AUDIO/VISUAL INDEPENDENT COMPONENTS

Paris Smaragdis[*]

Media Laboratory
Massachusetts Institute of Technology
Cambridge MA 02139, USA
paris@media.mit.edu

Michael Casey

Department of Computing
City University
Northampton Square, London EC1V 0HB, UK
casey@soi.city.ac.uk

ABSTRACT

This paper presents a methodology for extracting meaningful audio/visual features from video streams. We propose a statistical method that does not distinguish between the auditory and visual data, but one that operates on a fused data set. By doing so we discover audio/visual features that correspond to events depicted in the stream. Using these features, we can obtain a segmentation of the input video stream by separating independent auditory and visual events.

1. INTRODUCTION

Perceiving objects in the real world is a process that integrates cues from multiple modalities. Our mental representation of many things is not just an image, but also a sound or a smell, or an experience from any other sensory domain. Objects exist in this multidimensional space and we are very well tuned to parsing it and understanding such multiple modalities of an object. Computer recognition on the other hand is mostly limited to individual domains, sometimes heuristically combining findings at some higher level. Recently some work has emerged in the audio/visual domain, trying to address this issue. Hershey and Movellan (2000), made an introduction to this field by observing that audio and visual data off a video stream exhibit some statistical regularity that can be employed for joint processing. Slaney and Covell (2001), in a system designed to improve the synchrony of audio and video, refined that statistical link between audio and video. Finally Fisher *et al.* (2001), demonstrated an audio-visual system that successfully correlated audio and visual activity by use of information theory, thereby bypassing an implicit assumption in the previous work that the audio/visual data are Gaussian distributed. In this paper, we pursue a similar approach; however we hope to present a more general and compact methodology that is based on well-known algorithms. Additionally, unlike this past work, we seek to perform object extraction from the audio/visual space and not just correlate auditory with visual cues. Finally we will try to place our work in the larger framework of machine perception and redundancy reduction (Barlow 1989) and not limit its scope to the audio/visual domain.

[*]Currently in Mitsubishi Electric Research Labs, Cambridge, MA 02139 USA

2. SUBSPACE PROJECTIONS

Subspace projections are an efficient method of data reduction. When paired with powerful optimization criteria, they uncover a lot of the structure of the data. In this paper we will employ the subspace independent component methodology proposed for audio segregation by Casey (2001), and extended for video by Smaragdis (2001). This procedure is divided into two steps: 1) a dimensionality reduction, and 2) an independence transform step.

2.1. DIMENSIONALITY REDUCTION

In our introduction we will assume a multidimensional input data set $\mathbf{x}(t) \in \mathbb{R}^n$ with zero mean[1]. Dimensionality reduction is performed by principal components analysis (PCA), a linear transformation $\mathbf{W}_o$ that will project our input $\mathbf{x}(t)$, to make its variates orthonormal, that is:

$$\mathbf{x}_o(t) = \mathbf{W}_o \cdot \mathbf{x}(t),$$

so that $E\{\mathbf{x}_o \cdot \mathbf{x}_o^T\} = \mathbf{I}$ ($\mathbf{I}$ being the identity matrix, and $E\{\cdot\}$ the expectation operator). PCA algorithms usually organize the output $\mathbf{x}_o(t)$ in order of variance, so that the first dimension exhibits maximal variance, whereas the last dimension exhibits the least. In order to perform the dimensionality reduction we keep the dimensions that exhibit maximal variance, that is the first few dimensions of $\mathbf{x}_o$ so that $\mathbf{x}_r(t) = \mathbf{x}_o^{(1\ldots m)}(t)$. The superscript denotes the dimensions of $\mathbf{x}$ that we select resulting in $\mathbf{x}_r(t) \in \mathbb{R}^m$. The

[1]The zero mean constraint is not mandatory, but it simplifies the presentation of this process. For our examples later we enforce this constraint by removing the mean from all input data.

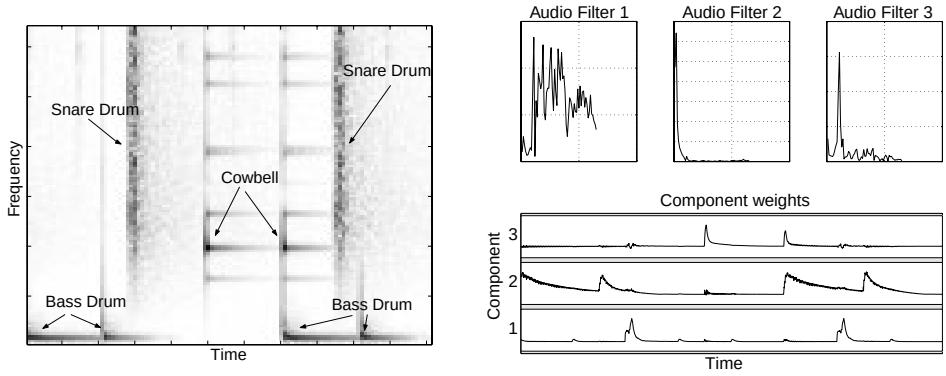

Fig. 1. The left plot is the magnitude spectrum $\mathbf{f}_m$ of a drum loop composed of a bass drum, a snare drum, and a cowbell. The bottom right plot displays the component weights $\mathbf{f}_{mi}$ that we extracted from it. The top left plots display the three subspace independent component bases $\mathbf{W}$ that mapped the input $\mathbf{f}_m$ to the components $\mathbf{f}_{mi}$.

complete data reduction transform can then be expressed as $\mathbf{W}_r = \mathbf{W}_o^{(1\ldots m)}, \mathbf{W}_r \in \mathbb{R}^{m\times n}$.

2.2. INDEPENDENCE TRANSFORM

For the subsequent independence transform, we employ independent components analysis (ICA) (Hyvärinen 1999), which ensures that the variates of its input, $\mathbf{x}_r$, will be maximally statistically independent. This is also a linear transform:

$$\mathbf{x}_i(t) = \mathbf{W}_i \cdot \mathbf{x}_r(t)$$

To estimate $\mathbf{W}_i$ we used a natural gradient algorithm (Amari *et al.* 1995). This is an iterative algorithm in which the update of $\mathbf{W}_i$ is defined as:

$$\Delta\mathbf{W}_i \propto (\mathbf{I} - g(\mathbf{x}_r(t)) \cdot \mathbf{x}_r(t)) \cdot \mathbf{W}_i,$$

where for $g(\cdot)$ we used the hyperbolic tangent function. Upon convergence of $\mathbf{W}_i$, the resulting $\mathbf{x}_i(t)$, will contain elements such that their mutual information will be minimized.

2.3. COMBINING, UNDERSTANDING AND INVERTING

The overall two-step process can also be described by a single linear transformation $\mathbf{W} = \mathbf{W}_i \cdot \mathbf{W}_r, \mathbf{W} \in \mathbb{R}^{m\times n}$. The inverse transform of this process will be $\mathbf{A} = \mathbf{W}^+, \mathbf{A} \in \mathbb{R}^{n\times m}$, where the $^+$ operator denotes the generalized matrix inverse.

The quantities $\mathbf{x}_i(t), \mathbf{A}$ and $\mathbf{W}$ have a special interpretation that we will use. $\mathbf{x}_i(t)$ is a set of maximally independent time series which carry information to make an reconstruction of the original $\mathbf{x}(t)$, by projecting them through the transform $\mathbf{A}$. $\mathbf{W}$ contains a set of basis functions that will create these independent time series from the original input. The quality of the reconstruction depends on how much smaller the original dimensionality n is from the reduced dimensionality m. How we determine m, the number of dimensions we keep, is a complex issue which is not yet automated, and for which we employ heuristics. If we wish to reconstruct the original input using only the ith component of the analysis, we can do so by setting all the elements of $\mathbf{x}(t)$, except the ith, to zero and synthesizing by $\mathbf{A} \cdot \mathbf{x}_i(t)$. In the remainder of this paper we will refer to $\mathbf{x}_i(t)$ as the component weights, and to the rows of $\mathbf{W}$ as the component bases. This procedure allows us to decompose a high dimensional input to a smaller set of independent time series. If the input contains a highly correlated and redundant mix of time series, this operation will remove the correlation and the redundancy so as to expose the content using a sparse description.

For some of the examples presented later, the dimensionality of the data was in the order of several tens of thousands, which requires a prohibitive amount of computational power for the dimensionality reduction step. In order to deal with this issue we instead employed either Lanczos methods, or fast approximate PCA algorithms (Roweis 1997, Partridge and Calvo 1998), which qualitatively give the same results.

3. AUDIO SUBSPACES

To use the above technique in the audio domain we compute a frequency transform of the input sound $s(t)$:

$$\mathbf{f}(t) = f\{[s(t) \cdots s(t+n)]^T\},$$

with $\mathbf{f} \in \mathbb{C}^n$, and $f\{\cdot\}$ is an arbitrary transform (e.g. a DFT). From it we extract the magnitude $\mathbf{f}_m = |\mathbf{f}|$, and the phase $\mathbf{f}_a = \measuredangle\mathbf{f}$ components of the signal. The magnitude data is then factored using the above process to obtain:

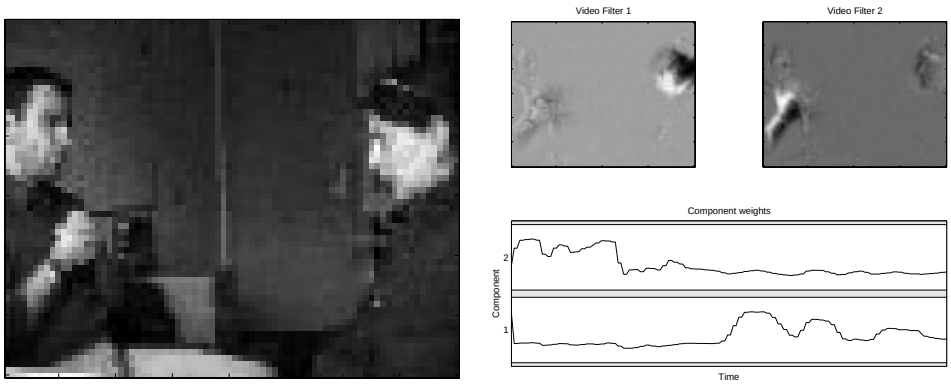

Fig. 2. The left plot is a frame of a dialog movie. The speaker on the left was only moving his hands, whereas the speaker on the right was only moving her head. The plots on the right are the basis functions of the two subspace independent component bases ($\mathbf{W}$), and the extracted component weights $\mathbf{m}_i(t)$.

$$\mathbf{f}_{mi}(t) = \mathbf{W} \cdot \mathbf{f}_m(t),$$

Where $\mathbf{W} \in \mathbb{R}^{m \times n}$ and $\mathbf{f}_{mi} \in \mathbb{R}^m$. The resulting set of time series $\mathbf{f}_{mi}(t)$ will contain the energy evolution of the set of the independent subspace components in the signal. To illustrate this consider the set of magnitude spectra in Figure 1.

By observing the resulting $\mathbf{f}_{mi}$ and $\mathbf{W}$ we can see that the structure of the scene has been compactly described. Component number one was tuned to the snare drum, component two to the bass drum, and component three to the cowbell. $\mathbf{f}_{mi}$ contains their temporal evolution, whereas $\mathbf{W}$ contains their spectral profile. Had we wished to separate the individual components, we could do by reconstructing the original spectrum using only one component at a time. To do so we set the remaining component weights to zero and invert the analysis:

$$\mathbf{f}_m^{(j)}(t) = \mathbf{a}_j \cdot \mathbf{f}_{mi}^{(j)}(t),$$

where $\mathbf{a}_j$ is the jth column of $\mathbf{A} = \mathbf{W}^+$ and parenthesized superscript denotes selection of the jth element. To obtain the time domain signal we modulate the amplitude spectrum by the phase $\mathbf{f}_a$ of the original signal and invert the frequency transformation. This technique has been described and demostrated in greater detail by Casey and Westner (2000), and Smaragdis (2001), and has been successfully used to extract multiple auditory sources off complex monophonic and stereo real-world auditory scenes.

4. VIDEO SUBSPACES

Using the same process we can estimate the independent components of video streams. We begin with a set of input frames $\mathbf{M}(t), \mathbf{M} \in \mathbb{R}^{m \times n}$, in which the element (i, j) of the matrix $\mathbf{M}(t)$ contains the intensity of the pixel at position i, j at time t. We reshape $\mathbf{M}(t)$ to a vector $\mathbf{m}(t)$, so that $\mathbf{m} \in \mathbb{R}^{mn}$ and process it to obtain:

$$\mathbf{m}_i(t) = \mathbf{W} \cdot \mathbf{m}(t),$$

where $\mathbf{m}_i \in \mathbb{R}^k$ are the component weights of the scene and $\mathbf{W} \in \mathbb{R}^{k \times n}$ the component bases by which to extract them. To visualize the bases in $\mathbf{W}$, we reshape each of its rows to the original size of the input frames. To illustrate this process consider the example in Figure 2. The input movie was composed of 165 frames of size 80×60, sampled at 30 frames per sec.

From the results we can see that the component bases in $\mathbf{W}$ represent the principal objects in the scene. The first component's basis is tuned to the head movements of the right speaker and the second is tuned to the arm and hand movement of the left speaker. Their temporal evolution $\mathbf{m}_i(t)$ reflects this, correctly showing the left speaker active at first, and the second speaker nodding three times afterward. As in the previous example we can reconstruct parts of the movie corresponding to the various components by inverting the process. Doing so provides us with a set of movies featuring only one of the extracted components.

5. AUDIO/VISUAL SUBSPACES

Traditionally audio/visual processing takes place in either domain separately, and results are often correlated afterward. In our work we will treat both the audio and the visual streams as one set of data, from which we will extract the subspace independent components. As our results show these components often correspond to objects in the scene that have simultaneous audio/visual presence.

For our examples, the soundtrack of the input video streams will be processed by a short time Fourier transform, so as to

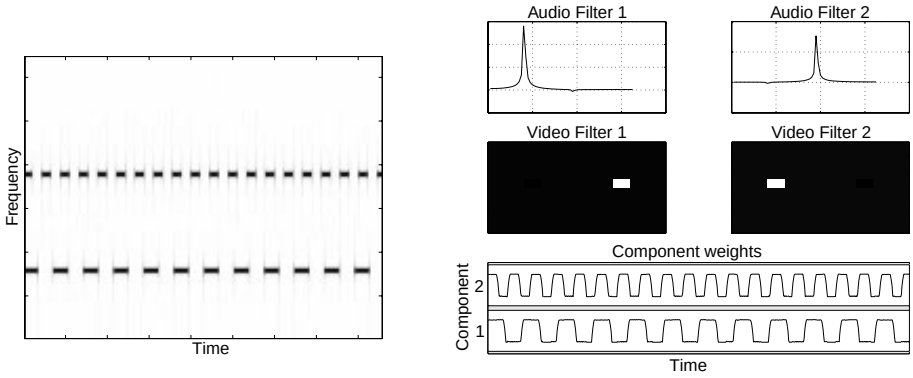

Fig. 3. Simple video example. The left plot is a spectrogram of the soundtrack, which consists of two periodically gated sine waves. The audio segment of the component bases $\mathbf{W}_a$ is shown at the top right plots, and video segment $\mathbf{W}_v$ at the middle right. The component weights $\mathbf{x}_i(t)$ are shown on the bottom right.

obtain a time-frequency representation $\mathbf{f}(t) \in \mathbb{C}^{n_a}$. From this we will extract its magnitude $\mathbf{f}_m = |\mathbf{f}|$, and phase $\mathbf{f}_a = \angle\mathbf{f}$. The video frames will be reshaped as vectors $\mathbf{m}(t) \in \mathbb{R}^{n_v}$. The two data sets will then be combined into one data vector:

$$\mathbf{x}(t) = \begin{bmatrix} \alpha \cdot \mathbf{f}(t) \\ \beta \cdot \mathbf{m}(t) \end{bmatrix},$$

so that $\mathbf{x}(t) \in \mathbb{R}^{n_v+n_a}$ is the result of the vertical concatenation of $\mathbf{f}(t)$ and $\mathbf{m}(t)$. In order to ensure that this concatenation is possible the audio data can be either sampled in synchrony with the video frame rate, or either domain can be appropriately resampled. We will then process the compound signal $\mathbf{x}(t)$ and extract its subspace independent components. The two scalars α and β are used for variance equalization. Since the first step of our operation is variance based, we can adjust these values to have the results influenced more by the video component or the audio component of the scene. A greater α would use more of the soundtrack to localize objects in time, whereas a greater β would do the inverse. There is no right setting for these numbers, for our simulations we picked one so that the overall variance of $\mathbf{f}(t)$ was approximately equal to the variance of $\mathbf{m}(t)$.

The bases $\mathbf{W}$ that we will extract will now exist in the audio/visual space. In order to understand the results and get a better idea of what these bases mean we can separate each of them to an audio and a video segment. Recall that the audio/visual analysis takes part on a compound matrix $\mathbf{x}(t)$. We can rewrite the analysis equation in a segmented form to show how the audio and video inputs are handled:

$$\mathbf{x}_i(t) = \mathbf{W} \cdot \mathbf{x}(t) \Rightarrow \begin{bmatrix} \mathbf{f}_i(t) \\ \mathbf{m}_i(t) \end{bmatrix} = [\mathbf{W}_a, \mathbf{W}_v] \cdot \begin{bmatrix} \mathbf{f}(t) \\ \mathbf{m}(t) \end{bmatrix},$$

where $\mathbf{f}_i, \mathbf{m}_i \in \mathbb{R}^k$ are the audio and video component weights, and $\mathbf{W}_a \in \mathbb{R}^{k \times n_a}$ and $\mathbf{W}_v \in \mathbb{R}^{k \times n_v}$ are the bases corresponding to the audio and video part of the input. Our estimation takes place using $\mathbf{W} = [\mathbf{W}_a, \mathbf{W}_v] \in \mathbb{R}^{k \times n_v + n_a}$, not on separate audio and visual bases. This results in components that have the same weight for both their audio and visual basis, forcing these two segments of the bases to be statistically related, therefore capturing the features of the same object.

To visualize and evaluate the results we will do the following. For the audio segment we will plot the rows of $\mathbf{W}_a$ which due to our representation of $\mathbf{f}(t)$ will be spectral profiles. Likewise, to visualize the video bases we will plot each row of $\mathbf{W}_v$ reshaped back to the size of the input frames. The component weights $\mathbf{x}_i(t)$ will indicate how present each audio/visual component is at any time.

5.1. A SIMPLE EXAMPLE

A very simple example on which we can build intuition is the following video scene. The soundtrack consists of two gated sine waves (Figure 3), and the video was two visual spots that were each blinking in synchrony with a corresponding sine. Putting the data through our procedure we obtain a set of component weights $\mathbf{x}_i(t)$, and a set of component bases $\mathbf{W}$. The results for this particular example are shown in Figure 3.

By observing the results, we can clearly see that the two audio bases have latched on the spectral profile of the two sines, and that the video bases have done the same for their visual counterparts. The component weights are correctly highlighting the components temporal evolution. Due to the common amplitude modulation of the audio and video signals, the pairs of audio/visual that were discovered highlight the cross-modal structure of the scene. Since each sine was statistically related to one of the visual configurations, our attempt to reduce common information between two com-

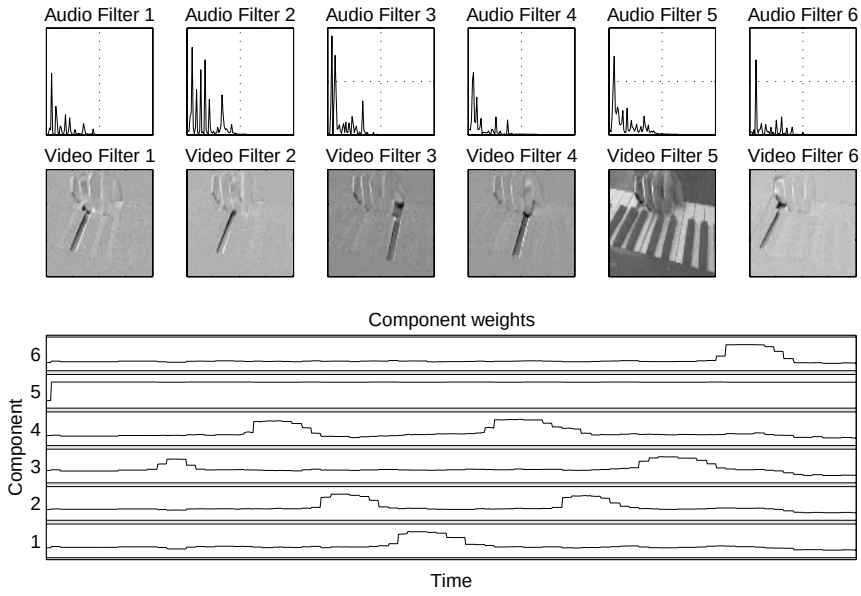

Fig. 4. Analysis results from the piano video. The audio segment of the component bases $\mathbf{W}_a$ is shown at the top plots, and video segment $\mathbf{W}_v$ at the middle. The component weights $\mathbf{x}_i(t)$ are shown on the bottom.

ponents resulted in this partitioning. Since in a video stream there is often some correlation between the visual and the auditory part we can form audio/visual objects using this method.

5.2. REAL-WORLD DATA

The above example was overly simple and was meant to be an intuitive introduction. This technique has been also applied to real-world video streams with satisfying results. Here we present an example of such a case. The input video was a shot of a hand playing notes on a piano keyboard, the movie was 85 frames sized at 120×160 pixels and recorded at 30 frames per sec, with a soundtrack sampled at $11025Hz$. The frequency transform was a short time Fourier transform of 128 points, with a hop of 64 samples with no windowing. Putting the data through our procedure we obtain the $\mathbf{x}_i(t)$, $\mathbf{W}_a$ and $\mathbf{W}_v$ shown in Figure 4.

From observation of the component bases we can represent source components of the scene. One component has a constant weight value and is the background term of the scene. The remaining component bases are tuned to the individual keys that have been pressed. This is evident from their visual part highlighting the key pressed, and their audio part roughly tuned to the harmonic series of the notes of each key. The component weights offer a temporal transcription of the piece played, providing the correct timing of the performance.

Using this decomposition is it possible to reconstruct the original input as is, albeit with the familiar compression artifacts that the PCA data reduction creates. Alternatively, given the highly semantic role of the extracted bases, we can tamper with the component weights so as to create a video of a hand playing different melodies on the piano.

6. DISCUSSION

This technique has been inspired by the works on redundancy reduction and sensory information processing (Barlow 1989). We are using computational techniques that have been used extensively for perceptual models (Linsker 1988, Bell and Sejnowski 1997, Smaragdis 2001), and that we think correlate well with what a perceptual system might do. Our hope is to link all this past work with a common conceptual and computational core, toward the development of a perceptual machine. In this paper we have limited our demonstrations to an audio/visual format, however this is a technique can work equally well on any time based modes which can carry sensory information. Such cases can include combinations or audio, video, radar/sonar, magnetic field sensing, and various other more exotic domains.

One of the major issues of this approach is that although it works well for scenes with static objects, it is not designed to work with dynamic scenes. An object moving across the field of vision for example cannot be tracked by only one component and it will be distributed among many visual bases. This will raise the number of components needed and it will weaken the association of the visual component with

say a more static sound. This can be remedied by having a moving window of analyses and keeping track of component changes from frame to frame. This is an issue beyond the scope of this paper that we intend to address in future publications.

7. CONCLUSIONS

We have presented a methodology to extract independent objects from complex multi-modal scenes. The main advantage of our approach is that the operation takes place on a fused data set, instead of individual processing of every mode. We have demonstrated the usefulness of this technique on various audio/visual data showing that the presence of objects in both domains can be extracted as a feature. We also presented some of the research directions that this approach points to, issues we look forward to addressing in the near future. This is by no means a complete scene analysis system; we hope however that it will serve as a stepping stone for multi-modal analysis research using independence criteria.

References

Amari S-I., A. Cichocki and H.H. Yang (2000). A New Learning Algorithm for Blind Signal Separation. In D.S. Touretzky and M.C. Mozer and M.E. Hasselmo (eds.), *Advances in Neural Information Processing Systems 8*, MIT Press, Cambridge MA.

Barlow, H.B. (1989) Unsupervised learning. In *Neural Computation* **1** pp. 295-311. MIT Press, Cambridge MA.

Bell, A. J. and Sejnowski, T. J. (1997). The índependent componentsóf natural scenes are edge filters. In *Vision Research*, **37**(23) pp. 3327-3338.

Casey, M., and Westner, W., (2000) Separation of Mixed Audio Sources by Independent Subspace Analysis, in *Proceedings of the International Computer Music Conference*, Berlin August 2000.

Casey, M. (2001). Reduced-Rank Spectra and Minimum Entropy Priors for Generalized Sound Recognition. In *Proceedings of the Workshop on Consistent and Reliable Cues for Sound Analysis*, EUROSPEECH 2001, Aalborg, Denmark.

Fisher, J.W. III, T. Darrell, W.T. Freeman, and P. Viola. (2000) Learning joint statistical models for audio-visual fusion and segregation. In T. K. Leen, T. G. Dietterich, and V. Tresp, (eds.), *Advances in Neural Information Processing Systems 13*. MIT Press, Cambridge MA.

Hershey, J. and J. Movellan. (2001) Using audio-visual synchrony to locate sounds. In S.A. Solla, T.K. Leen, and K-R.Müller, (eds.), *Advances in Neural Information Processing Systems 12*. MIT Press, Cambridge MA.

Hyvärinen, A. (1999) Survey on independent component analysis. In *Neural Computing Surveys*, **2**, pp. 94-128.

Linsker, R. (1988). Self-organization in a perceptual network. In *Computer*, **21**.

Partridge, M.G. and R.A. Calvo. (1998) Fast dimensionality reduction and simple PCA. In *Intelligent Data Analysis*, **2**(3).

Roweis, S. (1997) EM Algorithms for PCA and SPCA. In M.I. Jordan,M. Kearns andS. Solla (eds.), *Neural Information Processing Systems 10*. MIT Press, Cambridge MA.

Slaney, M. and M. Covell. (2000) Facesync: A linear operator for measuring synchronization of video facial images and audio tracks. In T. K. Leen, T.G. Dietterich, and V. Tresp, (eds.), *Advances in Neural Information Processing Systems 13*. MIT Press, Cambridge MA.

Smaragdis, P. (2001) Redundancy reduction for computational audition, a unifying approach. *Doctoral dissertation*, MAS department. Massachusetts Institute of Technology, Cambridge MA.