Audio source separation using sparse representations

Andrew Nesbit, Maria Jafari, Emmanuel Vincent, Mark Plumbley

Audio Source Separation using Sparse Representations

Andrew Nesbit[1], Maria G. Jafari[1], Emmanuel Vincent[2] and Mark D. Plumbley[1]
[1]*Queen Mary University of London, United Kingdom*
[2]*INRIA, Centre Inria Rennes – Bretagne Atlantique, France*

ABSTRACT

We address the problem of audio source separation, namely, the recovery of audio signals from recordings of mixtures of those signals. The sparse component analysis framework is a powerful method for achieving this. Sparse orthogonal transforms, in which only few transform coefficients differ significantly from zero, are developed; once the signal has been transformed, energy is apportioned from each transform coefficient to each estimated source, and, finally, the signal is reconstructed using the inverse transform. The overriding aim of this chapter is to demonstrate how this framework, as exemplified here by two different decomposition methods which adapt to the signal to represent it sparsely, can be used to solve different problems in different mixing scenarios.

To address the instantaneous (neither delays nor echoes) and underdetermined (more sources than mixtures) mixing model, a lapped orthogonal transform is adapted to the signal by selecting a basis from a library of predetermined bases. This method is highly related to the windowing methods used in the MPEG audio coding framework. In considering the anechoic (delays but no echoes) and determined (equal number of sources and mixtures) mixing case, a greedy adaptive transform is used based on orthogonal basis functions that are learned from the observed data, instead of being selected from a predetermined library of bases. This is found to encode the signal characteristics, by introducing a feedback system between the bases and the observed data. Experiments on mixtures of speech and music signals demonstrate that these methods give good signal approximations and separation performance, and indicate promising directions for future research.

KEYWORDS

Source separation, sparsity, lapped orthogonal transform, dictionary learning

INTRODUCTION

The problem of *audio source separation* involves recovering individual audio *source* signals from a number of observed *mixtures* of those simultaneous audio sources. The observations are often made using microphones in a live recording scenario, or can be taken, for example, as the left and right channels of a stereo audio recording. This is a very challenging and interesting problem, as evidenced by the multitude of techniques and principles used in attempts to solve it. Applications of audio source separation and its underlying principles include audio remixing (Woodruff, Pardo, & Dannenberg, 2006), noise compensation for speech recognition (Benaroya, Bimbot, Gravier, & Gribonval, 2003), and transcription of music (Bertin, Badeau, & Vincent, 2009). The

choice of technique used is largely governed by certain constraints on the sources and the mixing process. These include the number of mixture channels, number of sources, nature of the sources (e.g., speech, harmonically related musical tracks, or environmental noise), nature of the mixing process (e.g., live, studio, using microphones, echoic, anechoic, etc), and whether or not the sources are moving in space.

The type of mixing process that generates the observed sources is crucially important for the solution of the separation problem. Typically, we distinguish between *instantaneous*, *anechoic* and *convolutive* mixing. These correspond respectively to the case where the sources are mixed without any delays or echoes, when delays only are present, and when both echoes and delays complicate the mixing. Source separation for the instantaneous mixing case is generally well understood, and satisfactory algorithms have been proposed for a variety of applications. Conversely, the anechoic and convolutive cases present bigger challenges, although they often correspond to more realistic scenarios, particularly for audio mixtures recorded in real environments. Algorithms for audio source separation can also be classified as *blind* or *semi-blind*, depending on whether a priori information regarding the mixing. Blind methods assume that nothing is known about the mixing, and the separation must be carried out based only on the observed signals. Semi-blind methods incorporate a priori knowledge of the mixing process (Jafari et al., 2006) or the sources' positions (Hesse & James, 2006).

The number of mixture channels relative to the number of sources is also very important in audio source separation. The problem can be *overdetermined*, when more mixtures than sources exist, *determined*, with equal number of mixtures and sources, and *underdetermined*, when we have more sources than mixtures. Since the overdetermined problem can be reduced to a determined problem (Winter, Sawada, & Makino, 2006), only the determined and underdetermined situations have to be considered. The latter is particularly challenging, and conventional separation methods alone cannot be applied. An overview of established, statistically motivated, model-based separation approaches are presented elsewhere in this book (Vincent et al., 2010), which can also serve as an introduction to audio source separation for the non-expert reader. Another useful introduction is the review article by O'Grady, Pearlmutter, & Rickard (2005).

A widely used class of model-based source separation algorithms that exploits the sparsity of the source signals in some time-frequency (TF) transform domain is *sparse component analysis* (SCA). It entails transforming the signals into a domain in which they are *sparse*, estimating the mixing matrix from the transform coefficients, estimating the source representations, and finally, inverting the transform representation of the estimated sources. A *sparse* signal representation is one which conveys the information within the signal using only a few elementary components, denoted as *atoms*, which are selected from a *dictionary* to form a sparse signal decomposition. This often helps to uncover hidden structure in the analysed signal by characterising the original signal using only a small number of large coefficients. The short-time Fourier transform (STFT), for instance, decomposes a time-domain signal using a dictionary of windowed Fourier (complex exponential) atoms, and will reveal the frequency content of the signal even though this might not be evident from the temporal waveform. Fixed-basis transforms such as the STFT or the fixed-basis modified discrete cosine transform (MDCT) are often used in audio (Ravelli & Daudet, 2006). They have the advantageous property of being easily invertible and providing a unique signal representations.

However, transforms based on a fixed dictionary fail to match all signal features present, such as fast-varying transients and slower components (Daudet & Torrésani, 2002), and they are often

based on a rigid structure that prevents the compact representation of some signals (Davis, 1994). In response to this, *redundant* or *overcomplete* dictionaries are often used, where the number of atoms is greater than the dimensionality of the signal space. An alternative approach is to construct an orthogonal dictionary directly from the observed data, so that it captures features that are exactly relevant to the analysed signal, and introduces a feedback system between the signal and the dictionary (Górecki & Domanski, 2007). Examples of dictionary learning algorithms include independent component analysis (ICA) (Abdallah & Plumbley, 2004) and K-SVD (Aharon, Elad & Bruckstein, 2006).

In this chapter, SCA for audio source separation is considered under two mixing scenarios, and in each case, a different sparse decomposition technique for SCA is used. The instantaneous, underdetermined problem is addressed using a class of adaptive *lapped orthogonal transforms*, which select from a dictionary of localised cosine basis functions, those which yield an orthogonal, linear transform. There are many possible orthogonal bases to choose from (i.e., the *library* of bases for the entire time-domain signal is large), due to the fact that there are many ways in which the signal may be segmented in time by overlapping windows. In the conceptual SCA framework, once the mixture signals have been transformed using this method, the sources are estimated by assigning energy from each of the transform domain coefficients to the source estimates (it is assumed here that the mixing matrix is either known, or has been estimated, i.e., the semi-blind case). Finally, the time-domain source estimates are recovered by applying the inverse transform.

We then direct our attention to the anechoic, determined audio source separation problem. We present an orthogonal transform that is used to sparsify the data in the first step of the SCA procedure. The transform is a greedy algorithm which adaptively learns a dictionary from data blocks taken from the observed signal. This maximises the ℓ^2-norm of the data, while minimizing its ℓ^1-norm, hence resulting in a sparse representation for the signal. The transform is forced to be orthogonal by removing all the components lying in the direction of a particular vector (corresponding to the selected data frame) at each iteration. Since the atoms are extracted from the observed data, the greedy adaptive dictionary (GAD) algorithm finds atoms that are directly relevant to the data being analysed. Thus, we apply the transform to the audio source separation problem, within the SCA framework, and compare its performance to that of ICA and K-SVD within the same framework, as presented in (Jafari et al., 2008), and (Jafari & Plumbley, 2008).

SOURCE SEPARATION

The problem of source separation arises when two or more signals (sources) are mixed by passage through an unknown medium, and the objective of source separation algorithms is to recover the original sources from only the available mixtures. We consider the separation of J sources, from an equal number of mixtures, generated according to the anechoic mixing model, defined as (Saab et al, 2005)

$$x_i(n) = \sum_{j=1}^{J} a_{i,j} s_j(n - \tau_{i,j}), \;\; i = 1, \ldots, J \tag{1.1}$$

where $x_i(n)$ is the observed mixture, and $a_{i,j}$, and $\tau_{i,j}$ are the real-valued attenuation coefficients ($a_{i,j} > 0$) and time delays relating to the path from source j to mixture i.

In this chapter, we also consider the underdetermined and instantaneous case, where the problem becomes one of estimating $J > 2$ sources when the number of mixture channels is two. Thus, equation (1.1) becomes

$$x_i(n) = \sum_{j=1}^{J} a_{i,j} s_j(n), \quad i = 1, 2 \tag{1.2}$$

which in matrix can be written as

$$\mathbf{x}(n) = \mathbf{A}\mathbf{s}(n) \tag{1.3}$$

where $\mathbf{x}(n) = \begin{bmatrix} x_1(n) & x_2(n) \end{bmatrix}^T$ and $\mathbf{s}(n) = \begin{bmatrix} s_1(n) & \cdots & s_J(n) \end{bmatrix}^T$ are the mixture and source vectors respectively, $\mathbf{A} = [a_{i,j}]_{2\times J}$ is the instantaneous mixing matrix with real-valued entries, and the discrete-time index ranges as $0 \le n < N$, where N is the length of the signal.

Sparse Component Analysis

Sparse component analysis methods, based on the assumption that the source signals are sparse in some transform domain, are frequently applied to source separation, since working with signal representations that do not overlap simplifies the separation problem (Gribonval & Lesage, 2006). Moreover, many approaches to audio source separation based on, for instance, ICA, assume that the number of mixture channels is equal to the number of sources. As this condition is not always satisfied, we look for a different solution. Generally, the SCA procedure is comprised of the following four conceptual stages.

Firstly, the mixture signals are transformed so that they lie in a domain in which they are sparse; this typically entails the use of orthogonal transforms such as the wavelet transform or MDCT or nonorthogonal transforms such as the STFT, but learned dictionaries are acquiring popularity. Examples of sparsifying transforms based on learned dictionaries are ICA (Jafari et al., 2008), and K-SVD (Aharon, Elad & Bruckstein, 2006). Secondly, the mixing matrix is estimated, typically by clustering coefficients in the sparse transform domain. Thirdly, the sources in the transform domain are estimated by apportioning energy from each source coefficient to the source estimates according to their mixing parameters determined in the previous stage. Finally, the sources are reconstructed by applying the inverse transform.

It should be noted that these four stages are used for conceptualising the SCA procedure, but in practical implementations the various stages might have varying dependencies upon each other.

SPARSE COMPONENT ANALYSIS BASED ON LAPPED ORTHOGONAL TRANSFORMS

Adaptive *lapped orthogonal transforms* (LOTs), which adapt to the time-varying signal structures in the TF domain, have the potential to yield sparser representations and superior performance compared to commonly used transforms such as the STFT or fixed-basis MDCT (Nesbit, Vincent & Plumbley, 2009). This section describes their construction and the way they naturally fit within the SCA framework.

Sparsifying Step: Adaptive Lapped Orthogonal Transforms

Adapting a LOT to the mixture channels $x_i(n)$ entails forming an appropriate partition of their domain $[0, \ldots, N-1]$, that is, a collection of K strictly increasing points n_k such that

$$0 = n_0 < n_1 < \cdots < n_k < \cdots < n_{K-1} = N - 1. \tag{2.1}$$

This segments the domain of $x_i(n)$ into adjacent intervals $\mathrm{I}_k = [n_k, n_{k+1} - 1]$ which should be relatively long over durations which require good frequency resolution, and relatively short over durations requiring good time resolution. It is well known that simply using rectangular windows to divide the signal along its time axis at these points leads to the familiar 'ringing' artifacts at the window boundaries. However, by using a differentiable window of compact support, which does not have such objectionable discontinuities, these border artifacts can be alleviated. In the context of adaptive LOTs, this means that any two adjacent windows will overlap by a certain amount. To specify this amount of overlap, augment the aforementioned partition by associating with each n_k a *bell parameter* η_k, so that the partition becomes a finite set of ordered pairs $\lambda = \{(n_k, \eta_k)\}$. Figure 1 is a schematic representation of the way $x_i(n)$ is windowed with windows $\beta_k^\lambda(n)$ according to some particular partition λ. Each window $\beta_k^\lambda(n)$ is supported in $[n_k - \eta_k, n_{k+1} + \eta_{k+1} - 1]$, thus partly overlapping with its immediately adjacent windows $\beta_{k-1}^\lambda(n)$ and $\beta_{k+1}^\lambda(n)$ by η_k and η_{k+1} points respectively.

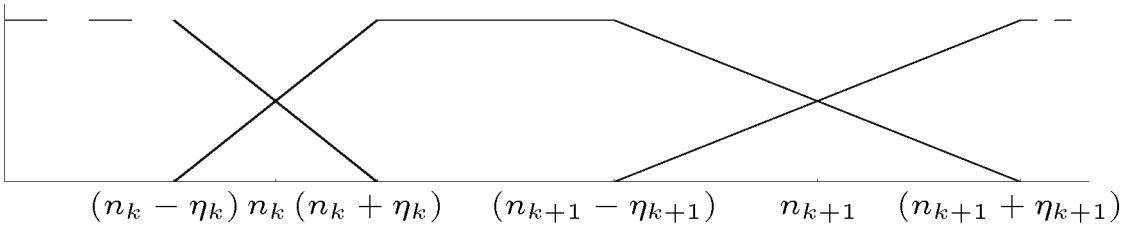

Figure 1. Schematic representation of window denoted by β_k^λ. The partition points are given by n_k and n_{k+1} and the bell parameters by η_k and η_{k+1}.

These bell parameters η_k are thus subject to the constraint

$$n_{k+1} - n_k \geq \eta_{k+1} + \eta_k. \tag{2.2}$$

Note that $\eta_0 = \eta_{K-1} = 0$ and appropriate border modifications need to be made for this special case (Mallat, 1999). For every partition λ we form its associated windows according to the following function:

$$\beta_k^\lambda(n) = \begin{cases} r\left(\dfrac{n - \left(n_k - \frac{1}{2}\right)}{\eta_k}\right) & \text{if } n_k - \eta_k \leq n < n_k + \eta_k, \\ 1 & \text{if } n_k + \eta_k \leq n < n_{k+1} - \eta_{k+1}, \\ r\left(\dfrac{\left(n_{k+1} - \frac{1}{2}\right) - n}{\eta_{k+1}}\right) & \text{if } n_{k+1} - \eta_{k+1} \leq n < n_{k+1} + \eta_{k+1}, \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

where the *bell function* $r(t)$ satisfies $r^2(t)+r^2(-t)=1$ for $-1 \le t \le 1$, $r(t)=0$ for t < -1 and r(t) = 1 for $t>1$, where t is real-valued and satisfies various differentiability properties (Mallat, 1999). The bell parameters η_k and η_{k+1} determine how quickly the window monotonically rises on its left side and monotonically falls on its right side. Although there are many possible bell functions which satisfy these constraints, in practice we use a sine bell; refer to Mallat (1999) for its definition.

The local cosine basis associated with the interval I_k is then given by modulating $\beta_k^{\lambda}(n)$ by functions from a cosine-IV basis as follows:

$$\mathrm{B}_k^{\lambda} = \left\{ \beta_k^{\lambda} \sqrt{\frac{2}{n_{k+1}-n_k}} \cos\left[\pi\left(m' + \frac{1}{2} \right) \frac{n-(n_k-\frac{1}{2})}{n_{k+1}-n_k} \right] \right\} \tag{2.4}$$

where $m' \in [0, n_{k+1}-n_k-1]$ is the discrete cosine frequency index. This defines the basis B^{λ} for the orthogonal LOT, adapted to the partition λ, for the space of signals of length N:

$$B^{\lambda} = \bigcup_{k=0}^{K-1} \mathrm{B}_k^{\lambda}. \tag{2.5}$$

Since our aim is to find the *best orthogonal basis* (BOB) of all possibilities, we will consider all admissible partitions $\lambda \in \Lambda$ subject to some relatively lenient constraints, each of which determines a different orthogonal basis. Thus we obtain a *library* of bases for this space of signals of length N:

$$\mathrm{L} = \bigcup_{\lambda \in \Lambda} B^{\lambda}. \tag{2.6}$$

As such, the union of all bases in the library constitutes an overcomplete dictionary from which we obtain our sparse representation. Each admissible basis $B^{\lambda} \in \mathrm{L}$ has an associated *cost* of representing a particular signal in that basis, given by an additive cost function. Finding the BOB amounts to minimizing this cost, which, ideally, should maximize the separation performance criterion. Examples of suitable cost functions are the ℓ^1-norm (useful in blind and semi-blind cases), and the *oracle* benchmarking criterion (useful for algorithm evaluation); each of these cost functions is defined and described later in this section, because, in this particular framework for SCA, the computation of the BOB is intimately tied in with estimating the source coefficients.

Given any additive cost function, the BOB is determined by applying one of several partitioning schemes and associated algorithms based on dynamic programming (Huang, Pollak, Bouman, & Do, 2006; Xiong, Ramchandran, Herley, & Orchard, 1997). In previous work (Nesbit, Plumbley, & Vincent, 2009) a *flexible segmentation* (FS) scheme was described, which admits all possible partitions λ with some 'resolution' L, so that if the signal length N is an integral multiple of L, then each partition point can be written as $n_k = cL$ for $c \ge 0$, and where η_k is subject only to the condition (2.2). Provided that both L and N are powers of two, any library L admitted by FS is a superset of the library admitted by the less flexible, *dyadic* partitioning scheme, in which intervals are recursively formed by 'splitting' already existing intervals at their middles (Mallat, 1999). Although FS gives excellent separation results, its library L is very large due to a combinatorial explosion between the range of allowed interval lengths, interval onsets and bell parameters. Therefore, its computation time is impractically high. As we wish to maintain flexible partitioning on the domain of the signal, yet decrease the time required for estimation of $s(n)$, we are motivated by the corresponding ideas from the MPEG-4

AAC audio coding framework (ISO, 2005) and introduce the following `MPEG-like' partitioning schemes:

- **Long-Short (LS).** We restrict the range of allowable partitions to admit intervals I_k of only two lengths, that is, a *long interval* of length L_L and a *short interval* of length $L_S = L$, where L_L is an integral multiple of L_S, and we admit only bell parameters such that $2\eta_k \in \{L_L, L_S\}$. Apart from this restriction of interval lengths and bell parameters, there are no additional constraints, and LS is otherwise the same as FS.
- **Window Shapes (WS).** This is equivalent to LS with the additional constraint that if I_k is long, then at most one of η_k and η_{k+1} is short. In other words, the four different window shapes admitted (compared to five in LS) correspond to a long window ($2\eta_k = 2\eta_{k+1} = L_L$), a short window ($2\eta_k = 2\eta_{k+1} = L_S$), a long-short *transition window* ($2\eta_k = L_L, 2\eta_{k+1} = L_S$), and a short-long ($2\eta_k = L_S, 2\eta_{k+1} = L_L$) transition window in the MPEG-4 framework.
- **Onset Times (OT).** This is equivalent to LS with the additional constraint if any interval I_k is long, then n_k must satisfy $n_k = cL_L$ for some $c = 0, \ldots, \frac{N}{L_L} - 1$.
- **WS/OT.** This scheme imposes both the WS and OT constraints simultaneously.
- **WS/OT/Successive Transitions (WS/OT/ST).** This scheme imposes the WS/OT constraints in addition to disallowing adjacent transition windows, i.e., a transition window must be adjacent to a long window and a short window. This implements the windowing scheme used by MPEG-4, apart from the choice of the bell function $r(t)$.

Even though the sizes of the libraries become significantly smaller as we impose more constraints, we expect that the MPEG-like partitioning schemes are nevertheless sufficiently flexible so that benefits gained in computation time will outweigh any decrease in separation performance.

Estimating the Mixing Matrix

In the SCA framework, the mixing matrix, $\mathbf{A}$, is typically estimated by clustering TF coefficients. For example, the method of Bofill (2008) applies a linear, sparsifying transform to each of the mixture channels and selects only those coefficients whose spatial 'direction', given by the ratio of the magnitudes of the two mixture channels at each point in the transform domain, remain constant over time.[1] The idea is that such coefficients are more likely to belong to a single source, and can then be clustered using a weighted histogram to estimate $\mathbf{A}$. However, as the description of the particular source separation framework described in this section is more concerned with evaluating the adaptive nature of LOTs, only semi-blind experiments will be performed, and it is assumed that $\mathbf{A}$ is either already known or has been estimated.

[1] Although Bofill (2008) specifically uses the STFT, the mixing matrix estimation algorithm can readily be adapted to use the LOT framework.

Estimating the Sources and Inverting the Transform

Let $\tilde{\mathbf{x}}(m) = \begin{bmatrix} \tilde{x}_1(m) & \tilde{x}_2(m) \end{bmatrix}^T$ be the vector of a linear, orthogonal, TF transform of each channel of $\mathbf{x}(n)$, and let $\tilde{\mathbf{s}}(m) = \begin{bmatrix} \tilde{s}_1(m) & \cdots & \tilde{s}_J(m) \end{bmatrix}^T$ be the transform of $\mathbf{s}(n)$, where m indexes the coefficients in the TF domain and $0 \le m < N$. In the present case, the transform used is an LOT, as described above.

In the semi-blind case, by assuming that the source coefficients in the transform domain follow a Laplacian distribution independently and identically for all j and m, the maximum a posteriori estimation of $\mathbf{s}(n)$ is equivalent to minimising the ℓ^1-norm of the sources coefficients given by the following:

$$C\left(\hat{\tilde{\mathbf{S}}}\right) = \sum_{m=0}^{N-1} \sum_{j=1}^{J} \left| \hat{\tilde{s}}_j(m) \right| \tag{2.7}$$

where $\hat{\tilde{\mathbf{S}}}$ is a $J \times N$ matrix of estimated source coefficients in the transform domain (Bofill & Zibulevsky, 2001). The primary implication of this, in the present case where there are two mixture channels, is that exactly two sources are assumed to be active at each m; incidentally, this gives better performance than the simpler *binary masking* case which allows only one active source (Bofill & Zibulevsky, 2001; Yilmaz & Rickard, 2004). Furthermore, it is known that minimising the ℓ^1-norm promotes sparsity of the estimated coefficients; as such, it is an appropriate estimation criterion for this implementation of SCA (Zibulevsky & Pearlmutter, 2001).

The set of both source indices contributing to $\tilde{\mathbf{x}}(m)$ is denoted by $\mathrm{J}_m = \{ j : \tilde{s}_j(m) \neq 0 \}$, and is called the *local activity pattern* at m. Given a particular ϑ_m, (1.3) then reduces to a determined system:

$$\tilde{\mathbf{x}}(m) = \mathbf{A}_{\mathrm{J}_m} \tilde{\mathbf{s}}_{\mathrm{J}_m}(m) \tag{2.8}$$

where $\mathbf{A}_{\mathrm{J}_m}$ is the 2×2 submatrix of $\mathbf{A}$ formed by taking columns $\mathbf{A}_j$, and $\tilde{\mathbf{s}}_{\mathrm{J}_m}(m)$ is the subvector of $\tilde{\mathbf{s}}(m)$ formed by taking elements $\tilde{s}_j(m)$, whenever $j \in \vartheta_m$. As such the activity patterns need to be estimated according to

$$\mathrm{J}_m^{\mathrm{sb}} = \arg\min_{\mathrm{J}_m} \sum_{j=1}^{J} \left| \hat{\tilde{s}}_j(m) \right| \tag{2.9}$$

which depends implicitly on the following:

$$\begin{cases} \hat{\tilde{s}}_j(m) = 0 & \text{if } j \notin \mathrm{J}_m, \\ \hat{\tilde{\mathbf{s}}}_{\mathrm{J}_m}(m) = \mathbf{A}_{\mathrm{J}_m}^{-1} \tilde{\mathbf{x}}(m) & \text{otherwise} \end{cases} \tag{2.10}$$

where $\mathbf{A}_{\mathrm{J}_m}^{-1}$ is the matrix inverse of $\mathbf{A}_{\mathrm{J}_m}$ (Gribonval, 2003). Finally, the estimated source vectors in the time domain $\hat{\mathbf{s}}(n)$ are recovered by using the inverse transform.

Experiments and Results

We performed two sets of experiments to test our algorithms. Performance is measured through the *signal to distortion ratio* (SDR), which is defined as the following in this section (Vincent, Gribonval, & Plumbley, 2007):

$$\text{SDR [dB]} = 10\log_{10}\frac{\sum_{n=0}^{N-1}\sum_{j=1}^{J}\left(s_j(n)\right)^2}{\sum_{n=0}^{N-1}\sum_{j=1}^{J}\left(\hat{s}_j(n)-s_j(n)\right)^2}. \tag{2.11}$$

This particular (yet standard) definition of SDR is chosen for this section because *oracle* source estimation depends on it (see below).

In the first set of experiments, we applied our methods to twenty mixtures in total (ten music mixtures and ten speech mixtures), where each mixture had $J = 3$ sources at a sampling rate of 16 kHz, with a resolution of 16 bits per sample, and length of $N = 2^{18}$ (approx. 11 s). The sources were mixed according to following mixing matrix:

$$\mathbf{A} = \begin{bmatrix} 0.21 & 0.95 & 0.64 \\ 0.98 & 0.32 & 0.77 \end{bmatrix}. \tag{2.12}$$

For each mixture, we performed semi-blind estimations of $s(n)$ for each of the LS, WS, OT, WS/OT and WS/OT/ST partitioning schemes, with long intervals $L_L = 2^c$, where $c \in \{8,\ldots,11\}$ (12 ms to 93 ms), and short intervals $L_S = 2^c$ where $c \in \{4,\ldots,9\}$ (0.73 ms to 23 ms). We exclude all long-short combinations where $L_L \le L_S$. Results are presented in Table 1, where each entry is the average over the twenty different mixtures corresponding to a particular transform scheme with given block lengths. We also compare the MPEG-like schemes to the baseline fixed basis (FB) transform (where $L_L = L_S$ and $2\eta_k = L_L$ for all k) and find that the maximum average SDR is 12.06 dB at $L_L = L_s = 2^{10}$.

For the results in Table 1, the best average SDR is approximately 12.3 dB for each transform scheme. Admittedly, there is no significant difference in performance between the different segmentation schemes, which indicates that more flexible schemes, e.g., LS, do not offer enough of a performance improvement to justify their increased computational burden. Previous results demonstrated oracle performance of approximately 25 dB, but the differences between the two cases are not surprising; in contrast to the semi-blind estimation criterion (ℓ^1-norm), the oracle estimation criterion is optimised for the performance measurement criterion (SDR). The greatest variability in average SDR occurs with changing the long interval length L_L. The SDR improvements in the demonstrated range of 1-2 dB may be significant in high fidelity applications. In each case in Table 1, the best average SDR is achieved at the greatest length for the short intervals ($L_S = 2^9$).

Scheme	L_L	L_S					
		2^4	2^5	2^6	2^7	2^8	2^9
LS	2^8	10.45	10.50	10.51	10.55	-	-
	2^9	11.72	11.71	11.72	11.72	11.79	-
	2^{10}	12.14	12.10	12.19	12.16	12.23	12.29
	2^{11}	11.70	11.59	11.73	11.77	11.92	**12.34**
WS	2^8	10.45	10.51	10.52	10.55	-	-
	2^9	11.76	11.71	11.74	11.74	11.80	-
	2^{10}	12.16	12.14	12.18	12.16	12.23	**12.28**
	2^{11}	11.62	11.66	11.69	11.75	11.91	12.22
OT	2^8	10.68	10.66	10.65	10.64	-	-
	2^9	11.83	11.83	11.85	11.85	11.83	-
	2^{10}	12.07	12.07	12.07	12.06	12.15	12.19
	2^{11}	11.65	11.56	11.60	11.61	11.86	**12.29**
WS/OT	2^8	10.68	10.67	10.66	10.64	-	-
	2^9	11.84	11.83	11.85	11.85	11.83	-
	2^{10}	12.07	12.07	12.08	12.08	12.16	12.20
	2^{11}	11.62	11.56	11.59	11.61	11.83	**12.29**
WS/OT/ST	2^8	10.69	10.68	10.67	10.64	-	-
	2^9	11.84	11.84	11.85	11.85	11.85	-
	2^{10}	12.05	12.04	12.06	12.08	12.16	12.21
	2^{11}	11.57	11.52	11.53	11.55	11.77	**12.28**

Table 1. Average results for MPEG-like transforms for semi-blind separation on music and speech mixtures. Long and short interval sizes are given by L_L and L_S respectively, and LS, WS, OT, WS/OT and WS/OT/ST correspond to each of the MPEG-like partitioning schemes. The best average SDR for each scheme is highlighted in bold.

For the second set of experiments, we indicate the performance achievable on particular types of mixtures. We applied the best transform scheme as determined by Table 1 (that is, LS) to each instantaneous mixture in the *dev1* data set of the *Signal Separation Evaluation Campaign* (SiSEC 2008) [2] and present the results in Table 2. Also shown in the Table are *oracle estimation* results, where the L_L and L_S which give best results were determined in previous work (Nesbit, Vincent, & Plumbley, 2009). The aim of oracle estimation is to determine those J_m and $B^\lambda \in \mathrm{L}$ which give the best possible separation performance for every TF index m. This allows us to judge the difficulty of estimating the sources $s(n)$ from a given mixture $x(n)$, and to gain insight into the upper performance bounds of our class of separation algorithms (Vincent, Gribonval, & Plumbley, 2007). Oracle results are computed by jointly determining the local activity patterns J_m and the best orthogonal basis $B^\lambda \in \mathrm{L}$ which maximise the SDR. As oracle estimation depends on knowing the reference source signals $s(n)$ and the mixing matrix

[2] Available online at http://sisec.wiki.irisa.fr/tiki-index.php.

A it is intended to be used for algorithm evaluation rather than for practical (semi-)blind separation applications.

Mixture	J	Semi-blind			Oracle		
		L_L	L_S	Avg SDR [dB]	L_L	L_S	Avg SDR [dB]
3 Female Speakers	3	2^9	2^5	10.35	2^{10}	2^4	24.09
4 Female Speakers	4	2^{11}	2^9	7.04	2^{10}	2^4	18.61
3 Male Speakers	3	2^9	2^9	8.41	2^{10}	2^4	18.56
4 Male Speakers	4	2^{10}	2^9	5.62	2^{10}	2^4	14.37
Music with no drums	3	2^{10}	2^7	16.33	2^{10}	2^4	34.26
Music with drums	3	2^9	2^4	11.95	2^{10}	2^4	28.06

Table 2. Results for LS scheme for semi-blind and oracle separation on SiSEC 2008 data.

In contrast to Table 1, Table 2 shows individual, rather than average, results. Previous oracle results for the LS and WS schemes show that the best average SDR was obtained at the least length for the short intervals ($L_S = 2^4$), where we suggested that a library which allows fine-grained placement of the long windows improves performance (Nesbit, Vincent, & Plumbley, 2009). The current ℓ^1-norm criterion does not lead to such a basis being selected, but a semi-blind criterion which admits such fine-grained placement will be a good step towards closing the performance gap between semi-blind and oracle performance.

SPARSE COMPONENT ANALYSIS BASED ON A LEARNED DICTIONARY

In this section we consider the problem of audio source separation when a set of anechoic mixtures generated by the same number of sources are observed. The problem is again considered within the SCA framework, where the dictionary used for the sparsifying transform is now learned from the observed data, rather than selected from a fixed set of pre-existing bases.

Dictionaries that are inferred from the training data have the advantage of being more finely tuned to the data itself. Their main disadvantage is the limit on the size of the dictionary that can be trained, due to the complexity of the training algorithms, and the limit on the size of the signal that can be analysed (Rubinstein, Zibulevsky & Elad, 2009). Two pre-existing dictionary learning methods are the ICA and K-SVD algorithms. The reader is referred to (Abdallah & Plumbley, 2004) and (Aharon, Elad & Bruckstein, 2006), respectively, for more details on these techniques. They were applied to the audio separation problem in (Jafari et al., 2008), and (Jafari & Plumbley, 2008) respectively, and therefore will be used later in comparisons with the separation approach based on the *greedy adaptive dictionary* (GAD) algorithm presented here. In this section, we also summarise the other SCA steps necessary to separate the sources.

Sparsifying Step: Learning the Dictionary

We consider a data vector, $\mathbf{x}(n)$, which contains two observed signals. The GAD algorithm is used to find a basis set that encodes both spatial and temporal correlations in the observed data. To do this, the data vector $\mathbf{x}(n)$ is reshaped into a matrix $\mathbf{X}$, such that sample pairs from the

former are stacked to form the columns of the latter. Reshaping the input in this fashion allows correlations between microphones and across time to be modelled.

Thus, $\mathbf{x}(n)$ is reshaped into a $P \times Q$ matrix, where successive blocks of $P/2$ sample pairs are taken from the mixture vector, with an overlap of T samples. Then the (p,q)-th element of the matrix $\mathbf{X}$ is given by

$$[\mathbf{X}]_{p,q} = \begin{cases} x_1\left((q-1)Z + (p+1)/2\right) & : p \text{ odd} \\ x_2\left((q-1)Z + p/2\right) & : p \text{ even} \end{cases}$$

where $Z = P/2 - T$, and $p \in \{0,\ldots,P-1\}$, and $q \in \{0,\ldots,Q-1\}$.

The q-th column of the newly constructed matrix is represented by the signal block $\mathbf{x}_q = [x_1 \ldots x_P]^T$, with $Q > P$, and the dictionary is learned from the columns of $\mathbf{X}$. Therefore, the sparse representation problem can be stated as follows: given a real valued signal $\mathbf{x}_q = [x_0 \ldots x_{P-1}]^T$, and an orthogonal dictionary D, we seek a decomposition of $\mathbf{x}_q$, such that

$$\mathbf{x}_q = \sum_{m=0}^{N-1} \alpha_q(m)\psi(m), \quad \forall q \in \{0,\ldots,Q-1\} \tag{3.1}$$

where $\alpha_q(m)$ is the coefficient of expansion relating to the q-th column of $\mathbf{X}$.

Greedy Adaptive Dictionary Algorithm (GAD)

The GAD algorithm is a greedy method that adaptively learns a data dependent dictionary by sequentially extracting the columns of the matrix $\mathbf{X}$. At each iteration, the column of $\mathbf{X}$ with highest ℓ^2-norm becomes a dictionary element; all the columns of $\mathbf{X}$ are decreased by an appropriate amount, determined by the currently selected atom and the expansion coefficients. As a result, the column corresponding to the current atom is set to zero, thus reducing the space dimension by 1. Then, each atom subsequently extracted is orthogonal to the current atom. Hence, the GAD algorithm yields an orthogonal transform.

At ach iteration, extraction of a new atom depends on finding the column of $\mathbf{X}$ that satisfies:

$$\max_q \frac{\left\|\mathbf{x}_q\right\|_2}{\left\|\mathbf{x}_q\right\|_1} \tag{3.2}$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the ℓ^1- and ℓ^2-norm respectively. Thus at each iteration, the method reduces the energy of the data by a maximum amount, across all frames, while ensuring that the ℓ^1-norm is reduced by a minimum amount. It is interesting to note how the expression in equation (3.2) relates to the sparsity index ξ for a signal vector $\mathbf{s}_q$, defined in (Tan & Fevotte, 2005) as

$$\xi_q \triangleq \frac{\|\mathbf{s}_q\|_1}{\|\mathbf{s}_q\|_2}. \tag{3.3}$$

The sparsity index quantifies the sparsity of a signal, and is such that the smaller ξ_q is, the sparser the vector $\mathbf{s}_q$. Clearly, equation (3.2) is equivalent to

$$\min_q \xi_q. \tag{3.4}$$

Thus, by ensuring that the ℓ^1-norm is minimum, the proposed algorithm seeks a sparse dictionary by construction. The proposed sparse adaptive dictionary algorithm solves the maximization problem in equation (3.2) according to the following steps:

1. Initialisation:
   - set $m = 0$;
   - ensure that the columns of $\mathbf{X}$ have unit ℓ^1-norm

$$\mathbf{x}'_q = \frac{\mathbf{x}_q}{\| \mathbf{x}_q \|_1} \tag{3.5}$$

   - initialise the residual matrix

$$\mathbf{R}^0 = \mathbf{X}' \tag{3.6}$$

   where $\mathbf{R}(m) = [\mathbf{r}_0(m), \ldots, \mathbf{r}_{Q-1}(m)]$, and $\mathbf{r}_q(m) = [r_0(m) \ldots r_{Q-1}(m)]$ is a residual column vector corresponding to the q-th column of $\mathbf{R}(m)$.

2. Compute the ℓ^2-norm of each frame

$$E(q) = \left\| \mathbf{r}_q(m) \right\|_2 = \sum_{q=0}^{Q-1} \left| \mathbf{r}_q(m) \right|^2 \tag{3.7}$$

3. Find the index $\hat{q}$ corresponding to the signal block with largest ℓ^2-norm, $\mathbf{r}_{\hat{q}}(m)$

$$\hat{q} = \arg\max_{q \in \mathrm{Q}} \left(E(q) \right) \tag{3.8}$$

   where $\mathrm{Q} = \{0, \ldots, Q-1\}$ is the set of all indices pointing to the columns of $\mathbf{R}(m)$.

4. Set the m-th dictionary element $\boldsymbol{\psi}(m)$ to be equal to the signal block with largest ℓ^2-norm $\mathbf{r}_{\hat{q}}(m)$

$$\boldsymbol{\psi}(m) = \frac{\mathbf{r}_{\hat{q}}(m)}{\left\| \mathbf{r}_{\hat{q}}(m) \right\|_1} \tag{3.9}$$

5. For all the columns of the residual matrix $\mathbf{R}(m)$, evaluate the coefficients of expansion $\alpha_q(m)$, given by the inner product between the residual vector $\mathbf{r}_q(m)$, and the atom $\psi(m)$

$$\alpha_q(m) = \left\langle \mathbf{r}_q(m), \boldsymbol{\psi}(m) \right\rangle, \ \forall q = 0, \ldots, Q-1 \tag{3.10}$$

6. For all the columns of the residual matrix $\mathbf{R}(m)$, compute the new residual, by removing the component along the chosen atom, for each element q in $\mathbf{r}_q(m)$

$$\mathbf{r}_q(m+1) = \mathbf{r}_q(m) - \frac{\alpha_q(m)}{\left\langle \boldsymbol{\psi}(m), \boldsymbol{\psi}(m) \right\rangle} \boldsymbol{\psi}(m), \ \forall q = 0, \ldots, Q-1 \tag{3.11}$$

7. Repeat from step 2, until $m = N - 1$.

The term in the denominator of $\alpha_q(m) / \left\langle \boldsymbol{\psi}(m), \boldsymbol{\psi}(m) \right\rangle$ in equation (3.11), is included to ensure that the coefficient of expansion $\alpha_q(m)$ corresponding to the inner product between the selected atom $\boldsymbol{\psi}(m)$ and the frame of maximum ℓ^2-norm $\mathbf{r}_{\hat{q}}(m)$, is normalised to 1. Then, the corresponding column of the residual matrix $\mathbf{R}(m)$ is set to zero, since the whole atom is removed. This is the step that ensures that the transform is orthogonal. This implies that the

inverse transform is evaluated straightforwardly from $\mathbf{X}^L = \mathbf{DY}$, where $\mathbf{X}^L$ is the $L \times L$ matrix which approximates $\mathbf{X}$ using the first L atoms, and $\mathbf{D} = [(\boldsymbol{\psi}(0))^T, \ldots, (\boldsymbol{\psi}(Q-1))^T]$ is the dictionary matrix. The method has the implicit advantage of producing atoms that are directly relevant to the data being analyzed.

Since the algorithm operates upon a stereo signal, whose data samples have been reshaped into the matrix, $\mathbf{X}$, as described earlier, GAD learns a set of stereo atoms, $\boldsymbol{\psi}_{(i)}(m),\ i = 1,2$ from the columns of $\mathbf{X}$. For comparison purposes, we also use ICA and K-SVD to construct stereo dictionaries from the data, and apply the remaining SCA steps in all cases, to obtain estimates for the separated sources. The reshaping of the data allows modelling of correlations between the microphones and across time, and therefore the stereo atoms that are learned encode information regarding the mixing process. The clustering approaches outlined below aim at exploiting this property.

Estimating the Mixing Matrix by Clustering the Atom Pairs

It was shown in (Jafari et al., 2008), that the atom pairs encode information regarding the time delays for the two source signals, and therefore the atoms can be clustered according to the time delay existing between the atoms $\boldsymbol{\psi}_{(j)}(m),\ j = 1,2; m = \{0, \ldots, N-1\}$, in the pair. The time delay, or direction of arrival (DOA), is evaluated according to the generalized cross-correlation with phase transform (GCC-PHAT) algorithm in (Knapp & Carter, 1976).

GCC-PHAT typically results in a function that exhibits a single sharp peak at the lag corresponding to the time delay between the two signals, which is consistent with the learned atom pairs exhibiting a dominant DOA. Thus, we seek to find the delay at which the peak occurs for each atom pair, and use the K-means clustering algorithm on this data, in order to group the atoms. K-means will identify two clusters, whose centers correspond to the time delay for each source $\Upsilon_j,\ j = 1,2$. This can then be used to identify those atoms that relate to one source or the other, by finding a set of indices $\gamma_j,\ j = 1,2$, that map the m-th atom to the source to which it belongs

$$\gamma_j = \{m \mid (\Upsilon_j - \Delta) \leq \tau_m \leq (\Upsilon_j + \Delta)\} \tag{3.12}$$

within some threshold Δ of the cluster centroid. We also define a 'discard' cluster

$$\gamma_0 = \{m \mid m \notin \gamma_j,\ j = 1,2\} \tag{3.13}$$

for those atoms that will not be associated with any of the j sources.

Estimating the Source and Inverting the Transform

Reconstruction of the source is performed using binary masking, followed by inverting the sparsifying transform. Two mask matrices $\mathbf{H}_j(m),\ j = 1,2$ are defined as

$$\mathbf{H}_j(m) = \mathrm{diag}(h_j(0), \ldots, h_j(N-1)) \tag{3.14}$$

where

$$h_j(m) = \begin{cases} 1 & \text{if } m \in \gamma_j \\ 0 & \text{otherwise} \end{cases} \tag{3.15}$$

for $m = 0, \ldots, N-1$. Thus, the diagonal elements of $\mathbf{H}_j(m)$ set to one or zero depending on whether an atom is considered to belong to the j-th source. Then, the estimated image $\hat{\tilde{\mathbf{X}}}_j$ of the j-th source is given by

$$\hat{\tilde{\mathbf{X}}}_j = \mathbf{D}^T \mathbf{H}_j(m) \mathbf{D} \mathbf{X}_j \tag{3.16}$$

Finally, the vector of images of the j-th source at both microphones is obtained by transforming the matrix $\hat{\tilde{\mathbf{X}}}_j$ back into a vector, to find the source image $\hat{\mathbf{x}}^j(n) = \left[\hat{x}_1^j(n), \hat{x}_2^j(n) \right]^T$. This entails reversing the reshaping process that was carried out on the data before applying the SCA method.

Experimental Results

In this section we compare the GAD, ICA, and K-SVD algorithms, for the analysis of a male speech signal, and in all cases we look for a dictionary containing 512 atoms. The KSVD Matlab Toolbox was used to implement the K-SVD algorithm[3]. The number of nonzero entries T_0 in the coefficient update stage was set to 10.

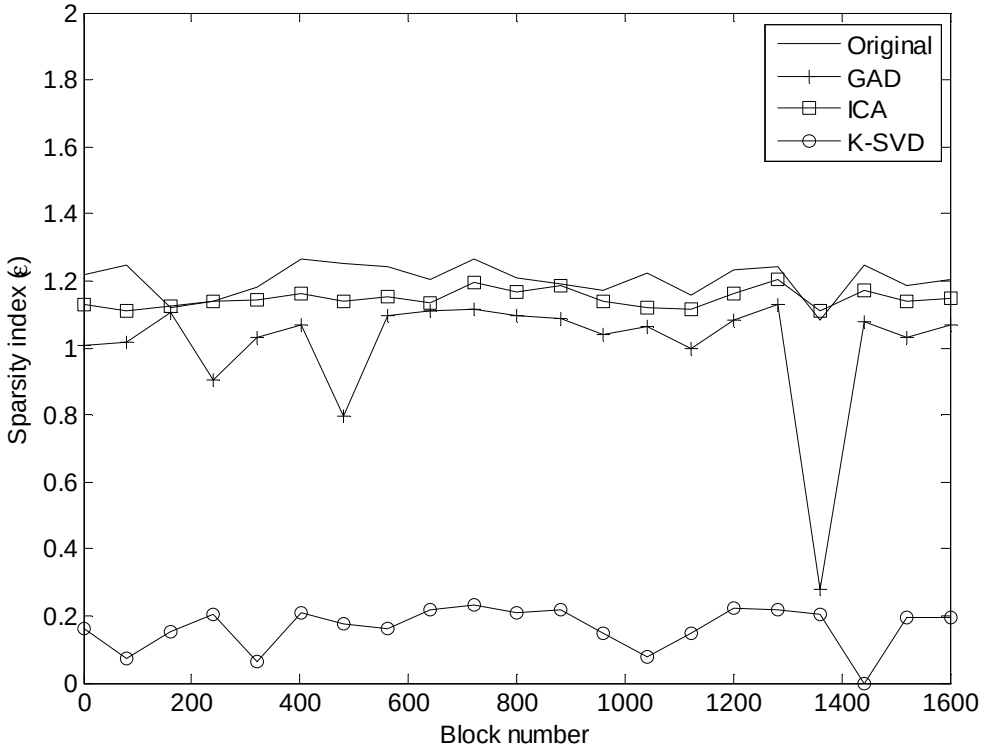

Figure 2 Sparsity index for the GAD algorithm, compared to the original signal, and to ICA and K-SVD for every block in the observation matrix. The GAD algorithm consistently achieves a sparser representation than ICA, but not as sparse as K-SVD.

[3] The K-SVD Matlab Toolbox is available at http://www.cs.technion.ac.il/~elad/software/

To determine how sparse the representation obtained with the proposed approach is, we plot the sparsity index for the transform coefficients obtained with the three methods. The sparsity index of a signal y is defined in equation (3.3) as $\xi = \| y \|_1 / \| y \|_2$; generally, the lower the sparsity index is, the sparser the signal y. Figure 2 shows a plot of the sparsity index for the original signal blocks in $\mathbf{X}$, and for the coefficients of expansion obtained with the GAD, ICA and K-SVD algorithms. We can see that the signal transformed with the GAD algorithm is sparser than in the time domain, and than the coefficients obtained with ICA.

K-SVD yields significantly sparser results, thanks to the strong sparsity constraint it imposes. Nonetheless, while such a strong constraint leads to a very sparse signal decomposition, the accuracy of the approximation decreases proportionally with T_0. This can be seen by considering the approximation error δ obtained when the function f is approximated by $\tilde{f}$,

$$\delta = \| \tilde{f} - f \|. \tag{3.17}$$

It was found that for K-SVD with $T_0 = 3$ nonzero entries, $\delta = 0.0036$, while with $T_0 = 10$, $\delta = 0.0010$, the approximation error becomes almost a third. Therefore, in what follows, we use K-SVD with $T_0 = 10$.

Method	Number of Atoms					
	512	400	300	200	100	50
GAD	0.0000	0.0007	0.0017	0.0028	0.0053	0.0068
ICA	0.0000	0.0022	0.0043	0.0078	0.0122	0.0151
K-SVD	0.0010	0.0052	0.0069	0.0093	0.0103	0.0135

Table 3 Approximation error for the GAD, ICA and K-SVD algorithms. All values are expressed in decibels (dB).

Table 3 shows the approximation error for all algorithms, describing the accuracy of the approximation as the number of atoms used in the signal reconstruction decreases from 512 to 50. The results indicate that the GAD algorithm performs better, while ICA and K-SVD yield signal approximations that suffer most from the reduction in number of atoms, with ICA performance suddenly worsening as the number of atoms goes from 200 to 100. This behaviour is a result of the way GAD works. Since the atoms with highest ℓ^2-norm are extracted first, as new atoms are found, they have less information to convey. This, however, is not the case with ICA and K-SVD. Therefore, the GAD algorithm results in good signal approximations even when the number of atoms is reduced.

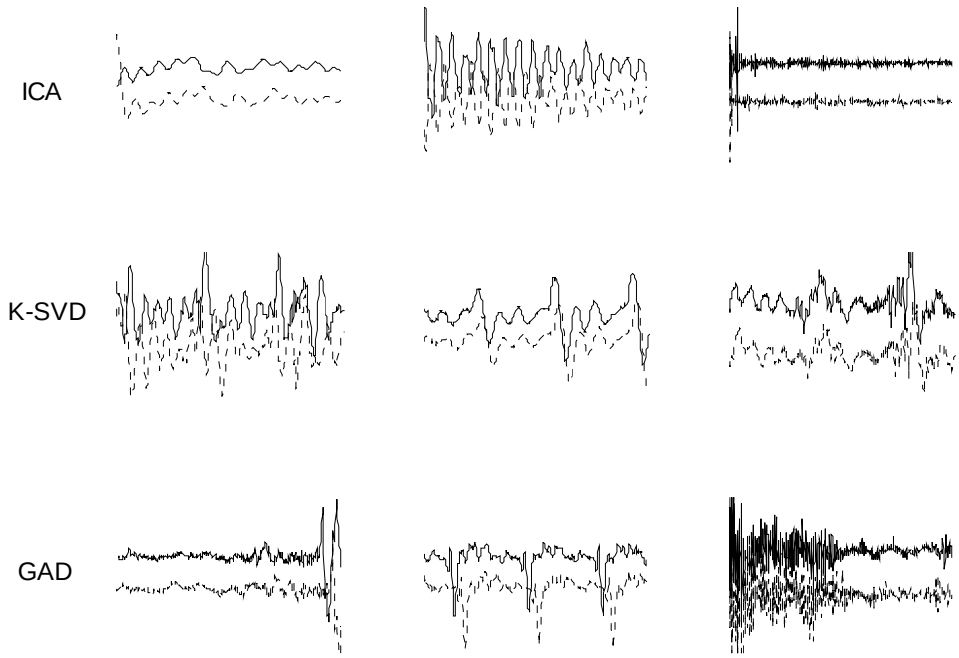

Figure 3 Examples of the atom pairs learned with the ICA, K-SVD and GAD methods. Within each pair the signals are delayed relative to each other in time.

The three methods were then used to address the audio source separation problem. A stereo mixture was generated when a male and female speech signals were synthetically mixed according to the anechoic mixing model in equation (1.1), with delays of 7 and -29 samples. Separation was performed with GAD, K-SVD and ICA as the sparsifying transforms in the SCA procedure described earlier. The plots in Figure 3 show some of the atom pairs obtained with the three algorithms. Comparing these, we see that all algorithms extract atoms that capture information unique to the analyzed signal, with the ICA-based method finding atoms that appear to represent very localised and elementary components. The GAD method yields much less elementary atoms, that appear to capture more information about the signal, and which are still fairly localized. The atoms extracted with K-SVD are the least localized, and do not appear to be capturing any particular features of the speech signal, but perhaps more general characteristics. Moreover, the atom pairs obtained with all methods were found to encode how the extracted features are received at the microphone, that is, they capture information about time-delays and amplitude differences.

Figure 4 shows estimates for the time delays, and their histograms, obtained by the three algorithms from all atom pairs. All methods were found to correctly identify the time delays as 7 and -29 samples. Their performance was also evaluated using a slightly modified version of the the signal-to-distortion ratio compared to the previous section, which required that particular definition for oracle estimation of source within the adaptive LOT framework. Here, the modified signal-to-distortion ratio is denoted SDR*, and is combined with two additional separation

performance metrics: the *signal-to-interference ratio* (SIR) and *signal-to-artifacts ratio* (SAR) measuring, respectively, the distortion due to interfering sources and the distortion due to artifacts resulting from the separation process itself (Fevotte, Gribonval & Vincent, 2005). Table 4 shows the criteria obtained, where the single figures were obtained by averaging across all sources and microphones. The low SAR and SDR* values indicate that large artifacts are present on the recovered source, which dominate the distortion. The high SIR values, on the other hand, suggest that the desired source can now be heard clearly, or more clearly than the other source. Informal listening tests suggest that in all cases the separation algorithm has the effect of making each source more audible within the mixture, but do not clearly separate them. This indicates that perhaps sparsity alone is not a sufficient criterion for separation.

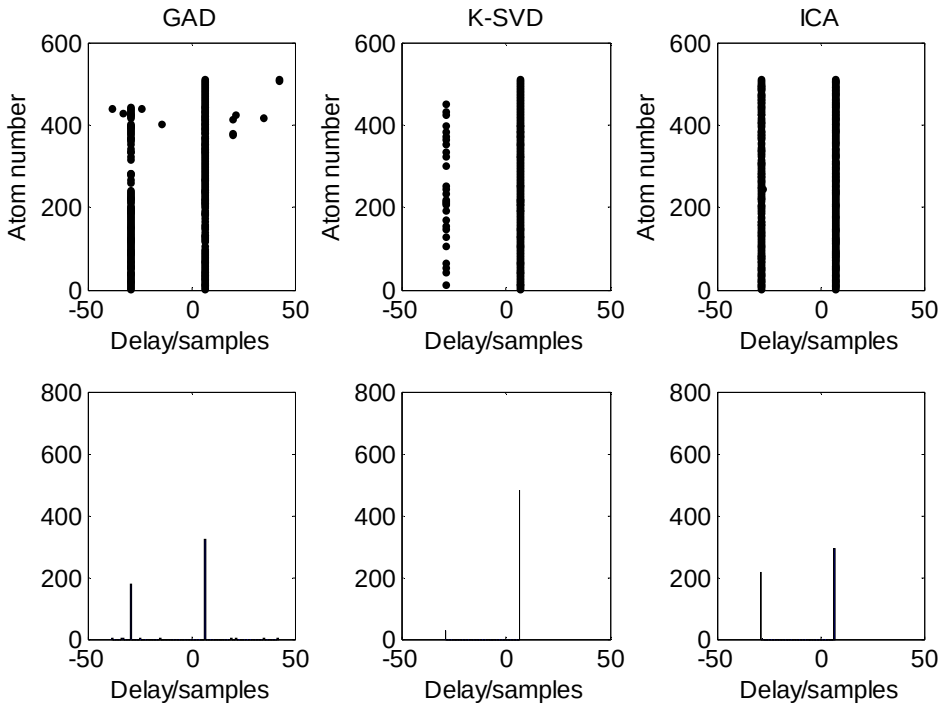

Figure 4 Upper plots: Scatterplots of the estimated time delays for all basis vectors, where the delays between signals within the atoms are apparent. Lower plots: Histograms associated with of the upper plots.

Method	SDR	SAR	SIR
GAD	3.10	3.07	9.80
ICA	3.73	3.87	8.43
K-SVD	1.31	2.58	7.59

Table 4 Objective performance of GAD, K-SVD and ICA-based separation algorithms. All values are expressed in decibels (dB).

CONCLUSIONS

In this chapter we have addressed the audio source separation problem within a sparse component analysis (SCA) framework. When the mixing is instantaneous and underdetermined, sparse signal representations are learned with adaptive lapped orthogonal transforms (LOTs). This method demonstrated average SDR performance of 12-13 dB on mixtures of music and speech signals. Further work includes extending this technique from the semi-blind separation case considered here, to the blind situation; preliminary experiments have shown very promising results, and we intend to incorporate that framework into our adaptive transform schemes.

SCA has also been applied to the anechoic, determined mixing problem. In this case, a greedy adaptive dictionary (GAD) learning algorithm was presented, and it was compared to the ICA and K-SVD methods; it was found to give good signal approximations, even as the number of atoms in the reconstructions decreases considerably. The GAD algorithm can correctly identify the directions of arrival of the source signals, but an objective assessment of the separation performance indicated that while each source become more clearly audible within the mixture, they were not completely separated. This result was corroborated by an informal listening test.

Hence, we have presented two transform techniques that fit within SCA in a similar fashion. They have important similarities, in that they both represent the sources sparsely by adapting directly to the estimated sources and observation mixtures. However, their differences underpin their intended applications; the adaptive LOT scheme, which adapts the transform given a predefined library of bases, operates within an underdetermined, instantaneous mixing framework, whereas the GAD algorithm, which adaptively learns the dictionary atoms per se, is intended for the anechoic case. In our future work, we intend to look at the extension of the GAD method to the underdetermined case, and the LOT scheme to anechoic and convolutive mixing problems, and compare directly the performance of the two methods.

ACKNOWLEDGMENTS

Part of this work was supported by EPSRC grants GR/S82213/01, GR/S85900/01, EP/E045235/1, & EP/G007144/1, and EU FET-Open project 225913 “SMALL”. Part of this work was done when AN was hosted as a visiting researcher at INRIA.

REFERENCES

Abdallah, S. A., & Plumbley, M. D. (2004). Application of geometric dependency analysis to the separation of convolved mixtures. In *Proceedings of the International Conference on Independent Component Analysis and Signal Separation* (pp. 22-24).

Aharon, M., Elad, M., & Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representations. *IEEE Transactions on Signal Processing, 54*, 4311-4322.

Benaroya, L., Bimbot, F., Gravier, G., & Gribonval, R. (2003). Audio source separation with one sensor for robust speech recognition. In *NOLISP-2003*, paper 030.

Bertin, N., Badeau, R., & Vincent, E. (2009). *Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription.* (Report No. 2009D006). Paris, France: Telecom ParisTech

Bofill, P. & Zibulevsky, M. (2001). Underdetermined blind source separation using sparse representations. *Signal Processing*, 81, 2353-2362.

Daudet, L., & Torrésani, B. (2002). Hybrid representations for audiophonic signal encoding. *Signal Processing*, *82*, 1595-1617.

Davis, G. (1994). *Adaptive nonlinear approximations.* Unpublished doctoral dissertation, New York University.

Fevotte, C., Gribonval, R., & Vincent, E. (2005). *BSS_EVAL Toolbox User Guide* (Technical Report No. 1706). http://www.irisa.fr/metiss/bss_eval/: IRISA.

Górecki, L., & Domanski, M. (2005). Adaptive dictionaries for matching pursuit with separable decompositions. In *Proceedings of the European Signal Processing Conference.* (pp.786-790).

Gribonval, R. (2003). Piecewise linear source separation. In M. A. Unser, A. Aldroubi, & A. F. Laine (Eds.), *Wavelets: Applications in Signal and Image Processing X*, *Proceedings of SPIE* (pp. 297-310). San Diego, USA: SPIE.

Gribonval, R., & Lesage, S. (2006). A survey of sparse component analysis for blind source separation: principles, perspectives, and new challanges. In *Proceedings of the European Symposium on Artificial Neural Networks* (pp. 323-330).

Hesse, C. W., & James, C. J. (2006). On semi-blind source separation using spatial constraints with applications in EEG analysis. *IEEE Transactions on Biomedical Engineering*, 53, 2525-2534.

Huang, Y., Pollak, I., Bouman, C. A., & Do, M. N. (2006) Best basis search in lapped dictionaries. *IEEE* Transactions *on Signal Processing*, 54, 651-664.

Jafari, M. G., Wang, W., Chambers, J. A., Hoya, T., & Cichocki, A. (2006). Sequential blind source separation based exclusively on second-order statistics developed for a class of periodic signals. *IEEE Transactions on Signal Processing*, 54, 1028-1040.

Jafari, M. G., Vincent, E., Abdallah, S. A., Plumbley, M. D., & Davies, M. E. (2008). An adaptive stereo basis method for convolutive blind audio source separation. *Neurocomputing*, *71*, 2087-2097.

Jafari, M. G., & Plumbley, M. D. (2008). Separation of stereo speech signals based on a sparse dictionary algorithm. In *Proceedings of the European Signal Processing Conference.* (pp.786-790).

Knapp, C., & Carter, G. (1976). The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustic, Speech, and Signal Processing*, *24*, 320-327.

Mallat, S. (1999). *A Wavelet Tour of Signal Processing*. (2nd ed.). San Diego, USA: Academic Press.

Nesbit, A., Vincent, E., & Plumbley, M. D. (2009). Extension of sparse, adaptive signal decompositions to semi-blind audio source separation. In *Proceedings of the Independent Component Analysis and Signal Separation* (pp. 605-612).

O'Grady, P. D., Pearlmutter, B. A., & Rickard, S. T. (2005). Survery of sparse and non-sparse

methods in source separation. *International Journal of Imaging Systems and Technology, 15, 18-33.*

Ravelli, E., & Daudet, L. (2006). Representations of audio signals in overcomplete dictionaries: what is the link between redundancy factor and coding properties? In *Proceedings of the International Conference on Digital Audio effects* (pp. 267-270).

Rubinstein, R., Zibulevsky, M., & Elad, M. (2009). Learning sparse dictionaries for sparse signal representation. Submitted to *IEEE Transactions on Signal Proceedings.*

Saab, R., Yilmaz, O., McKeown, M. J., & Abugharbieh, R. (2005). Underdetermined sparse blind source separation with delays. In *Proceedings of the Workshop on Signal Processing with Adaptive Sparse Structured Representations.*

Tan, V., & Fevotte, C. (2005). A study of the effect of source sparsity for various transforms on blind audio source separation performance. In *Proceedings of the Workshop on Signal Processing with Adaptive Sparse Structured Representations.*

Vincent, E., Gribonval, R., & Plumbley, M. D. (2007). Oracle estimators for the benchmarking of source separation algorithms. *Signal Processing*, 87, 1933-1950.

Vincent, E., Jafari, M. G., Abdallah, S. A., Plumbley, M. D., and Davies, M. E. (2010). Probabilistic modeling paradigms for audio source separation. In *Machine Audition: Principles, Algorithms and Systems*. IGI Global.

Winter, S., Sawada, H. & Makino, S. (2006). Geometrical interpretation of the PCA subspace approach for overdetermined blind source separation. *EURASIP Journal on Applied Signal Processing*, 2006, 176-186.

Woodruff, J., Pardo, B., & Dannenberg, R. (2006). Remixing stereo music with score-informed source separation. In *Proceedings of the Seventh International Conference on Music Information Retrieval* (pp. 314-319).

Xiong, Z. and Ramchandran, K., & Herley, C. and Orchard, M. T.(1997). Flexible tree-structured signal expansions using time-varying wavelet packets. *IEEE Transactions on Signal Processing*, 43, 333-345.

Yilmaz, Ö., & Rickard, S. (2004). Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, *52*, 1830-1847.

Zibulevsky, M., & Pearlmutter, B. A. (2001). Blind source separation by sparse decomposition in a signal dictionary. *Neural Computation, 13* (4), 863-882.