UNIVERSITY OF MIAMI

AUDITORY COMPONENT ANALYSIS
USING PERCEPTUAL PATTERN RECOGNITION TO IDENTIFY AND EXTRACT INDEPENDENT COMPONENTS FROM AN AUDITORY SCENE

By

Jonathan Boley

A RESEARCH PROJECT

Submitted to the Faculty
of the University of Miami
in partial fulfillment of the requirements for
the degree of Master of Science in Music Engineering Technology

Coral Gables, FL

May, 2005

UNIVERSITY OF MIAMI

A thesis submitted in partial fulfillment of
the requirements for the degree of
Master of Science in Music Engineering Technology

AUDITORY COMPONENT ANALYSIS
USING PERCEPTUAL PATTERN RECOGNITION TO IDENTIFY AND EXTRACT INDEPENDENT COMPONENTS FROM AN AUDITORY SCENE

Jonathan Boley

Approved:

Ken C. Pohlmann Professor of Music Engineering	Dr. Edward P. Asmus Associate Dean of Graduate Studies
Colby N. Leider Assistant Professor of Music Engineering	Dr. Christian A. Duncan Assistant Professor of Computer Science

BOLEY, JONATHAN (M.S. in Music Engineering Technology)

(May 2005)

Auditory Component Analysis: Using Perceptual Pattern Recognition to Identify and Extract Independent Components from an Auditory Scene

Abstract of a Master's Research Project at the University of Miami

Research project supervised by Professor Ken Pohlmann

Number of pages in text: 71

The *cocktail party effect*, our ability to separate a sound source from a multitude of other sources, has been researched in detail over the past few decades, and many investigators have tried to model this on computers. Two of the major research areas currently being evaluated for the so-called sound source separation problem are Auditory Scene Analysis (Bregman 1990) and a class of statistical analysis techniques known as Independent Component Analysis (Hyvärinen 2001). This paper presents a methodology for combining these two techniques. It suggests a framework that first separates sounds by analyzing the incoming audio for patterns and synthesizing or filtering them accordingly, measures features of the resulting tracks, and finally separates sounds statistically by matching feature sets and making the output streams statistically independent. Artificial and acoustical mixes of sounds are used to evaluate the signal-to-noise ratio where the signal is the desired source and the noise is comprised of all other sources. The proposed system is found to successfully separate audio streams. The amount of separation is inversely proportional to the amount of reverberation present.

ACKNOWLEDGEMENTS

This work would not be what it is without the encouragement from so many of my friends and family members. First, to my parents, without whom I wouldn't be here. To my dad for instilling in me a passion for math and science at a young age, and to my mom for her unfailing encouragement and support. To my friends- to Vib who has taught me so much and who has always been there to bounce ideas off of. Send. To Vishu, whose hard work and passion for his work have motivated me to keep at it.

To my professors, Ken Pohlmann and Colby Leider, for their guidance on this journey. A special thanks to Dan Ellis, who provided some thoughtful discussions and his collection of Matlab source code for generating the wefts.

Table of Contents

Index of Figures

Index of Tables

Introduction

The cocktail party effect was first described by Cherry (1953) as our ability to listen to a single person in an environment where multiple people are speaking at once. This talent is related to localization, binaural masking level differences, auditory scene analysis, and perhaps several other psychoacoustic phenomena.

Our ability to focus on a single sound source is due, in part, to our ability to localize sounds and to recognize patterns. Our ability to localize sounds stems from the fact that we have two ears, and the brain processes these sounds in such a way that sounds emitted from a particular location actually seem to be separated from other sounds within the brain. The goal of the field of auditory scene analysis is to understand how the human auditory system analyzes our environment, and much of the research in this field directly relates to the problem of sound source separation. Independent component analysis is another active area of research and is being utilized for its ability to separate sounds such that they become statistically independent.

Traditionally, the fields of computational auditory scene analysis (CASA) and independent component analysis (ICA) have been used separately to approach the problem of sound source segregation. These approaches appear to work well under very different conditions, thus illustrating the need for a combined system.

Perhaps the most profitable application for sound separation is automatic speech recognition in crowded environments. However, sound separation could also be used to improve spatialization of multichannel audio, to separate musical instruments, to aid in forensic audio investigations, to improve assisted hearing devices, and many other applications.

Objectives

The objective of this paper is to evaluate the benefits of combining the grouping heuristics of auditory scene analysis with the statistics of independent component analysis. A stereo pair of channels will be evaluated for individual streams according to various rules of auditory scene analysis. Feature vectors will be calculated for each stream, and these features will be combined to form a Gaussian mixture model (GMM). These original streams will also be processed with an independent component analysis (ICA) algorithm that separates the sound described by the GMM such that the stream and the residual are statistically independent. This combined algorithm will be evaluated with subjective listening tests to determine the perceived quality of the output streams.

Structure

Chapter one discusses several of the conceptual rules for grouping of auditory events and presents some algorithms for the identification and separation of these events. Chapter Two presents the mathematical theory behind ICA and an efficient algorithm for separation of a specific audio signal. Chapter three presents a method for calculating feature vectors based on auditory scene analysis and a method for combining the ASA and ICA algorithms. Chapter Four discusses the experimental methodology for evaluating the quality of the separated signals, and Chapter Five presents the results of the subjective experiments. Chapter Six draws conclusions and suggests future work.

1. Auditory Scene Analysis

Auditory Scene Analysis is the process by which we are able to make sense of the complex auditory environment that we experience every day. At any given moment, our ears may pick up sound from dozens of individual sources. These sounds are mixed acoustically and arrive at our ears in a series of sound waves that are drastically different from the sound waves originating from any of the individual sources. Due to processing within the brain, we are able to focus our attention on the portions originating from an individual source. Two of the principal low-level processes that account for much of this source discrimination are localization and stream segregation.

1.1 Localization

People are able to localize sounds along the horizontal plane by evaluating the intensity or time difference between the signals arriving at each ear. At high frequencies, there will be a significant difference in intensity if the sound source is coming from the side of the head. At low frequencies, the intensity difference may be negligible, so the brain evaluates the time difference between the signals at each ear. By measuring both the interaural level difference (ILD) and the interaural time difference (ITD), we are able to localize sounds fairly accurately (less than 1° in some situations). Figure 1 shows how ILD and ITD vary with angle.

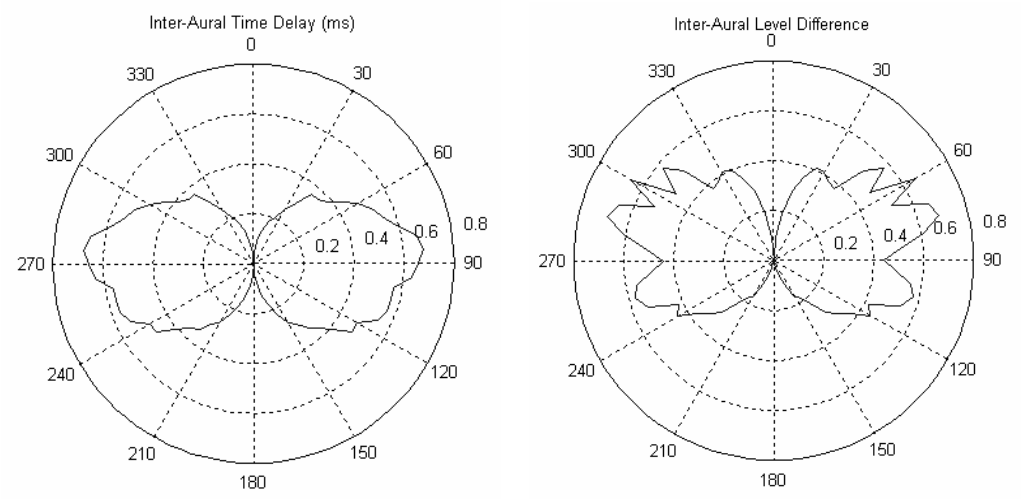

Figure 1. ITD and ILD values corresponding to various angles.*

Physiologically, binaural processing begins in the superior olivary complex (Johnson 1997). Here, the brain calculates the interaural level differences (ILD) and interaural time differences (ITD). The brain appears to be wired such that sounds are separated according to left/right origin rather than at which ear the sound arrives. The inferior colliculus is thought to provide a spatial map of all sounds. This information is passed on to higher levels of brain processing, so it makes sense that spatial separation should be performed before attempting any higher level audio processing in a computer system.

Localization is the first step of auditory scene analysis and thus perhaps the greatest contribution to perceptual organization. In a natural environment, sounds do not normally jump great distances instantaneously but rather stay stationary or move relatively slowly. If we were to hear a sound suddenly jump a great distance from right to left and back to the right again, we would likely perceive a second source coming from the left.

1.2 Stream Segregation

Most of the research within the field of auditory scene analysis has been related to auditory stream segregation. Stream segregation is the result of applying several grouping rules to organize a conglomeration of sounds into several streams. These streams correspond to the individual sound sources.

1.2.1 Spectral Integration

Perhaps the simplest feature by which sounds are grouped is harmonicity. Natural sounds do not consist of a single tone, but rather a combination of tones that are often harmonically related. A musical instrument playing an A440 note will not only produce a 440 Hz sine wave, but also sine waves at integer multiples of the fundamental frequency. However, we are also able to hear two instruments playing the same note as two separate instruments. In fact, we use several other grouping rules for discriminating between sound sources.

One of these grouping rules is that of common fate. Harmonics that are modulated similarly in amplitude or frequency tend to be grouped together. For instance, if a series of harmonics maintains constant frequency ratios with respect to each other, and then suddenly only two of the harmonics begin oscillating in frequency, the listener will hear a single sound split into two– a steady sound and an oscillating sound. This separation will also occur if multiple sounds share harmonic frequencies but begin oscillating (in either amplitude or frequency) differently. In this case, each sound source is identified according to the phase of the modulation. Figure 2 shows an example of a stream splitting into two then fusing back into one, based on common frequency modulation.

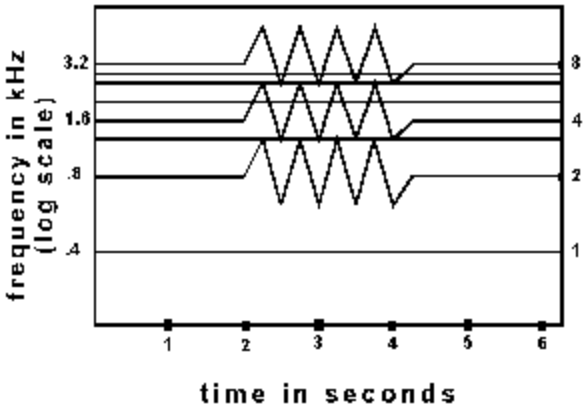

Figure 2. Common frequency modulation is a strong indication of a common source.[†]

Another phenomenon, known as *comodulation masking release*, exemplifies the power of amplitude modulation. If a pure-tone target is played in the presence of a narrow band of noise centered on the target frequency, the pure tone may be masked. However, if the noise masker is amplitude modulated, the pure tone will be detected. The target tone may be made even more audible by placing a second narrow band of noise just outside the critical band of the first. If the two bands of noise are comodulated (amplitude modulated in phase with each other), the target tone is more easily heard and is identified as a separate stream.

Perhaps one of the greatest cues to identifying a sound lies in the onset of that sound. For example, a piano note is more difficult to identify if the characteristic attack is removed. When multiple harmonics are played together, they may be grouped together according to the rules described above, but the time and rate of onset also play a large role in source discrimination. If four harmonically related tones have similar temporal envelopes, as shown in Figure 3, and are played together, they may be grouped together perceptually. Even if some tones start later than others, they may be grouped together if

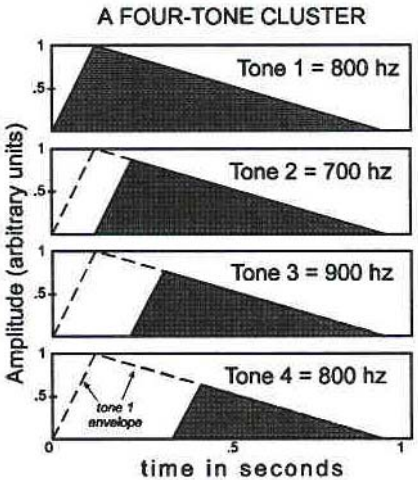

Figure 3. A four-tone cluster may be grouped together perceptually, depending on the rate of onset. A slow rate of onset will allow perceptual fusing, but a rapid onset will make each component sound distinct.[†]

they slowly become audible. But as the rate of onset becomes faster (more percussive), the tones begin to separate into individual streams.

1.2.2 Sequential Integration

Sequences of sound events are grouped into perceptual streams when these sounds can be grouped according to various criteria. For example, a series of tone pulses alternating between a high frequency and a low frequency will sound like a single stream when played slowly. However, if this sequence is played fast enough, it will separate into two streams: one of low-frequency tones and another of high-frequency tones, as shown in Figure 4.

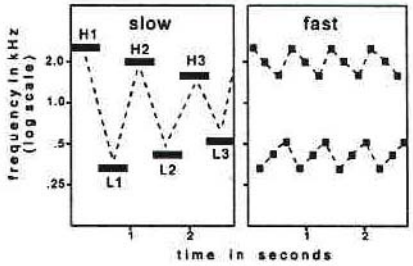

Figure 4. An example of two possible streaming patterns. Here, the streams are determined by proximity of the tones in both time and frequency.[†]

It is not only the temporal spacing that determines the perceptual streams but also the frequency separation. The relative spacing between sounds in the frequency domain determines the grouping of these sounds. In the above example, if the high-frequency tones are only slightly higher in pitch than the low-frequency tones, the separation would be more difficult. But if the relative spectral spacing is great, separation is much easier.

Another very important auditory cue for source discrimination is the timbre of a sound. Timbre is difficult to define, but a key aspect of timbre is the position of the spectral peaks. Two sounds containing the exact same frequencies can be separated into two streams if the spectral envelopes differ. The positions of the peaks of these envelopes determine the brightness of the sound and can be used to distinguish between the sounds.

1.2.3 Old Plus New Heuristic

Spectral and sequential integration were introduced above, but it is often the competition of these two types of grouping that produces some of the most interesting effects of auditory scene analysis. If a pure tone is followed by a complex tone, the perceived streams depend on the frequency proximity of the pure tone to a component of the complex one. If the frequency of the pure tone is close enough to a frequency component of the complex tone, that component may be perceptually removed from the complex tone and grouped with the pure tone. However, if the frequency of the pure tone is not very close to any part of the complex tone, a component of the complex tone may still be grouped with the pure tone if that component begins slightly before the other components. Here, we see that the grouping rules of spectral integration may indicate

one possible set of streams, while sequential integration may indicate an entirely different set. By using the stronger grouping, the brain attempts to identify the streams arriving from individual sources. Figure 5 shows an example of this effect with bands of noise.

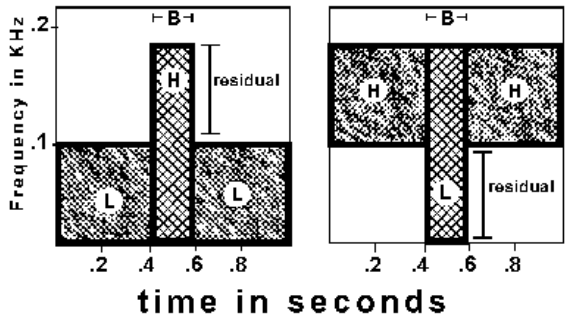

Figure 5. The apparent continuity of the initial sound results in a residual sound composed of only those frequencies not in the initial sound.[†]

Another grouping phenomenon is that of apparent continuity in the presence of a masking noise. An interesting trick that we can play is to insert a short period of silence into the middle of a pure tone. We will quickly identify this break unless the silence is replaced by a loud noise masker. Because this noise includes components that stimulate the brain in a way similar to the stimulation caused by the pure tone, the tone is perceived as being continuous though the noise burst.

Similarly, if a complex tone is followed by another complex tone containing only the low-frequency components of the first, the low frequencies may all be grouped into a single stream. If the temporal gap between the two complex tones is long, each will be perceived as separate, but as the two are placed closer in time, the likelihood that the low frequencies will be grouped together becomes greater. If the gap is reduced to zero, the low frequencies become a single stream, and the high frequencies are left as a residual stream.

1.3 Computational Models

Although people are relatively adept at focusing their attention on a single sound source in a complex mixture, machines have a very difficult time doing this. Engineers in the field that has become known as *computational auditory scene analysis* (CASA) have been working on these problems for approximately two decades. Work actually began in the field of computer vision with David Marr's "computational theory of vision" (1982), which attempted to explain vision in terms of perceptual objects. Psychologist Albert Bregman performed research into the heuristics and schemata that we use to separate sounds in our minds. Researchers in speech processing and recognition soon realized the importance of ASA and began work to incorporate the principles presented by Bregman into their algorithms. Unfortunately, merely analyzing a spectrogram is not usually enough. Natural sounds often overlap in both frequency and time, making it even more difficult for a computer to identify the individual sounds. Dozens of methods have been proposed that attempt to model various aspects of auditory scene analysis.

While working on his PhD dissertation, Dan Ellis developed the concept of the *weft*. According to Ellis, "'Weft' is the Anglo-Saxon word for parallel fibers in woven cloth, giving the idea of a connected set of threads" (Ellis 1997). In contrast to previous techniques, wefts allow tracking of sounds that share frequency bands. The idea is based on the 3D correlogram, which maps frequency vs. autocorrelation lag and time.

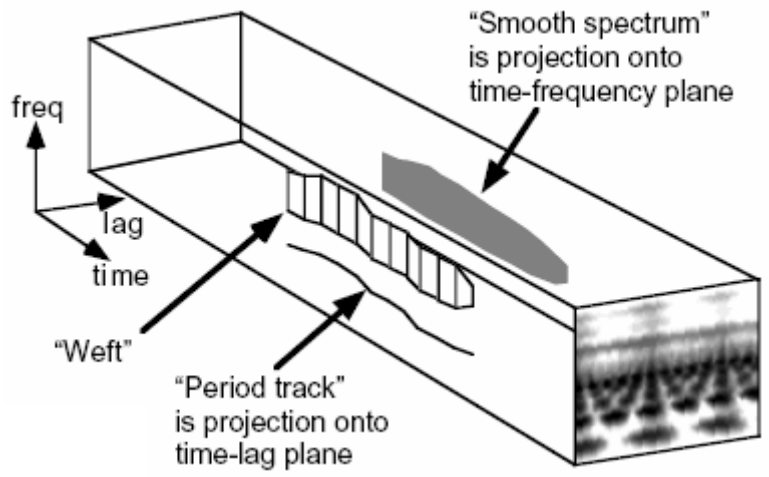

Figure 6. A weft can be found within the 3D correlogram.[‡]

The correlogram is generated by first passing the sound through a cochlear filterbank, half-wave rectifying the resulting signals, and smoothing over a 1ms window. For each subband, the short-time autocorrelation is calculated such that the samples are spaced logarithmically to approximate human pitch perception. The resulting three-dimensional correlogram then displays the autocorrelation of each subband, with various lag periods, at a number of time frames, as shown in Figure 6.

The periodogram is than computed from the autocorrelation by normalizing the autocorrelation values (relative to zero-lag) and summing across subbands, resulting in a two-dimensional representation of autocorrelation lag vs. time. A threshold may then be set to identify strong modulation periods and thus possible weft elements. A search algorithm starts at the shortest period and looks for peaks. Each time it finds a peak, any peaks occurring at integer multiples of that period are subtracted to get rid of the subharmonic aliases created by the autocorrelation process. A hysteresis may be used to

allow an existing period track to continue existing for a short period even if the peaks are slightly below the threshold.

The spectra for each of the period tracks are then extracted from the 3D correlogram at points corresponding to the period in question. Subharmonic aliases are also then subtracted from the correlogram so that the remaining value of the peak is roughly proportional to the energy of the modulation (Ellis 1997).

The presence of noise can negatively effect the accuracy of the weft representation if appropriate steps are not taken. It can be shown (Ellis 1997) that, if the noise is assumed to be periodic impulses, noise can be detected by first calculating the ratio for the noiseless case:

$$d = A / P \tag{1}$$

where A is the average autocorrelation and P is the peak value of the autocorrelation. In practice, a table of d values is calculated for all frequency channels and autocorrelation lags. If the measured ratio is greater than the calculated value in the table, it is assumed that noise is present and may be factored out using the following equations:

$$A = (d \cdot M + (1-d) \cdot N)^2 \tag{2}$$

$$P = d \cdot M^2 + (1-d) \cdot N^2 \tag{3}$$

where N is the level of the noise floor and M is the maximum level of idealized noise.

The weft representation of sounds is very useful for identifying sound sources, and resynthesis of these signals will be discussed in Chapter Three.

---

* HRTF data courtesy of (Gardner 1994)
† Images reproduced, with permission, from (Bregman and Ahad 1996)
‡ Image courtesy of Dan Ellis (1995)

2. Blind Source Separation

2.1 Basic Theory

Blind source separation (BSS) is the technique used to separate independent signals from a set of mixed signals without any prior knowledge of the signals. The source signals $\mathbf{s}(t)=[s_1(t),s_2(t),\ldots,s_m(t)]^T$ are to be estimated from the observed signals $\mathbf{x}(t)=[x_1(t),x_2(t),\ldots,x_m(t)]^T$. The system is modeled as

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \tag{4}$$

where the mixing matrix, **A**, is unknown.

It is assumed that the original source signals are independent and the mixing matrix is nonsingular; thus it is possible to compute a demixing matrix **W** as follows:

$$\mathbf{y}=\mathbf{W}\mathbf{x} \tag{5}$$

where **y** is the output signal vector. The signals represented by **y** are similar to **s** but they may be scaled and the order may be different.

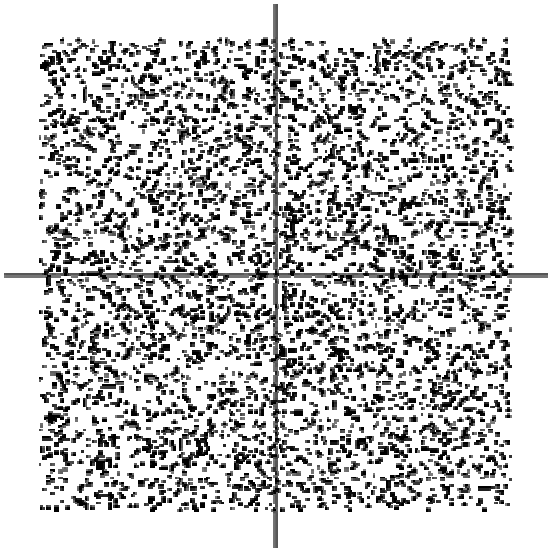

Figure 7. Joint density of two sources with uniform distribution*

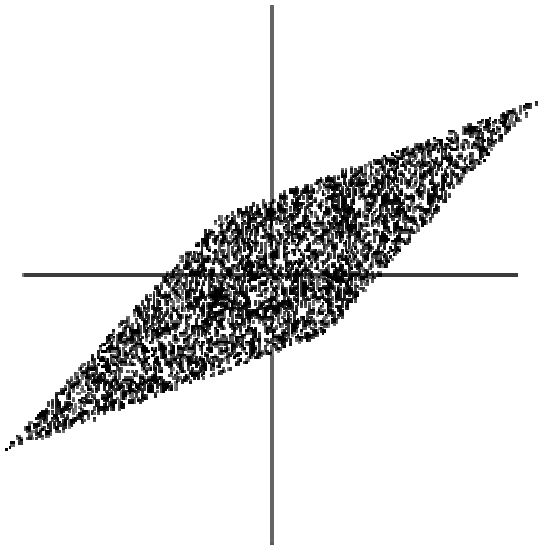

Figure 8. Joint density of two mixtures of the uniformly distributed sources*

Figure 7 shows the joint density of two sources with uniform distribution, and Figure 8 shows the joint density of the mixtures of these two sources. Notice that the square has become rotated and skewed. The goal of BSS is to return this joint density back to its original shape.

Separation of component signals may be easily performed using blind source separation if the number of observed signals is equal to the number of original source signals. (Notice that this implies that the mixing matrix, **A**, is square.) Blind source separation has been successfully used in speech recognition, communications systems, and medical applications.

2.1.1 Principal Component Analysis

Principal component analysis (PCA), also known as the Karhunen-Loève transform or the Hotelling transform, uses the idea of correlation to exploit the redundancy in the various input signals and thus determine the input signals. The first step of PCA is to center the data by subtracting the mean:

$$\mathbf{x} \leftarrow \mathbf{x} - \mathrm{E}\{\mathbf{x}\} \tag{6}$$

The vector **x** is then linearly transformed such that the redundancy is removed. In other words, a transform is done which rotates the coordinate system such that the elements of **x** become uncorrelated (Hyvärinen 2001). Two variables, *a* and *b*, are said to be uncorrelated if their covariance is equal to zero:

$$c_{a,b} = \mathrm{E}\{(a - m_a)\,(b - m_b)\} = 0 \tag{7}$$

or equivalently

$$\mathrm{E}\{ab\} = \mathrm{E}\{a\}\mathrm{E}\{b\} = m_a m_b \tag{8}$$

There are a few ways to do perform this transform, but perhaps the simplest is the covariance method (Wikipedia 2005). PCA calculates the matrix $\mathbf{W^T}$ by first establishing that $\mathbf{W^T}$ is an $n \times n$ orthonormal projection matrix (i.e., it represents an orthogonal coordinate system):

$$\mathbf{y} = \mathbf{W^T x} \tag{9}$$

PCA also requires that cov(**y**) is a diagonal matrix and that $\mathbf{W^{-1}} = \mathbf{W^T}$.

Solving for cov(**y**):

$$\begin{aligned} \text{cov}(\mathbf{y}) &= E[\mathbf{y}\,\mathbf{y^T}] \\ &= E[(\mathbf{W^T x})(\mathbf{W^T x})^T] \\ &= E[(\mathbf{W^T x})(\mathbf{W x^T})] \\ &= \mathbf{W^T}\, E[\mathbf{x}\,\mathbf{x^T}]\, \mathbf{W} \\ \text{cov}(\mathbf{y}) &= \mathbf{W^T}\, \text{cov}(\mathbf{x})\, \mathbf{W} \end{aligned} \tag{10}$$

We then get:

$$\begin{aligned} \mathbf{W}\, \text{cov}(\mathbf{y}) &= \mathbf{W}\, \mathbf{W^T}\, \text{cov}(\mathbf{x})\, \mathbf{W} \\ \mathbf{W}\, \text{cov}(\mathbf{y}) &= \text{cov}(\mathbf{x})\, \mathbf{W} \end{aligned} \tag{11}$$

W can then be rewritten as a combination of n $n \times 1$ column vectors:

$$\mathbf{W} = [\mathbf{W_1}, \mathbf{W_2}, \ldots, \mathbf{W_n}]$$

And cov(**y**) can be written as:

$$\text{cov}(\mathbf{y}) = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix}$$

Substituting into equation (11), we get:

$$[\lambda_1 \mathbf{W_1}, \lambda_2 \mathbf{W_2}, \ldots, \lambda_n \mathbf{W_n}] = [\mathrm{cov}(\mathbf{x})\mathbf{W_1}, \mathrm{cov}(\mathbf{x})\mathbf{W_2}, \ldots, \mathrm{cov}(\mathbf{x})\mathbf{W_n}] \quad (12)$$

Thus, for i=1...n

$$\lambda_i \mathbf{W_i} = \mathrm{cov}(\mathbf{x})\mathbf{W_i} \quad (13)$$

which implies that $\mathbf{W_i}$ is an eigenvector of cov($\mathbf{x}$). Therefore, by finding the eigenvectors of cov($\mathbf{x}$), we can find the projection matrix $\mathbf{W}$.

There are many other methods of performing PCA including variance maximization, minimum mean-square error compression, and several adaptive algorithms, such as the PAST algorithm (Hyvärinen 2001).

2.1.2 Whitening

One useful preprocessing step for BSS is known as whitening. A vector is said to be white if all its elements are uncorrelated and have unit variances. For instance, a vector of white noise, $\mathbf{z}$, is in fact white because

$$E\{\mathbf{z}\,\mathbf{z}^T\} = \mathbf{I}$$

where $\mathbf{I}$ is the identity matrix.

Another word for white is *sphered*. If the density of $\mathbf{z}$ is radially symmetric and suitably scaled (for instance, white Gaussian noise), it is sphered (Hyvärinen 2001, p.140). See Figure 9 for an example. However, a sphered vector is not necessarily radially symmetric. If the density of the white noise is uniform rather than Gaussian, the two-dimensional plot of the density is a rotated square., as shown in Figure 10.

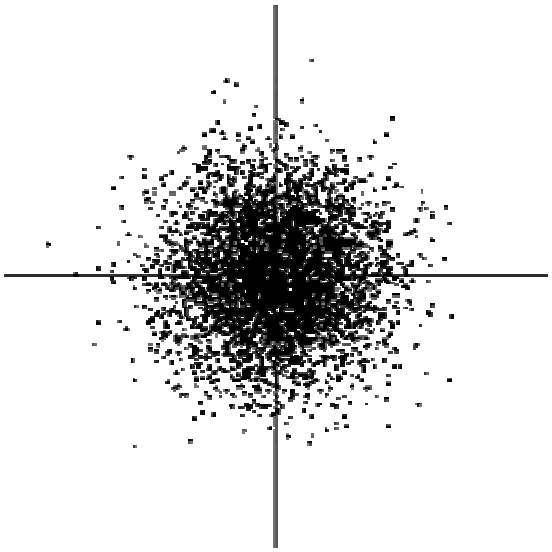

Figure 9. A sphered Gaussian distribution*

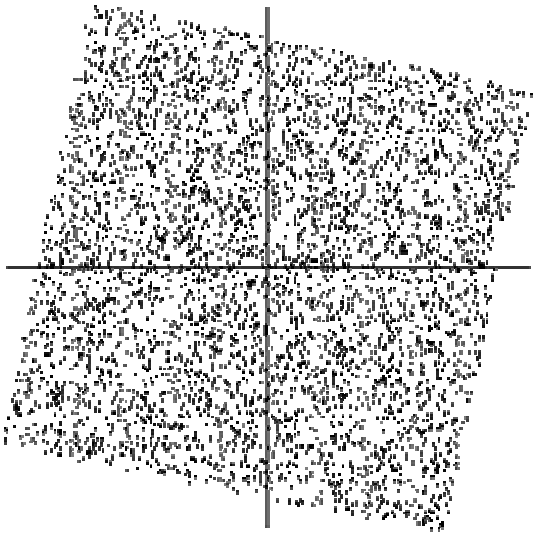

Figure 10. A sphered uniform distribution*

The process of whitening is simply to decorrelate the input vectors and scale them such that the variance is unity, which is very similar to what PCA does. Therefore, whitening can be accomplished using PCA by solving for **z** such that

$$\mathbf{z} = \mathbf{V}\,\mathbf{x}$$

is sphered.

Let the columns of the matrix **E** be the eigenvectors of the covariance matrix $\mathbf{C_x} = E\{\mathbf{x}\mathbf{x}^T\}$ and let **D** be the diagonal matrix of the eigenvalues of **C**. We can then solve for **V**:

$$\mathbf{V} = \mathbf{D}^{-1/2}\,\mathbf{E}^T \qquad (14)$$

2.2 Independent Component Analysis

Independent component analysis (ICA) is a method of blind source separation that is gaining tremendous popularity. Originally developed to tackle the cocktail party effect,

ICA has found great use in brain imaging, telecommunications, financial analysis and various other fields.

The ICA model relies on several basic assumptions and restrictions that make signal separation possible (Hyvärinen 2001, p.152). The first assumption is that the independent components are statistically independent. This assumption is the underlying basis for the operation of ICA. More details on the technical requirements for statistical independence will be outlined in the next section.

Another constraint is that the independent components must be non-Gaussian. ICA utilizes relatively high-order information for its calculations, and Gaussian distributions do not have such high-order cumulants.

The number of observed mixtures must be equal to the number of independent components. In other words, the mixing matrix is square. Assuming the mixing matrix is also invertible, it is possible to determine the independent components from the observed mixes and the inverse of the estimated mixing matrix.

It is not possible to determine the energies of the independent components. Because both the independent components and the mixing matrix are unknown, it is also unknown whether scalar multipliers should be placed in the mixing matrix or applied to the resultant output signals.

The order of the independent components is unknown. Again, because both **s** and **A** in equation (4) are unknown, the terms of each may be rearranged to obtain the same result.

2.2.1 Definition of Independence

Two random variables are considered independent if they do not affect one another in any way. In other words, knowing the outcome of one process does not provide any information about the outcome of the other. Mathematically, independence is defined in terms of the probability density functions:

$$p_{x,y}(x,y) = p_x(x)\, p_y(y) \tag{15}$$

where $p_{x,y}$ is the joint probability of x and y, p_x is the probability of x, and p_y is the probability of y.

Thus, it can also be stated that

$$E\{g(x)h(y)\} = E\{g(x)\}E\{h(y)\}$$

where g(x) and h(y) are absolutely integrable. Notice that this is very similar to the requirements for two signals to be uncorrelated. However, it should be noticed that the equation defining uncorrelatedness is a special case of the independence equation where g(x) and h(y) are linear functions and it only takes into account second-order statistics such as correlation or covariance. (Note that independence is equivalent to uncorrelatedness if the signals are Gaussian.)

2.2.2 Nongaussianity Measures

When whitening was discussed in section 2.1.2, it was noted that the density function for white Gaussian noise is radially symmetric but the density function for a uniform distribution was represented by a square. Similarly, it can be shown that any distribution other than Gaussian will result in a density function that is not radially symmetric. It was also pointed out in section 2.1.1that PCA essentially rotates the coordinate system to

obtain whitening. We can see that rotating a radially symmetric density function will have no effect. Therefore, it is obvious that Gaussian signals should be forbidden. In fact, we would like to minimize Gaussianity altogether.

To minimize Gaussianity, or rather to maximize non-Gaussianity, we need to establish measures of how Gaussian or non-Gaussian a signal is. Two measures have proven to be quite effective: kurtosis and negentropy.

Kurtosis is the fourth-order cumulant of a random variable and is defined by the equation

$$\text{kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2 \tag{16}$$

This is simply a normalized measure of the fourth moment. It works well as a measure of non-Gaussianity because the fourth moment of a Gaussian distribution is equal to $3(E\{y^2\})^2$, which produces a kurtosis value of zero for a Gaussian signal. If the variance of y has been normalized to one ($E\{y^2\}=1$), the kurtosis equation simplifies to

$$\text{kurt}(y) = E\{y^4\} - 3 \tag{17}$$

The kurtosis values can be either positive (super-Gaussian) or negative (sub-Gaussian), but for the purposes of measuring non-Gaussianity, the absolute value or square of the kurtosis value is usually considered.

If kurtosis must be calculated based on a measured sample, problems may sometimes arise. Kurtosis is, in fact, quite sensitive to outliers so this measure can sometimes be erroneous. Negentropy, however, does not have this problem.

The *entropy* of a vector **y** is defined as

$$H(\mathbf{y}) = -\int p(\mathbf{y}) \log p(\mathbf{y})\, d\mathbf{y} \tag{18}$$

where $p(y)$ is the probability density function of y.

Because a Gaussian distribution, by definition, has the greatest entropy for a given variance, the negentropy J can be defined as

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}) \tag{19}$$

where $\mathbf{y}_{gauss}$ is a Gaussian random vector with the same correlation matrix as $\mathbf{y}$.

This calculation is much more robust than kurtosis, but it is also much more complex. To curtail the complexity of the negentropy calculations, several approximations have been introduced. One of the best approaches is to use expectations of higher-order functions. By replacing the polynomials y^3 and y^4 with other functions G_i, we can approximate the negentropy with the following equation:

$$J(\mathbf{y}) \approx k_1(E\{G_1(\mathbf{y})\})^2 + k_2(E\{G_2(\mathbf{y})\} - E\{G_2(v)\})^2 \tag{20}$$

where k_1 and k_2 are positive constants and v is a Gaussian variable with zero mean and unit variance. By carefully choosing the functions G_1 and G_2, robust calculations can be obtained. The following functions have been shown to work well (Hyvärinen 2000):

$$\begin{aligned} G_1(y) &= \frac{1}{a_1}\log\cosh a_1 y \\ G_2(y) &= -\exp\left(\frac{-y^2}{2}\right) \end{aligned} \tag{21}$$

Where $1 \le a_1 \le 2$ is a constant, usually chosen to be one.

2.2.3 The ICA Algorithm

The first principle of ICA estimation is that of nonlinear decorrelation. Not only should the components be uncorrelated, but the transformed components should also be uncorrelated. The second principle is that of maximum non-Gaussianity. By keeping the variance of y constant, and finding the local maxima of the non-Gaussianity of $y = \sum_i b_i x_i$, the independent component may be found. Each local maximum of the non-Gaussianity corresponds to one independent component.

There are many different ways to perform ICA, each focusing on a different aspect of the ideas presented above. Historically, some of the earliest algorithms used nonlinear decorrelation and nonlinear PCA, but these have become fairly obsolete, giving way to newer, more efficient algorithms. Some other implementations have used tensor mathematics, maximization of non-Gaussianity, minimization of mutual information, and maximum likelihood estimation. Tensor mathematics is a difficult subject (this is how Einstein mathematically formulated his theory of general relativity), so we will leave the task of explaining the relationship to ICA for a braver soul. See (Cardoso 1996) for details. Maximization of non-Gaussianity was briefly explained in the paragraph above.

Minimization of mutual information uses the concept of entropy to define the mutual information I between y_i, $i = 1 \ldots m$ as

$$I(y_1, y_2, \ldots, y_m) = (\sum H(y_i)) - H(\mathbf{y}) \qquad (22)$$

This measure is zero if and only if the variables are statistically independent and positive otherwise. A property of mutual information is that for $\mathbf{y} = \mathbf{W}\mathbf{x}$,

$$I(y_1, y_2, \ldots y_m) = \sum H(y_i) - H(\boldsymbol{x}) - \log |\det \mathbf{W}| \quad (23)$$

If y_i is uncorrelated and of unit variance,

$$\mathbf{I} = E\{\mathbf{y}\mathbf{y}^T\} = \mathbf{W}E\{\mathbf{x}\mathbf{x}^T\}\mathbf{W}^T$$

$$1 = \det \mathbf{I} = (\det\mathbf{W}\, E\{\mathbf{x}\mathbf{x}^T\}\mathbf{W}^T) = (\det\mathbf{W})(\det E\{\mathbf{x}\mathbf{x}^T\})(\det\mathbf{W}^T)$$

which implies that det$\mathbf{W}$ is a constant because $E\{\mathbf{x}\mathbf{x}^T\}$ does not depend on $\mathbf{W}$. It can also be noted that negentropy can be written as

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}) = C - H(\mathbf{y}) \quad (24)$$

where C is a constant. The mutual information can thus be calculated as follows:

$$I(y_1, y_2, \ldots y_m) = C - \sum J(y_i) \quad (25)$$

Notice that maximizing the sum of the negentropies is equivalent to minimizing mutual information.

Maximum likelihood estimation is a very popular method of approximating the ICA model. For this method, the density p_x of the mixture vector is formulated as

$$p_x(\mathbf{x}) = |\det\mathbf{W}|\, p_s(\mathbf{s}) = |\det\mathbf{W}| \prod p_i(\mathbf{s}_i)$$

where $\mathbf{W} = \mathbf{A}^{-1}$ and p_i denote the densities of the independent components. The likelihood (as a function of $\mathbf{W}$) is then defined as

$$L(\mathbf{W}) = \prod_{t=1}^{T} \prod_{i=1}^{n} p_i\left(\mathbf{w}_i^T \mathbf{x}(t)\right) |\det \mathbf{W}| \quad (26)$$

or

$$\log L(\mathbf{W}) = \sum_{t=1}^{T} \sum_{i=1}^{n} \log p_i\left(\mathbf{w}_i^T \mathbf{x}(t)\right) + T \log|\det \mathbf{W}| \quad (27)$$

Consider now the expectation of the log-likelihood:

$$\frac{1}{T}E\{L\} = \sum_{i=1}^{n} E\left\{\log p_i\left(\mathbf{w}_i^T \mathbf{x}(t)\right)\right\} + \log\left|\det \mathbf{W}\right| \qquad (28)$$

Now, if p_i is equal to the distribution of $\mathbf{w}_i^T\mathbf{x}$, the first term evaluates to $-\sum H(\mathbf{w}_i^T\mathbf{x})$. The likelihood is then the negative of the mutual information plus some constant.

Hyvärinen proposed an efficient algorithm for estimating the ICA model which he calls FastICA (Hyvärinen 1999). It is based on an iterative routine that finds a weight vector $\mathbf{w}$ that maximizes the negentropy $J(\mathbf{w}^T\mathbf{x})$. Recall that the variance of $\mathbf{w}^T\mathbf{x}$ is to be kept at unity, which is equivalent to requiring the norm of $\mathbf{w}$ to be unity. The basic steps for FastICA are then as follows:

1) Initialize the weight vector $\mathbf{w}$ (with random values);
2) Let $\mathbf{w}^+ = E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - E\{g'(\mathbf{w}^T\mathbf{x})\}\mathbf{w}$ …, where g is the derivative of G in (20);
3) Let $\mathbf{w} = \mathbf{w}^+ / \|\mathbf{w}^+\|$;
4) If not converged, return to step 2.

Figure 11 shows the flowchart for this algorithm. For the derivation of the equation in step 2, refer to (Hyvärinen 2000). This process is known as a one-unit learning algorithm.

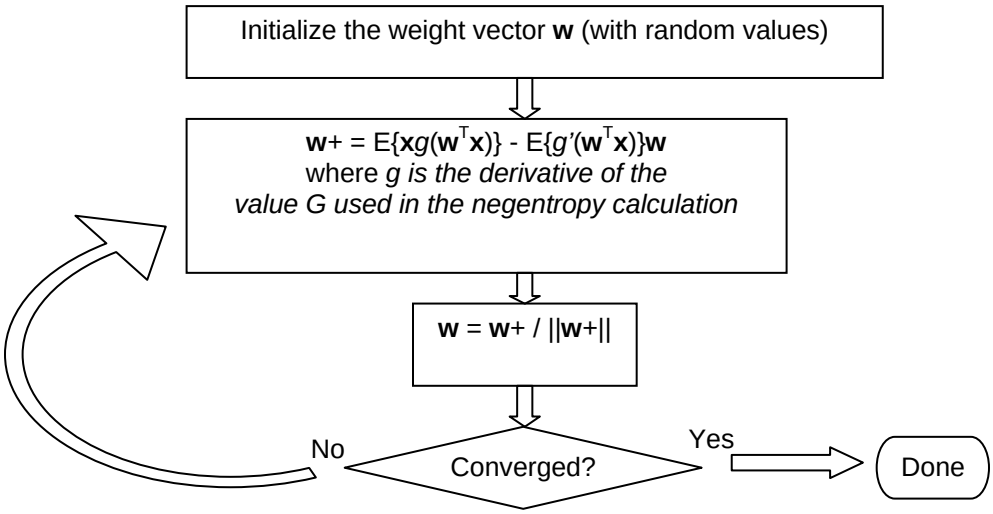

Figure 11. Fast ICA Algorithm

However, the above algorithm is only designed to estimate one of the independent components. In practice, several independent components are desired, but the algorithm must ensure that they do not converge to the same maxima. To compute a matrix **W** without giving preference to any particular independent component, the following algorithm may be added to the end of each iteration of the above algorithm (Hyvärinen 2000):

3.1) Let $\mathbf{W} = \mathbf{W} / \sqrt{\left\| \mathbf{W}\mathbf{W}^T \right\|}$

3.2) Let $\mathbf{W} = \frac{3}{2}\mathbf{W} - \frac{1}{2}\mathbf{W}\mathbf{W}^T\mathbf{W}$

3.3) Repeat step 2 until converged.

2.3 Intelligent ICA

Intelligent ICA was presented by Mitianoudis and Davies as a conference paper at the May 2002 Audio Engineering Society Convention. They applied ICA to instrument recognition (Mitianoudis 2002), extracting several feature vectors from solo recordings of instruments that they desired to recognize. The authors then created a Gaussian mixture model (GMM) for each instrument, and used this to model the timbre of that particular instrument. The Intelligent ICA algorithm then used the information in the GMM of the desired instrument to locate this instrument and separate it from the mix.

2.3.1 Source Identification

Source identification has been used in various applications from speaker verification to instrument recognition. The process is usually divided into two phases: the training phase and the recognition phase. The training phase consists of passing a significant amount of audio data through the system which in turn makes various measurements and stores information about that data set. In the recognition phase, a much smaller audio clip is analyzed and compared to the data obtained during the training phase. Based on the similarity to the training data, the system decides whether or not they are the same source.

Prior to analysis, some preprocessing is usually performed. This typically consists of removing any DC offset and applying a pre-emphasis filter. The pre-emphasis filter is often a simple high-pass first order filter such as

$$H(z) = 1 - 0.97z^{-1}$$

The audio is then split into several overlapping, windowed frames. From each frame several features are extracted and organized into a feature vector. The choice of features has a significant effect on the performance of the source identification system.

In his work on musical timbre, John Grey (1977) pointed out that the three greatest contributing factors to timbre are the spectral envelope, how the spectrum changes over time, and the attack characteristics of the sound. Since then, several algorithms have been designed to extract these features (and more) from a sound. The frequency envelope can be captured using linear prediction coefficients (LPC), mel-frequency cepstral coefficients (MFCC), or perceptual linear predictive coefficients (PLPC). The MFCC has proven to be an effective measure for various sound sources. To calculate this, the Fourier transform is first performed on the frame of audio data, the amplitudes are converted to a logarithmic scale, the frequencies are converted to the perceptual mel-frequency scale, and finally the discrete cosine transform is performed.

Another feature is the spectral centroid (SC), which generally corresponds to the perceived brightness. The power spectrum is calculated, converted to a logarithmic frequency scale, and the spectral peak is found. The normalized value of the SC is the frequency of the peak (in Hz) divided by the fundamental frequency. The max, mean, and standard deviation may all be used as features. (Eronen 2001)

Feature vectors can be combined into a model of the timbre of a particular sound. This can be done with a neural network classifier, a Hidden Markov Model, or a

Gaussian Mixture Model. Intelligent ICA prescribes the use of a Gaussian mixture model (GMM).

A GMM represents the feature vectors as a sum of Gaussian distributions, resulting in a composite probability density function. Given a feature vector $\boldsymbol{v}$, the probability given by a GMM is defined by the following equations:

$$P(\mathbf{v} \mid \lambda) = \sum_{i=1}^{M} p_i b_i(\mathbf{v}) \tag{29}$$

$$b_i(\mathbf{v}) = \frac{\exp(-0.5(\mathbf{v}-\mathbf{m}_i)^Y C_i^{-1}(\mathbf{v}-\mathbf{m}_i))}{\sqrt{(2\pi)^D |C_i|}} \tag{30}$$

where M is the number of Gaussians and p_i, $\boldsymbol{m_i}$, C_i are the weight, mean vector and covariance matrix of each Gaussian. The GMM may be described by the notation $\lambda=\{p_i,\boldsymbol{m}_i,C_i\}$.

To use the GMM for source identification, given S source models λ_k and a set of feature vectors $V=\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_T\}$, the correct model maximizes the probability

$$\max_{1\le k\le S} P(\lambda_k \mid V) = \max_{1\le k\le S} P(V \mid \lambda_k) \tag{31}$$

In other words, the source is matched to the best fitting GMM and identified as that source.

2.3.2 Combining Non-Gaussianity and Probabilistic Inference

Intelligent ICA takes advantage of a one-unit learning rule (discussed in section 3.2.3) by obtaining an estimate of the weight vectors, choosing the one that corresponds to the desired source, and repeating the process until convergence. As discussed in the section

on ICA, the non-Gaussianity measure may have local maxima as well as a global maximum. For our purposes, we want to find the local maximum that corresponds to a specific source.

After the first iteration of the algorithm, an estimate of **w** corresponding to the most non-Gaussian component is obtained. We can also obtain estimates of the other sources because the prewhitening has caused their directions to be orthogonal to the first estimate. Feature vectors can then be extracted from these estimates and compared to the GMM for the desired source. Once the source is identified, the learning rule continues by using the vector **w** that corresponds to the direction of that source. Figure 12 shows the flowchart for the *Intelligent ICA* algorithm.

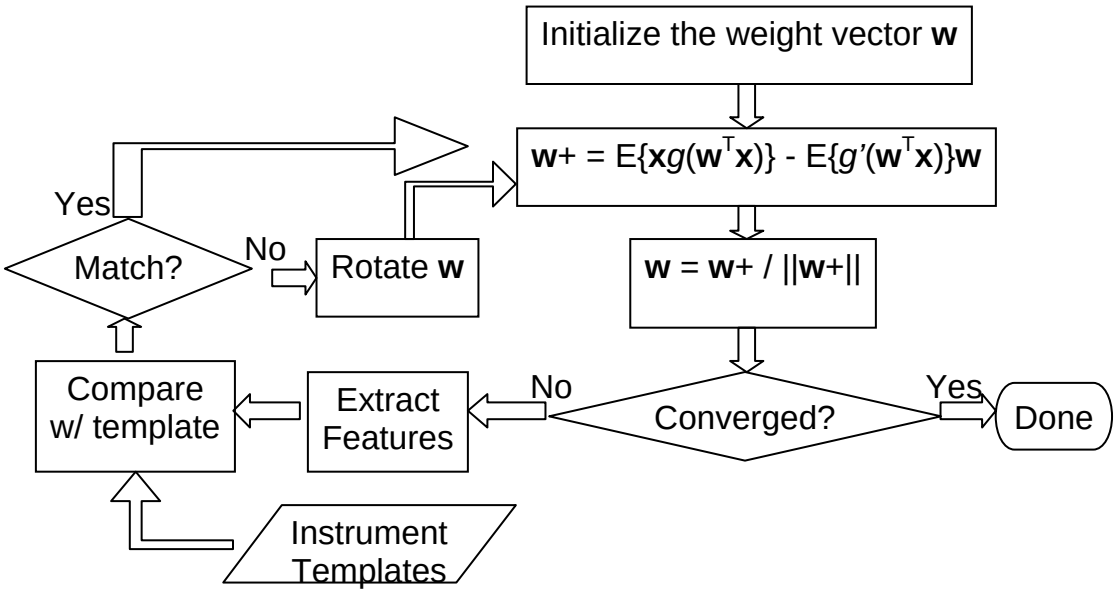

Figure 12. Intelligent ICA Algorithm

* Images courtesy of Aapo Hyvärinen (Hyvärinen et al 2000)

3. Auditory Component Analysis

Computational methods for auditory scene analysis have been shown to be successful for identification and even separation of simple harmonic sounds, but more complex audio streams remain a difficult problem. Independent component analysis works great on mixtures for which the number of sources is equal to the number of sensors (microphones), but does not work well with the underdetermined case (fewer sensors than sources). The proposed system combines computational ASA with ICA to separate several components from a stereo signal.

3.1 Analyzing the Auditory Scene

Several methods for identifying audio sources were presented in Chapter One. Any number of features could be used for grouping and distinguishing auditory units, including, but by no means limited to, spatial location, phase coherence, spectral proximity, temporal proximity, and common onset.

For simplicity, the proposed algorithm considers only phase coherence and spectral/temporal proximity. The Weft representation discussed in the first chapter is well suited for both of these measurements. A Weft is calculated by tracking peaks in the autocorrelogram which are in phase, so phase coherence is an inherent factor. After peaks have been found, tracks are formed which each represent a periodic signal moving through time. This tracking is reminiscent of the sinusoidal tracking by MacAuley and Quatieri (1986), but is accomplished in a slightly different manner. A track may be allowed to jump small distances across the lag-time plane, and this distance may be

determined according to psychoacoustic rules. For now, a simple Euclidean distance measure is used. The Euclidean distance is defined as:

$$\xi = \sqrt{\alpha(t_1 - t_2)^2 + \beta(k_1 - k_2)^2} \tag{32}$$

where t_1 and t_2 are the time indices, k_1 and k_2 are the autocorrelation lag periods, and $\alpha=\beta=1$ for now. Extending this into the psychoacoustic realm would simply require multiplying the time and lag components by appropriate coefficients (α and β respectively), but determining such coefficients would require extensive experimentation.

Tracking of the periodogram peaks may then be accomplished by first setting a maximum number of possible tracks and setting all points below a specified threshold in the periodogram to zero. (A value of 0.01 was empirically determined to work well.) All points above this threshold are set to one, such that the resulting two-dimensional periodogram is binary, as shown in Figure 13.

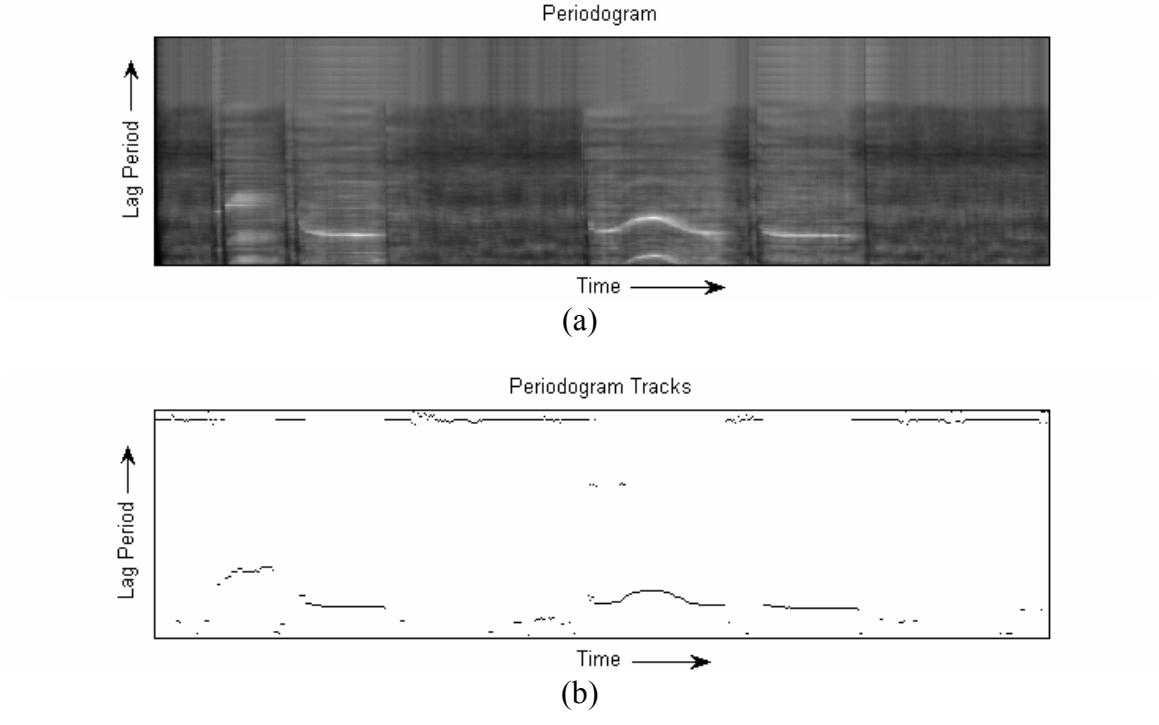

Figure 13. (a) Periodogram and (b) Tracks formed by extracting the peaks.

The locations of all these peaks are then noted and declared as possible track points. The location of the first point is noted and deleted from the list of possible track points. The following loop is iteratively performed until all possible track points have been exhausted:

```
Is the next point within the critical distance, ξ ?
        If so, add this point to the current track.
        If not, have the maximum number of tracks been created?
                If so, is the current track longer than any of the previous tracks?
                        If so, replace the shortest track with the new one.
                        If not, delete the current track
                If not, add a new track
Delete the next point from the list of possible track points
```

The resulting tracks are shown in Figure 14.

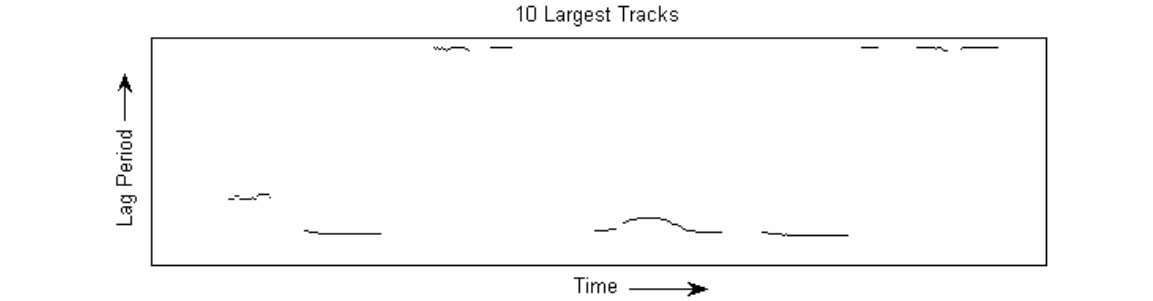

Figure 14. The 10 largest tracks will be used to create the wefts

3.1.1 Source Resynthesis

Borrowing the source-filter model from the field of speech synthesis, we can model sounds like speech as a train of impulses followed by a filter (Rabiner 1978). Because we know the period of each track, we can resynthesize each by creating an impulse train defined by:

$$e(t) = \sum_i \delta(t - t_i) \tag{33}$$

where

$$t_i = \arg_t \left\{ \int_0^t \frac{2\pi}{p(\tau)} d\tau = 2\pi \cdot i \right\} \tag{34}$$

and $p(\tau)$ is the period at time τ. However, the phase will not be exactly a multiple of 2π, so impulses are created at times when the phase passes 2π.

The source-filter model is based on the fact that the spectrum of glottal impulses from the vocal chords is shaped by the acoustic cavity formed by the throat, mouth, and nose. (See Figure 15.) As all vowel sounds are formed in this manner, similar harmonic sounds may also be modeled in this manner.

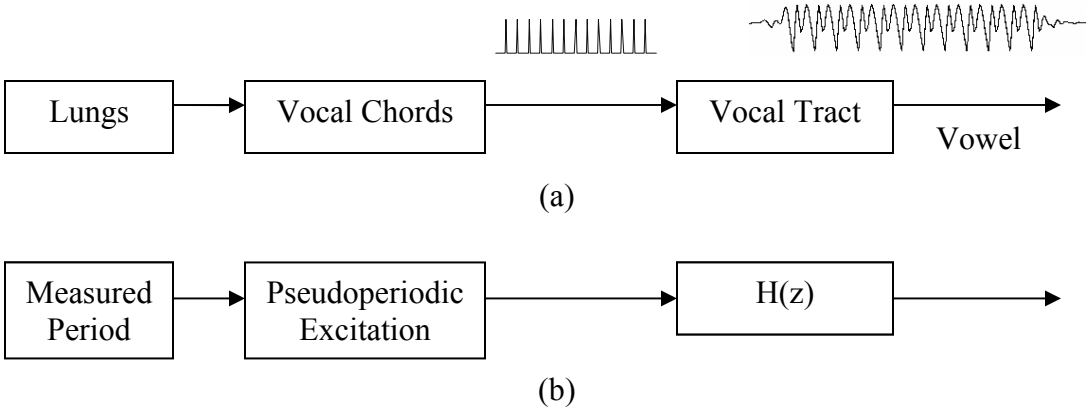

Figure 15. (a) Source-Filter Model for speech synthesis
(b) Resynthesis model for periodic signals

The periods of each of the tracks shown in Figure 14 are shown below in Figure 16. Note that the period follows the inverse of the frequency contour. In other words, where the frequency increases, the period decreases. This corresponds to a high frequency being modeled with closely spaced impulses. Figure 17 shows the pseudoperiodic excitation signal.

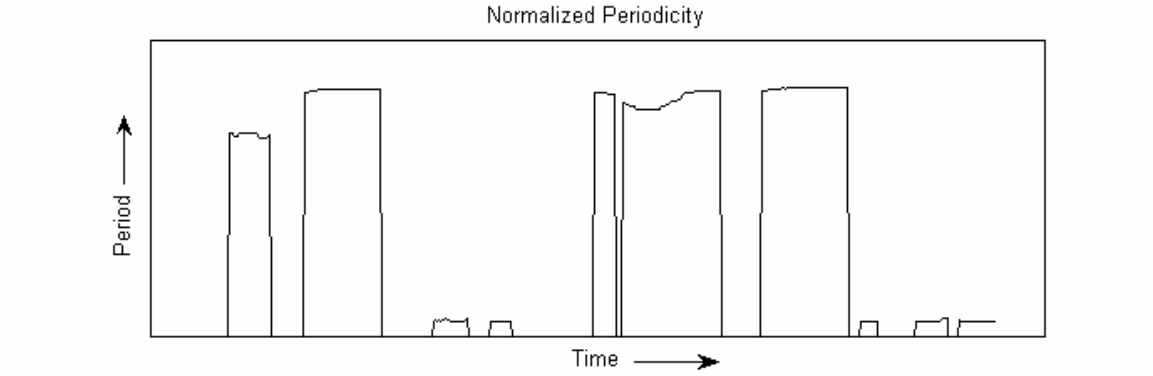

Figure 16. Normalized periodicity

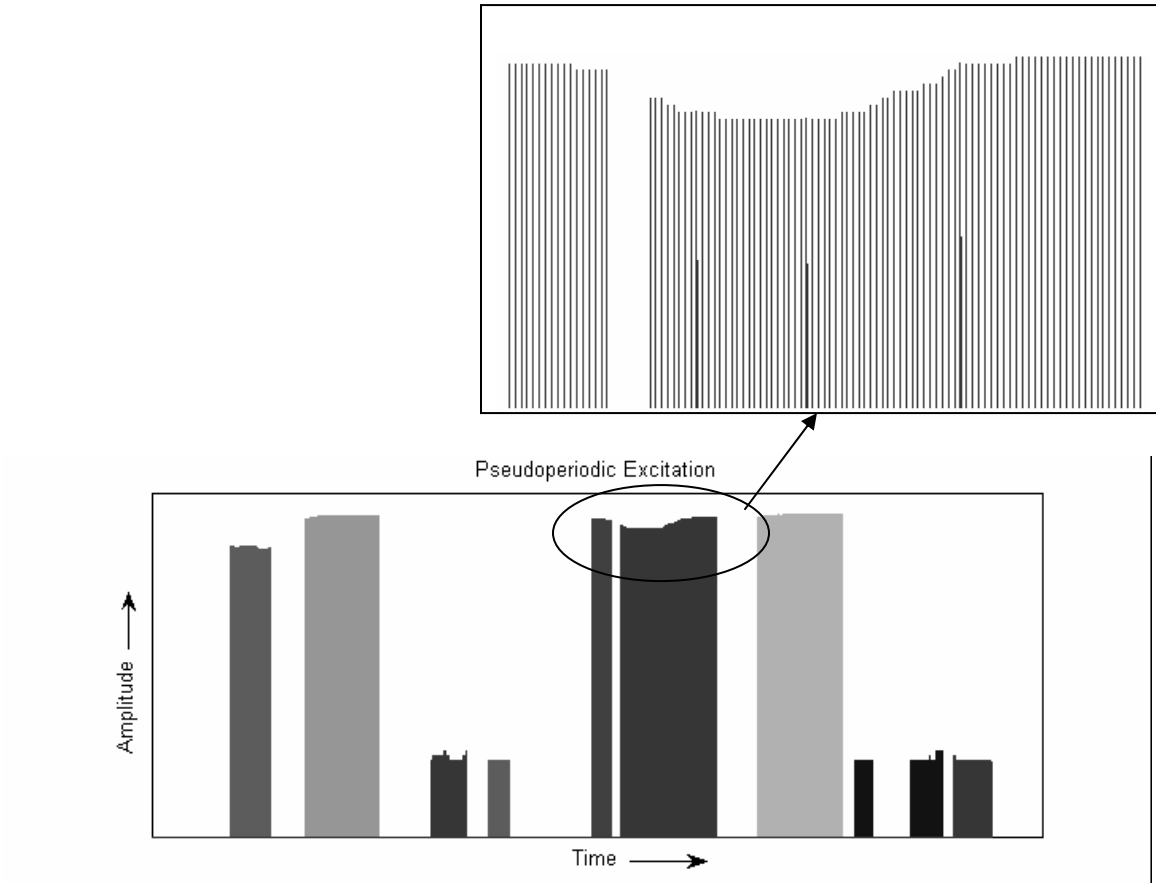

Figure 17. Pseudoperiodic excitations used for resynthesis

The excitation signals are each windowed with a sine-squared window with a length of approximately 20ms. The window is designed such that, with a 50% overlap, the sum of the two windows is equal to one, as shown in Figure 18.

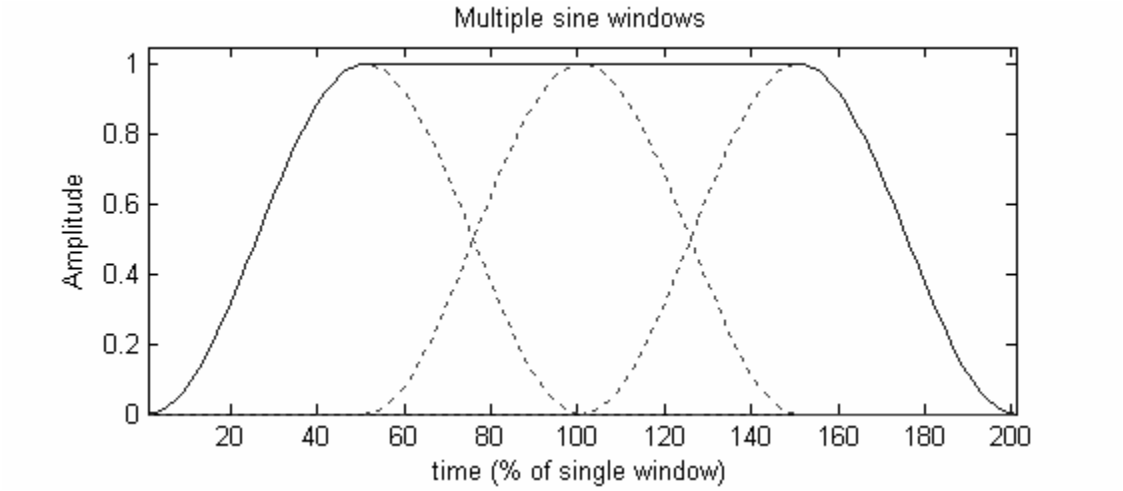

Figure 18. Combination of multiple sine windows

For each frame, a scaling function is derived by taking a slice of the 3D spectrogram to form a spectrum for that frame, as shown in Figure 19. Each excitation signal is then filtered by multiplying the FFT of the excitation signal with the scaling function for that frame. After scaling, the Inverse FFT is applied, and the frames are recombined using the traditional overlap-add technique (Crochiere 1980) to synthesize each audio stream.

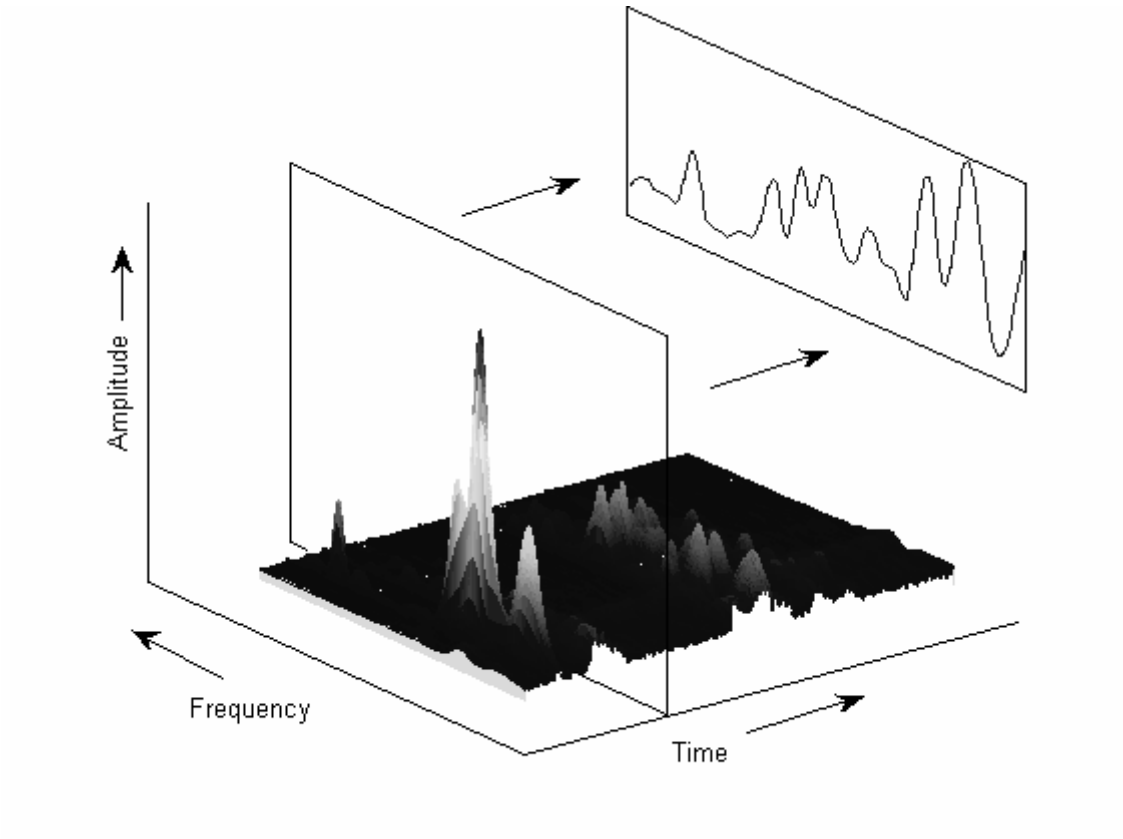

Figure 19. Scaling function derived from original 3D spectrogram

3.1.2 Source Filtering

Alternatively, the spectrum corresponding to each track may be found by extracting the appropriate points from the 3D correlogram and the peaks within the tenth frame of each track's spectrogram are recorded. (This number is chosen merely to reduce computational complexity.) For each of the chosen frames, a cochlear filter bank is designed such that the filters are centered on the frequencies corresponding to the spectral peaks. This results in a series of filter banks for each track.

The original audio tracks are added together and windowed with a sine-squared window (refer back to Figure 11) such that the length of the window is twice the amount of time covered by 10 frames of the correlogram.

The filter bank corresponding to each frame is applied to the audio within that frame, and the frames are added back together. The resulting audio stream coincides with a perceived audio source.

3.2 Analyzing the Auditory Components

Researchers in the field of speech processing have developed several techniques for voice identification. These algorithms include long-term averaging of acoustic features (Markel 1977), Hidden Markov Models (Tishby 1991), and Neural Networks (Rudasi 1991). Reynolds (1995) argues that these algorithms have various drawbacks, and has proposed an algorithm that analyzes the audio based on a psychoacoustically inspired feature vector known as mel-frequency cepstral coefficients, or MFCCs (Mermelstein 1980). These coefficients are used to create a Gaussian probability model (or Gaussian mixture model, GMM) that represents an individual speaker. This algorithm has been used successfully for speaker identification and has been shown to work quite well for instrument identification and separation (Mitianoudis 2002).

The human auditory system does not hear frequencies in a linear fashion, but rather on a logarithmic scale. (See Figure 20.) Several scales have been proposed, one of which is the mel scale. The mel scale is commonly used in speech processing, and is based on a psychological study that asked participants to judge frequencies that were equally distant from one another. Conversion to the mel scale may be accomplished through a filterbank, where the conversion from Hertz to mel is described by the equation

$$m = 1127.01048 \log_e(1 + f / 700) \qquad (35)$$

where f is the frequency in Hertz and m is the mel-frequency.

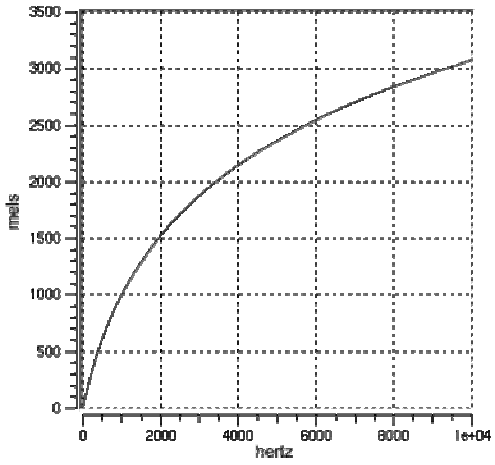

Figure 20. Relationship of Hertz to mel-frequency*

The audio signal may be converted to a frequency-domain signal via the fast Fourier transform (FFT). The coefficients are then converted to a logarithmic scale (in terms of amplitude) and passed through a high-pass pre-emphasis filter to flatten the spectrum, thus assigning significance to the upper harmonics:

$$H(z)=1-0.97z^{-1} \tag{36}$$

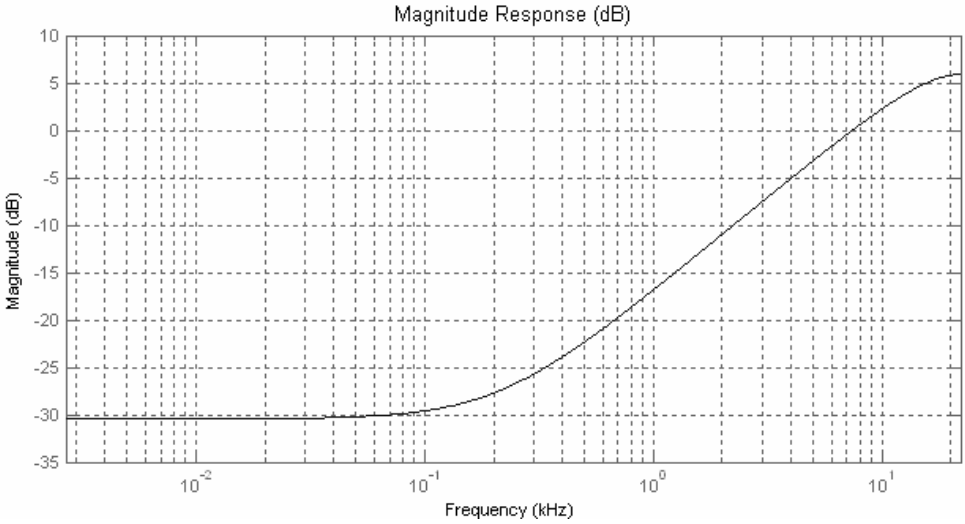

Figure 21. High-pass pre-emphasis filter

The frequency coefficients are grouped and weighted on the mel scale in such a fashion that the sum of the coefficients for a given group is equal to unity, as shown in Figure 22.

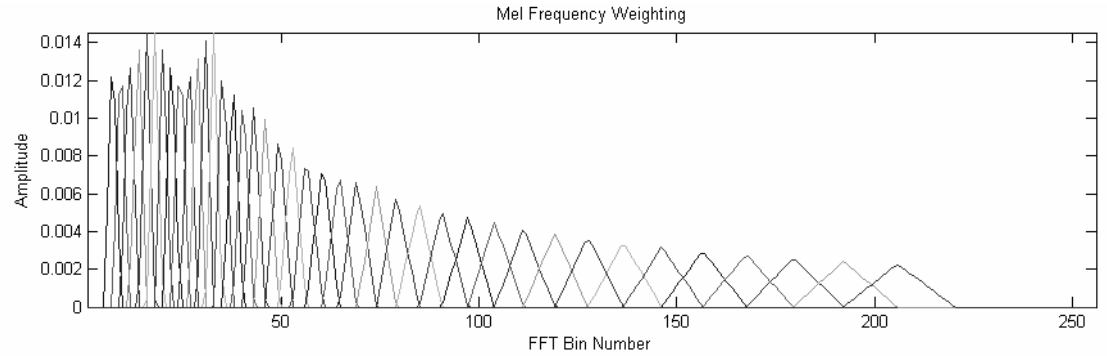

Figure 22. Simulated mel-scale filterbank

The resulting coefficients are then run through a cosine transform, and the output of this is a set of mel-frequency cepstral coefficients (MFCCs). This feature vector was shown to work well for musical instrument recognition in (Eronen 2001).

A Gaussian mixture model (GMM) is defined by equations (29) and (30) in Chapter 2. It is essentially a weighted mixture of several Gaussian distributions, each representing the distribution of a given feature vector. As shown in Figure 23, the mixture of several Gaussian models allows for a more precisely controlled model of the probability density function than a single Gaussian model.

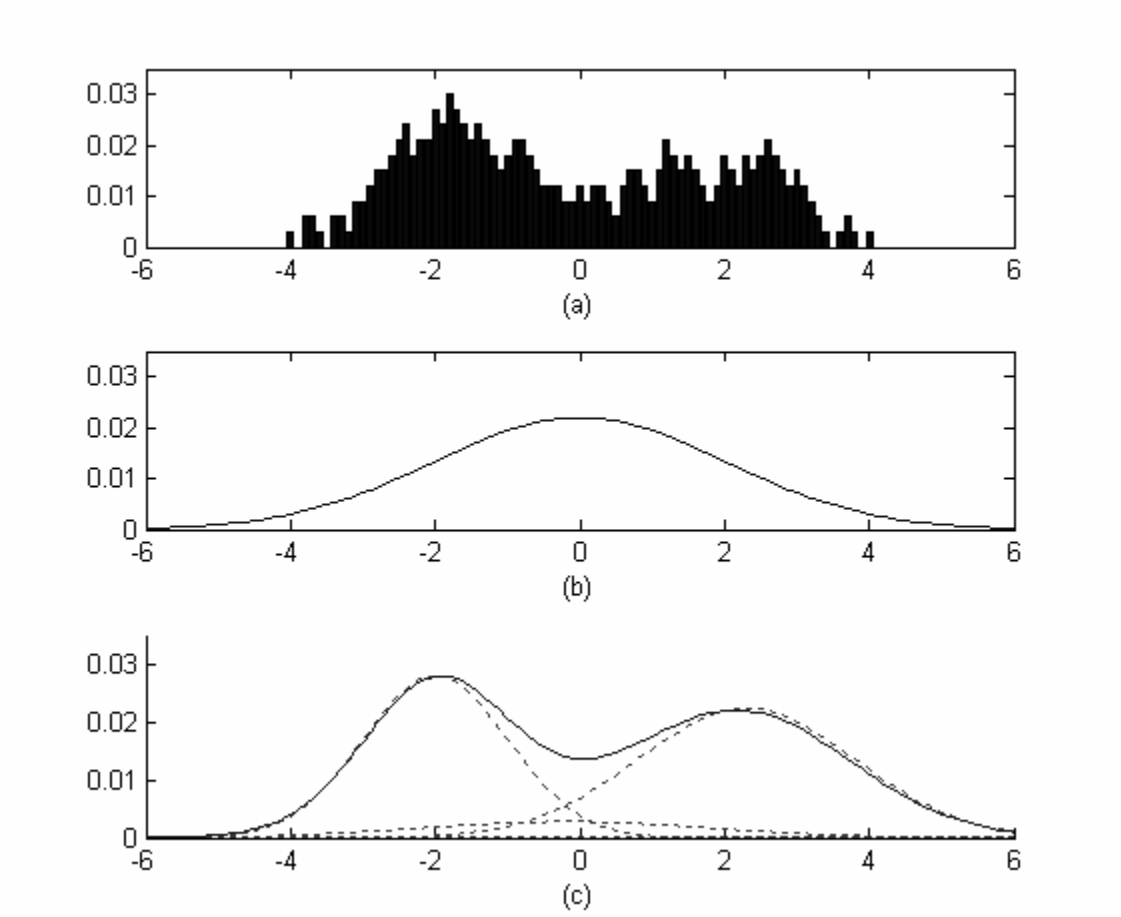

Figure 23. Comparison of distribution models: (a) Histogram of a cepstral coefficient; (b) Maximum likelihood using a single Gaussian model; (c) GMM with several underlying density functions

The GMM is adapted using the *Expectation Maximization* algorithm (Dempster 1977), which attempts to find the parameters that maximize the likelihood of the GMM. Given an initial model λ, this algorithm finds a new model $\overline{\lambda}$, such that $p\left(X \mid \overline{\lambda}\right) \geq p\left(X \mid \lambda\right)$. The GMM is iteratively adapted until a convergence threshold is attained (0.001 for experimentation) or until a specified number of iterations (10) is reached. For each iteration, the parameters are recalculated as follows (Reynolds 1995):

Mixture Weights:

$$\bar{p}_i = \frac{1}{T}\sum_{t=1}^{T} p\left(i \mid \vec{x}_t, \lambda\right) \tag{37}$$

Means:

$$\vec{\bar{u}}_i = \frac{\sum_{t=1}^{T} p\left(i \mid \vec{x}_t, \lambda\right)\vec{x}_t}{\sum_{t=1}^{T} p\left(i \mid \vec{x}_t, \lambda\right)} \tag{38}$$

Variances:

$$\bar{\sigma}_i^2 = \frac{\sum_{t=1}^{T} p\left(i \mid \vec{x}_t, \lambda\right)x_t^2}{\sum_{t=1}^{T} p\left(i \mid \vec{x}_t, \lambda\right)} - \bar{u}_i^2 \tag{39}$$

where the *a posteriori* probability for an auditory object is given by

$$p\left(i \mid \vec{x}_t, \lambda\right) = \frac{p_i b_i(\vec{x}_t)}{\sum_{k=1}^{M} p_k b_k(\vec{x}_t)} \tag{40}$$

3.3 Separating the Auditory Components

Independent Component Analysis is based on a square mixing matrix, which implies that there are an equal number of sources and sensors. If information is needed about an individual source within a standard stereo recording, only two channels may be available. However, in this case, it is possible to separate one source at a time using the Intelligent ICA algorithm presented in the previous chapter.

The proposed system identifies the auditory components using the CASA-based feature vectors described in the previous section and uses this information to drive the ICA engine and remove a particular source. Figure 24 shows how the different algorithms are combined to form the source separation system.

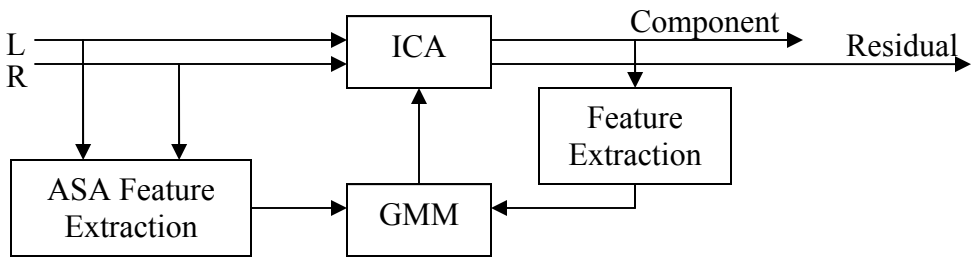

Figure 24. Combining ASA and ICA

In contrast to the Intelligent ICA algorithm described in (Mitianoudis 2002), this algorithm does not require any training period. As in Intelligent ICA, the GMM is adapted according to the *Expectation Maximization* algorithm, but instead of a prolonged training phase, the learning takes place during the separation process. Because the training is derived from the same input signal, the probability density function described by the associated GMM models the desired source very closely.

The GMM is initialized according to the harmonic signal extracted by means of the ASA processing block. This information is used to identify the independent component and extract it from the mix using the following algorithm. After each iteration of the FastICA algorithm, feature vectors are calculated for the separated component and compared with the GMM corresponding to the desired source. While the independent component matches the desired source, the mixing matrix continues to adapt until it converges. However, if the independent component does not match the desired source, the estimated mixing matrix is rotated such that an orthogonal signal becomes the next estimate for the output. (As discussed in the previous chapter, this is possible because the input signals have been whitened.) The process is repeated until the mixing matrix converges to a point such that the output independent component matches the

desired GMM, or until a maximum number of iterations have been performed. Twenty iterations appear to be more than enough for convergence. The resulting two independent components correspond to the desired source and the remaining mixture of all other sources.

---

[*] Image copied from Wikipedia article entitled "mel frequency" under the GNU license agreement

4. Experimentation

4.1 Measurements

Evaluation criteria for separation algorithms, both CASA and BSS systems, have often been subjective and nearly always inconsistent. Many have evaluated their systems based on speech recognition rate, while others have used signal-to-noise ratios or simple subjective listening tests. A standardized set of tests has been proposed (Schobben 1999) which defines measures of distortion and separation, and prescribes the use of specific audio test files, both real and synthetic.

Distortion is defined as follows:

$$D_j = 10\log\left(\frac{E\left\{\left(x_{j,s_j} - \alpha_j y_j\right)^2\right\}}{E\left\{\left(x_{j,s_j}\right)^2\right\}}\right) \tag{41}$$

where

$$\alpha_j = \frac{E\left\{x_{j,s_j}^2\right\}}{E\left\{y_j^2\right\}}. \tag{42}$$

The indices j are chosen such that output y_j corresponds to an original audio source signal s_j. Notice that this distortion measure is minimal when the output y_j is equal to $x_{j,sj}$ (the contribution of the source signal s_j to the observed signal x_j). The distortion may also be calculated as a function of frequency by applying the short-time Fourier transform:

$$D_j(\omega) = 10\log\left(\frac{E\left\{\left(STFT\left\{x_{j,s_j} - \alpha_j y_j\right\}\right)^2\right\}}{E\left\{\left(x_{j,s_j}\right)^2\right\}}\right) \tag{43}$$

The quality of separation is defined as

$$S_j = 10\log\left(\frac{E\left\{\left(y_{j,s_j}\right)^2\right\}}{E\left\{\left(\sum_{i\neq j} y_{j,s_i}\right)^2\right\}}\right) \tag{43}$$

where $y_{j,si}$ is the output when only source s_i is active. This too may be expressed as a function of frequency:

$$S_j = 10\log\left(\frac{E\left\{\left(STFT\left\{y_{j,s_j}\right\}\right)^2\right\}}{E\left\{\left(STFT\left\{\sum_{i\neq j} y_{j,s_i}\right\}\right)^2\right\}}\right) \tag{44}$$

4.2 Test Data

The first set of audio test files is derived from a recording of two speakers, one male and one female, in the presence of background noise. The room measured 3.1 meters high, 4.2 meters wide, and 5.5 meters deep. The recordings were made with both speakers in the presence of the two omni-directional microphones, but with only one person speaking at a time. The contributions of the two speakers to each microphone may be added together to create the observed signals x_1 and x_2. The data set consists of the four audio files described in Table 1.

File Name	Description
spc1s1m1.wav	Contribution of Source 1 to Microphone 1
spc1s1m2.wav	Contribution of Source 1 to Microphone 2
spc1s2m1.wav	Contribution of Source 2 to Microphone 1
spc1s2m2.wav	Contribution of Source 2 to Microphone 2

Table 1. Audio Test Data Set #1*

The remaining sets of test data have been created synthetically and include the sounds described in Table 2. Each signal is accompanied by a corresponding gated signal which turns the signal on and off at various points.

File Name	Description
sine.wav	Sine wave oscillating in frequency (Fig. 23)
sawtooth.wav	Sawtooth wave oscillating in frequency
square.wav	Square wave oscillating in frequency
gaussn.wav	Gaussian noise
cauchyn.wav	Cauchy noise
gsine.wav	Gated sine wave oscillating in frequency
gsawtooth.wav	Gated sawtooth wave oscillating in frequency
gsquare.wav	Gated square wave oscillating in frequency
ggaussn.wav	Gated Gaussian noise
gcauchyn.wav	Gated Cauchy noise (Fig. 26)

Table 2. Synthetic Audio Test Data*

These sounds may be mixed in a number of ways. The simplest option is to add signals together to create composite mixes. The sources may also be filtered by a simple FIR or Head Related Transfer Function (HRTF) filter before mixing. Yet another possibility is to apply a simulated or measured room response, specifying the location of sources and sensors.

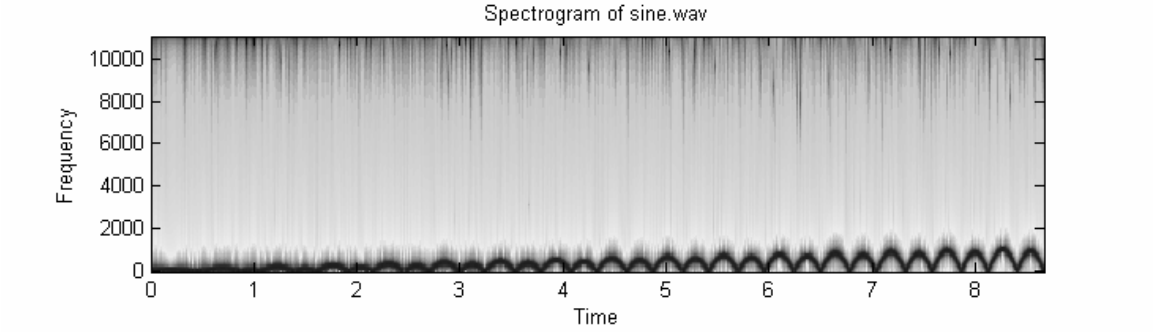

Figure 25. Spectrogram of oscillating sine wave

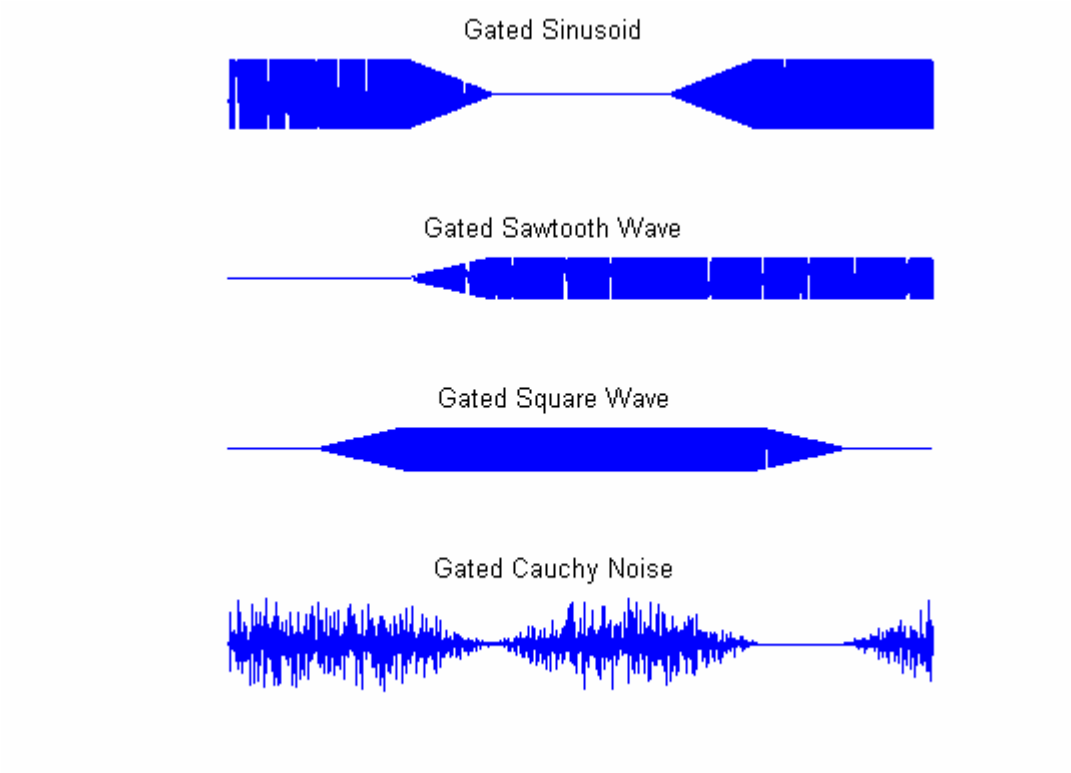

Figure 26. Gated test signals

4.3 Evaluation Procedure

Distortion and separation measures are calculated for several different variations of sources as well as assorted source and sensor locations. Five test cases are described below.

Test Case #1: An oscillating sine tone mixed with Cauchy noise.

Test Case #2: The stereo recording of the male and female speakers.

Test Case #3: Gated sawtooth wave, gated sine wave, gated Cauchy noise filtered with Head Related Impulse Responses (HRIR) at 30° left of center, center, and 30° right of center, respectively. (These are simply used to act as low-order mixing filters, not because they are related to spatial listening.) As shown in Figure 27, there are two HRIRs for each angle. Both HRIRs are applied to the signal. The signals corresponding to the left ear are added together and the signals corresponding to the right ear are added together. The center channel is not actually filtered with an HRIR.

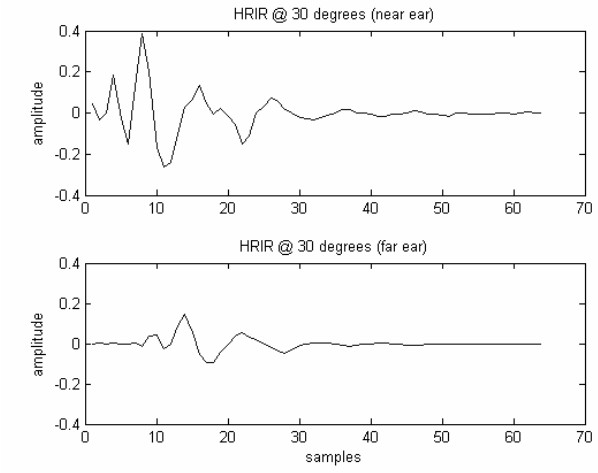

Figure 27. Head Related Impulse Responses

Test Case #4: Sawtooth wave, gated sine wave, and Cauchy noise, filtered by a dense mixing FIR filter, resembling the response of a room. The FIR filters are designed such that none of the coefficients are zero. The coefficients are generated from an exponentially decaying Cauchy noise.

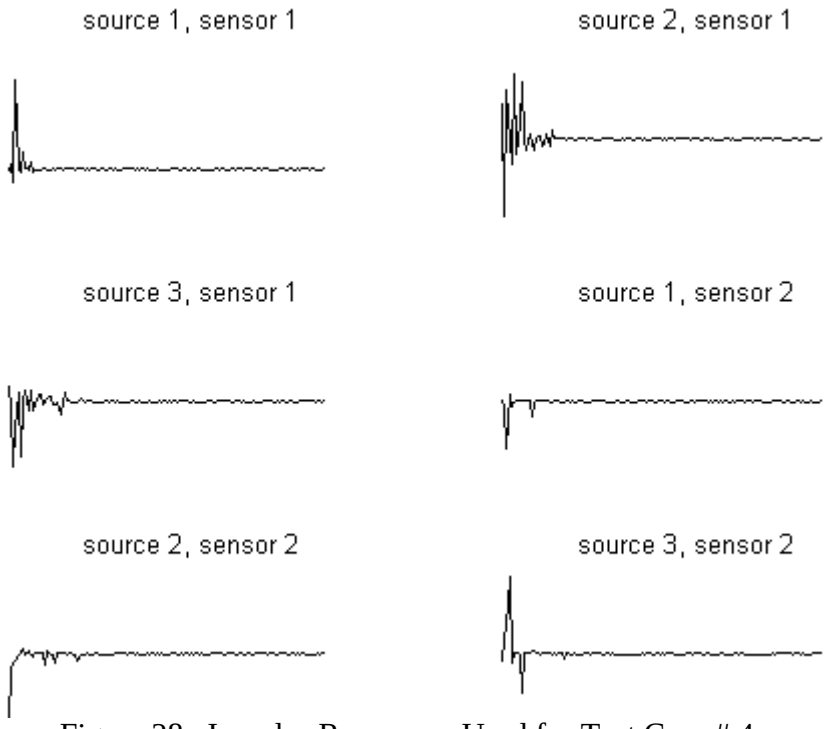

Figure 28. Impulse Responses Used for Test Case # 4

Test Case #5: Gated square wave, sine wave, sawtooth wave, and gated Cauchy noise, each placed in corners of a virtual room (1 meter from each wall), with a pair of virtual sensors in the center of the room 1 meter apart, as shown in Figure 29.

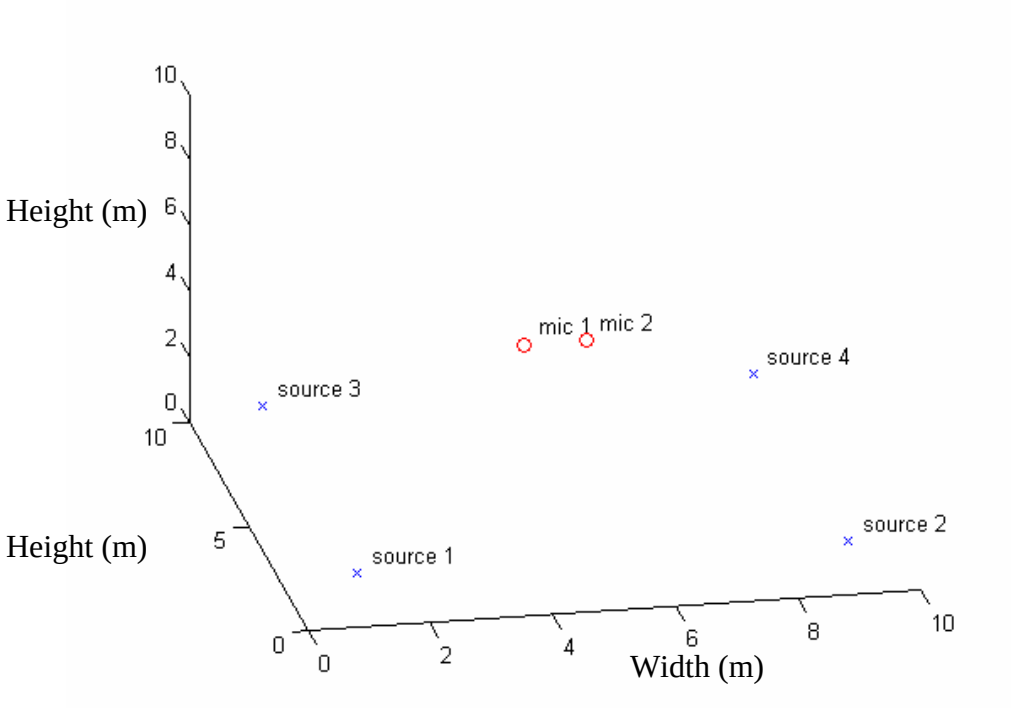

Figure 29. Source and microphone placements for Test Case #5

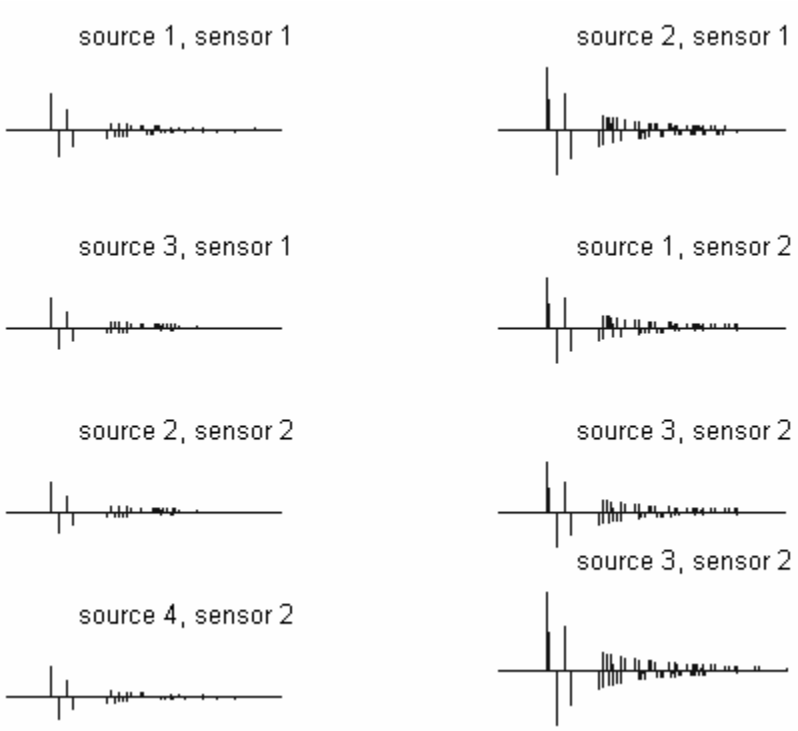

Figure 30. Impulse Responses Used For Test Case #5

[*] Recordings and Matlab code for synthesis/mixing taken from http://www.ele.tue.nl/ica99

5. Results

5.1 Data

Test Case #1 (An oscillating sine tone instantaneously mixed with Cauchy noise)

The oscillating sine tone was extracted from the mixture by choosing the GMM that corresponds to the appropriate period track, and the separation was measured to be 116dB according to (43). The original and separated waveforms are shown in Figure 31.

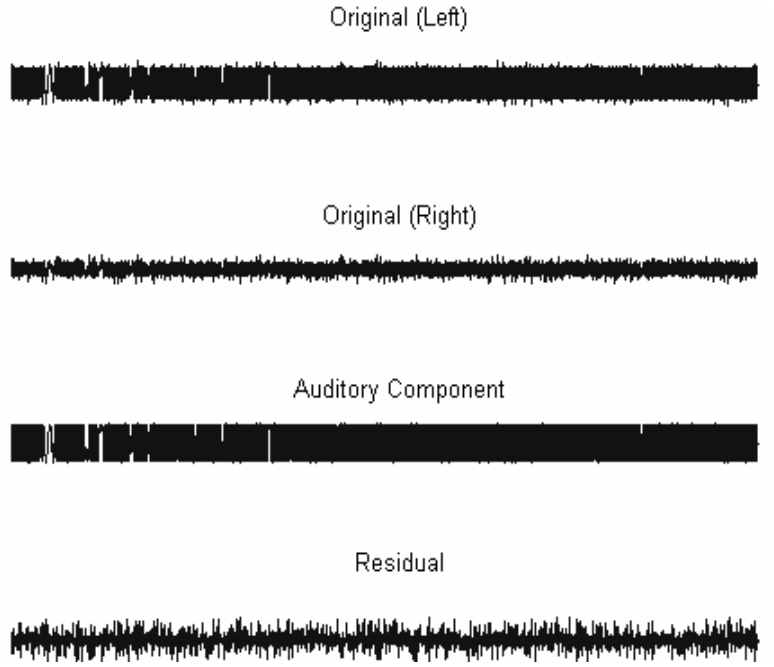

Figure 31. Original and Separated Waveforms for Test Case #1

Figure 32 shows the spectrogram of the original and separated waveforms. Both of the original waveforms include a substantial amount of broad-frequency noise. However, the auditory component contains virtually no noise, and the residual is composed primarily of the noise.

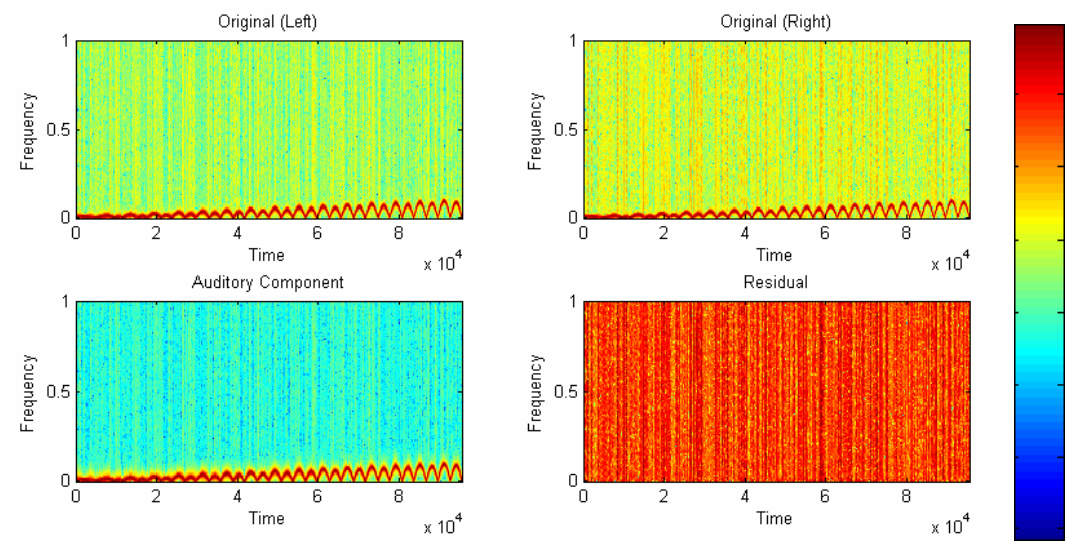

Figure 32. Spectrograms of Original and Separated Sounds for Test Case #1

Figure 33 shows the distortion of each channel for the first test case. It can be seen that the separated component (*BSS output 2*) has virtually no distortion from *source 2*. Instead, nearly all of the energy is due to *source 1*.

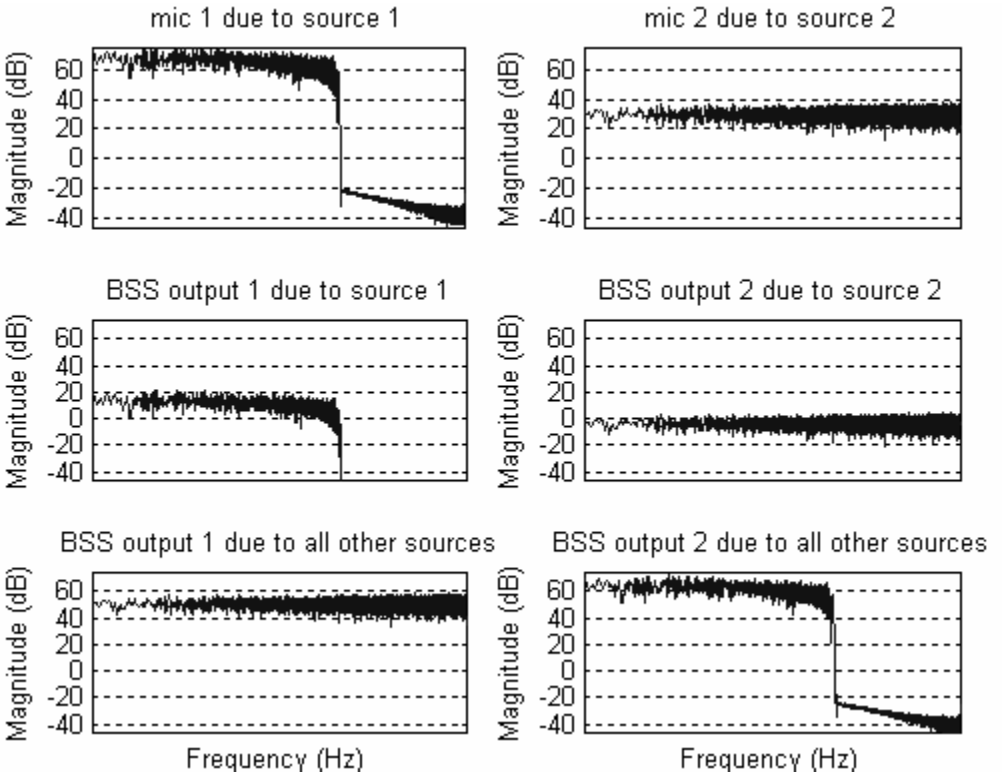

Figure 33. Distortion Measures for Test Case #1

Test Case #2 (Stereo recording of the male and female speakers)

The male speaker was extracted from the mixture, and the separation was measured to be 0.2dB. The original and separated waveforms are shown in Figure 34.

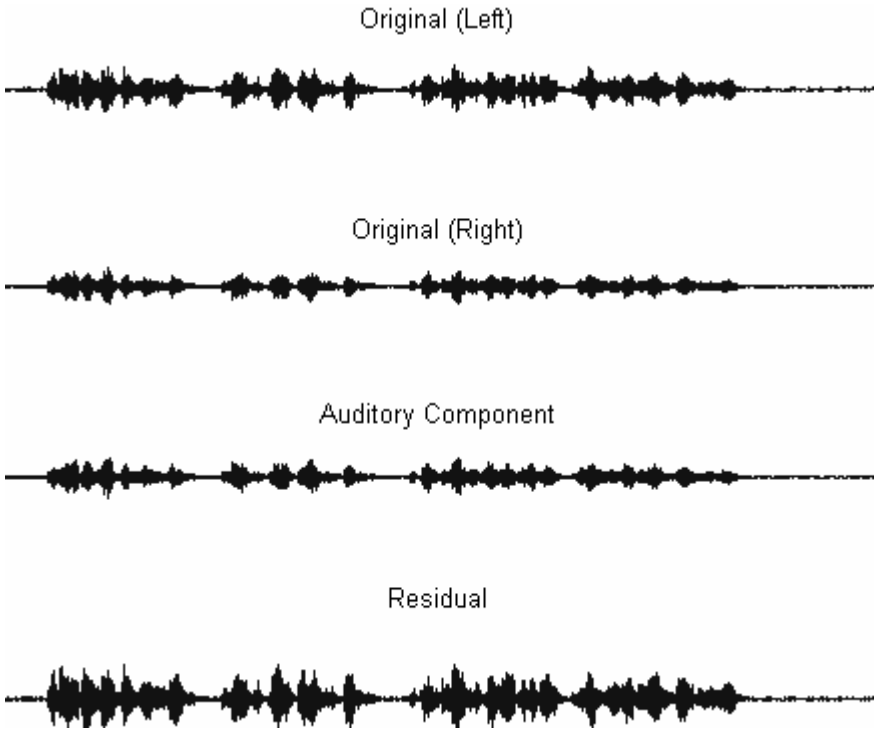

Figure 34. Original and Separated Waveforms for Test Case #2

Figure 35 shows the spectrogram of the original and separated waveforms. Notice that the auditory component and residual signals look very similar to the original waveforms.

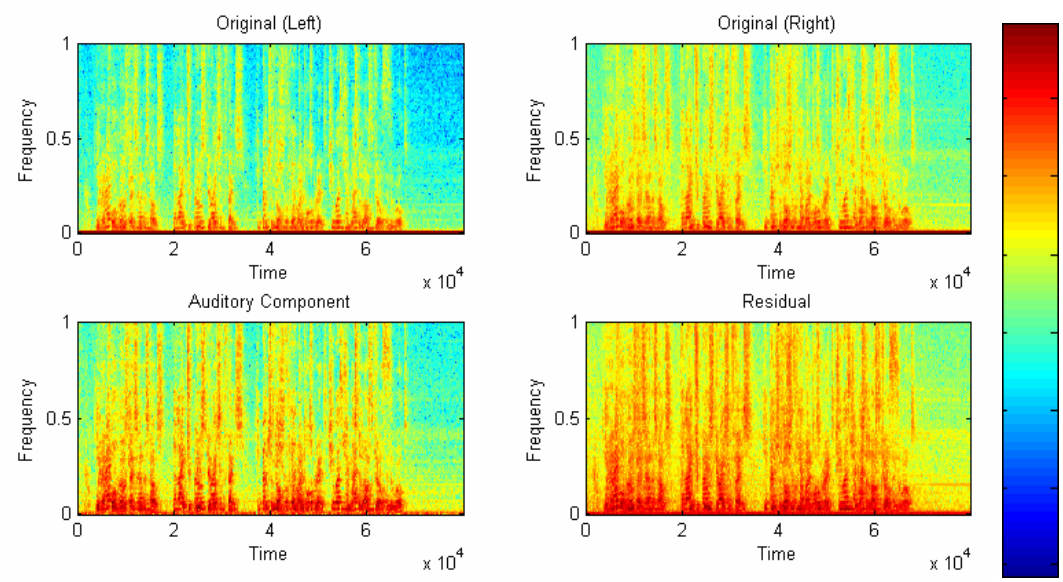

Figure 35. Spectrograms of Original and Separated Sounds for Test Case #2

Figure 36 shows the distortion of each channel for the second test case. Notice that each output has about an equal amount of energy from both sources.

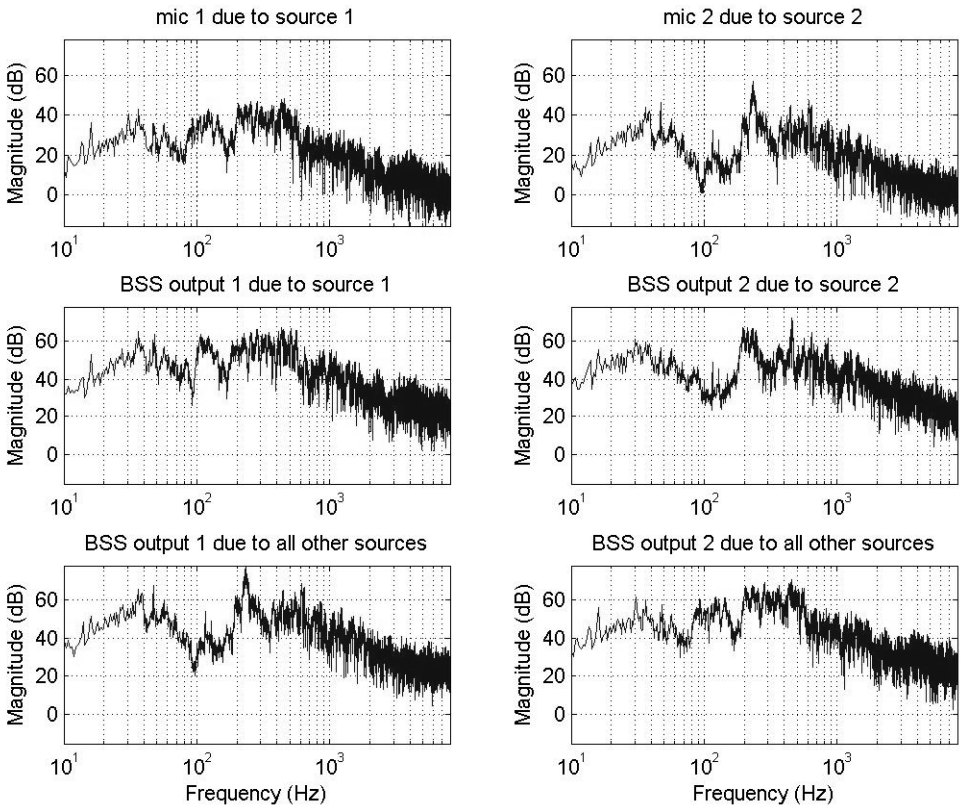

Figure 36. Distortion Measures for Test Case #2

Test Case #3 (Gated sawtooth wave, gated sine wave, gated Cauchy noise filtered with Head Related Impulse Responses at 30° left of center, center, and 30° right of center, respectively.)

The sawtooth wave was extracted from the mixture, and the separation was measured to be 43.7dB. The original and separated waveforms are shown in Figure 37.

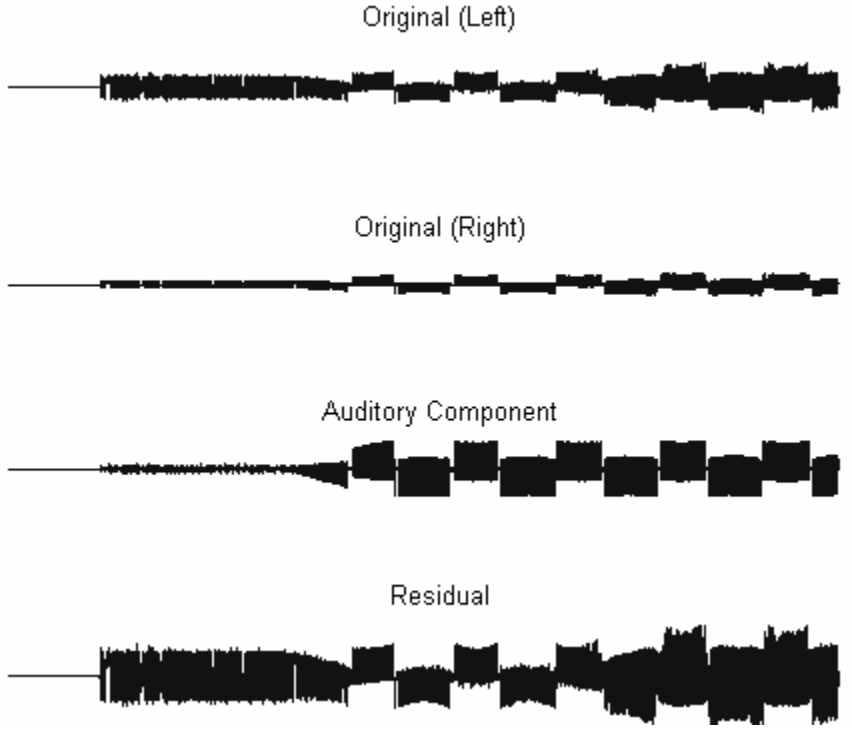

Figure 37. Original and Separated Waveforms for Test Case #3

Figure 38 shows the spectrogram of the original and separated waveforms. Notice that the spectrum of the oscillating sine tone is virtually absent from the auditory component, and the spectrum of the sawtooth wave is harder to see in the residual spectrogram.

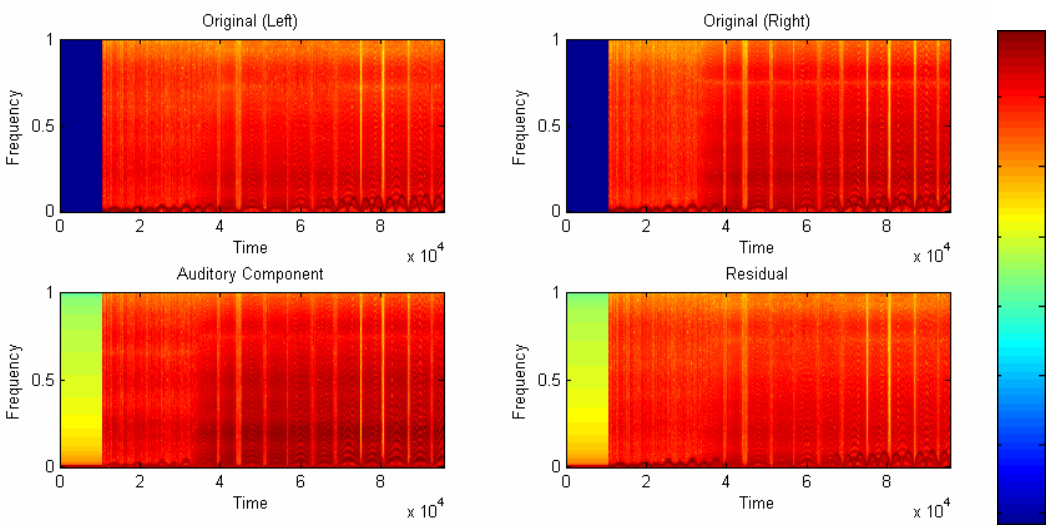

Figure 38. Spectrograms of Original and Separated Sounds for Test Case #3

Figure 39 shows the distortion of each channel for the third test case. Note that the energy in *BSS output 1* due to the combination of *source* 1 and *source* 2 is lowest, signifying that *source 3* (the sawtooth wave) is most prevalent.

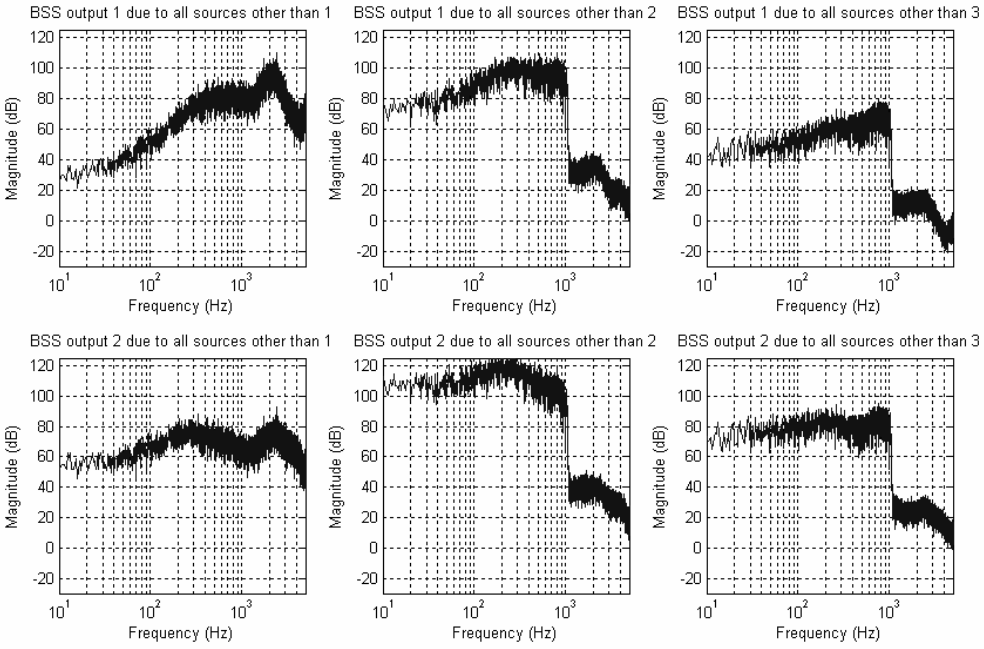

Figure 39. Distortion Measures for Test Case #3

Test Case #4 (Sawtooth wave, gated sine wave, and Cauchy noise, filtered by a dense mixing FIR filter, resembling the response of a room.)

The sawtooth waveform was extracted from the mixture, and the separation was measured to be 15.5dB. The original and separated waveforms are shown in Figure 40.

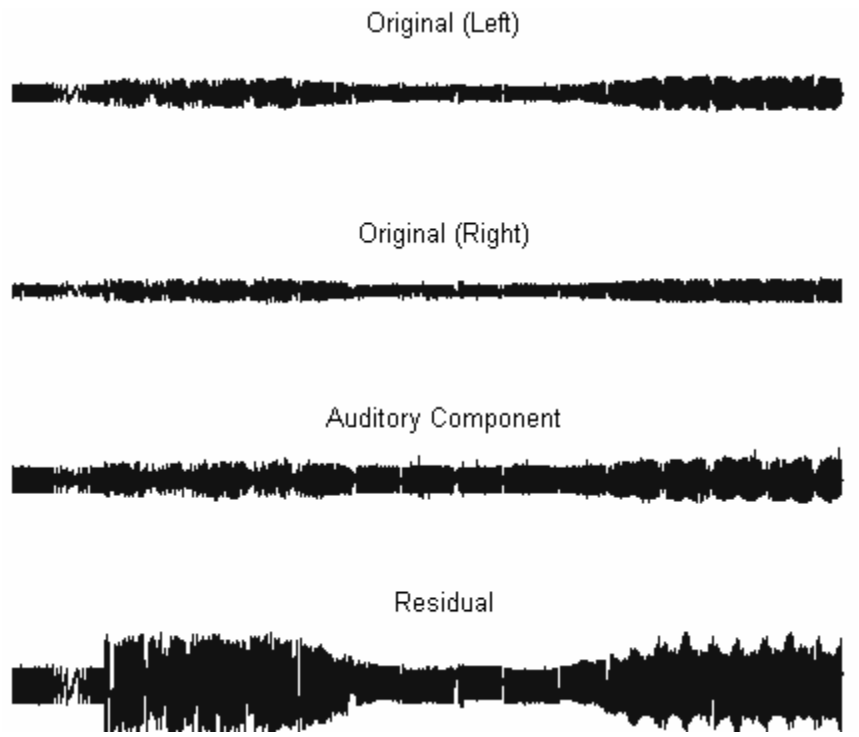

Figure 40. Original and Separated Waveforms for Test Case #4

Figure 41 shows the spectrogram of the original and separated waveforms. While some of the sine wave and Cauchy noise are still present, a significant amount of high-frequency energy is also there, signifying that the sawtooth wave is prevalent. (Notice that the high-frequency content is not present in the residual.)

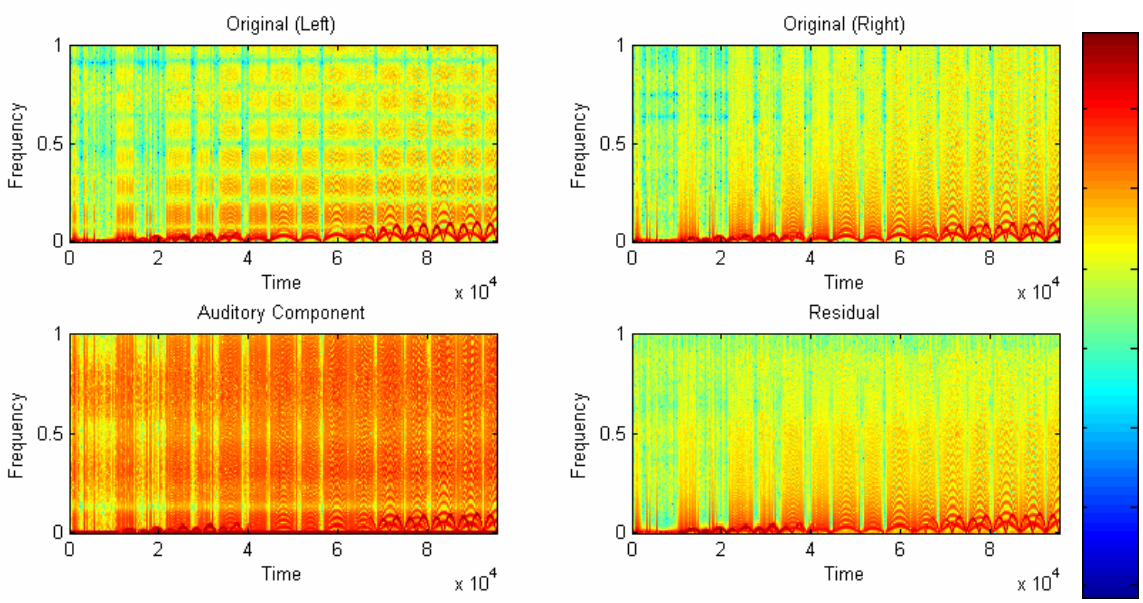

Figure 41. Spectrograms of Original and Separated Sounds for Test Case #4

Figure 42 shows the distortion of each channel for the fourth test case. Note that the energy in *BSS output 2* due to the combination of *source* 1 and *source* 2 is lowest, signifying that *source 3* (the sawtooth wave) is most prevalent.

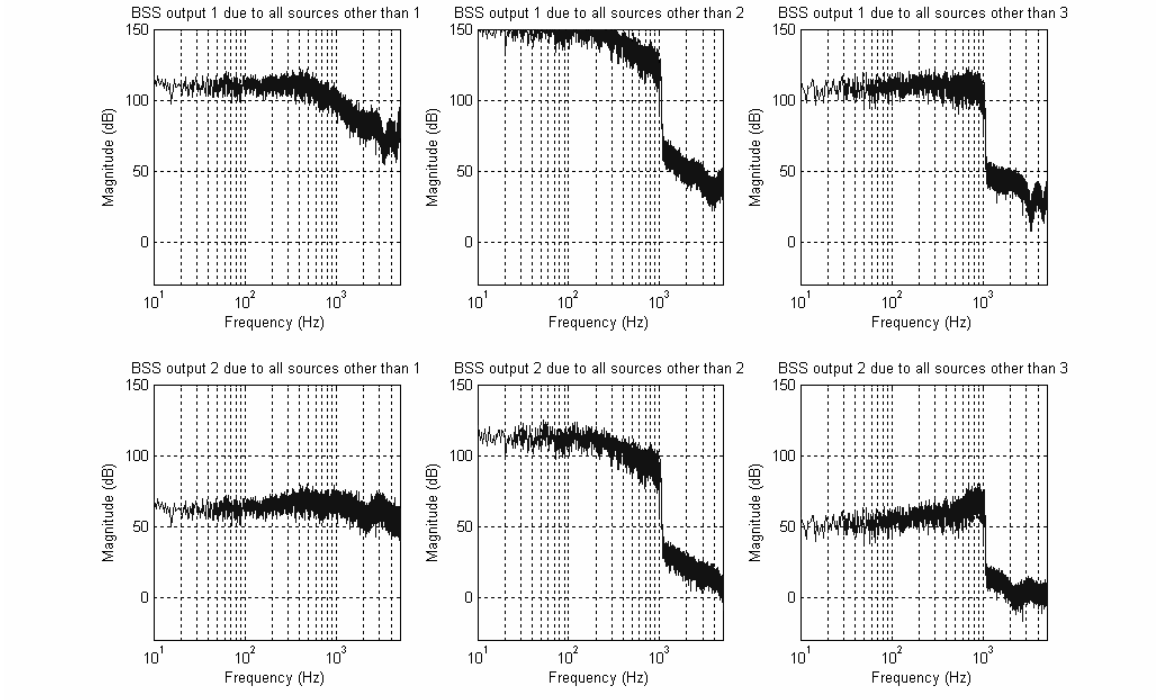

Figure 42. Distortion Measures for Test Case #4

Test Case #5: (Gated square wave, sine wave, sawtooth wave, and gated Cauchy noise, each placed in corners of a virtual room (1 meter from each wall), with a pair of virtual sensors in the center of the room 1m apart.)

The square wave was extracted from the mixture, and the separation was measured to be approximately 1.6dB. The original and separated waveforms are shown in Figure 43.

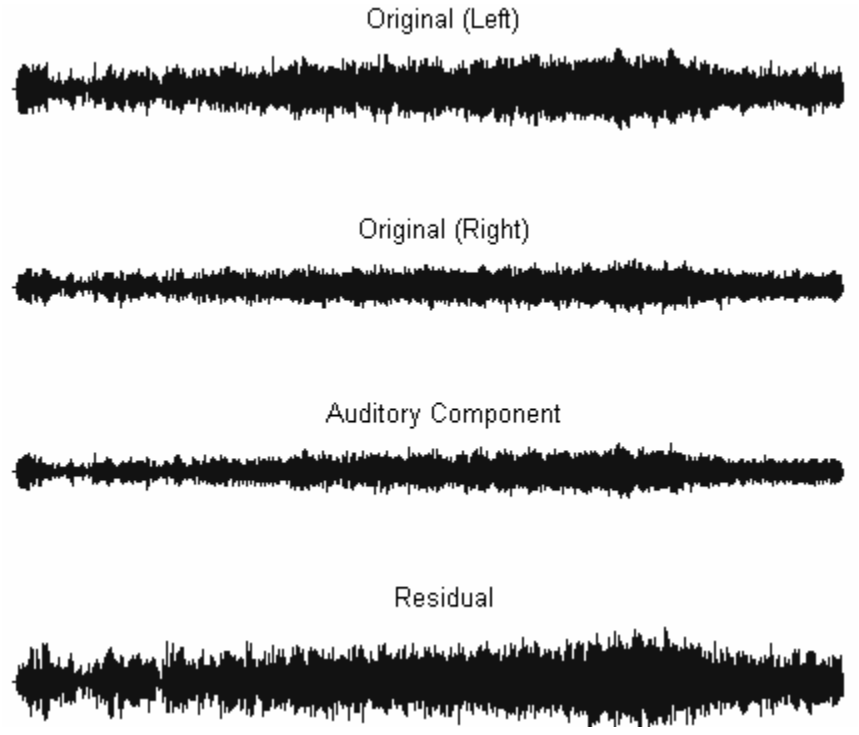

Figure 43. Original and Separated Waveforms for Test Case #5

Figure 44 shows the spectrogram of the original and separated waveforms. Practically nothing is visible here. The four sources combined with the reverberation noise have clouded the spectrogram so that identification of sources of very difficult.

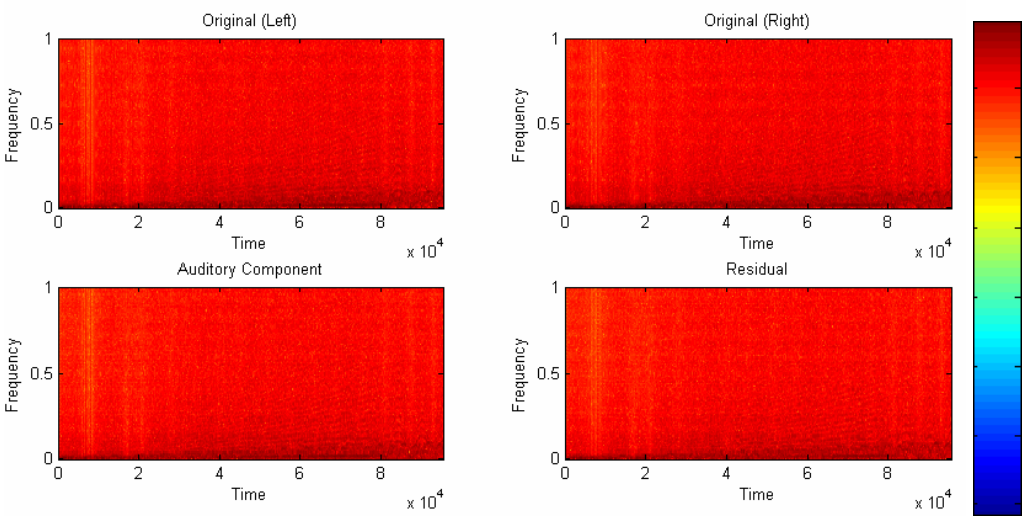

Figure 44. Spectrograms of Original and Separated Sounds for Test Case #5

Figure 45 shows the distortion of each channel for the fifth test case. The energy distribution of the various sources is roughly equivalent, telling us that separation has not occurred.

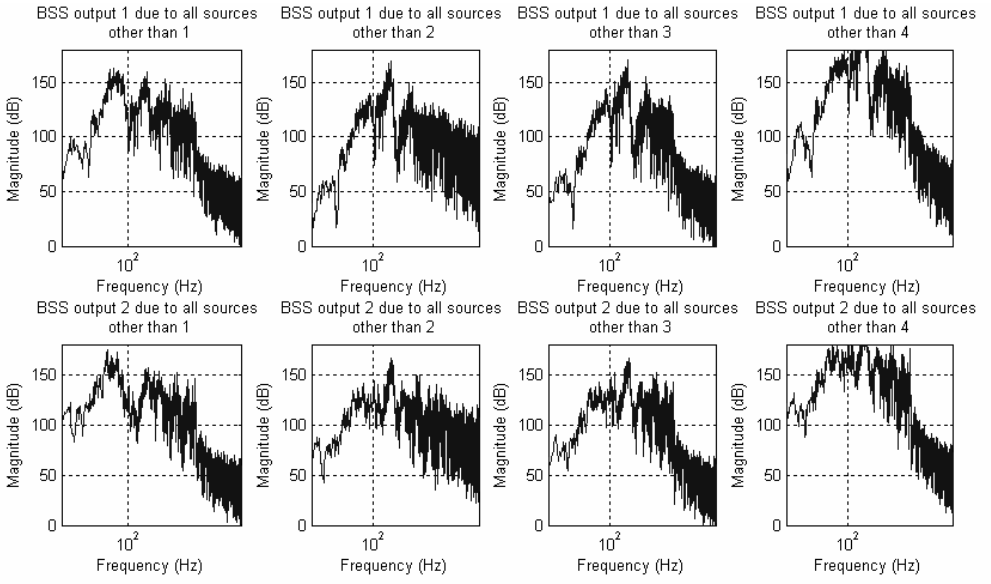

Figure 45. Distortion Measures for Test Case #5

Additional Tests

A few additional tests were performed to obtain more data. The first such test was a simple experiment to see if anechoic recordings of musical instruments could be separated. The recordings were randomly panned. Table 3 shows the various combinations of mixtures tested and the resulting separation.

Separated Instrument	Flute (Left 38%)	Bassoon (Left 28%)	Alto Sax (Right 20%)	Bass Clarinet (Right 18%)	Violin (Left 46%)
Alto Sax (Right 20%)	12.5dB	x	x	x	x
Alto Sax (Right 20%)	6.8dB		x	x	x
Bass Clarinet (Right 18%)	14.8dB			x	x
Trumpet (Right 40%)	x	x	x	3dB	

Table 3. Instrument mixtures and corresponding quality of separation

Because panning seemed to have an effect on the performance of the separation algorithm, another test was performed to ascertain just how panning affects the amount of separation. Four cello recordings were panned to varying degrees in the stereo soundstage. The cellos were placed at -15°, -5°, 5°, and 15°. The separation results are shown in Table 4.

-15°	-5°	5°	15°	Separation
	X	X		6 dB
X			X	79 dB
X	X	X		16 dB
X	X	X	X	3 dB

Table 4. Separation of panned cellos

Finally, a third test was added to make certain that the separation algorithm would work with speech. Using three anechoic speech samples panned far left, center, and far right, 15.6dB of separation was achieved.

5.2 Remarks

By far the best performing of the five original test scenarios was the simplest- the separation of the sine wave from the instantaneous mixture with noise was 116dB. The next best performance was with the Head-Related Transfer Functions at 43.7dB of separation. The Head-Related Impulse Responses for the near and far ears respectively are shown in figure 27. These create a relatively complex mixture that includes an interaural level difference and an interaural time difference between the sensors. The *2x2* mixing matrix used for the independent component analysis essentially restricts the model to a simple weighted sum of the inputs. The fact that the proposed system worked well in this situation shows promise.

The FIR room response method also did fairly well with 15.5dB of separation, but the mixture that simulated a virtual room response did not do so well, with only 1.6dB of separation. The FIR filters used to model a simple room response are shown in figure 28. This system is a little more complex than the HRTFs, and this may account for the decreased amount of separation.

The scenario which used the simulated room performed rather poorly. This could be due in part to the fact that this system had four sources, whereas the HRTF and FIR mixtures were composed of only three sources. However, it also more closely modeled an actual stereo recording, with longer reverberation tails and a phase difference between

the microphone signals. The impulse responses used for this scenario are shown in Figure 43. This more complex system presents a significant challenge for sound-separation tasks, and the proposed system will need more fine tuning before it handles this situation well.

The most difficult scenario for this system was the stereo recording of two speakers talking simultaneously. With only 0.2dB of enhancement, it did not provide any significant separation. This is simply a more complex situation than the simulated room because it was recorded in an actual room (in the presence of background noise and a significant amount of reverberation).

In the additional tests, instruments were separated from several different mixtures of a variety of instruments. The first test case, an alto saxophone was separated from a mixture of the alto sax and a flute. The resulting output was separated 12.5dB. However, when a bassoon was added to the mix, the separation of the alto sax was reduced to only 6.8dB. This would suggest that as the number of sources increases, the amount of separation decreases. However, when the bass clarinet was added to the mix (for a total of four instruments) then removed, the separation improved to become 14.8dB. This drastic change suggests that something about the bassoon is drastically different. The bassoon was panned right 18%, while the alto sax, flute, and bassoon were panned right 20%, left 38%, and left 28% respectively. In the previous test, the alto sax was spatially separated from the flute and bassoon, but in this test the bassoon and the alto sax are very close. Therefore, it appears that the spatial location was not the determining factor in this test, so further testing is needed to determine the specific cause. Most likely, the reason for the increased separation in this test is due to some statistical properties of the bassoon.

If the bassoon signal is less Gaussian than the other signals, the separation should be far simpler. Because the ICA algorithm attempts to maximize non-Gaussianity, separation of a pure tone (minimum entropy) from a random sequence (much higher entropy) should result is more separation than two signals with similar entropies.

To evaluate the importance of spatial separation, various combinations of stereo panned cellos were mixed together and a single instrument separated from each mixture. When two cellos were panned just slightly (5° left and right), the separation was only 6dB, but when the stereo separation was increased to 15° left/right, the separation increased to 79dB. Apparently, a slight increase in spatial separation can have a significant effect on the amount of separation possible.

When this is extended to three and four cellos, the separation diminishes to 16dB then 3dB respectively, showing that the number of sources also has an effect on the amount of separation that may be achieved. As the number of sources increases, the separation task becomes more difficult.

6. Conclusion

6.1 Analysis of results

Over the past several years, great advances have been made in the field of audio processing. New techniques have been created for engineers to analyze and process audio. Recent research into computational auditory scene analysis has given us tools to better understand the perception of our auditory environment. Advances in statistical signal processing, such as independent component analysis, have enabled us to separate conglomerations of measurements into their respective components. However, computational ASA has not given us a framework for effectively separating the many sounds that surround us and ICA generally requires a large number of sensors to allow separation of the independent components.

The proposed algorithm combines these two areas by first examining the perceived audio streams then, using information gathered from these streams, separating the audio such that the output audio files are statistically independent. It was shown that separation of a single sound source from a mixture of many is possible with the proposed algorithm. Separation ranged from 116dB to no separation, depending on the mixing conditions. The results presented in the previous chapter show that as the complexity of the listening environment increases, the sound separation problem becomes more difficult. However, simple mixtures, such as an instantaneous mix or a low-order filtering, can be separated considerably. It was also shown that separation of stereo panned sounds can be accomplished with relative ease. However, the degree of separation depends not only on the reverberation time, but also the spatial separation and the number of sources present. As spatial separation increases, more separation is

possible. As with all blind source separation problems, separation of many sources is much more difficult than two or three.

6.2 Future Work

Given more time, some improvements may be done to improve the performance of this separation algorithm. The current algorithm only separates one component, but it may be extended to separate out more auditory components. One possible method would be to separate several components in parallel, as shown in Figure 46.

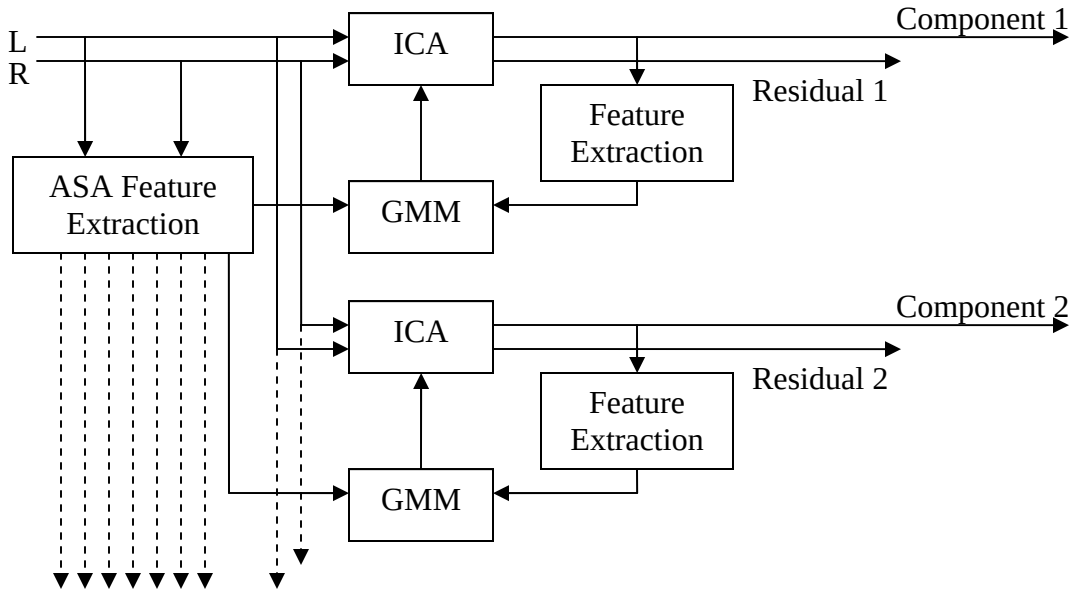

Figure 46. Removing several auditory components in parallel.

Another such improvement would be to extract more features. A set of cepstral coefficients could be calculated for the attack of each sound in addition to a set of coefficients for the steady-state portion. This would better model the tremendous affect

of attack characteristics on the timbre of a sound. In addition to calculating the cepstral coefficients, features such as the spectral centroid, crest factor, onset asynchrony, and amplitude envelope may allow even greater precision in the separation process. A perceptual model that encompasses and prioritizes these features could be very useful for future sound separation endeavors.

It may even be possible to improve separation by considering sound source location. Incorporating localization information into the auditory scene analysis algorithm would be a challenging project that might produce significant improvements. After all, sound localization is a very strong cue for the human auditory scene analysis problem, especially in noisy, crowded, and reverberant environments. Including such information in a computational model might alleviate some of the difficulties that arise from having multiple sources and long reverberation tails.

References

Blauert J., *Spatial Hearing, The Psychophysics of Human Sound Localization*, MIT Press, Cambridge, Massachusetts, 1997.

Bregman, Albert S. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, Cambridge, Massachusetts, 1990.

Bregman, Albert and Pierre Ahad. *Demonstrations of Auditory Scene Analysis: The Perceptual Organization of Sound*. Audio Compact Disk. MIT Press, Cambridge, Massachusetts, 1996.

Cardoso, J.-F. and P. Comon. *Independent component analysis, a survey of some algebraic methods*. In *Proc. ISCAS'96*, vol. 2, pp. 93-96, 1996.

Cherry, E. C., "*Some experiments on the recognition of speech, with one and with two ears*," Journal of Acoustic Society of America, vol. 25, pp. 975-979, 1953.

Crochiere, R. E. "*A weighted overlap-add method of shorttime Fourier analysis/synthesis.*" IEEE Transactions on ASSP, 28(1), pp. 99-102, 1980.

Dempster, A. P., Laird, N.M., and Rubin, D.B., "*Maximum Likelihood from Incomplete Data via the EM Algorithm*," J. of the Royal Stat. Soc., Series B, 39(1):1-38, 1977.

Ellis, Dan and David Rosenthal. "*Mid-level Representation for computational auditory scene analysis*". In Proc. Of the Computational Auditory Scene Analysis Workshop, 1995.

Ellis D.P.W. "*The Weft: A representation for periodic sounds*." In Proc. of Int. Conference on Acoustic, Speech & Sig. Proc. ICASSP-97, Munich, Vol. 2 pp. 1307-1310, 1997.

Eronen, A. "*Comparison of features for musical instrument recognition*". In Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2001.

Friedlander, B., and B. Porat, "*The Modified Yule-Walker Method of ARMA Spectral Estimation,"* IEEE Transactions on Aerospace Electronic Systems, AES-20, No. 2 (March 1984), pp. 158-173.

Gardner, Bill and Keith Martin. *HRTF Measurements of a KEMAR Dummy-Head Microphone*. 18 May 1994. 29 January 2003. < http://sound.media.mit.edu/KEMAR.html>.

Grey J. M., "*Multidimensional perceptual scaling of musical timbres*," J. Acoust. Soc. Amer., vol. 61, pp. 1270-1277, 1977.

Hyvärinen, A. "*Fast and Robust Fixed-Point Algorithms for Independent Component Analysis.*" IEEE Transactions on Neural Networks 10(3):626-634, 1999.

Hyvärinen, Aapo and Erkki Oja, "*Independent Component Analysis: Algorithms and Applications,"* Neural Networks*, 13(4-5):411-430, 2000.*

Hyvärinen, Aapo, Juha Karhunen and Erkki Oja. *Independent Component Analysis*. John Wiley & Sons, Inc., New York, 2001.

Johnson, Don H. *The Binaural Pathway*. 16 March 1997. 22 February 2005 < http://www-ece.rice.edu/~dhj/binaural.html>.

Marr, David. *Vision, A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman Press, San Francisco, 1982.

Markel, J., B. Oshika, and A. Gray, Jr., "*Long-term feature averaging for speaker recognition*," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-25, pp. 30-37, Aug. 1977.

McAulay, Robert J. and Thomas F. Quatieri, "*Speech analysis /synthesis based on a sinusoidal representation*," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. ASSP-34, no. 4, pp. 744-754, Aug. 1986.

Mermelstein, Paul, and Steven B. Davis, "*Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences*," IEEE Trans. Acoust., Speech, Signal Processing, vol. 28(4), pp. 357–366, 1980.

Mitianoudis, Nikolaos and Mike Davies, "*Intelligent Audio Source Separation Using Independent Component Analysis.*" In Proceedings of the 112th Convention of the Audio Engineering Society. Munich, Germany, May 10-13 2002.

Rabiner, L. R. and R. W. Schafer, *Digital Processing of Speech Signals*, PTR Prentice Hall, Inc., Englewood Cliffs, NJ, USA, 1978.

Reynolds D.A., Rose R.C., "*Robust Text-Independent Speaker Identification using Gaussian Mixture Speaker Models*," IEEE transactions on Speech and Audio Processing, Vol.3, No. 1, January 1995.

Rosenthal, D. and H. G. Okuno, Eds., *Computational Auditory Scene Analysis*, Lawrence Erlbaum Associates, 1998.

Rudasi, L., and S.A. Zahorian, "*Text-independent talker identification with neural networks*," In Proc. of Int. Conference on Acoustic, Speech & Sig. Proc. ICASSP-91, pp. 385-388, May 1991.

Schobben, D., K. Torkkola and P. Smaragdis, *"Evaluation of Blind Signal Separation Methods"*, in Proceedings Int. Workshop Independent Component Analysis and Blind Signal Separation, Aussois, France, January 11-15 1999.

Tishby, N. Z., "*On the application of mixture AR hidden Markov models to text independent speaker recognition*," IEEE Trans. Signal Processing, vol. 39, pp. 563-570, March 1991.

Wikipedia, Mel Scale. 5 March 2005. Wikimedia Foundation. 18 March 2005. <http://en.wikipedia.org/wiki/Mel_scale>.

Wikipedia, Principle Components Analysis. 31 January 2005. Wikimedia Foundation. 22 February 2005. <http://en.wikipedia.org/wiki/Principal_components_analysis>.