then ignore, the steady-state frequency components that belong to undesired room noise. The frequency components of speech, for example, smoothly change over a small duration of, say, 500ms, while noise bands, like the ones shown in Figure 3, persist for as much as five seconds.

Another aspect of our BSSD algorithm that we're looking to improve is the initialization step, which, right now, essentially zeros the unmixing filters. Intuitively, a better way to initialize would be to start the filters out at a state where they more closely resemble the ultimate solution that the BSSD algorithm is trying to find. We could get a better initial estimation for these filters by finding the time delays between the sound sources [18]. Then, in a manner similar to beamforming, we could incorporate these delays into the unmixing filters so that the desired sources would be time-aligned [19].

In controlled laboratory experiments, BSSD algorithms perform very well. In the real world, however, they flounder. If we focus our research more on the practical acoustic engineering necessary to obtain signals with less reverberation and noise, then our current algorithms will thrive.

References

[1] Cherry, E. Collin. (1953) Some Experiments on the Recognition of Speech, with One and with Two Ears, *Journal of the Acoustical Society of America*, ***25***, pp. 975-979.

[2] Hérault, Jeanny and Christian Jutten. (1986) Space or time adaptive signal processing by neural network models, *AIP conference proceedings*, ***151***, pp. 206-211.

[3] Orfanidis, Sophocles, J. (1996) *Introduction to Signal Processing*, Upper Saddle River, New Jersey: Prentice Hall.

[4] Haykin, Simon. (1994) *Blind Deconvolution*, Englewood Cliffs, New Jersey: PTR Prentice Hall.

[5] Torkkola, Kari. (1996) Blind Separation of Convolved Sources Based on Information Maximization, *Proceedings of the 1996 IEEE Workshop on Neural Networks for Signal Processing*, pp. 423-432.

[6] Lambert, Russell H. (1996) *Multichannel Blind Deconvolution: FIR Matrix Algebra and Separation of Multipath Mixtures*, Ph.D. Thesis, University of Southern California.

[7] Ikeda, Shiro and Noboru Murata. (1998) A method of blind separation based on temporal structure of signals, to appear in *Proceedings of the International Conference on Neural Information Processing*, October 1998.

[8] Bell, Anthony J. and Terrence J. Sejnowski. (1995a) An information-maximization approach to blind separation and blind deconvolution, *Neural Computation*, ***7***, No. 6, pp. 1129-1159.

[9] Lee, Te-Won, Anthony J. Bell and Russell H. Lambert. (1996) Blind separation of delayed and convolved sources, *Advances in Neural Information Processing Systems*, ***9***, pp. 758-764.

[10] Smaragdis, Paris. (1998) Blind Separation of Convolved Mixtures in the Frequency Domain, to appear in *Proceedings of the International Workshop on Independence & Artificial Neural Networks*.

[11] Westner, Alex. (1999) *Object-Based Audio Capture: Separating Acoustically-Mixed Sounds*, M.S. Thesis, Massachusetts Institute of Technology.

[12] Bell, Anthony J. and Terrence J. Sejnowski. (1995b) Fast blind separation based on information theory, *Proceedings of the International Symposium on Nonlinear Theory and Applications (NOLTA)*, ***1***, pp. 43-47.

[13] Smaragdis, Paris. (1997) *Information Theoretic Approaches to Source Separation*, M.S. Thesis, Massachusetts Institute of Technology.

[14] Westner, Alex and V. Michael Bove, Jr. (1999) Blind separation of real-world audio signals using overdetermined mixtures. *Proceedings of ICA'99*, Jan. 11-15, Aussois, France.

[15] Ward, W., G. Elko, R. Kubli, and W. McDougal. (1994) The new varechoic chamber at AT&T Bell Labs, *Proceedings of the Wallance Clement Sabine Centennial Symposium*, pp. 343-346.

[16] Renomeron, Richard J. (1997) *Spatially Selective Sound Capture for Teleconferencing Systems*, M.S. Thesis, Rutgers University.

[17] Dunn, H. K. and D. W. Farnsworth. (1939) Exploration of Pressure Field Around the Human Head During Speech, *Journal of the Acoustical Society of America, **10***, pp. 184-199.

[18] Jian, Ming, Alex C. Kot, and Meng H. Er. (1998) Performance Study of Time Delay Estimation in a Room Environment, *IEEE International Symposium on Circuits and Systems, **5***, pp. 554-557.

[19] Rabinkin, Daniel V., Richard J. Renomeron, Arthur Dahl, Joseph C. French, James L. Flanagan and Michael H. Bianchi. (1996) A DSP Implementation of Source Location Using Microphone Arrays, *Proceedings of the SPIE, **2846***, pp. 88-99.

This work was funded by the Digital Life Consortium at the MIT Media Lab.

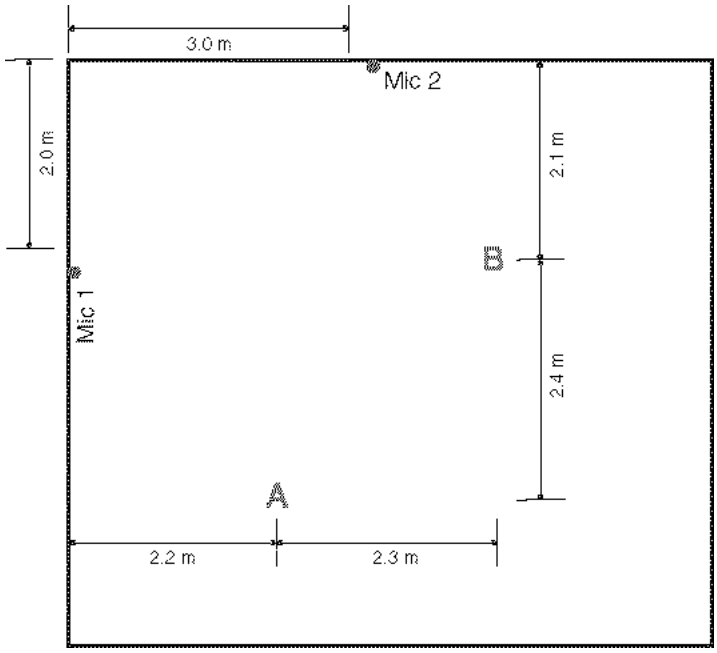

Figure 1. Physical layout of the microphones and loudspeaker locations that we used from the Varechoic Chamber data. Loudspeaker A played a male speech sample and loudspeaker B played a female speech sample.

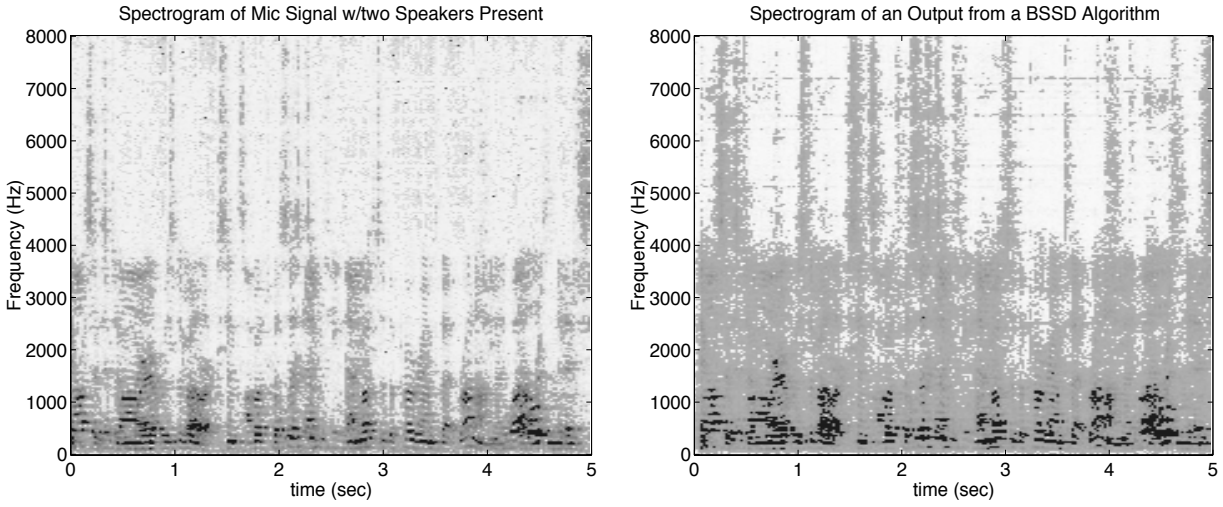

Figure 2. The spectrogram on the left is of a microphone signal taken from a recording of 2 males speaking in a conference room. This signal was one of the two inputs into a BSSD algorithm. The spectrogram on the right is the corresponding output from output of the algorithm.

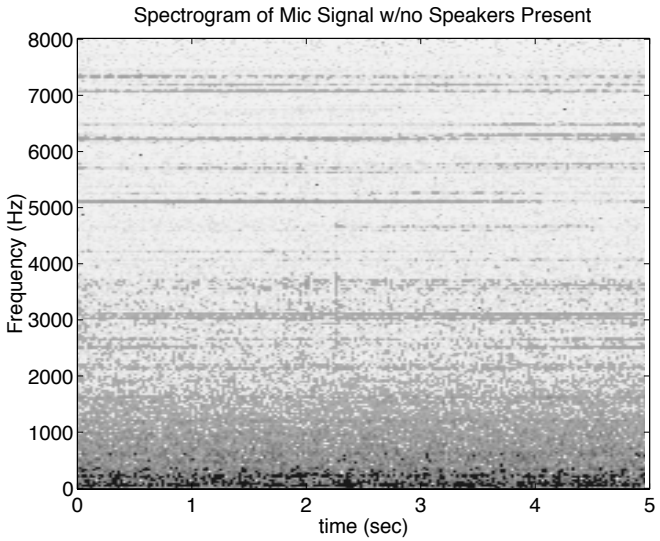

Figure 3. Spectrogram of the noise in the conference room.