Audio Source Separation by Time-Frequency Masking using a Signal-Adaptive Local Cosine Transform

Andrew Nesbit [a,*,1] Mark D. Plumbley [a] Mike E. Davies [b]

[a] *Queen Mary, University of London*
Centre for Digital Music, Department of Electronic Engineering
Mile End Road, London, E1 4NS, UK

[b] *University of Edinburgh*
IDCOM & Joint Research Institute for Signal and Image Processing
King's Buildings, Mayfield Road, Edinburgh, EH9 3JL, UK

Abstract

Audio source separation of instantaneous, two-channel mixtures by time-frequency masking depends on (approximately) disjoint representations of the sources in some transform domain. We investigate the application of cosine packet (CP) trees to perform this transform. A computationally efficient best basis algorithm is applied to trees of local cosine bases to determine an appropriate transform. We concentrate on demixing the sources by binary masking, and assume the mixing parameters are known. We develop a heuristically motivated cost function which maximises the energy of the transform coefficients associated with a particular source. Finally, we evaluate our proposed transform method by comparing it against more well-known transforms such as the short-time Fourier transform and modified discrete cosine transform. It is shown that in some circumstances, our method of adaptively selecting local cosine bases can give better results than fixed-basis representations.

[*] Corresponding author.
Email addresses: andrew.nesbit@elec.qmul.ac.uk (Andrew Nesbit), mark.plumbley@elec.qmul.ac.uk (Mark D. Plumbley), mike.davies@ed.ac.uk (Mike E. Davies).

[1] Supported by the Department of Electronic Engineering, Queen Mary, University of London, and the *Semantic Interaction with Music Audio Contents (SIMAC)* project (EU-FP6-IST-507142). This work is also partially supported by EPSRC grants GR/S75802/01 and GR/S85900/01.

Preprint submitted to Elsevier *31 July 2006*

1 Introduction

The problem of *audio source separation* involves recovering individual audio *sources* from a number of observed *mixtures* of those audio source signals. When both the signals and mixing process are unknown, this problem is known as *blind audio source separation* (BASS).

Many different approaches to audio source separation have been investigated. For example, computational auditory scene analysis [5] aims to model the ways by which the human auditory system perceives individual sounds in mixtures. Beamforming [27] observes the mixtures with an array of sensors (commonly microphones) and takes advantage of time delays between those sensors to determine the spatial direction from which a desired source is arriving—it increases the gain in that direction while decreasing the gains of all unwanted sources. Frameworks based on independent component analysis (ICA) [14] try to find a linear transformation, by maximising some function that measures statistical independence, so that the recovered sources are as independent as possible. Time-frequency masking [30,23,1] transforms the sources and forms (possibly weighted) clusters of transform coefficients corresponding to each source.

In this article, we investigate audio source separation using different time-frequency transforms as part of the time-frequency masking method. We concentrate on separating instantaneous stereo mixtures (*pan-potted stereo*) where the mixing parameters are known (i.e. non-blind) or have been estimated by an earlier identification process within a blind algorithm. We perform source separation using a signal-adaptive local cosine transform in the form of a cosine packet tree. To adapt the local cosine transform we introduce two cost functions that attempt to represent the sources sparsely, and hence to attempt to improve source separation performance.

1.1 The instantaneous, two-channel mixing model

The number of sources present in any system of observed mixtures, and the number of observed mixtures, is one fundamental constraint on the applicability of the various source separation frameworks. Cases in which the number of mixtures is equal to the number of sources are called *overdetermined*, and those in which the numbers of sources and mixtures are equal are called *determined*. These situations have been well studied, commonly through the application of ICA [14]. If delays between sensors are present, then beamforming [27] or time-frequency masking [30,1] are possible methods. In contrast to the overdetermined case, *underdetermined* blind source separation considers cases in which there are more sources than mixtures.

In this work, we deal with underdetermined, *instantaneous*, *two-channel* mixtures of $n > 2$ time-domain audio sources:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \end{pmatrix} \begin{pmatrix} s_1 \\ \vdots \\ s_n \end{pmatrix} \tag{1}$$

where s_j is the jth source, x_i is the ith mixture, a_{ij} is the positive real amplitude (mixing parameter) of the jth source in the ith mixture (observation), and $1 \leq j \leq n$ and $i = 1, 2$.

The model given by Equation (1) describes mixtures in which each source has only a relative amplitude difference between the two channels. Neither relative delays nor reverberant recording conditions are considered. This means that the model may represent, for example, an approximation of an audio signal mixed with a *pan-potted stereo* method. In this representation, we assume that each source corresponds exactly to one (mono) input channel on a physical mixing desk, mixed down directly to stereo with no extra processing (such as compression or reverberation) applied to the mix. This is a very simple model, but it is nevertheless applicable to some real musical recordings. Our experiments will concentrate on music signal mixtures created by simulated pan-potted stereo.

The blind source separation problem may be split conceptually into two successive subproblems [29]. Estimation of the a_{ij} constitutes the *identification* phase, while extraction of the s_j, to yield estimated sources $\hat{s}_j$ is the called *filtering* phase. In this article, we are concentrating primarily on the filtering phase. We assume that the mixing parameters have been determined already by some other method, such as forming a histogram of mixing parameter estimates [30] or clustering in high-dimensional spaces [12]. This means that our methods are equally applicable to other *non-blind* scenarios, in which the mixing parameters are known or have already have determined.

The structure of this article is as follows: in Section 2 we introduce the method of source separation by time-frequency masking, including an overview of the DUET method and a discussion of alternative transforms. In Section 3 we introduce the cosine packet (CP) tree approach, together with our proposed cost functions for its use in source separation. In Section 4 the proposed method is evaluated and compared to the short-time Fourier transform (STFT) and the modified discrete cosine transform (MDCT), and is followed by a discussion of further work and conclusions.

2 Source separation by time-frequency masking

Time-frequency masking is a powerful technique for separating underdetermined mixtures [29]. One of the requirements of most types of time-frequency masking systems, is that the sources have a disjoint representation. It is well-known that time-domain representations of audio signals are not, in general, disjoint. Therefore, time-frequency masking methods transform the mixtures to produce (approximately) disjoint representations. One commonly used transform is the short-time Fourier transform (STFT). The STFT is suitable for representing anechoic mixtures of speech signals disjointly, and is commonly used in time-frequency masking algorithms such as DUET [30].

DUET is one method which may be applied to mixtures in the form of Equation (1). It was originally developed for blind source separation of *anechoic* mixtures, in which the mixture may include relative delays as well as relative amplitude gains, but may be applied to instantaneous mixtures by setting all relative delays to zero.

2.1 Representing mixtures by short-time Fourier transform

The DUET algorithm uses the STFT to represent the mixtures before masking. Let $\tilde{x}_i$ represent the STFT of the ith mixture of length N:

$$\tilde{x}_i(k,l) = \sum_{r=0}^{N-1} x_i(r) w(r-k) \exp\left(\frac{-i2\pi lk}{N}\right) \tag{2}$$

where w is a suitably chosen window function which satisfies the overlap-add condition for resynthesis [16].

The STFT is a linear operation. This means that after transforming both mixtures, the following holds:

$$\begin{pmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{pmatrix} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \end{pmatrix} \begin{pmatrix} \tilde{s}_1 \\ \vdots \\ \tilde{s}_n \end{pmatrix} \tag{3}$$

where $\tilde{x}_i$ and $\tilde{s}_i$ are the respective transforms of the signals x_i and s_i. The mixing parameters a_{ij} are the same as in Equation (1) and $\tilde{s}_j$ is the transformed jth source.

If the STFT has successfully transformed the mixtures so that the sources have disjoint representations, then at any given time-frequency point, the contribution from at most one source will dominate. If source s_j dominates at time-frequency index pair (k,l) then Equation (3) reduces to the following simple linear system:

$$\begin{pmatrix} \tilde{x}_1(k,l) \\ \tilde{x}_2(k,l) \end{pmatrix} \approx \begin{pmatrix} a_{1,j} \\ a_{2,j} \end{pmatrix} \tilde{s}_j(k,l). \tag{4}$$

This means that we can estimate the mixing parameters of each source by forming ratios of the transformed mixtures, since $\frac{\tilde{x}_2(k,l)}{\tilde{x}_2(k,l)} \approx \frac{a_{1,j}}{a_{2,j}}$ where j is the index of the dominant source.

2.2 Identification of mixing parameters

Recall that for each source, we assume that the ratio of its mixing coefficients $\frac{a_{2,j}}{a_{1,j}}$ is unique to that source. That is, $\frac{a_{2,j}}{a_{1,j}} \neq \frac{a_{2,j'}}{a_{1,j'}}$ whenever $j \neq j'$. This means the ratio $\frac{|\tilde{x}_2(k,l)|}{|\tilde{x}_1(k,l)|}$ at each time-frequency point is an estimate of the ratio of the mixing parameters for some, as yet unknown, source. Each estimate is likely to be close to the true mixing parameters of some source, so by forming clusters of these points, we can estimate the mixing parameters. In the DUET algorithm we form a histogram of these ratios, with each peak in the histogram corresponding to one source. Note that it is only possible to estimate the *ratio* of the mixing parameters for each source, not the mixing parameters themselves. Nevertheless, this information is sufficient to extract the sources, relative to some unknown scaling factor.

In this article, we are concentrating on the filtering phase, so we will assume that the mixing parameters are known, or have already been estimated.

2.3 Filtering by time-frequency masking

We assume that the mixtures have been transformed into an approximately disjoint representation of the sources. The sources may therefore be extracted by a conceptually simple clustering technique called *binary time-frequency masking*. Each cluster has a centre given by the ratios of mixing parameters which are given or have already been estimated. The mixtures' time-frequency coefficients are partitioned into these clusters, thus extracting the sources. Finally, an inverse STFT transforms the source estimates back into the time domain, giving the separated source estimates. Typically the overlap-add method is used to invert the STFT [16].

2.4 Alternative transforms

DUET uses the STFT to yield disjoint representations, to which the time-frequency masking is applied. Its properties are well understood and it is simple to implement, and has been shown to perform well on speech signals. Furthermore, the STFT gives extra information about the relative phase of the signals, allowing separation of anechoic mixtures. This means that in addition to relative amplitude differences between sources, sources which have relative delays between mixtures can be separated. However, since we are working with instantaneous mixtures, in which no delays are present, we do not necessarily need this extra information.

Even though the STFT is a popular choice in many time-frequency masking techniques [4,1], its use is not ubiquitous. The designers of the DUET algorithm recognise that any transform which satisfies certain properties[2] will suit a time-frequency masking framework [30]. An initial study of the performance effects of different transforms appears in [25], and time-frequency masking has been successfully applied to instantaneous mixtures of speech in the modified discrete cosine transform (MDCT) domain [9,25].

Our motivation for exploring different transforms is to give a representation which represents the sources more disjointly than the STFT and hence might give better separation. When block transforms, such as the STFT, are computed with discontinuous rectangular windows, artefacts occur around the block boundaries. This is because the truncated basis functions cause discontinuities in the reconstruction of the signal. Smooth, compactly supported windows help avoid these artefacts, but it can be shown that a Fourier basis resulting from such windowing cannot be orthonormal, and is, in fact, overcomplete. An overcomplete transform contains redundant information which will reduce the sparsity of the representation.

To try to overcome these limitations, *lapped transforms* [18] were developed. The main idea is to construct an orthogonal basis such that each windowed cosine basis function has a smooth transition from and to zero at its start and end. The commonly used MDCT [22,21] can also be classified as a lapped transform. The MDCT is linear, critically sampled, has well-known energy compaction (sparsifying) properties, and allows perfect reconstruction through overlap-add synthesis. These properties make it a suitable candidate for disjoint representation of sources, and so we compare it to other transforms (Section 4).

[2] These properties are invertibility, disjoint representations and linearity. In the case of anechoic mixtures with delays, the narrowband assumption for array processing is also assumed.

Let us generalise the concept of time-frequency masking to other linear transforms. The first step in time-frequency masking is to choose an appropriate representation for the mixtures. We apply a real- or complex-valued linear transform T to the mixtures x_1 and x_2 to give a pair of transformed mixtures $\tilde{x}_1 = Tx_1$ and $\tilde{x}_2 = Tx_2$, forming the mixing structure given by Equation (3). In Equation (1) T was the STFT.

We have already established that we want the transform to give a disjoint representation of the sources. Now we address the question of how to obtain such a representation, through the use of *sparse* transforms [20,25]. A sparse transform has most coefficients very close to zero and only a few large coefficients. The probability that two sources are active at any given time-frequency index is very small. This means that a sparse transform will represent the mixtures in such a way that the sources have approximately disjoint support in the transform domain.

Sparse transforms which we will evaluate (Section 4) include the STFT and the modified discrete cosine transform (MDCT) [21,22], each of which has a fixed basis set. In this article, we will also evaluate adaptive transforms whose basis functions are localised cosines, in which the basis functions are adapted to match the signal structures (Section 3.1).

Individual sources can be estimated from $\tilde{x}_1$ and $\tilde{x}_2$ by constructing binary time-frequency masks. This assumes that at each point in the transform domain, energy from at most one source dominates, that is, it assumes a sparse transform which represents the sources disjointly. Then, the mask can be used to filter (extract) the coefficients belonging to a particular source.

Given that we have the ratios of mixing coefficients, or estimates of these, for example using the methods proposed in [12] or [30], the masks are constructed as follows. Given the linear system described by Equation (3), the ratio of mixing coefficients for the jth source can be associated with an angle

$$\theta_j = \arctan\left(\frac{a_{2j}}{a_{1j}}\right) \tag{5}$$

where the inverse tangent is computed in the first quadrant of the plane. If the transform $Ts_j = \tilde{s}_j$ of the jth source is sparse and real-valued, then its coefficients tend to cluster along the line defined by θ_j. A binary time-frequency mask $M_{\theta_j,u}$ captures the coefficients which fall 'close' to the line corresponding to θ_j and discards all others.

For simplicity we will consider manually-defined masks based on symmetric thresholds on the ratio angle $\hat{\theta} = \arctan(\tilde{x}_2/\tilde{x}_1)$. This allows us to set the thresholds in such a way that we can test the sparsity of the transform. For example, setting a very small threshold allows us to evaluate easily whether or not the transform coefficients for a particular source are close to the radial line described by its mixing coefficients. Secondly, it enables us to compare different transforms given the same binary mask. We therefore use the mask

$$M_{\theta_j,u} = \begin{cases} 1 \text{ if } \theta_j - \frac{u}{2} < \arctan\left(\frac{\tilde{x}_2}{\tilde{x}_1}\right) < \theta_j + \frac{u}{2} \\ 0 \text{ otherwise.} \end{cases} \tag{6}$$

This mask $M_{\theta_j,u}$ estimates the mixture coefficients carrying most of the energy for source j, for the given symmetric threshold u.

Once the masks have been constructed, they can be used to determine $\hat{\tilde{s}}_j$, an estimation of the transformed jth source:

$$\hat{\tilde{s}}_j = M_{\theta_j,u} \cdot (\tilde{x}_1 \cos\theta_j + \tilde{x}_2 \sin\theta_j). \tag{7}$$

Finally, we need to apply the inverse transform to recover the time-domain sources s_j. In the case of a complete, invertible transform, we simply use the inverse transform $s_j = T^{-1}\tilde{s}_j$. For overcomplete transforms, a pseudo-inverse or other dimension reduction is also involved. For the STFT, which is normally an overcomplete transform, the overlap-add procedure is one way to perform the required 'pseudo-inverse' transform back to the time domain.

3 Adapting the representation using the Cosine Packet Tree

Transforms such as the STFT and MDCT have constant-length analysis windows and fixed bases for the entire duration of the signal, giving fixed time-frequency resolution. In order to better match the time-varying characteristics of the mixtures and sources, some researchers have used adaptive transforms [16], whose bases and window lengths are adapted to the input signals, to try to represent signals more sparsely. Examples include ICA-like approaches which use local cosine bases (Section 3.1) and wavelet packets to represent the signals [13], and clustering of wavelet transform and wavelet packet coefficients [15].

In the following sections, we describe two local cosine packet transforms in which the transform adapts to the input mixtures, giving longer windows over intervals requiring fine frequency resolution, at the expense of coarser time resolution, and shorter windows over intervals with broadband frequency content, giving finer time resolution. If a signal is decomposed in such a basis, then we anticipate that the resulting transform may be sparser than transforms which decompose the signal in a fixed basis.

3.1 Trees of local cosine bases

The basis functions of the cosine packet transform [16] are defined over dyadic-length (powers of 2) intervals $[c_{pd}, c_{p+1,d}]$. The endpoints are given by

$$c_{pd} = 2^{-d}Np - \frac{1}{2} \tag{8}$$

where N is the length of the (time-domain) signal, and these define a binary tree structure where the depth of a node is given by d up to a maximum depth D ($0 \leq d \leq D$), and the position of a node at level d is given by p ($0 \leq p < 2^d$). A pair of indices (p, d) corresponds to a node in the tree, and identifies a signal space $\mathbf{W}_d^p$ spanned by an orthogonal *local cosine basis*:

$$\left\{ w_{pd}[n] \sqrt{\frac{2}{2^{-d}N}} \cos \left[\pi \left(k + \frac{1}{2} \right) \frac{n - c_{pd}}{2^{-d}N} \right] \right\} \tag{9}$$

where $0 \leq k < 2^{-d}N$ indexes the functions in the basis. The smooth window w_{pd} localises the basis functions over a dyadic interval $[c_{pd}, c_{p+1,d}]$ and partly overlaps with its immediately adjacent windows $w_{p-1,d}$ and $w_{p+1,d}$. Furthermore, the window must satisfy special properties [16].

Each signal space $\mathbf{W}_d^p$ is orthogonal to $\mathbf{W}_d^q$ whenever $p \neq q$, and $\mathbf{W}_j^p = \mathbf{W}_{j+1}^{2p} \oplus \mathbf{W}_{j+1}^{2p+1}$. This means that the union of the bases corresponding to the children of any node is an orthogonal basis of the space corresponding to that (parent) node. The length-N signal being analysed is in the signal space $\mathbf{W}_0^0$. Bases occurring deeper in the tree correspond to shorter time intervals and so are better for representing sections of the signal with highly time-varying characteristics; bases occurring higher in the tree are better for representing sections which need better frequency resolution at the cost of coarser time resolution. We attempt to use this framework to our advantage to represent the sources more disjointly. Moreover, this tree structure offers a computationally efficient method for computing a good basis.

3.2 Selecting the best basis

A tree of local cosine bases describes many possible orthogonal bases for representing a signal. A complete binary tree provides a *dictionary* of more than one orthogonal basis from which the optimal basis can be adaptively chosen to represent the signal. This is in contrast to transforms such as the STFT or MDCT, whose dictionaries include exactly one basis set. The l^1 cost of

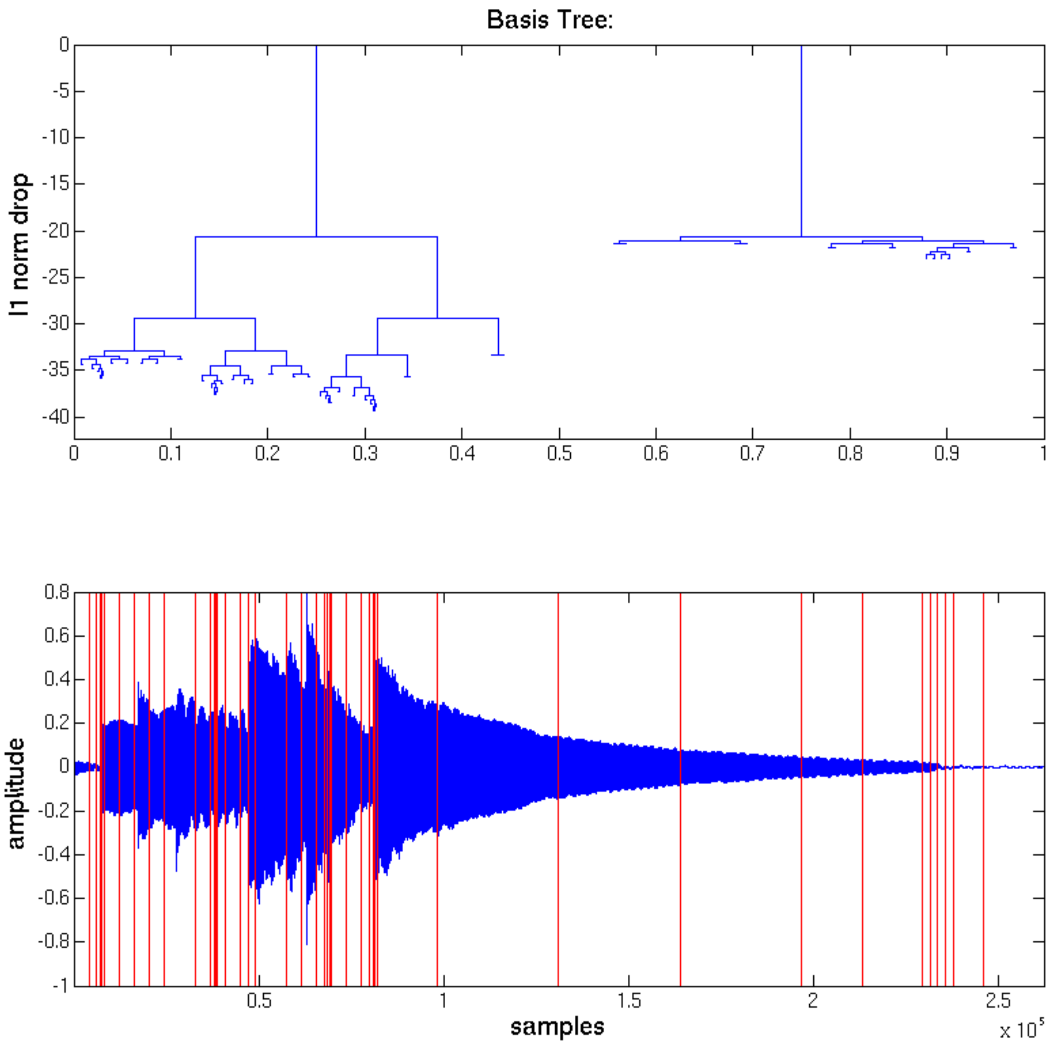

Fig. 1. Recording of a glockenspiel. Upper plot is a local cosine best basis tree computed by minimising the l^1 norm to a maximum depth $D = 10$. Lower plot is the time-domain signal partitioned into intervals; the width of each interval is determined by the depth of the corresponding basis in the tree.

representing a length-N signal x in the basis $B = \{b_m\}$ is given by

$$C(x, B) = \sum_{m=1}^{N} \frac{|\langle x, b_m \rangle|}{\|x\|} \tag{10}$$

and provides a convenient measure of sparsity [6]. We choose the *best basis* as the one which minimises this cost. The computationally efficient Coifman-Wickerhauser algorithm takes advantage of the binary structure and determines the best basis in $O(N \log_2 N)$ time [7].

Figure 1 depicts a tree of local cosine bases adapted to an audio recording of a glockenspiel showing the original time-domain signal partitioned into dyadic intervals, each of which correspond to a basis in the tree. The bars of the glockenspiel are struck in the first half of the signal and so relatively short basis functions have been adapted to capture the transients. The notes all ring out and decay in the second half of the signal; here, long basis functions have been chosen because the signal varies relatively slowly over time.

3.3 Adapting to the input

We consider two natural ways by which the local cosine basis may be adapted. The first method attempts to maximise the sparsity of the average of the two mixtures $\tilde{x}_1$ and $\tilde{x}_2$

$$x_a = \frac{1}{2}(x_1 + x_2) \tag{11}$$

by minimising the l^1 cost described in Section 3.2. This method will be referred to as *CP1*. Results for CP1, are given in Section 4.

3.4 Adapting to a single source

One issue with the CP1 method is that it models mixtures of the sources rather than the sources themselves. For example, in a music signal, if a percussive note with broadband frequency content and a tonal note with fine frequency content occur at same time, then the basis selected to cover that time interval may not be particularly well adapted to either. Furthermore, the basis may not adapt to transients well as tonal content tends to have more energy.

To overcome this possible limitation, we propose to adapt one basis to the expected output of the time-frequency mask for each source. This will select a basis for each source with the intention that each such basis will capture the time-frequency structures of that source better than the basis determined by CP1.

We propose a heuristically motivated cost function based on this intuitive reasoning. Whereas the CP1 method minimises the l^1 cost of expressing a signal in some basis, here we maximise the energy of the local cosine coefficients associated with a particular source angle θ_j. The mixing parameters for a given source are known; the representation which has greatest sparsity for this source has local cosine coefficients clustered around these mixing parameters. By selecting a basis which maximises the energy of coefficients that cluster around θ_j we would expect that a sparse representation will be generated.

Therefore we use the following cost function:

$$C(x_1, x_2, B, \theta_j, u) = -\sum_{m=1}^{N} \Lambda_{\theta_j,u} \langle (x_1 \cos\theta_j + x_2 \sin\theta_j), b_m \rangle^2 \tag{12}$$

where

$$\Lambda_{\theta_j,u} = \begin{cases} 1 \text{ if } \theta_j - \dfrac{u}{2} < \arctan\left(\dfrac{\langle x_2, b_m\rangle}{\langle x_1, b_m\rangle}\right) < \theta_j + \dfrac{u}{2} \\ 0 \text{ otherwise} \end{cases} \tag{13}$$

and $B = \{b_m\}$ is a basis from the dictionary of bases derived from the complete local cosine tree. The binary mask $\Lambda_{\theta_j,u}$ has a similar form to Equation (6), but instead of masking a transformed mixture $\tilde{x}_1$ or $\tilde{x}_2$, it masks local cosine coefficients in the basis B. Again, the fast tree-searching algorithm of Coifman and Wickerhauser [7] finds the best basis corresponding to this cost function. For the rest of this article, this method will be referred to as *CP2*.

This method learns an overcomplete dictionary of bases adapted to different sources. In this sense, it may be considered to be equivalent to techniques based on, for example, the matching pursuit algorithm [17]. However, the advantage of this method stems from the representation of local cosine bases as tree structures which allows us to apply the fast tree-searching algorithm to determine the best basis.

4 Evaluation

We obtained eight pieces of multitracked music by several artists, released under Creative Commons licenses, with access to the original multitracked digital audio data [24,3,10,2,19,8,26,11]. This provided us with sources from which to synthesize instantaneous mixtures. For each mixture, the pitched sources were harmonically related and so overlapping partial frequencies were expected. Each source had a sample rate of 22.05 kHz at a resolution of 16 bits per sample. An extract of 2^{18} samples was taken from each source, giving approximately 11.9 s of audio.

For each experiment, the mixtures x_1 and x_2 were generated by instantaneously mixing, with the same mixing parameters in each case:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0.83\ 0.60\ 0.40\ 0.83 \\ 0.17\ 0.40\ 0.60\ 0.17 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{pmatrix}. \tag{14}$$

The resulting mixtures are simulations of pan-potted stereo mixes, which we have found to be relatively challenging for standard signal extraction methods.

This mixing matrix ensures that all sources are evenly spaced in the first quadrant of the plane. The motivation behind this choice of mixing matrix is so that we can test the same set of symmetric threshold constants u over all mixtures. The values of u that we tested were 0.1, 0.2, 0.3, and 0.39. It is clear from Equation (6) that $u' > u$ implies $M_{\theta_j,u'} \geq M_{\theta_j,u}$, so the set of time-frequency points covered by the masks form a sequence of supersets.

For the STFT and MDCT methods, block sizes $K = 2^m$ with $m = 6,\ 7, \ldots,\ 15$ were tested, i.e. $K = 64$ up to $K = 32768$ in powers of two. For the CP1 and CP2 methods, we tested maximum tree search depths $D = 3, 4, \ldots, 12$.

4.1 Measuring performance

To measure the performance of the source separation methods, we used the criteria discussed in [28]. For each estimated source, we wish to make numerical evaluations of the contribution of unwanted sources (interference) and the distortion due solely to the separation process (artefacts). We do this by measuring the *Source to Interference Ratio (SIR)* and the *Source to Artefacts Ratio (SAR)*. Furthermore, in order to simplify direct comparisons, the *Source to Distortion Ratio (SDR)* is computed; this combines both the SIR and SAR into a single numerical measure of total relative distortion. Methods for computing these measurement criteria are explained in detail in [28]. Whenever these measures are used, they will be stated in units of decibels (dB).

4.2 Experiments

Time-frequency masks were constructed and applied to the mixture channels x_1 and x_2 represented by the following transforms:

- **STFT** at block sizes K, Hamming-windowed, and with $K/2$ overlap on consecutive blocks. This is essentially the filtering component of DUET [30]. The STFT is a complex-valued transform, so the binary masks were determined based on the magnitude of the STFT.
- **MDCT** critically sampled, with various block sizes K
- **CP1** at various maximum tree search depths D
- **CP2** at various maximum tree search depths D

The lengths of the CP1 and CP2 basis functions at each D correspond to block sizes $K = 2^{M-D}$, where $M = 18$ because the length of the input signal is 2^{18}

transform	SDR	SIR	SAR	total
STFT	24 (75%)	18 (56%)	22 (69%)	64 (67%)
MDCT	2 (6%)	5 (16%)	1 (3%)	8 (8%)
CP1	0 (0%)	6 (19%)	0 (0%)	6 (6%)
CP2	6 (19%)	3 (9%)	9 (28%)	18 (19%)
total	32 (100%)	32 (100%)	32 (100%)	96 (100%)

Table 1
Performance of the various transforms. Each cell in the table indicates the number of times the transform scored best for that performance measure.

samples. This allows us to compare methods based on the lengths of their analysis windows. The depth d of a node in a local cosine tree corresponds to basis functions of length 2^{18-d} (the example mixture has length 2^{18}). The maximum tree depths tested were $D = 12$, so that the smallest basis functions have length 64 (equal to the smallest K).

The results are presented in Tables A.1,A.2 and A.3. They indicate that the effects of the artefacts dominate interference, since the SAR values are typically significantly lower than SIR. This is not too surprising, since these methods are based on binary masking, and we would expect the masking process to introduce some artefacts.

The number of 'best' results for each transform are shown in Table 1. Overall, the STFT showed best separation performance for the majority of sources and pieces, with our proposed CP2 showing best performance on most of the remainder. The MDCT and our proposed CP1 method performed best on only a few sources. Nevertheless, since our proposed CP2 method is a complete, orthogonal transform while the STFT is an overcomplete, non-orthogonal transform, with double the representation size of CP2, we consider these results to represent competitive performance for our proposed CP2 method. We could find no immediately apparent relationship between the nature of an extracted source and the performances of the various transforms.

We noticed that some pieces tended to be separated better with short frames or deep trees, while others were separated better with long frames or shallow trees. For example, *Blue* [24] (more steady-state content) was separated best with long frame sizes (shallow trees) while *Carol* [3] (more transient content) was separated best with short frame sizes (deep trees). These observations generally reflect what we might expect from this type of music, although this is by no means consistent across all methods.

Informal listening tests indicate that in general our proposed CP2 method, when giving reasonable performance, appears to produce less audible 'pipe noise' when compared to the STFT. These also suggest that the noise is least objectionable for mid-range tree depths (around $D = 6$, $K = 4096$), with more pipe noise for deeper trees (large D, small K), while shallower trees (small D, large K) are associated with pre-echo and apparent note timing jitter. We intend to carry out other listening tests in future to further investigate these effects.

5 Further work

Results have shown that adapting a local cosine basis to the output can give good results. However, the energy-based cost function (Equation (12)) is derived from heuristic reasoning. It may be the case that a more subtle cost function is required to represent the estimated source more sparsely. In particular, the current energy-based cost function considers only coefficients of the estimated source without regarding the coefficients of the other sources. Therefore, the next step is to manually examine the basis functions which are adapted to a particular source direction and determine the most suitable cost function for this sort of joint adaptation of local cosine bases. Similarly, the CP1 technique (Section 3.3) minimises the l^1 cost of $\tilde{x}_a$, the average of the input mixtures. Alternatively, one could minimise the average l^1 cost of both $\tilde{x}_1$ and $\tilde{x}_2$.

The tree structure described in Section 3 is not necessarily tied to local cosine bases. It should be possible to apply a tree-like framework to other transforms, such as the STFT. This would give access to phase information so the framework could be used to separate anechoic mixtures.

All techniques in this article assume the mixing parameters are already known (the non-blind case). In practical situations this information may not be available and so the mixing structure would need to be identified. It would be interesting to study the sensitivity of the sparse representations to the accuracy of the mixing parameter estimates.

Finally, the performance measures, SIR, SAR and SDR, may not correspond well to a subjective human assessment of separation performance. Informal listening tests show that each representation imparts a noticeably different timbre to the extracted sources. Therefore, we believe that listening tests would give a more meaningful, practical measure of separation performance.

6 Conclusions

We have described a time-frequency masking approach to stereo audio source separation using local cosine packet representations. We proposed two versions, one (CP1) with a cost function calculated from the mean of the observation, and another (CP2) which adapts the basis set to the time-frequency mask used for separating each separate source. Searching a tree of local cosine bases is fast and gives promising results.

We compared the performance of our proposed time-frequency methods to the short-time Fourier transform (STFT) and modified discrete cosine transform (MDCT) on a set of instantaneous stereo musical audio mixtures ('panpotted stereo'). The STFT gives the best performance for separation of most sources from most mixtures. Nevertheless, our results indicate that the performance our proposed CP2 method is competitive, and exhibits better performance than the MDCT. Informal listening tests suggest that the cosine packet method can exhibit less objectionable noise than the STFT. We consider the cosine packet method to be an interesting method for time-frequency source separation, and we will continue to investigate this in future work.

Acknowledgements

The authors would like to thank Emmanuel Vincent for many suggestions and discussions during this work.

References

[1] Parham Aarabi, Guangji Shi, and Jahromi Omid. Robust speech separation using time-frequency masking. In *Proceedings of the 2003 IEEE Conference on Multimedia and Expo (ICME 2003)*, Baltimore, MD, USA, 6–9 July 2003.

[2] AlexQ. Jiggly (2000 version). Multitrack audio recording. Accessed online at `http://www.archive.org/details/alexqjiglive` subject to the *Creative Commons Attribution-NonCommercial-ShareAlike 2.0* license.

[3] AlexQ. Carol of the Bells. Multitrack audio recording, 2003. Accessed online at `http://www.archive.org/details/alexqcaroltrax` subject to the *Creative Commons Attribution-NonCommercial-ShareAlike 2.0* license.

[4] Dan Barry, Bob Lawlor, and Eugene Coyle. Real-time sound source separation: Azimuth discrimination and resynthesis. In *Proceedings of the AES 117th Convention*, San Francisco, CA, USA, 28–31 October 2004.

[5] Albert S. Bregman. *Auditory Scene Analysis.* MIT Press, 1994.

[6] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Review*, 43(1):129–159, 2001.

[7] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2):713–718, March 1992.

[8] Apple Computer. Apple Loops for Soundtrack. Collection of audio loops bundled with computer software, 2004.

[9] M. Davies and N. Mitianoudis. Simple mixture model for sparse overcomplete ICA. *IEE Proceedings on Vision, Image and Signal Processing*, 151(1):35–43, February 2004.

[10] Another Dreamer. Dreams. Multitrack audio recording, 2004. Accessed online at `http://www.anotherdreamer.net` subject to the *Creative Commons Attribution-NonCommercial 1.0* license.

[11] Another Dreamer. We Weren't There. Multitrack audio recording, 2004. Accessed online at `http://www.anotherdreamer.net` subject to the *Creative Commons Attribution-NonCommercial 1.0* license.

[12] Pando Georgiev, Fabian Theis, and Andrzej Cichocki. Sparse component analysis and blind source separation of underdetermined mixtures. *IEEE Transactions on Neural Networks*, 16(4):992–996, July 2005.

[13] Rémi Gribonval. Piecewise linear source separation. In Michael A. Unser, Akram Aldroubi, and Andrew F. Laine, editors, *Proceedings of the SPIE (Wavelets: Applications in Signal and Image Processing X)*, volume 5207, pages 297–310. SPIE—The International Society for Optical Engineering, WA, USA, November 2003.

[14] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis.* Wiley-Interscience, 2001.

[15] Pavel Kisilev, Michael Zibulevsky, Yehoshua Y. Zeevi, and Barak A. Pearlmutter. Multiresolution framework for blind source separation. Technical Report CCIT 317, Technion University, Israel, June 2001.

[16] Stéphane Mallat. *A Wavelet Tour of Signal Processing.* Academic Press, second edition, 1999.

[17] Stéphane G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, December 1993.

[18] Henrique S. Malvar. *Signal Processing with Lapped Transforms.* Artech House, Norwood, MA, USA, 1992.

[19] Mister Mouse. Nat Min, 2005. Accessed online at `http://members.home.nl/mistermouse/natmin/index.htm` subject to the *Creative Commons Attribution-NonCommercial 2.0* license.

[20] Paul D. O'Grady, Barak A. Pearlmutter, and Scott T. Rickard. Survey of sparse and non-sparse methods in source separation. *International Journal of Imaging Systems and Technology (Special Issue: Blind Source Separation and De-convolution in Imaging and Image Processing)*, 15(1):18–33, 2005.

[21] J. P. Princen, A. W. Johnson, and A. B. Bradley. Subband/transform coding using filter bank designs based on time domain aliasing cancellation. In *Proceedings of IEEE 1987 International Conference on Acoustics, Speech and Signal Processing (ICASSP'87)*, volume 12, pages 2161–2164, Dallas, TX, USA, April 1987.

[22] John P. Princen and Alan Bernard Bradley. Analysis/synthesis filter bank design based on time domain aliasing cancellation. *IEEE Transactions ASSP*, 34(5):1153–1161, October 1986.

[23] Nicoleta Roman. *Auditory-Based Algorithms for Sound Segregation in Multisource and Reverberant Environments.* PhD thesis, Ohio State University, USA, 2005.

[24] Brian Smith. Blue Backdrop. Multitrack audio recording, 2005. Accessed online at `http://www.archive.org/details/bluebackdropneedsvox`, subject to the *Creative Commons Attribution-NonCommercial 2.0* license.

[25] Vincent Y. F. Tan and Cédric Févotte. 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 (SPARS'05)*, Rennes, France, 16–18 November 2005.

[26] Espi Twelve. Sun Under Shadows. Multitrack audio recording, 2004. Accessed online at `http://www.projektwerkstatt.de/krach/download.html` subject to the *Creative Commons Attribution-ShareAlike 2.0* license.

[27] Barry D. Van Veen and Kevin M. Buckley. Beamforming: a versatile approach to spatial filtering. *IEEE ASSP Magazine*, 5(2):4–24, April 1988.

[28] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE Transactions on Speech and Audio Processing*, 2004. Preprint, to appear.

[29] Emmanuel Vincent, Maria G. Jafari, Samer A. Abdallah, Mark D. Plumbley, and Mike E. Davies. Blind audio source separation. Technical Report C4DM-TR-05-01, Centre for Digital Music, Queen Mary, University of London, November 2005. Available online `http://www.elec.qmul.ac.uk/people/emmanuelv/VincentEtAl05_bass_tutorial.pdf`.

[30] Özgür Yılmaz and Scott Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, 52(7):1830–1847, July 2004.

A Tables of results

src	trans	K	D	u	SDR	SIR	SAR
s_1	STFT	16384	-	0.30	**21.93**	**38.97**	**22.02**
	MDCT	16384	-	0.20	18.03	34.48	18.14
	CP1	2048	7	0.20	18.69	36.64	18.76
	CP2	8192	5	0.30	16.90	27.43	17.31
s_2	STFT	32768	-	0.39	**9.82**	**38.26**	**9.83**
	MDCT	8192	-	0.39	6.91	29.71	6.94
	CP1	8192	5	0.39	6.58	31.77	6.60
	CP2	1024	8	0.39	7.82	32.97	7.83
s_3	STFT	32768	-	0.10	**2.26**	**12.66**	**2.90**
	MDCT	16384	-	0.10	-0.42	12.17	0.08
	CP1	64	12	0.10	-0.20	11.90	0.34
	CP2	16384	4	0.10	-1.72	9.56	-0.93
s_4	STFT	16384	-	0.39	1.52	20.88	1.60
	MDCT	16384	-	0.39	2.82	20.44	2.94
	CP1	16384	4	0.39	2.82	20.44	2.94
	CP2	512	9	0.39	**3.49**	**22.29**	**3.57**

(a) *Blue* [24]

src	trans	K	D	u	SDR	SIR	SAR
s_1	STFT	4096	-	0.39	1.71	**16.11**	1.97
	MDCT	2048	-	0.39	1.65	14.80	2.00
	CP1	1024	8	0.39	1.70	15.19	2.03
	CP2	1024	8	0.39	**1.90**	14.56	**2.29**
s_2	STFT	2048	-	0.20	**0.74**	**11.81**	**1.37**
	MDCT	1024	-	0.30	-1.17	9.69	-0.36
	CP1	1024	8	0.30	-1.22	9.68	-0.40
	CP2	512	9	0.20	-2.46	8.57	-1.54
s_3	STFT	4096	-	0.39	**0.82**	5.70	**3.56**
	MDCT	1024	-	0.39	-0.36	5.79	1.86
	CP1	1024	8	0.39	-0.24	**6.07**	1.87
	CP2	512	9	0.39	-1.02	4.12	1.99
s_4	STFT	4096	-	0.39	**3.87**	**18.40**	4.09
	MDCT	2048	-	0.39	3.41	15.64	3.79
	CP1	1024	8	0.39	3.56	16.25	3.91
	CP2	1024	8	0.39	3.85	15.70	**4.26**

(b) *Carol* [3]

src	trans	K	D	u	SDR	SIR	SAR
s_1	STFT	32768	-	0.39	**11.14**	33.69	**11.17**
	MDCT	32768	-	0.39	10.33	**36.17**	10.35
	CP1	2048	7	0.39	8.64	31.70	8.66
	CP2	8192	5	0.39	10.72	28.34	10.80
s_2	STFT	8192	-	0.20	**7.83**	**30.50**	**7.86**
	MDCT	32768	-	0.30	6.05	20.12	6.27
	CP1	*	*	0.20	5.55	20.01	5.75
	CP2	*	*	0.20	5.55	20.01	5.75
s_3	STFT	32768	-	0.39	**6.49**	19.99	**6.73**
	MDCT	32768	-	0.39	5.42	**23.38**	5.50
	CP1	2048	7	0.39	4.91	16.94	5.27
	CP2	16384	4	0.39	6.16	21.03	6.33
s_4	STFT	32768	-	0.39	**5.79**	**21.47**	**5.94**
	MDCT	32768	-	0.39	4.06	16.59	4.40
	CP1	1024	8	0.39	1.73	16.00	2.01
	CP2	4096	6	0.39	3.30	13.76	3.89

(c) *Dreams* [10]

Table A.1: Results of source extraction. The STFT and MDCT block sizes are specified by K. The maximum depths of local cosine trees are given by D. For each maximum depth D there is a corresponding minimum block size; this is written in the K column. Rows marked with an asterisk (*) indicate that equal best results were obtained for several different search depths. For CP1, these were 3, 4, 9, 10, 11 and 12. For CP2, these were 11 and 12.

src	trans.	K	D	u	SDR	SIR	SAR
s_1	STFT	2048	-	0.39	2.26	**23.45**	2.31
	MDCT	4096	-	0.39	4.00	20.34	4.14
	CP1	1024	8	0.39	4.19	20.89	4.32
	CP2	256	10	0.39	**6.54**	22.68	**6.68**
s_2	STFT	2048	-	0.10	**-5.40**	4.28	-3.53
	MDCT	2048	-	0.20	-6.41	3.81	-4.47
	CP1	8192	5	0.10	-7.94	**5.61**	-6.69
	CP2	32768	3	0.30	-10.32	-4.82	**-2.81**
s_3	STFT	8192	-	0.39	**3.08**	11.41	**4.07**
	MDCT	2048	-	0.39	2.19	**11.90**	2.96
	CP1	2048	7	0.39	2.27	11.27	3.16
	CP2	2048	7	0.39	2.63	10.95	3.65
s_4	STFT	8192	-	0.39	**1.00**	**16.73**	**1.21**
	MDCT	4096	-	0.39	-0.74	10.64	-0.06
	CP1	4096	6	0.39	-0.87	11.43	-0.30
	CP2	512	9	0.39	-2.47	4.66	-0.26

(a) *Jiggly* [2]

src	trans.	K	D	u	SDR	SIR	SAR
s_1	STFT	32768	-	0.39	6.50	**27.61**	6.54
	MDCT	16384	-	0.39	**6.55**	26.88	6.60
	CP1	16384	4	0.39	6.33	25.26	6.40
	CP2	16384	4	0.39	6.52	24.21	**6.62**
s_2	STFT	16384	-	0.30	**3.44**	14.47	**3.95**
	MDCT	8192	-	0.30	2.23	15.88	2.53
	CP1	8192	5	0.20	2.38	**19.91**	2.50
	CP2	4096	6	0.20	1.64	16.12	1.90
s_3	STFT	32768	-	0.30	**-0.19**	**16.84**	**-0.01**
	MDCT	32768	-	0.39	-2.34	12.05	-1.91
	CP1	32768	3	0.39	-2.14	13.01	-1.79
	CP2	32768	3	0.39	-2.66	11.71	-2.21
s_4	STFT	32768	-	0.39	**18.16**	**29.52**	**18.49**
	MDCT	32768	-	0.39	16.85	28.85	17.14
	CP1	32768	3	0.39	17.44	29.13	17.75
	CP2	32768	3	0.39	17.44	29.13	17.75

(b) *Natmin* [19]

src	trans.	K	D	u	SDR	SIR	SAR
s_1	STFT	2048	-	0.39	9.14	**20.78**	9.49
	MDCT	2048	-	0.39	9.18	20.33	9.56
	CP1	2048	7	0.39	9.17	20.48	9.54
	CP2	1024	8	0.39	**9.41**	18.10	**10.11**
s_2	STFT	4096	-	0.39	**-1.52**	6.03	**0.29**
	MDCT	2048	-	0.39	-2.93	**6.87**	-1.64
	CP1	1024	8	0.39	-2.67	6.64	-1.28
	CP2	1024	8	0.39	-3.46	4.58	-1.42
s_3	STFT	4096	-	0.30	**-2.93**	**13.64**	**-2.65**
	MDCT	2048	-	0.39	-4.36	10.30	-3.82
	CP1	2048	7	0.39	-4.55	9.97	-3.98
	CP2	1024	8	0.30	-5.38	8.98	-4.70
s_4	STFT	2048	-	0.39	**15.44**	**28.47**	**15.67**
	MDCT	2048	-	0.39	14.51	25.62	14.87
	CP1	2048	7	0.39	14.47	25.70	14.82
	CP2	2048	7	0.39	14.36	23.69	14.92

(c) *Scoring* [8]

Table A.2: Results of source extraction (continued). The STFT and MDCT block sizes are specified by K. The maximum depths of local cosine trees are given by D. For each maximum depth D there is a corresponding minimum block size; this is written in the K column.

src	trans.	K	D	u	SDR	SIR	SAR
s_1	STFT	2048	-	0.39	10.21	25.45	10.35
	MDCT	1024	-	0.39	**10.31**	28.09	**10.39**
	CP1	512	9	0.39	9.92	**28.87**	9.98
	CP2	1024	8	0.30	9.31	22.47	9.55
s_2	STFT	2048	-	0.39	**8.88**	17.67	**9.57**
	MDCT	1024	-	0.39	8.06	20.15	8.38
	CP1	512	9	0.39	8.07	19.56	8.44
	CP2	1024	8	0.30	7.99	**22.04**	8.20
s_3	STFT	2048	-	0.10	**2.14**	19.46	**2.28**
	MDCT	256	-	0.10	-0.17	18.14	-0.04
	CP1	256	10	0.10	0.38	**21.21**	0.44
	CP2	256	10	0.10	-5.31	8.04	-4.48
s_4	STFT	2048	-	0.39	**7.77**	**35.92**	**7.78**
	MDCT	1024	-	0.39	6.66	26.09	6.72
	CP1	512	9	0.39	7.18	29.47	7.21
	CP2	512	9	0.39	5.81	18.76	6.10

(a) *Sun* [26]

src	trans.	K	D	u	SDR	SIR	SAR
s_1	STFT	4096	-	0.39	8.51	29.94	8.55
	MDCT	2048	-	0.39	8.88	29.53	8.93
	CP1	1024	8	0.39	8.72	**29.97**	8.76
	CP2	512	9	0.39	**9.89**	26.96	**9.99**
s_2	STFT	4096	-	0.20	**10.46**	24.35	**10.66**
	MDCT	1024	-	0.30	8.23	20.64	8.52
	CP1	1024	8	0.30	8.16	20.59	8.45
	CP2	512	9	0.10	7.59	**24.90**	7.69
s_3	STFT	4096	-	0.39	**1.54**	10.96	**2.40**
	MDCT	1024	-	0.39	-0.61	**11.72**	-0.07
	CP1	4096	6	0.39	-1.08	11.06	-0.48
	CP2	1024	8	0.39	-1.81	7.23	-0.48
s_4	STFT	4096	-	0.39	7.40	**26.09**	7.47
	MDCT	4096	-	0.39	7.07	23.05	7.21
	CP1	4096	6	0.39	7.08	23.09	7.21
	CP2	1024	8	0.39	**7.54**	16.22	**8.28**

(b) *We* [11]

Table A.3: Results of source extraction (continued). The STFT and MDCT block sizes are specified by K. The maximum depths of local cosine trees are given by D. For each maximum depth D there is a corresponding minimum block size; this is written in the K column.