SEMI-BLIND AUDIO SOURCE SEPARATION OF LINEARLY MIXED TWO-CHANNEL RECORDINGS VIA GUIDED MATCHING PURSUIT

Dimitri Zantalis, *

Audio Lab, Department of Electronics,
University of York
York, UK
dimitri@zantalis.com

Jeremy J. Wells

Music Research Centre, Department of Music,
University of York
York, UK
jez.wells@york.ac.uk

ABSTRACT

This paper describes a source separation system with the intent to be used in high quality audio post-processing tasks. The system is to be used as the front-end of a larger system capable of modifying the individual sources of existing, two-channel, multi-source recordings. Possible applications include spatial re-configuration such as up-mixing and pan-transformation, re-mixing, source suppression/elimination, source extraction, elaborate filtering, time-stretching and pitch-shifting. The system is based on a new implementation of the Matching Pursuit algorithm and uses a known mixing matrix. We compare the results of the proposed system with those from `mpd-demix` of the 'MPTK' software package and show that we get similar evaluation scores and in some cases better perceptual scores. We also compare against a segmentation algorithm which is based on the same principles but uses the STFT as the front-end and show that source separation algorithms based on adaptive decomposition schemes tend to give better results. The novelty of this work is a new implementation of the original Matching Pursuit algorithm which adds a pre-processing step into the main sequence of the basic algorithm. The purpose of this step is to perform an analysis on the signal and based on important extracted features (e.g frequency components) create a mini-dictionary comprising atoms that match well with a specific part of the signal, thus leading to focused and more efficient exhaustive searches around centres of energy in the signal.

1. INTRODUCTION

In the sound engineering field, sometimes the post-processing of an already made stereophonic recording is necessary. For example, in a live studio setting, a system that modifies spatial information contained in a pre-existing two-channel recording could be an invaluable tool to the engineer, saving time and money. The engineer could up-mix [1], [2] the recording making it suitable for reproduction over different formats or apply panning transformation [3], [4], e.g from level panning to delay-based panning. Spatial re-configuration could also benefit consumers in the domestic listening environment. It is a fact that listening trends tend to vary and evolve over time thus it is highly desirable to be able to modify pre-recorded material. Apart from spatial effects other types of processing are source suppression/elimination (e.g. Karakoe system) and individual source modification such as filtering, changing/correcting pitch of single/multiple instrument(s), time stretching/compressing etc.

* This work was funded by EPSRC

All the aforementioned problems and probably many more, could be solved using a three stage approach: source separation followed by post-processing and finally remixing. A simple example system would have a coincident pair stereo recording as its input, separate the sources within the mixture and re-mix the separated sources using a different spatial configuration (e.g. by applying delays to produce time-difference panning) [5], [6]. It is clear that the crux of the system is the source separation step which should produce high quality results as this would most probably affect the quality of any subsequent processing.

Source separation is one of the trickiest types of signal processing and it is a vast field. Of course it would be extremely difficult to devise a solution that handles all cases of source separation and if such a system comes to life it would probably be a hybrid of parametrized and statistical modeling techniques and everything in between. Because of the complexity of the problem we need to state some assumptions, minimize the requirements and design a system that is realizable and scalable.

1.1. Mixing Model

In this work we assume an instantaneous mixing model with no delay-time parameter and no noise term:

$$x_m[n] = \sum_{p=1}^{P} \alpha_{mp} s_p[n] \,, \quad 1 \leq m \leq M \tag{1}$$

where $x_m[n]$ are the mixture signals, $s_p[n]$ the original source signals α_{mp} are the mixing coefficients, M the number of mixtures and P the number of sources. This can be expressed more compactly using matrix notation:

$$\mathbf{x} = \mathbf{A} \cdot \mathbf{s} \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^{M \times N}$ are the mixture signals, $\mathbf{A} \in \mathbb{R}^{M \times P}$ is the mixing matrix where each element is a mixing coefficient α_{mp} and $\mathbf{s} \in \mathbb{R}^{P \times N}$ are the source signals. Equation 2 also makes the connection between the problem at hand and linear algebra where, the mixture signals can be seen as linear combinations of the source signals. The problem of source separation is essentially the estimation of the mixing matrix $\mathbf{A}$ and recovery of the sources $\mathbf{s}$ given the mixture $\mathbf{x}$. The problem of estimating both $\mathbf{A}$ and $\mathbf{s}$ is known as 'Blind Source Separation (henceforth BSS) and when audio signals are involved as 'Blind Audio Source Separation' (BASS).

1.2. Number of Mixtures & Sources

Source separation problems are classified based on the number of mixtures and sources. In this work we deal with two-channel

recordings comprising multiple sources, thus $M = 2$ and $P > M$. This leads to the under-determined case of source separation which in general is not a trivial task. In this case the inverse mixing matrix cannot be directly used to estimate the original sources. Instead other ways are employed to estimate the mixing matrix and separate the sources. Usually techniques based on 'Sparse Component Analysis' (henceforth SCA) are used since they produce good results for this source separation case [7]. In this work we use ideas from this field of study.

1.3. Known Mixing Matrix

The proposed system has knowledge of the mixing matrix $\mathbf{A}$. In this case we have a semi-blind source separation problem (SBSS or SBASS for audio signals). Estimation of the mixing matrix is usually treated as a separate problem to recovery of sources and many algorithms can handle this specific task with very good results (TIFROM [8], DUET [9], DEMIX [10], [11], [12] etc.). In fact we could estimate the mixing matrix with a modification of the proposed algorithm but this is outside the scope of this paper and will not be pursued any further. Generally speaking though we can safely assume that the mixing matrix is known or can be estimated accurately.

We also assume that the recording was made using a Blumlein pair, i.e. two 'figure-of-eight' microphones angled at 90°. This is a famous microphone technique that produces very accurate imaging of sources in the front quadrant and is widely used. A source is positioned in space, in-front of the listener, using inter-channel level differences alone, thus it is in accord with the mixing model presented in section 1.1. For the particular case of two-channel mixtures the mixing coefficients are given by:

$$\alpha_{1p} = \frac{\psi_p}{1+\psi_p}, \quad \alpha_{2p} = 1 - \psi_p \tag{3}$$

with

$$\psi_p = \frac{1+tan(\theta_p)}{1-tan(\theta_p)}, \quad -45^\circ \leq \theta_p \leq 45^\circ \tag{4}$$

where θ_p is the direction of the p^{th} source.

1.4. Sparsity of Sources

Another important assumption is that the source signals can be sparsely represented in a suitable domain. A signal is considered sparse in a domain if only few coefficients are needed to represent that signal in that domain. For example speech signals can be sparse in the time domain (e.g. two speakers talking in turns) whereas music cannot. Music exhibits sparsity in different domains such as the time-frequency domain. The notion of sparsity plays a central role in many signal processing fields including BSS and SCA techniques. In fact, regarding BSS, it has been shown that better separation can be achieved by exploiting sparse representations of signals [13]. By representing the signals in a sparse domain we hope that the coefficients of individual sources will be much more distinguishable (i.e we can see time-frequency regions where a source dominates) thus much easier to separate. After separation in the sparse domain, usually performed using frequency masks or clustering algorithms, we invert the separated sources back into the time domain to get the estimates. This is the main idea behind many SCA techniques.

Based on these assumptions we propose a semi-blind audio source separation algorithm that deals with linear, instantaneous mixtures and uses a new software implementation of Matching Pursuit (MP)[14] as its front-end. Similar algorithms that use MP for source separation are the 'Stereo Matching Pursuit'[15], the algorithms proposed in [16] which work with knowledge of the mixing matrix and the algorithm in[17].

Some other systems that are designed for active listening applications but with some overlapping goals are DReaM [18] and MPEG Spatial Audio Object Coding (SAOC) [19]. The fundamental difference against our algorithm is that these systems are based on encoder/decoder schemes. In particular inaudible meta-parameters are embedded within a mixture during the encoding stage and then used in the decoding stage for post processing such as re-mixing and respatialisation. Source separation algorithms based on encoder/decoder schemes are referred to as 'Informed Source Separation' and differ significantly from BASS and SBASS algorithms.

The rest of the paper is organized as follows: In section 2, we describe the original MP algorithm. In section 3, we introduce a modified MP version and make comparisons with basic MP algorithms. In section 4, we show how to apply the new proposed MP implementation in a source separation context. The final sections are for results obtained from our experiments, discussion on future work and conclusion.

2. BASIC MATCHING PURSUIT

Matching Pursuit is a recursive, adaptive algorithm for sparse signal decompositions. It belongs to a family of techniques known as 'Atomic Decompositions' (aka sparse decompositions or sparse atomic decompositions) that aim to decompose a given signal $\mathbf{x}$ as a linear combination of elementary waveforms $(g_\gamma)_{\gamma\in\Gamma}$, called atoms, taken from a dictionary $\mathbf{D}$. This can be formally expressed as [20]:

$$\mathbf{x} = \sum_{\gamma\in\Gamma} c_\gamma g_\gamma \tag{5}$$

where γ is a set of parameters characterizing each atom, g_γ are the individual atoms and c_γ are the expansion coefficients. We can also get an approximate decomposition for a fixed number of atoms m:

$$\mathbf{x} = \sum_{i=1}^{m} c_{\gamma i} g_{\gamma i} + R^{(m)} \tag{6}$$

where $R^{(m)}$ is a residual after an m-*term* decomposition. Matching Pursuit and similar algorithms, aim to find a sub-optimal solution to (5).

For basic MP we let $\mathbf{D} = \{g_\gamma | \gamma \in \Gamma\}$ be a dictionary comprising atoms of unit norm, $\|g_\gamma\| = 1$, for all $g_\gamma \in \mathbf{D}$. We also let the set of atoms in $\mathbf{D}$ to be redundant, i.e we have an over-complete dictionary. Decompositions in over-complete dictionaries are not unique since some atoms might be linearly dependent. MP will recursively build the approximation signal, one atom at a time, choosing at every iteration step the atom that minimizes $\|R^m\|$ in (6). In basic MP we first choose/create a dictionary $\mathbf{D}$, initialize $R^0 = \mathbf{x}$ and for each iteration step i we proceed as follows:

1. Compute inner products $\left\langle R^{i-1}, g_\gamma \right\rangle$, for all $g_\gamma \in \mathbf{D}$.
2. Select best atom $g_{\gamma i} = \arg\max_{g_\gamma \in \mathbf{D}} \left|\left\langle R^{i-1}, g_\gamma \right\rangle\right|$.
3. Get expansion coefficient $c_i = \left\langle R^{i-1}, g_{\gamma i} \right\rangle$.
4. Update the residual $R^i = R^{i-1} - c_i g_{\gamma i}$.
5. Check for exit conditions. If none is met continue to next iteration, otherwise stop decomposition.

We see that the standard inner product is used to compare the signal with the atoms in the dictionary. Thus the atom that maximizes the inner product is the one that minimizes the residual. MP is said to be a greedy algorithm in a sense that at every iteration it chooses the atom that removes the most energy from the residual[21]. We should note that MP can be configured to use other atom selection criteria and we will mention some of them in our proposed method. For a more detailed mathematical explanation of MP the reader is referred to [14].

Another important aspect of MP, and similar algorithms for that matter, is the choice or creation of the dictionary $\mathbf{D}$. The classic dictionary proposed in [14] is the Gabor dictionary which is parametric in nature; that is a set of parameters are needed to describe or create atoms of that type. A real Gabor atom is given by [15]:

$$g_{s,u,\xi,\phi} = K_{s,\xi,\phi} w(\frac{t-u}{s}) \cos(2\pi\xi(t-u)+\phi) \qquad (7)$$

where s is the scale parameter (i.e length of the atom), u the location parameter (i.e. location within the signal), ξ the frequency and ϕ the phase of the atom. $w(t)$ is generally any normalized window but for Gabor atoms a Gaussian window is used and can be defined as [22]:

$$w(t) = (\pi\sigma_s)^{-0.25} e^{-\frac{t^2}{2\sigma_s}} \qquad (8)$$

where σ_s is the variance of the window:

$$\sigma_s = (\frac{4}{\pi}) 2^{2(s_0-s)}, \quad s = 0, 1, 2, ...s_0 \qquad (9)$$

and s_0 depends on the application. In this work we set $s_0 =$ `nextpow2(N)-2`, where `N` is the maximum length of an atom in samples. Finally $K_{s,\xi,\phi}$ in (7) is a normalizing constant, set so that the atom has unit norm. Other parametric dictionaries are the Fourier dictionary, Chirplet dictionary, DCT and DST dictionaries, Gamma-tone and Gamma-chirp dictionaries etc. each having its own set of parameters. Other methods for creating non-parametric dictionaries exist but for the proposed MP implementation we are mostly interested in parametric ones.

Creating a parametric dictionary covering all possible parameter values would be impractical so usually the parameter space of a dictionary is discretised. For example for the Gabor dictionary we could include all atoms with scales $N = 2^s$ with $s = 1, ...S$, frequencies $\omega_k = 2\pi k/N$ for $k = 1...N/2$ and shift locations u every $N/4$ samples. Even so such dictionaries can become very large, especially when we consider joining multiple dictionaries, making the realization of MP almost impossible. The state of the art, of a publicly available MP implementation, is the 'Matching Pursuit ToolKit' (MPTK) [23] which is very fast and has support for multi-channel signals and multiple dictionaries. We use the MPTK in this work as a reference system.

3. GUIDED MATCHING PURSUIT

A modification of the basic MP algorithm is proposed where a pre-processing step is included as the first step in the main sequence of events. At every iteration, the pre-processing step performs some kind of analysis to the residual and extracts important information which is then used to create a mini-dictionary $\mathbf{D}_i$ containing a fraction of the atoms that exist in the original dictionary $\mathbf{D}$. For example the pre-processing step could be a Fourier analysis of the residual, where the frequency components with the maximum magnitude can be used to create the atoms in $\mathbf{D}_i$. In this particular example we choose the frequency components with maximum magnitude since these are most likely to contain a big portion of the signal energy. The idea is that the newly created atoms will correlate well with the corresponding frequency components of the residual. The pre-processing step acts as a guide for creating atoms that might best correlate with the features of the signal we are interested in, therefore we term this new approach as 'Guided Matching Pursuit' (henceforth GMP). Although this is a simple modification of basic MP, this approach has some interesting properties and allows for novel signal decompositions and transformations.

For this work we use the fast Fourier transform (FFT) (or short-time Fourier transform (STFT) for long signals) as the analysis of the pre-processing step, since this is a very simple and fast analysis we can perform on a signal and, as already mentioned, can give us information about the frequency of important partials in the residual. Note that the analysis part can be more elaborate; for example a phase vocoder analysis step or a re-assigned magnitude spectrum could be used to get the instantaneous frequencies of partials instead of the frequencies corresponding to the frequency bins of the FFT. Also other information could be used for the creation of atoms such as the phase obtained from the complex spectra. Some of these options have been tested and in some cases lead to much better results than when a simple FFT is used, but these are not consistent. This issue requires further study. Although we use steps similar to classic sinusoidal analysis systems, GMP differs significantly from these in that the resulting decomposition goes beyond the sinusoidal plus transient plus residual model usually proposed by these systems. The modeling of the underlying audio signal depends on the selection of the dictionary which can contain many different types of atoms (e.g. Gabor atoms, chirp atoms, harmonic atoms, wavelets, learned atoms etc.).

A good property of this method is that every $\mathbf{D}_i$ contains a set of atoms which is much smaller than an ordinary dictionary implementation. To put it in perspective a normal Gabor dictionary, discretized as mentioned earlier, would contain tens of thousands of atoms whereas with our method each $\mathbf{D}_i$ can contain as few as 5 atoms per iteration (e.g. $1\ frequency\ \times 5\ scales$). This is a big reduction in the number of atoms we need to correlate with, which for a 'textbook' implementation of MP is a huge reduction in computation. Also with our method it is much easier to create dictionaries comprising different atom types. Of course the atoms should be mathematically defined (i.e. have parameters that describe them) but this should not pose a problem since many interesting dictionaries exist that can represent a wealth of signal features and share a similar structure and parameter space. For example some possible atoms that can be used are Gabor atoms, Fourier atoms (i.e. Sine and Cosine atoms), DCT and DST atoms, Gaussian chirps, damped sinusoids, Gamma-tones, Gamma-chirps and FM atoms.

Another trick that we employ is the computation of groups of inner products using the FFT, as mentioned in [23]. We know that the inner product of two real, square-integrable functions $f(x)$ and $g(x)$, on an interval $[a, b]$ is given by:

$$\langle f, g \rangle = \int_a^b f(x)g(x)dx \tag{10}$$

We also know that the cross-correlation between two continuous functions f and g is the 'sliding inner product' of those two functions:

$$f(t) \star g(t) = \int_{-\infty}^{+\infty} f(\tau - t)g(t)d\tau \tag{11}$$

Finally comparing the cross-correlation with the convolution of two continuous functions f and g:

$$f(t) * g(t) = \int_{-\infty}^{+\infty} f(t - \tau)g(\tau)d\tau \tag{12}$$

we see that the only difference is the reversal of f for the convolution operation. Thus cross-corralation and convolution are related by:

$$f(t) \star g(t) = f(-t) * g(t) \tag{13}$$

and in the case where f is Hermitian (which implies a symmetric real part) then the cross-correlation and convolution operations are the same. By taking advantage of the convolution theorem and using a fast linear convolution algorithm we can compute groups of inner products really fast. Also since fast convolution computes the inner products between the atoms and the residual for every sample, the need to shift atoms along the residual is eliminated; that is the atom location parameter u is implied by the location of the correlation coefficient with the maximum absolute value (step 2 in basic MP).

Let $\mathbf{x} \in \mathbb{R}^N$ be a short duration, mono-channel input signal, N be the length of the signal in samples, $R(t)$ be the residual after the decomposition, $R(\omega)$ the Fourier transform of the residual, k a frequency bin index, $\mathbf{D}_i$ a mini-dictionary, g_γ the atoms in the dictionary, $\mathbf{C} \in \mathbb{R}^{N \times M}$ a matrix holding the cross-correlations between the residual and each atom in the dictionary, then the steps for a GMP implementation are as follows:

1. Initialise $R^0(t) = \mathbf{x}$, $i = 1$.
2. Compute FFT of residual: $R^{i-1}(\omega) = FFT\left(R^{i-1}(t), N\right)$ with N being the size of the FFT.
3. Select frequency bin with maximum magnitude $k = \arg\max_{k \in R(\omega)} \left|R^{i-1}(\omega)\right|$.
4. Create mini-dictionary $\mathbf{D}_i$ comprising real atoms (of possibly different types) with normalized frequency k/N and different scales.
5. Compute cross-correlations of the residual with all atoms in $\mathbf{D}_i$: $\mathbf{C}_i = XCORR\left(R^{i-1}(t), \mathbf{D}_i\right)$.
6. Select best atom $g_{\gamma i} = \arg\max_{g_\gamma \in \mathbf{D}_i} |\mathbf{C}_i|$.
7. Get expansion coefficient $c_i = \left\langle R^{i-1}(t), g_{\gamma i} \right\rangle$.
8. Update the residual $R^i(t) = R^{i-1}(t) - c_i g_{\gamma i}$.
9. Check for exit conditions. If none is met increase iteration number i by one and go to step 2, otherwise stop decomposition.

Steps 2, 3 and 4 collectively form the analysis step we discussed earlier in its simplest form. Note that these steps could be different depending on the information we want to extract from the residual. Also in this case we assume that the residual is a short signal (e.g. $N = 2048$ samples). If the residual is very long then we should apply the STFT and step 3 would select the frequency bin with the maximum magnitude from a single time frame or maybe select the frequency bins with maximum magnitude from each time frame, for multiple atom extraction (in contrast to single atom extraction of original MP). It is clear that adding a preprocessing step to the basic MP opens up new ways of looking for specific features in a given signal.

We compare the proposed algorithm against MATLAB's ® `wmpalg` [24] (hencefoth WMP) and MPTK [23]. A 2048 samples long snippet of the 'o' vowel, is decomposed using a Fourier dictionary for 20 iterations (i.e. 20 atoms in the decomposition). Table 1 shows the residual energy and the signal to residual ratio at the last iteration and the time taken for each MP implementation to complete. We can see that GMP and MPTK perform better compared to WMP with GMP giving better results overall. MPTK is faster but we should take into account that MPTK is written in C++ whereas GMP and WMP are written in MATLAB's ® `mcode` thus they are not optimised for speed. Having said that we can see that GMP performs much faster than WMP. Figure 1 shows how the energy of the residual decays with every iteration and figure 2 shows how the SRR increases with every iteration. Again we can see GMP performing better compared to MPTK and WMP.

A MATLAB® implementation of GMP with all files that produce these and subsequent results can be found in [25].

Table 1: *Metrics for different MP algorithms after 20 iterations.*

Algorithm	Res. enrg. (dB)	SRR (dB)	time (sec)
GMP	0.6441	13.83	0.89
MPTK	0.9748	12.03	0.39
WMP	1.8315	9.29	1.75

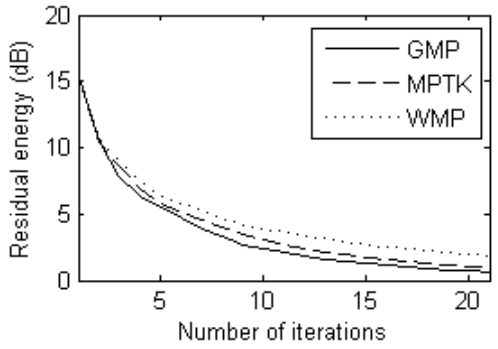

Figure 1: *Residual energy decay curves for three different MP implementations.*

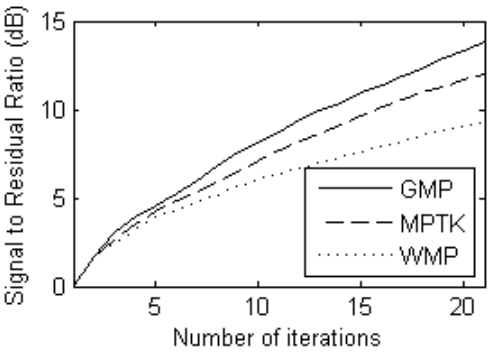

Figure 2: *Signal to residual ratio (SRR) for three different MP implementions.*

4. APPLICATION OF GMP IN SBASS

So far we have described how GMP works and we have shown that we get desirable results when the algorithm is used for signal decomposition. In this section we explain how to modify GMP to work with a source separation problem.

First of all we assume that we deal with two-channel recordings so the first modification regards an extension of the algorithm that works with multi-channel signals. In GMP this is easily achieved by applying the analysis steps to all channels of a signal. In particular we alter step 2 of GMP to compute the Fourier transform of both channels, then we add the resulting spectra and step 3 selects the frequency bin with the maximum magnitude from the new combined magnitude spectrum. Another approach could be to select frequency bins from each channel spectrum and the resulting dictionary would contain atoms with corresponding frequencies. The former approach was used since it makes sure that the frequency selection step is not biased by a particular source direction. The algorithm continues with the creation of the dictionary $\mathbf{D}_i$ and the cross-correlation of $\mathbf{D}_i$ with each channel of the residual.

Let us focus on step 5 of GMP. Assuming $\mathbf{D}_i$ holds M atoms then $\mathbf{C}$ is an $N \times M$ matrix where each column holds the N samples long cross-correlation of an atom with the residual. In the multi-channel case $\mathbf{C}$ becomes a $N \times M \times J$ matrix where the third dimension represents channels with $J = 2$ for a two-channel signal. Also remember that cross-correlation can be thought of as a 'sliding inner product', so every sample in each column of $\mathbf{C}$ is effectively the inner product between an atom and the residual signal at a particular time instance n , $\forall n \in \{1..N\}$. Thus the correlation samples in $\mathbf{C}$ are all potential expansion coefficients. We will therefore refer to that matrix as the coefficient matrix $\mathbf{C}$. Because of our instantaneous mixing model assumption in section 1.1 we can use the coefficient matrix to calculate the estimated directions of each expansion coefficient pair (i.e. left and right channel coefficients at same time instance) using:

$$\mathbf{\Theta} = \arctan\left(\frac{|\mathbf{C}_{n,m,2}|}{|\mathbf{C}_{n,m,1}|}\right) - \frac{\pi}{4}\,, \forall n \in \{1..N\},\ \forall m \in \{1..M\}. \tag{14}$$

where $\mathbf{\Theta} \in \mathbb{R}^{N \times M}$ and the constant $\pi/4$ is subtracted in order to bring the estimated directions in the range of $-\pi/4 \leq \mathbf{\Theta}_{n,m} \leq \pi/4$ which stems from our assumption in section 1.3. What we are interested in, is the distribution of the values in each column of $\mathbf{\Theta}$. Figure 3 shows the histograms of three columns of $\mathbf{\Theta}$ (which implies a mini-dictionary with three atoms) obtained by an example mixture with four sources equidistantly spaced in the front quadrant. We can clearly see that most values are clustered around specific directions; in this example around -11.25° and 11.25° which are two of the known mixing directions.

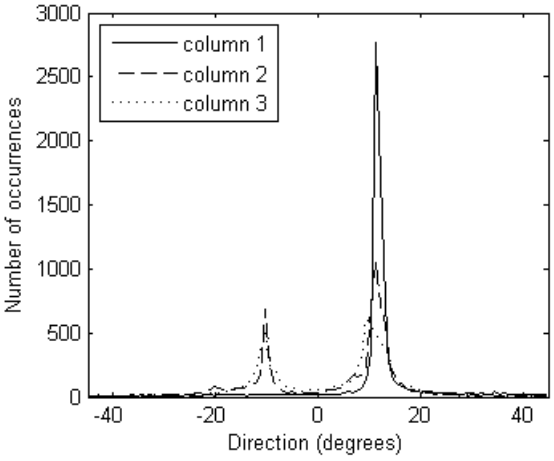

Figure 3: *Histograms of values in columns of* $\mathbf{\Theta}$

Having this information at hand we can proceed with the modification of step 6 of GMP. As already mentioned the original MP algorithm selects at each iteration, the atom that minimizes the energy of the residual, but this is not a strict requirement. What we are after are alternative selection criteria that take advantage of our known mixing matrix and the estimated atom directions. In this paper we test three different atom selection criteria.

In the first case we select the pair of expansion coefficients that are closest to a known direction and have the maximum absolute magnitude. The idea is that if a pair of coefficients is close to a known direction then there is a higher chance that the corresponding atom belongs to the source indicated by that direction. Also selecting an atom with the maximum magnitude (i.e. high energy) makes sure that the algorithm will converge fast. Let N be the maximum length of an atom (in samples), M be the number of atoms in the mini-dictionary and P the number of sources. Also let $\theta \in \mathbb{R}^P$ be a vector of P known directions. Then we calculate:

$$\mathbf{M} = \sum_{j=1}^{2} |\mathbf{C}_{n,m,j}| \tag{15}$$

$\forall n \in \{1..N\}\,, \forall m \in \{1..M\}$ and

$$\mathbf{d}_{n,m,p} = |\mathbf{\Theta}_{n,m} - \theta_p| \tag{16}$$

$\forall n \in \{1..N\}\,, \forall m \in \{1..M\},\ \forall p \in \{1..P\}$,
where $\mathbf{d} \in \mathbb{R}^{N \times M \times P}$ is a matrix holding the distances of each estimated direction in $\mathbf{\Theta}$ from the known directions θ. Then we calculate:

$$\mathbf{E} = \frac{\mathbf{M}}{\mathbf{d}} \tag{17}$$

where $\mathbf{E} \in \mathbb{R}^{N \times M \times P}$. The indices of the maximum value in $\mathbf{E}$ indicate the extraction location in the signal (in samples), the atom to choose from the dictionary (which implies the atom parameters such as type, scale, frequency, phase etc.) along with the

expansion coefficients (obtained from $\mathbf{C}$) and the source the atom belongs to. At this point we should mention that because of our sparsity assumption in section 1.4 each atom is allocated to only one source.

In the second case we favor atoms that appear to be coming from a dominant source. In order to find the dominant source we first find the maximum value of each histogram in $\mathbf{\Theta}$ and then subtract from the known directions θ. The minimum absolute value of that result indicates the source to allocate to. Then we use equations 15, 16 and 17 to obtain the atom to extract, the expansion coefficients and the location to extract from. Note that in this case $\mathbf{d}$ and $\mathbf{E}$ become $N \times M$ matrices since the source to allocate to has already been calculated, thus the third dimension reduces to 1.

In the final selection criterion we start by choosing the 'best' histogram for obtaining the expansion coefficients. The choice is based on the shape of the histograms obtained from $\mathbf{\Theta}$. In particular we favor histograms with values concentrated on one direction only. For example looking at figure 3 we see that the values of column 1 of $\mathbf{\Theta}$ are concentrated around direction 11.25° degrees whereas columns 2 and 3 produce peaks at two different directions. The idea is that if all or most of the estimated directions of a particular column of $\mathbf{\Theta}$ are clustered around one direction only then this is a strong indication that the corresponding atoms belong to the source indicated by that direction. Thus in this particular example the algorithm will select the expansion coefficients that correspond to column 1 of $\mathbf{\Theta}$ as candidates. We select the 'best' histogram h as follows:

$$h = \max \left(\frac{\sum_{n=1}^{N} \mathbf{M}_{n,m}}{\min \left(\frac{\sum_{n=1}^{N} \left| \mathbf{\Theta}_{n,m} - \theta_p \right|}{N} \right)} \right) \tag{18}$$

$\forall m \in \{1..M\}$, $\forall p \in \{1..P\}$. Having found the histogram to operate upon we use equations 15, 16, 17 with fixed $m = h$ and obtain $\mathbf{E}$ which is now a vector of length N. The index corresponding to the maximum value of $\mathbf{E}$ will give us the atom extraction location (in samples). Finally using the found location and h we can obtain the expansion coefficients from $\mathbf{C}$.

All three selection criteria presented here, provide us with the expansion coefficients and parameters of the atom to extract, the extraction location in the signal and finally the source that the atom belongs to. A new step is introduced where the selected atom is added to an approximated source. Note that the number of approximated sources will be the same as the number of mixing directions. Finally the algorithm proceeds with updating the residual and checking for exit conditions before continuing to the next iteration.

5. EXPERIMENTS

For the simulation a mixture comprising four sources was used. The sources were obtained from [26] and are anechoic recordings of a clarinet, a violin, a soprano and a viola. The sources are sampled at 44.1kHz and segments of 2^{17} samples (approx. 3 seconds) were used. The mixture was created using the mixing model in section 1.1 with a mixing matrix produced using equations 3 and 4. The sources were mixed so that they were equidistantly spaced in the front quadrant; a situation similar to a string quartet recording.

The same mixture was processed using three different source separation algorithms. All algorithms use the mixing matrix as prior information. We should also mention at this point that these algorithms have many parameters that can affect the outcome of the separation. In this experiment we tested all algorithms using various configurations and the best results are presented. The first algorithm used can be found in [5]. It uses the STFT as its front-end and performs source separation based on the directions estimated by the magnitude spectra of the mixture channels. It was found that a good setting for the particular mixture was an FFT size of 4096 with 75% overlap using a hanning window. The second algorithm we test against can be found in [16] and uses MP as its front-end. For this algorithm we used a Gabor dictionary (similar to that expressed by equation 7) comprising atoms with six scales, from $s = 512$ until 16384 samples with a 50% window-shift between atoms (see [23] for how to setup a dictionary in MPTK). The third algorithm is the one proposed in this paper. In order to be as fair as possible the proposed algorithm was set-up to use a Gabor dictionary comprising atoms up to six different scales. The algorithm also operated in the 'Short-Time Matching Pursuit' (STMP) mode were the signal is split into frames and each frame is processed separately in a similar fashion to the STFT. This is in contrast to MPTK where the signal is processed as a whole. Also for this example, GMP produced the 'best' results using the second atom selection criterion that was described in section 4

Because we are interested in high quality source separation for audio post-processing we used the PEASS toolkit [27], [28] for evaluating the performance of each algorithm. The toolkit produces the standard SDR,SIR,SAR and SNR measures but most importantly it calculates a set of perceptually motivated subjective measures which correlate better with human assessments. In particular it calculates the Overall Perceptual Score (OPS), Target related Perceptual Score (TPS), Interference related Perceptual Score (IPS) and Artifact related Perceptual Score (APS). Table 2 shows the SDR and SIR measures and table 3 the OPS, TPS and IPS scores produced by PEASS toolkit. The best scores are marked in bold.

Table 2: SDR and SIR performance measures (values in dB).

		SDR			SIR	
Src	STFT	MPD	GMP	STFT	MPD	GMP
1	5.83	**8.93**	7.16	**14.23**	10.93	12.16
2	2.75	2.98	**3.91**	**7.28**	5.23	6.09
3	5.77	10.24	**13.83**	**18.41**	18.13	17.21
4	4.97	9.96	**9.98**	**15.54**	12.96	14.51

By quick inspection of the tables there is no clear 'winner' algorithm. Having said that there are some interesting points we can talk about. First of all we can see that overall the algorithms that use MP as their front-end perform better. Also in the SIR case we see that the STFT algorithm produces better results by a small margin. This verifies to some extent the claim that algorithms which use adaptive decomposition schemes as their front-end tend to produce better results. We should also take into account that the mpddemix and GMP algorithms were used with their most basic settings, that is they use only one dictionary with limited number of atom variations. We expect the results to get better if we let the MP based algorithms run with multiple dictionaries comprising various atoms. These tests have yet to be performed. We should also mention that the STFT based algorithm was implemented to

Table 3: OPS, TPS and IPS performance scores (all scores out of 100)

		OPS			TPS			IPS	
Src	STFT	MPD	GMP	STFT	MPD	GMP	STFT	MPD	GMP
1	27.09	25.19	**32.57**	46.83	**53.41**	43.14	24.02	26.14	**59.76**
2	**25.81**	22.66	24	21.95	11.44	**30.82**	33.59	47.73	**66.02**
3	47.89	**70.36**	52.93	**80.21**	69.17	67.48	80.1	**80.94**	68.45
4	36.08	35.03	**36.13**	52.14	**57.83**	45.91	46.65	45.89	**51.68**

be used with this particular example. In particular the clustering of the coefficients that is performed in the STFT based algorithm is specifically designed to work with a mixing matrix that evenly places sources in the front quadrant. The MP based algorithms do not have this limitation and can operate using any mixing matrix.

Regarding the perceptually motivated evaluation scores we see again that the MP-based algorithms produce better results. Comparing MPD and GMP again does not show a clear 'winner' because sources obtain high scores in both algorithms. These are encouraging results for the proposed implementation since we test it against a well established source separation algorithm that uses MP. An interesting observation is that the IPS results show that GMP performs better on all sources but one. The interference related perceptual score is very important when we deal with high-quality source separation because it implies that the separated sources do not suffer from bleeding from other instruments. Informal listening tests have verified that to some extent. Audio examples can be found in [25].

6. CONCLUSION AND FUTURE WORK

In this paper we described a new method for decomposing multichannel audio signals using a variant of the basic Matching Pursuit algorithm. The new approach, which we term 'Guided Matching Pursuit', uses a pre-processing step to gather information about the signal and create a mini-dictionary comprising atoms that are expected to correlate well with the signal. We compared the new decomposition method with two accepted MP implementations and showed that we get better results regarding the signal to residual ratio and the residual energy decay rate. We further described how to apply the new decomposition method in a source separation problem by using three different atom selection criteria that take advantage of a known mixing matrix. We tested and compared the proposed algorithm against two available source separation algorithms that work using same principles and showed that we get similar results and in some cases better perceptual evaluation scores.

At the time, only one mixture has been tested. In particular this is an instrumental mixture comprising sources with quasi-periodic content which is expected to be decomposed well using a Fourier or Gabor dictionary. It will be of great interest to try the algorithm using mixtures that contain transients such as percussion content. To that extend we also want to try the algorithm using dictionaries comprising many atom types such as multi-scale Gabor atoms, windowed multi-scale Fourier atoms, damped sinusoids, chirplets, gamma-tones, gamma-chirps, and fm atoms, in which case we expect to see an increase in quality scores. Also at this point the algorithm takes a big amount of time to complete, which for our purposes at the moment does not pose a problem, but a faster implementation should be considered. This will be achieved by optimizing the code and possibly re-writing the algorithm using a faster language such as C. Finally, since our goal is audio post-production, a next step would be to try out the source separation algorithm in that context and perform subjective listening tests.

7. REFERENCES

[1] Avendano and Jot, "Frequency domain techniques for stereo to multichannel upmix," *Audio Engineering Society Conference: 22nd International Conference: Virtual, Synthetic, and Entertainment Audio*, Jun 2002.

[2] Avendano and Jot, "A frequency-domain approach to multichannel upmix," *J. Audio Eng. Soc.*, vol. 52, no. 7, pp. 740–749, 2004.

[3] Avendano C., "Frequency-domain source identification and manipulation in stereo mixes for enhancement, suppression and re-panning applications," in *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, Oct 2003, pp. 55–58.

[4] Wells J.J., "Modification of spatial information in coincident pair recordings," *J. Audio Eng. Soc.*, 2010.

[5] Jeremy J. Wells, "Directional segmentation of stereo audio via best basis search of complex wavelet packets," *J. Audio Eng. Soc.*, May 13-16 2011.

[6] Jeremy J. Wells, "A comparison of analysis and re-synthesis methods for directional segmentation of stereo audio," in *Proc. of the 14 th Int. Conference on Digital Audio Effects (DAFx-11), Paris, France*, Paris, France, Sept. 19-23 2011.

[7] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, chapter Sparse Component Analysis, pp. 367–412, Academic Press, 2010.

[8] F. Abrard and Y. Deville, "Blind separation of dependent sources using the "time-frequency ratio of mixtures" approach," in *Signal Processing and Its Applications, 2003. Proceedings. Seventh International Symposium on*, 2003, vol. 2, pp. 81–84.

[9] Yilmaz, "Blind separation using time-frequency masks," *IEEE Transactions on Signal Processing*, 2004.

[10] Arberet S. Gribonval R. Bimbot F., "A robust method to count and locate audio sources in a stereophonic linear instantaneous mixture," in *ICA*, 2006, pp. 536–543.

[11] Arberet S. Gribonval R. Bimbot F., "A robust method to count and locate audio sources in a stereophonic linear anechoic mixture," in *Proc. IEEE Intl. Conf. Acoust. Speech*

Signal Process (ICASSP'07), Honolulu, Hawai, Etats-Unis, 2007, pp. 745–748.

[12] Arberet S. Gribonval R. Bimbot F., "A robust method to count and locate audio sources in a multichannel underdetermined mixture," *Signal Processing, IEEE Transactions on*, vol. 58, no. 1, pp. 121–133, 2010.

[13] M. Zibulevsky B.A Pearlmutter P. Bofill and P Kisilev, *Independent Component Analysis: Principles and Practice*, chapter Blind source separation by sparse decomposition, S. J. Roberts and R.M. Everson eds., Cambridge, 2001.

[14] Mallat S.G. and Zhang Zhifeng, "Matching pursuits with time-frequency dictionaries," *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397 –3415, 1993.

[15] R. Gribonval, "Sparse decomposition of stereo signals with matching pursuit and application to blind separation of more than two sources from a stereo mixture," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, 2002, vol. 3, pp. 3057–3060.

[16] Sylvain Lesage, Sacha Krstulovic, and Remi Gribonval, *Under-Determined Source Separation: Comparison of Two Approaches Based on Sparse Decompositions*, vol. 3889, pp. 633–640, Springer Berlin Heidelberg, 2006.

[17] P. Sugden and N. Canagarajah, "Underdetermined noisy blind separation using dual matching pursuits," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, 2004, vol. 5, pp. 557–560.

[18] Marchand S. et al, "Dream: A novel system for joint source separation and multitrack coding," *Audio Engineering Society Convention 133*, Oct 2012.

[19] Breebaart J. et al, "Spatial audio object coding (saoc) - the upcoming mpeg standard on parametric object based audio coding," *Audio Engineering Society Convention 124*, May 2008.

[20] Scott Shaobing Chen, David L. Donoho, Michael, and A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33â–61, 1998.

[21] Seema Jaggi, William C. Karl, Stephane Mallat, and Alan S. Willsky, "High resolution pursuit for feature extraction," *Applied and Computational Harmonic Analysis*, vol. 5, no. 4, pp. 428–449, 1998.

[22] Shie Qian and Dapang Chen, "Signal representation using adaptive normalized gaussian functions," *Signal Processing*, vol. 36, no. 1, pp. 1–11, 1994.

[23] Sacha Krstulovic and Rémi Gribonval, "MPTK: Matching Pursuit made tractable," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP'06)*, Toulouse, France, May 2006, vol. 3, pp. 496–499.

[24] "Matching Pursuit," `http://www.mathworks.co.uk/help/wavelet/ref/wmpalg.html`, 2014, Last accessed 18-March-2014.

[25] "GMPcoder," `http://d.zantalis.com/gmpcoder`, 2014, Last accessed 18-March-2014.

[26] T. Lokki et al., "Anechoic recordings of symphonic music," `http://auralization.tkk.fi`.

[27] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 7, pp. 2046–2057, Sept 2011.

[28] Emmanuel Vincent, *Improved Perceptual Metrics for the Evaluation of Audio Source Separation*, vol. 7191 of *Lecture Notes in Computer Science*, pp. 430–437, Springer Berlin Heidelberg, 2012.