Hindawi Publishing Corporation
Computational Intelligence and Neuroscience
Volume 2008, Article ID 872425, 15 pages
doi:10.1155/2008/872425

Research Article

Extended Nonnegative Tensor Factorisation Models for Musical Sound Source Separation

Derry FitzGerald,[1] Matt Cranitch,[1] and Eugene Coyle[2]

[1] *Department of Electronic Engineering, Cork Institute of Technology, Cork, Ireland*
[2] *School of Electrical Engineering Systems, Dublin Institute of Technology, Kevin Street, Dublin, Ireland*

Correspondence should be addressed to Derry FitzGerald, derryfitz@eircom.net

Received 18 December 2007; Revised 3 March 2008; Accepted 17 April 2008

Recommended by Morten Morup

Recently, shift-invariant tensor factorisation algorithms have been proposed for the purposes of sound source separation of pitched musical instruments. However, in practice, existing algorithms require the use of log-frequency spectrograms to allow shift invariance in frequency which causes problems when attempting to resynthesise the separated sources. Further, it is difficult to impose harmonicity constraints on the recovered basis functions. This paper proposes a new additive synthesis-based approach which allows the use of linear-frequency spectrograms as well as imposing strict harmonic constraints, resulting in an improved model. Further, these additional constraints allow the addition of a source filter model to the factorisation framework, and an extended model which is capable of separating mixtures of pitched and percussive instruments simultaneously.

1. Introduction

The use of factorisation-based approaches for the separation of musical sound sources dates back to the early 1980s when Stautner used principal component analysis (PCA) to separate different tabla strokes [1]. However, it was not until the development of independent component analysis (ICA) [2] and techniques such as sparse coding [3, 4] and nonnegative matrix factorisation (NMF) [5, 6] that factorisation-based approaches received much attention for the analysis and separation of musical audio signals [7–11].

Factorisation-based approaches were initially applied to single channel separation of musical sources [7–10], where time-frequency analysis was performed on the input signal, yielding a spectrogram $\mathbf{X}$ of size $n \times m$. This spectrogram was then factorised to yield a reduced rank approximation

$$\mathbf{X} \approx \hat{\mathbf{X}} = \mathbf{AS}, \tag{1}$$

where $\mathbf{A}$ is of size $n \times r$ and $\mathbf{S}$ is of size $r \times m$, with r less than n and m. In this case, the columns of $\mathbf{A}$ contain frequency basis functions, while the corresponding rows of $\mathbf{S}$ contain amplitude basis functions which describe when the frequency basis functions are active. Typically this is done on a magnitude or power spectrogram, and this approach makes the assumption that the spectrograms generated by the basis function pairs sum together to generate the mixture spectrogram. This does not take into account the effects of phase when the spectrograms are added together, and in the case of magnitude spectrograms this assumption is only true if the sources do not overlap in time and frequency, while it holds true on average for power spectrograms. Where the various techniques differ is in how this factorisation is achieved. Casey and Westner [7] used PCA to achieve dimensional reduction and then performed ICA on the retained principal components to achieve independent basis functions, while more recent work has focused on the use of nonnegativity constraints in conjunction with a suitable cost function [8, 9].

A commonly used cost function is the generalised Kullback-Leibler divergence proposed by Lee and Seung [5]:

$$D(\mathbf{X}\|\hat{\mathbf{X}}) = \sum_{ij}\left(\mathbf{X}_{ij}\log\frac{\mathbf{X}_{ij}}{\hat{\mathbf{X}}_{ij}} - \mathbf{X}_{ij} + \hat{\mathbf{X}}_{ij}\right) \tag{2}$$

which is equivalent to assuming a Poisson noise model for the data [12]. This cost function has been widely used due to its ease of implementation, lack of parameters, and

the fact that it has been found to give reasonable results in many cases [13, 14]. A sparseness constraint can also be added to this cost function, and multiplicative update equations which ensure nonnegativity can be derived for these cost functions [15]. Other cost functions have been developed for factorisation of audio spectrograms such as that of Abdallah and Plumbley which assumes multiplicative gamma-distributed noise in power spectrograms [16]. A similar cost function recently proposed by Parry and Issa attempts to incorporate phase into the factorisation by using a probabilistic phase model [17, 18]. Families of parameterised cost functions have been proposed, such as the Beta divergence [19], and Csiszar's divergences [20]. The use of the Beta divergence for the separation of speech signals has been investigated by O'Grady [21], who also proposed a perceptually-based noise to mask ratio as a cost function.

Regardless of the cost function used, the resultant decomposition is linear, and as a result each basis function pair typically corresponds to a single note or chord played by a given pitched instrument. Therefore, in order to achieve sound source separation, some method is required to group the basis functions by source or instrument. Different grouping methods have been proposed in [7, 8], but in practice it is difficult to obtain the correct clustering for reasons discussed in [22].

1.1. Tensor Notation

When dealing with tensor notation, we use the conventions described by Bader and Kolda in [23]. Tensors are denoted using calligraphic uppercase letters, such as $\mathcal{A}$. Rather than using subscripts to indicate indexing of elements within a tensor or matrix, such as $\mathcal{X}_{i,j}$, indexing of elements is instead notated by $\mathcal{X}(i, j)$. When dealing with contracted product multiplication of two tensors, if $\mathcal{W}$ is a tensor of size $I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M$ and $\mathcal{Y}$ is a tensor of size $I_1 \times \cdots \times I_N \times K_1 \times \cdots \times K_P$, then contracted product multiplication of the two tensors along the first N modes is given by

$$\begin{aligned}\langle \mathcal{W}\mathcal{Y}\rangle_{\{1:N,1:N\}}(j_1,\ldots,j_m,k_1,\ldots,k_p) &= \sum_{i_1=1}^{I_1}\cdots\sum_{i_N=1}^{I_N}\mathcal{W}(i_1,\ldots,i_N,j_1,\ldots,j_M) \\ &\quad\times\mathcal{Y}(i_1,\ldots,i_N,k_1,\ldots,k_P),\end{aligned} \tag{3}$$

where the modes to be multiplied are specified in the subscripts that are contained in the angle brackets.

Elementwise multiplication and division are represented by $\otimes$ and $\oslash$, respectively, and outer product multiplication is denoted by $\circ$. Further, for simplicity of notation, unless otherwise stated, we use the convention that $: k$ denotes the tensor slice associated with the kth source, with the singleton dimension included in the size of the slice.

1.2. Tensor Factorisation

Recently, the above matrix factorisation techniques have been extended to tensor factorisation models to deal with stereo or multichannel signals by FitzGerald et al. [24] and Parry and Essa [25]. The signal model can be expressed as

$$\mathcal{X} \approx \widehat{\mathcal{X}} = \sum_{b=1}^{B} \mathcal{G}_{:b} \circ \mathcal{A}_{:b} \circ \mathcal{S}_{:b}, \tag{4}$$

where $\mathcal{X}$ is an $r \times n \times m$ tensor containing the spectrograms of the r channels, $\mathcal{G}$ is an $r \times B$ matrix containing the gains of the B basis functions in each channel, $\mathcal{A}$ is a matrix of size $n \times B$ containing a set of frequency basis functions, and $\mathcal{S}$ is a matrix of size $m \times B$ containing the amplitude basis functions. In this case, $: b$ is used to denote the bth column of a given matrix.

As a first approximation, many commercial stereo recordings can be considered to have been created by obtaining single-channel recordings of each instrument individually and then summing and distributing these recordings across the two channels, with the result that for any given instrument, the only difference between the two channels lies in the gain of the instrument [26]. The tensor factorisation model provides a good approximation to this case. The extension to tensor factorisation also provides another source of information which can be leveraged to cluster the basis functions, namely that basis functions belonging to the same source should have similar gains. However, as the number of basis functions increases it becomes more difficult to obtain good clustering using this information, as basis functions become shared between sources.

2. Shift-Invariant Factorisation Algorithms

The concept of incorporating shift invariance in factorisation algorithms for sound source separation was introduced in the convolutive factorisation algorithms proposed by Smaragdis [27] and Virtanen [28]. This was done in order to address a particular shortcoming of the standard factorisation techniques, namely that a single frequency basis function is unable to successfully capture sounds where the frequency content evolves with time, such as spoken utterances and drum sounds. To overcome this limitation, the amplitude basis functions were allowed to shift in time, with each shift capturing a different frequency basis function. When these frequency basis functions were combined, the result was a spectrogram of a given source that captured the temporal evolution of the frequency characteristics of the sound source.

Shift invariance in the frequency basis functions was later developed as a means of overcoming the problem of grouping the frequency basis functions to sources, particularly in the case where different notes played by the same instrument occurred over the course of a spectrogram [14, 29]. This shortcoming had been addressed by Vincent and Rodet using a nonlinear ISA approach [30], but this technique required pretraining of source priors before separation.

When incorporating shift invariance in the frequency basis functions, it is assumed that all notes played by a single pitched instrument consist of translated versions of a single frequency basis function. This single instrument basis function is then assumed to represent the typical frequency

characteristics of that instrument. This is a simplification of the real situation, where in practice, the timbre of a given instrument does change with pitch [31]. Despite this, the assumption does represent a valid approximation over a limited pitch range, and this assumption has been used in many commercial music samplers and synthesisers, where a prerecorded note of a given pitch is used to generate other notes close in pitch to the original note. The principal advantage of using shift invariance in the frequency basis functions is that instead of having basis functions which must be grouped to their respective sources before separation can occur, as in standard NMF, the frequency shift invariant model allows individual instruments or sources to be modelled explicitly with each source having an individual slice of the tensors to be estimated.

Up till now, the incorporation of shift invariance in the frequency basis functions required the use of a spectrogram with log-frequency resolution, such as the constant Q transform (CQT) [32]. Alternatively, a log-frequency transform can be approximated by weighted summation of linear-frequency spectrogram bins, such as obtained from a short-time Fourier transform. This can be expressed as

$$\mathbf{X} = \mathbf{R}\mathbf{Y}, \tag{5}$$

where $\mathbf{Y}$ is a linear-frequency spectrogram with f frequency bins and t time frames. $\mathbf{R}$ is a frequency weighting matrix of size $cf \times f$ which maps the f linear-frequency bins to cf log-frequency bins, with $cf < f$ and $\mathbf{X}$ is a log-frequency spectrogram of size $cf \times t$. It can be seen that $\mathbf{R}$ is a rectangular matrix and so no true inverse exists, making any mapping back from log-frequency resolution to linear frequency resolution only an approximate mapping.

If the frequency resolution of the log-frequency transform is set so that the center frequencies of the bands are given by $f_x = f_0\beta^{x-1}$, where f_x denotes the center frequency of the xth band, $\beta = 2^{1/12}$, and f_0 is a reference frequency, then the spacing of the bands will match that of the equal-tempered scale used in western music. A shift up or down by one bin will then correspond to a pitch change of one semitone.

In the context of this paper, translation of basis functions is carried out by means of translation tensors, though other formulations, such as the shift operator method proposed by Smaragdis [27] can be used. To shift an $n \times 1$ vector, an $n \times n$ translation matrix is required. This can be generated by permuting the columns of the identity matrix. For example, in the case of shifting a basis function up by one, the translation matrix can be obtained from $\mathbf{I}(:, [n, 1 : n-1])$, where the identity matrix is denoted by $\mathbf{I}$ and the ordering of the columns is contained in the square brackets where $[n, 1 : n-1]$ indicates that n is the first element in the permutation, followed by entries of $1 : n-1$. For Z allowable translations, these translation matrices are then grouped into a translation tensor of size $n \times Z \times n$.

Research has also been done on allowing more general forms of invariance, such as that of Eggert et al. on transformation invariant NMF [33], where all forms of transformation such as translation and rotation are dealt with by means of a transformation matrix. However, their model has only been demonstrated on translation or shift invariance. Further, while a transformation matrix could potentially be used to allow the use of linear frequency resolution through the use of a matrix that stretches the spectrum, it has been noted elsewhere that this stretching is difficult to perform using a discrete linear frequency representation [13].

2.1. Shifted 2D Nonnegative Tensor Factorisation

All of the algorithms incorporating shift invariance can be seen as special cases of a more general model, shifted 2D nonnegative tensor factorisation (SNTF), proposed by FitzGerald [34], and separately by [35]. The SNTF model can then be described as

$$\mathcal{X} \approx \sum_{k=1}^{K} \left\langle \mathcal{G}_{:k} \left\langle \langle \mathcal{T}\, \mathcal{A}_{:k} \rangle_{\{3,1\}} \langle \mathcal{S}_{:k} \mathcal{P} \rangle_{\{3,1\}} \right\rangle_{\{2:4,1:3\}} \right\rangle_{\{2,2\}}, \tag{6}$$

where $\mathcal{X}$ is a tensor of size $r \times n \times m$, containing the magnitude spectrograms of each channel of the signal. $\mathcal{G}$ is a tensor of size $r \times K$, containing the gains of each of the K sources in each of the r channels. $\mathcal{T}$ is an $n \times z \times n$ translation tensor, which translates the instrument basis functions in $\mathcal{A}$ up or down in frequency, where z is the number of translations in frequency, thereby approximating different notes played by a given source. $\mathcal{A}$ is a tensor of size $n \times K \times p$, where p is the number of translations across time. $\mathcal{S}$ is a tensor of size $z \times K \times m$ containing the activations of the translations of $\mathcal{A}$ which indicate when a given note played by a given instrument occurs, thereby generating a transcription of the signal. $\mathcal{P}$ is an $m \times p \times m$ translation tensor which translates the time activation functions contained in $\mathcal{S}$ across time, thereby allowing time-varying source or instrument spectra. These tensors, their dimensions, and functions are summarised in Table 1 for ease of reference, as are all tensors used in subsequent models. If the number of channels is set to $r = 1$, and the allowable frequency translations z are also set to one, then the model collapses to that proposed by Virtanen in [28]. Similarly, setting $p = 1$ results in the model proposed in [36], while setting both r and p to one results in the model described in (4). In [34], the generalised Kullback-Leibler divergence is used as a cost function, and multiplicative update equations derived for $\mathcal{G}$, $\mathcal{A}$, and $\mathcal{S}$.

When using SNTF, a given pitched instrument is modelled by an instrument spectrogram which is translated up and down in frequency to give different notes played by the instrument. The gain parameters are then used to position the instrument in the correct position in the stereo field. A spectrogram of the kth separated source can then be estimated from (6) using only the tensor slices associated with the kth source. This spectrogram can then be inverted to a time-domain waveform by reusing the phase information of the original mixture signal, or by generating a set of phase information using the technique proposed by Slaney [37]. Alternatively, the recovered spectrogram can be used

to generate a Wiener-type filter which can be applied to the original complex short-time Fourier transform.

As noted previously, the mapping from log-frequency to linear-frequency domains is an approximate mapping and this can have an adverse effect on the sound quality of the resynthesis. Various methods for performing this mapping and obtaining an inverse CQT have been investigated [38, 39]. However, a simpler method of overcoming this problem is to incorporate the mapping into the model. This can be done by replacing $\mathcal{T}$ in (6) with $\langle \mathcal{R}\mathcal{T} \rangle_{\{2,1\}}$, where $\mathcal{R}$ is an approximate map from log to linear domains. This mapping can simply be the transpose of $\mathbf{R}$, the mapping used in (5). Shift invariance is still implemented in the log-frequency domain, but the cost function is now measured in the linear-frequency domain. This is similar to the method proposed by O'Grady when using noise-to-mask ratio as a cost function [21]. O'Grady included the mapping from linear to Bark domain in his algorithm, as the cost function needed to be measured in the Bark scale domain. It was noted that this resulted in energy spreading in the magnitude spectrogram domain. In the modified SNTF algorithm, the opposite case applies, we wish to measure the cost function in the linear magnitude spectrogram domain, as opposed to a log-frequency domain, and the incorporation of the mapping results in less energy spreading in the frequency basis functions in the constant Q domain. It also has the advantage of performing the optimisation in the domain from which the final inversion to the time domain will take place. Despite this, the use of an approximate mapping still has adverse effects on the resynthesis quality.

In order to overcome these resynthesis problems, Schmidt et al. proposed using the spectrograms recovered to create masks which are then used to refilter the original spectrogram [40]. Schmidt et al. used a binary masking approach where bins were allocated to the source which had the highest power at that bin. In this paper, we use a refiltering method where the recovered source spectrogram is multiplied by the original mixture spectrogram as it was found that this gave better results than the previously described method.

3. Sinusoidal Shifted 2D Nonnegative Tensor Factorisation

While SNTF has been shown to be capable of separating mixtures of harmonic pitched instruments [34], a potential problem with the method is that there is no guarantee that the basis functions will be harmonic. A form of harmonic constraint, whereby the basis functions are only allowed to have nonzero values at regions which correspond to a perfectly harmonic sound, has been proposed by Virtanen [13] and later by Raczynski et al. [11], who used it for the purposes of multipitch estimation. However, with this technique, there is no guarantee that values returned in the harmonic regions of the basis functions will correspond to the actual shape that a sinusoid would have if present. It has also been noted by Raczynski that the structure returned when using this constraint may not always be purely harmonic as it is possible for the peaks to occur at points that are not at the centre of the harmonic regions.

Table 1: Summary of the tensors used, their dimensions, and function, in the various shift-invariant factorisation models included in this paper. Tensors that occur in multiple models are not repeated.

SNTF	$\mathcal{X}$	$r \times n \times m$	Signal spectrograms
	$\widehat{\mathcal{X}}$	$r \times n \times m$	Approximation of $\mathcal{X}$
	$\mathcal{G}$	$r \times K$	Instrument gains
	$\mathcal{T}$	$n \times z \times n$	Translation tensor (freq.)
	$\mathcal{A}$	$n \times K \times p$	Instrument basis functions
	$\mathcal{S}$	$z \times K \times m$	Note activations
	$\mathcal{P}$	$m \times p \times m$	Translation tensor (time)
SSNTF	$\mathcal{H}$	$n \times z \times h$	Harmonic dictionary
	$\mathcal{W}$	$h \times K \times p$	Harmonic weights
SF-SSNTF	$\mathcal{F}$	$n \times K \times n$	Formant filters
SF-SSNTF + N	$\mathcal{M}$	$r \times L$	Noise instrument gains
	$\mathcal{B}$	$n \times L \times q$	Noise basis functions
	$\mathcal{C}$	$L \times m$	Noise activations
	$\mathcal{Q}$	$m \times q \times m$	Noise translation tensor

An alternative approach to the problem of imposing harmonicity constraints on the basis functions is to note that the magnitude spectrum of a windowed sinusoid can be calculated directly in closed-form as a shifted and scaled version of the window's frequency response [41]. For example, using a Hann window, the magnitude spectrum of a sinusoid of frequency $f_0 = h2\pi/f_s$, where h is frequency in Hz, f_s is the sampling frequency in Hz, and N is the desired FFT, is given by

$$X(x) = \left| 0.5D(g) + 0.25\{D_1(g) + D_2(g)\} \right|, \tag{7}$$

where $g = f_x - f_0$, with $f_x = x2\pi/N$ being the centre frequency of the xth FFT bin and where D is defined as

$$D(g) = \frac{\sin(gN/2)}{\sin(g/2)}, \tag{8}$$

with $D_1(g) = D(g - 2\pi/N)$ and $D_2(g) = D(g + 2\pi/N)$. It is then proposed to use an additive synthesis type model, where each note is modelled as a sum of sinusoids at integer multiples of the fundamental frequency of the note, with the relative strengths of the sinusoids giving the timbre of the note played. This spectral domain approach has been used previously to perform additive synthesis, in particular the inverse FFT method of Freed et al. [42].

For a given pitch and a given number of harmonics, the magnitude spectra of the individual sinusoids can be stored in a matrix of size $n \times h$, where n is the number of bins in the spectrum, and h is the number of harmonics. This can be repeated for each of the allowed z notes, resulting in a tensor of size $n \times z \times h$. In effect, this tensor is a signal dictionary consisting of the magnitude spectra of individual sinusoids related to the partials of each allowable note. Again taking a

Hann window as an example, the tensor can then be defined as

$$\mathcal{H}(x,i,j) = \left|0.5D(g_{xij}) + 0.25\{D_1(g_{xij}) + D_2(g_{xij})\}\right|, \tag{9}$$

where $g_{xij} = f_x - f_{i,j}$ with $f_{i,j} = h_0\beta^{i-1}j2\pi/f_s$, h_0 is the frequency in hertz of the lowest allowable note and β is as previously defined in Section 2. This assumes equal-tempered tuning, but other tuning systems can also be used.

It is also possible to take into account inharmonicity in the positioning of the partials through the use of inharmonicity factors. For example, in the case of instruments containing stretched strings, $f_{i,j}$ can be calculated as

$$f_{i,j} = h_0\beta^{i-1}j2\pi\frac{\sqrt{1+(j^2-1)\alpha}}{f_s}, \tag{10}$$

where α is the inharmonicity factor for the instrument in question [43]. In practice, the magnitude spectra will be close to zero except in the regions around $f_{i,j}$, and so it is usually sufficient to calculate the values of $\mathcal{T}(x,i,j)$ for ten bins on either side of $f_{i,j}$ and to leave the remaining bins at zero. Further, the frequencies of the lowest partial of the lowest note, and the highest partial of the highest note place limits on the region of the spectrogram which will be modelled, and so spectrogram frequency bins outside of these ranges can be discarded. If a small number of harmonics are required, this can considerably reduce the number of calculations required, thereby speeding up the algorithm.

$\mathcal{H}$ contains sets of harmonic partials all of equal gain. In order to approximate the timbres of different musical instruments, these partials must be weighted in different proportions. These weights can be stored in a tensor of size $h \times K \times p$, where K is the number of instruments and p is the number of translations across time, thereby allowing the harmonic weights to vary with time. Labeling the weights tensor as $\mathcal{W}$, the model can be described as

$$\mathcal{X} = \sum_{k=1}^{K}\left\langle \mathcal{G}_{:k}\left\langle \langle \mathcal{H}\mathcal{W}_{:k}\rangle_{\{3,1\}}\langle \mathcal{S}_{:k}\mathcal{P}\rangle_{\{3,1\}}\right\rangle_{\{2:4,1:3\}}\right\rangle_{\{2,2\}}. \tag{11}$$

Using the generalised Kullback-Leibler divergence as a cost function, multiplicative update equations can be derived as

$$\mathcal{G}_{:k} = \mathcal{G}_{:k}\otimes\frac{\langle\langle\langle\langle \mathcal{D}\mathcal{H}\rangle_{\{2,1\}}\mathcal{W}_{:k}\rangle_{\{4,1\}}\mathcal{S}_{:k}\rangle_{\{3:4,1:2\}}\mathcal{P}\rangle_{\{2:4,3:1\}}}{\langle\langle\langle\langle \mathcal{O}\mathcal{H}\rangle_{\{2,1\}}\mathcal{W}_{:k}\rangle_{\{4,1\}}\mathcal{S}_{:k}\rangle_{\{3:4,1:2\}}\mathcal{P}\rangle_{\{2:4,3:1\}}},$$

$$\mathcal{W}_{:k} = \mathcal{W}_{:k}\otimes\frac{\langle\langle(\mathcal{G}_{:k}\circ\mathcal{H})\mathcal{D}\rangle_{\{[1,3],1:2\}}\langle\mathcal{S}_{:k}\mathcal{P}\rangle_{\{3,1\}}\rangle_{\{[1,2,4],[1,2,4]\}}}{\langle\langle(\mathcal{G}_{:k}\circ\mathcal{H})\mathcal{O}\rangle_{\{[1,3],1:2\}}\langle\mathcal{S}_{:k}\mathcal{P}\rangle_{\{3,1\}}\rangle_{\{[1,2,4],[1,2,4]\}}},$$

$$\mathcal{S}_{:k} = \mathcal{S}_{:k}\otimes\frac{\langle\langle\langle(\mathcal{G}_{:k}\circ\mathcal{H})\mathcal{A}_{:k}\rangle_{\{[2,5],[2,1]\}}\mathcal{D}\rangle_{\{1:2,1:2\}}\mathcal{P}\rangle_{\{2:3,[2,1]\}}}{\langle\langle\langle(\mathcal{G}_{:k}\circ\mathcal{H})\mathcal{A}_{:k}\rangle_{\{[2,5],[2,1]\}}\mathcal{O}\rangle_{\{1:2,1:2\}}\mathcal{P}\rangle_{\{2:3,[2,1]\}}}, \tag{12}$$

where $\mathcal{D} = \mathcal{X}\oslash\widehat{\mathcal{X}}$ and $\mathcal{O}$ is an all-ones tensor with the same dimensions as $\mathcal{X}$, and all divisions are taken as elementwise.

These update equations are similar to those of SNTF, just replacing $\mathcal{T}$ and $\mathcal{A}$, with a sinusoidal signal dictionary $\mathcal{H}$, and a set of harmonic weights $\mathcal{W}$, respectively. It is proposed to call this new algorithm *sinusoidal shifted 2D nonnegative tensor factorisation* (SSNTF) as it explicitly models the signal as the summation of weighted harmonically related sinusoids, in effect incorporating an additive synthesis model into the tensor factorisation framework. SSNTF can still be considered as shift invariant in frequency, as the harmonic weights are invariant to where in the frequency spectrum the notes occur.

An advantage of SSNTF is that the separation problem is now completely formulated in the linear-frequency domain, thereby eliminating the need to use an approximate mapping from log to linear frequency domains at any point in the algorithm, which removes the potential for resynthesis artifacts due to the mapping. Resynthesis of the separated time-domain waveforms can be carried out in a similar manner to that of SNTF, or alternatively, one can take advantage of the use of the additive synthesis model to reconstruct the separated signal using additive synthesis.

The SSNTF algorithm was implemented in Matlab using the Tensor Toolbox available from [44], as were all subsequent algorithms described in this paper. The cost function was always observed to decrease with each iteration. However, when running SSNTF, it was found that the best results were obtained when the algorithm was given an estimate of what frequency region each source was present in. This was typically done by giving an estimate of the pitch of the lowest note of each source. For score-assisted separation, such as that proposed by [45], this information will be readily available. The incorporation of this information has the added benefit of fixing the ordering of the sources in most cases. In cases where there is no score available, estimates can be obtained by running SNTF first and determining the pitch information from the recovered basis functions before running SSNTF. At present, research is being undertaken on devising alternate ways of overcoming this problem.

As an example of the improved reconstruction that SSNTF can provide, Figure 1 shows the frequency spectrum of a flute note separated from a single channel mixture of flute and piano. SNTF and SSNTF were performed on this example using 9 translations in frequency and 5 translations in time. All other parameters were set as described later in Section 6. The first spectrum is that of the flute note taken from the original unmixed flute waveform, the second spectrum is that of the recovered flute note using SNTF, with the mapping from log to linear domains included in the model, while the third spectrum is that returned by SSNTF. It can be appreciated that the spectrum returned by SSNTF is considerably closer to the original than that returned by SNTF. This demonstrates the utility of using an approach which is formulated in the linear frequency domain.

Figure 2 shows the original mixture spectrogram of piano and flute, while Figure 3(a) shows the unmixed flute spectrogram, with Figures 3(b), 3(c), and 3(d) showing the SNTF-separated flute spectrogram, the SNTF-separated flute spectrogram using refiltering, and the SSNTF-separated spectrogram, respectively. Figure 4(a) shows the unmixed

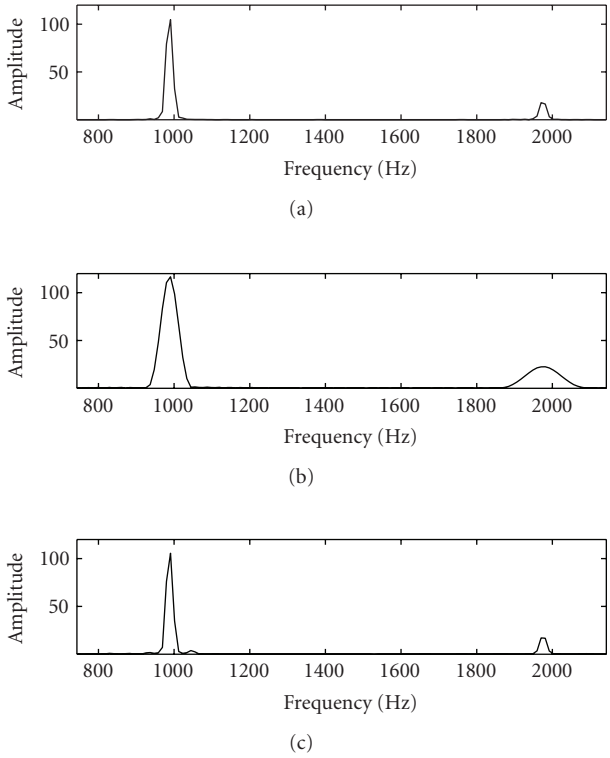

Figure 1: Spectra of flute note, original, SNTF, and SSNTF, respectively.

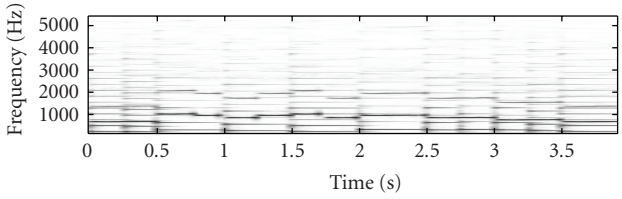

Figure 2: Spectrogram of piano and flute mixture.

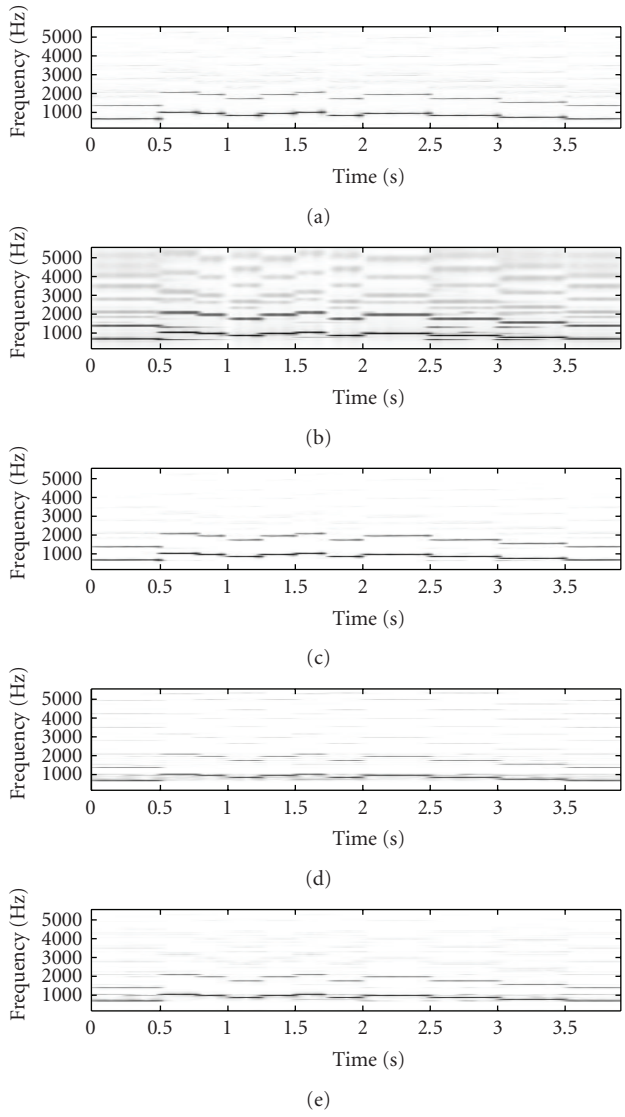

Figure 3: Spectrogram of flute signal, (a) original unmixed, (b) SNTF, (c) refiltered SNTF, (d) SSNTF, (e) source-filter SSNTF.

piano spectrogram, with Figures 4(b), 4(c), and 4(d) showing the SNTF-separated piano spectrogram, the SNTF-separated piano spectrogram obtained using refiltering, and the SSNTF-separated spectrogram, respectively. It can be seen that the spectrograms recovered using SSNTF are considerably closer to the original spectrograms than that recovered directly from SNTF, where the smearing due to the approximate mapping from log to linear domains is clearly evident. Considerably improved recovery of the sources was also noted on playback of the separated SSNTF signals in comparison to those obtained using SNTF directly. The spectrograms obtained using SNTF in conjunction with refiltering can be also seen to be considerably closer to the original spectrograms than any of the other methods. However, on listening, the sound quality is still less than that obtained using SSNTF. Further, as will be seen later, the SNTF-based methods are not as robust as SSNTF-based methods.

It should also be noted that the addition of harmonic constraints imposes restrictions on the solutions that can be returned by the factorisation algorithms. This is of considerable benefit when incorporating additional parameters into the models, as will be seen in the following sections.

4. Source-Filter Modelling

As noted previously in Section 2, the use of a single shifted instrument basis function to model different notes played by an instrument is a simplification. In practice, the timbre of notes played by a given instrument changes with pitch, and this restricts the usefulness of shifted factorisation models. Recently, Virtanen and Klapuri proposed the incorporation of a source-filter model approach in the factorisation method as a means of overcoming this problem [46]. In the source-filter framework for sound production, the source is typically a vibrating object, such as a violin string, and the filter

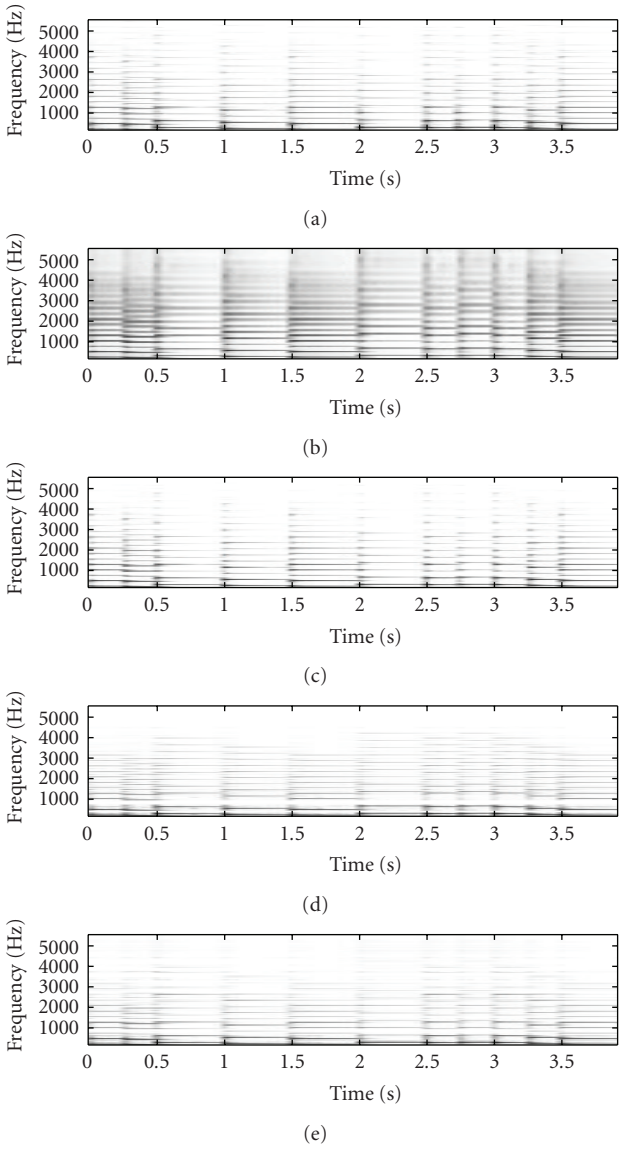

Figure 4: Spectrogram of piano signal, (a) original unmixed, (b) SNTF, (c) refiltered SNTF, (d) SSNTF, (e) source-filter SSNTF.

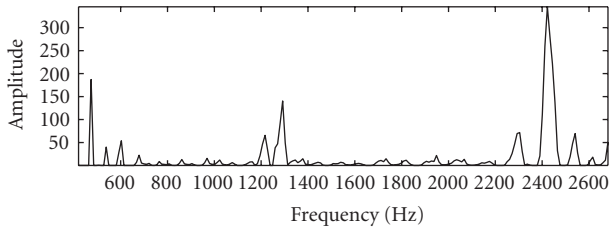

Figure 5: Filter returned for flute when using source-filter SSNTF.

accounts for the resonant structure of the instrument, such as the violin body, which alters and filters the sound produced by the vibrating object. This approach had been used previously in both sound synthesis and speech coding [47, 48], but not in a factorisation framework.

When applied in the context of shifted instrument basis functions, the instrument basis function represents a harmonic excitation pattern which can be shifted up and down in frequency to generate different pitches. A single fixed filter is then applied to these translated excitation patterns, with the filter representing the instrument's resonant structure. This results in a system where the instrument timbre varies with pitch, resulting in a more realistic model. The instrument formant filters can be incorporated into the shifted tensor factorisation framework through a formant filter tensor $\mathcal{F}$ of size $n \times K \times n$. In this case, the kth slice of $\mathcal{F}$ is a diagonal matrix, with the instrument formant filter coefficients contained on the diagonal.

Unfortunately, attempts to incorporate the source-filter model into the SNTF framework were unsuccessful. The resultant algorithm had too many parameters to optimise and it was difficult to obtain good separation results. However, the additional constraints imposed by SSNTF were found to make the problem tractable. The resultant model can then be described as

$$\mathcal{X} \approx \widehat{\mathcal{X}} = \sum_{k=1}^{K} \left\langle \mathcal{G}_{:k} \left\langle \langle \mathcal{R}_{:k} \mathcal{W}_{:k} \rangle_{\{[2,4],[2,1]\}} \mathcal{V}_{:k} \right\rangle_{\{2:4,[2,1,3]\}} \right\rangle_{\{2,2\}}, \tag{13}$$

where $\mathcal{R}_{:k} = \langle \mathcal{F}_{:k} \mathcal{H} \rangle_{\{3,1\}}$ and $\mathcal{V}_{:k} = \langle \mathcal{S}_{:k} \mathcal{P} \rangle_{\{3,1\}}$.

Again using the generalised Kullback-Lieber divergence as a cost function, the following update equations were derived:

$$\mathcal{G}_{:k} = \mathcal{G}_{:k} \otimes \frac{\left\langle \langle \mathcal{D} \langle \mathcal{R}_{:k} \mathcal{W}_{:k} \rangle_{\{[2,4],[2,1]\}} \rangle_{\{2,1\}} \mathcal{V}_{:k} \right\rangle_{\{2:5,[4,2,1,3]\}}}{\left\langle \langle \mathcal{O} \langle \mathcal{R}_{:k} \mathcal{W}_{:k} \rangle_{\{[2,4],[2,1]\}} \rangle_{\{2,1\}} \mathcal{V}_{:k} \right\rangle_{\{2:5,[4,2,1,3]\}}},$$

$$\mathcal{F}_{:k} = \mathcal{F}_{:k} \otimes \frac{\left\langle \langle \mathcal{G}_{:k} \mathcal{D} \rangle_{\{1,1\}} \left\langle \langle \mathcal{H} \mathcal{W}_{:k} \rangle_{\{3,1\}} \mathcal{V}_{:k} \right\rangle_{\{2:4,1:3\}} \right\rangle_{\{[1,3],2:3\}}}{\left\langle \langle \mathcal{G}_{:k} \mathcal{O} \rangle_{\{1,1\}} \left\langle \langle \mathcal{H} \mathcal{W}_{:k} \rangle_{\{3,1\}} \mathcal{V}_{:k} \right\rangle_{\{2:4,1:3\}} \right\rangle_{\{[1,3],2:3\}}},$$

$$\mathcal{W}_{:k} = \mathcal{W}_{:k} \otimes \frac{\left\langle \left\langle \langle \mathcal{G}_{:k} \mathcal{R}_{:k} \rangle_{\{2,2\}} \mathcal{D} \right\rangle_{\{[1,3],1:2\}} \mathcal{V}_{:k} \right\rangle_{\{[1,2,4],[2,1,4]\}}}{\left\langle \left\langle \langle \mathcal{G}_{:k} \mathcal{R}_{:k} \rangle_{\{2,2\}} \mathcal{O} \right\rangle_{\{[1,3],1:2\}} \mathcal{V}_{:k} \right\rangle_{\{[1,2,4],[2,1,4]\}}},$$

$$\mathcal{S}_{:k} = \mathcal{S}_{:k} \otimes \frac{\left\langle \langle \mathcal{G}_{:k} \langle \mathcal{R}_{:k} \mathcal{W}_{:k} \rangle_{\{[2,4],[2,1]\}} \rangle_{\{2,2\}} \langle \mathcal{D} \mathcal{P} \rangle_{\{3,1\}} \right\rangle_{\{[1,3,5],1:3\}}}{\left\langle \langle \mathcal{G}_{:k} \langle \mathcal{R}_{:k} \mathcal{W}_{:k} \rangle_{\{[2,4],[2,1]\}} \rangle_{\{2,2\}} \langle \mathcal{O} \mathcal{P} \rangle_{\{3,1\}} \right\rangle_{\{[1,3,5],1:3\}}}. \tag{14}$$

Figure 5 shows the filter recovered for the flute from the example previously discussed in Section 3. It can be seen that the recovered filter consists of a series of peaks as opposed to a smooth formant-like filter. This is due to a combination of two factors, firstly, the small number of different notes played in the original signal, and secondly, the harmonic constraints imposed by SSNTF. This results in a situation where large portions of the spectrum will have little or no energy, and accordingly the filter models these regions as having little or no energy.

On listening to the resynthesis, there was a marked improvement in the sound quality of the flute in comparison with SSNTF, with less high-frequency energy present. The resynthesis of the piano also improved, though less so than

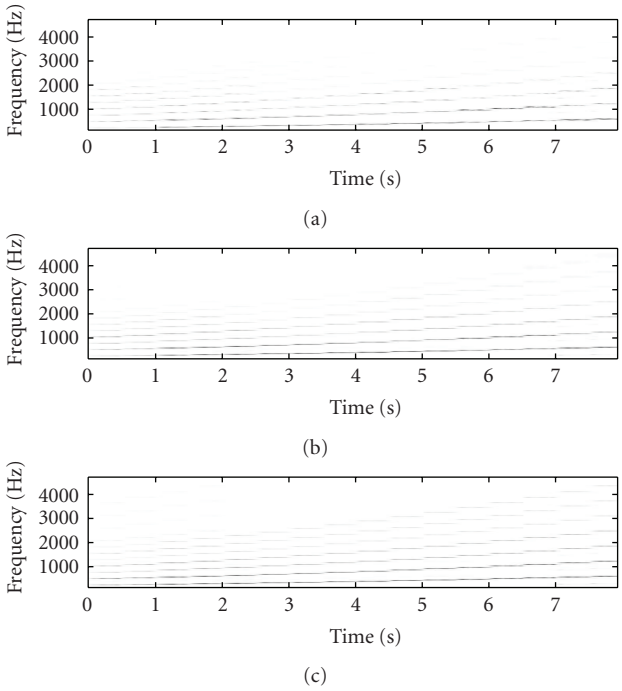

Figure 6: Spectrograms for (a) original flute spectrogram, (b) spectrogram recovered using source-filter SSNTF, and (c) spectrogram recovered using SSNTF.

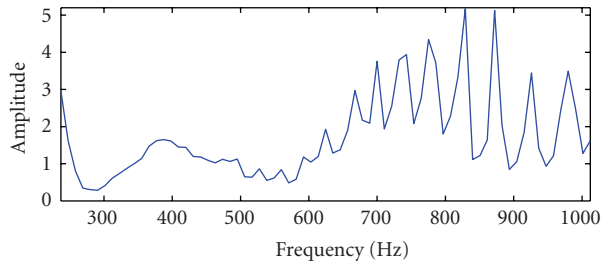

Figure 7: Filter returned for solo flute example in Figure 6 when using source-filter SSNTF.

that of the flute. Figures 3(e) and 4(e) show the spectrograms recovered using source-filter SSNTF for the flute and piano, respectively. It can be observed that the flute spectrogram is closer to the original than either SNTF or SSNTF, with no smearing and a reduced presence of higher harmonics in comparison to SSNTF, which is in line with what was observed on listening to the resynthesis. In comparison to the SNTF and refiltering approach, source-filter SSNTF has retained more high-frequency information than the refiltered approach, and can be seen to be closer to the original spectrogram. In the case of the piano, the refiltered spectrogram contains more high-frequency information than the source-filter SSNTF approach, which is closer to the original piano spectrogram. On listening, the source-filter SSNTF approach also outperforms the refiltered SNTF approach.

As a further example of source-filter SSNTF, Figure 6(a) shows the spectrogram of a flute signal consisting of 16 notes, one semitone apart played in ascending order, while Figures 6(b) and 6(c) show the spectrogram recovered using source-filter SSNTF and SSNTF, respectively. It can be seen that the source-filter method has returned a spectrogram closer to the original, with less high-frequency information than SSNTF. Figure 7 shows the source-filter associated with Figure 6(b). It can be seen that in this case, where 16 successive notes are played, the source-filter is smoother, as would be expected for a formant-like filter, but as the harmonics get further apart, evidence of peakiness similar to that in Figure 5 becomes more evident.

The above examples demonstrate the utility of using the source-filter approach as a means of improving the accuracy of the SSNTF model. This is bourn out in the improved resynthesis of the separated sources.

5. Separation of Pitched and Nonpitched Instruments

Musical signals, especially popular music, typically contain unpitched instruments such as drum sounds in addition to pitched instruments. While allowing shift invariance in both frequency and time is suitable for separating mixtures of pitched instruments, it is not suitable for dealing with percussion instruments such as the snare and kick drums, or other forms of noise in general. These percussion instruments can be successfully captured by algorithms which allow shift invariance in time only without the use of frequency shift invariance. In order to deal with musical signals containing both pitched and percussive instruments or contain additional noise, it is necessary to have an algorithm which handles both these cases. This can be done by simply adding the two models together. This has previously been done by Virtanen in the context of matrix factorisation algorithms [13], who also noted that the resulting model was too complex to obtain good results without the addition of additional constraints. In particular, the use of a harmonicity constraint was required, though in this case it was based on zeroing instrument basis functions in areas where no harmonic activity was expected, as opposed to the additive synthesis-based technique proposed in this paper.

Extending the concept to the case of tensor factorisation techniques results in a generalised tensor factorisation model for the separation of pitched and percussive instruments, which still allows the use of a source-filter model for pitched instruments. The model can be described by

$$\mathcal{X} \approx \widehat{\mathcal{X}} = \sum_{k=1}^{K} \left\langle \mathcal{G}_{:k} \left\langle \left\langle \mathcal{R}_{:k}\, \mathcal{W}_{:k} \right\rangle_{\{[2,4],[2,1]\}} \mathcal{V}_{:k} \right\rangle_{\{2:4,[2,1,3]\}} \right\rangle_{\{2,2\}} + \sum_{l=1}^{L} \left\langle \mathcal{M}_{:l} \left\langle \mathcal{B}_{:k} \left\langle \mathcal{C}_{:l} \mathcal{Q} \right\rangle_{\{2,1\}} \right\rangle_{\{2:3,1:2\}} \right\rangle_{\{2,2\}}, \tag{15}$$

where $\mathcal{M}$ is a tensor of size $r \times L$, which contains the gains of each of the L percussive sources, $\mathcal{B}$ is a tensor of size $n \times L \times q$, where q is the number of allowable

time shifts for the percussive sources, $\mathcal{C}$ is a tensor of size $L \times m$, and $\mathcal{Q}$ is a translation tensor of size $m \times q \times m$. Multiplicative update equations, based on the generalised Kullback-Leibler divergence can then be derived for these additional parameters, while update equations for all other parameters are as given in Section 4. The additional update equations are given by

$$\mathcal{M}_{:l} = \mathcal{M}_{:l} \otimes \frac{\left\langle \mathcal{D} \left\langle \mathcal{B} \langle \mathcal{C}\mathcal{Q} \rangle_{\{2,1\}} \right\rangle_{\{2:3,1:2\}} \right\rangle_{\{2:3,[1,3]\}}}{\left\langle \mathcal{O} \left\langle \mathcal{B} \langle \mathcal{C}\mathcal{Q} \rangle_{\{2,1\}} \right\rangle_{\{2:3,1:2\}} \right\rangle_{\{2:3,[1,3]\}}},$$

$$\mathcal{B}_{:l} = \mathcal{B}_{:l} \otimes \frac{\left\langle \left\langle \mathcal{M}_{:l}\mathcal{D} \right\rangle_{\{1,1\}} \langle \mathcal{C}\mathcal{Q} \rangle_{\{2,1\}} \right\rangle_{\{[1,3],[1,3]\}}}{\left\langle \left\langle \mathcal{M}_{:l}\mathcal{O} \right\rangle_{\{1,1\}} \langle \mathcal{C}\mathcal{Q} \rangle_{\{2,1\}} \right\rangle_{\{[1,3],[1,3]\}}}, \tag{16}$$

$$\mathcal{C}_{:l} = \mathcal{C}_{:l} \otimes \frac{\left\langle \left\langle \mathcal{M}_{:l}\mathcal{B} \right\rangle_{\{2,2\}} \langle \mathcal{D}\mathcal{Q} \rangle_{\{3,3\}} \right\rangle_{\{[1,3,4],[1,2,4]\}}}{\left\langle \left\langle \mathcal{M}_{:l}\mathcal{B} \right\rangle_{\{2,2\}} \langle \mathcal{O}\mathcal{Q} \rangle_{\{3,3\}} \right\rangle_{\{[1,3,4],[1,2,4]\}}}.$$

The individual sources can be separated as before, but the algorithm can also be used to separate the pitched instruments from the unpitched percussive instruments or vice-versa by resynthesising the relevant section of the model. It can also be used as a means of eliminating noise from mixtures of pitched instruments by acting as a type of "garbage collector," which can improve resynthesis quality in some cases. It can also be viewed as being analogous to the additive plus residual sinusoidal analysis techniques described by Serra [49] in that it allows the pitched or sinusoidal part of the signal to be resynthesised separately from the noise part of the signal.

As an example of the use of the combined model, Figure 8 shows the mixture spectrograms obtained from a stereo mixture containing three pitched instruments, piano, flute, and trumpet, and three percussion instruments, snare, hi-hats, and kick drum, while Figure 9 shows the original unmixed spectrograms for those sources, respectively. The piano, snare, and kick drum were all panned to the center, with the hi-hats and flute panned midleft and the trumpet midright. Figure 10 shows the separated spectrograms obtained using the combined model. It can be seen that the sources have been recovered well, with each individual instrument identifiable, though traces of other sources can be seen in the spectrograms. This is most evident where traces of the hi-hats are visible in the snare spectrogram, but the snare clearly predominates. On listening to the results, traces of the flute can also be heard in the piano signal, and the timbres of the instruments have been altered, but are still recognisable as being the instrument in question. The example also highlights another advantage of tensor factorisation models in general, namely the ability to separate instruments which have the same position in the stereo field. This is in contrast to algorithms such as Adress and DUET, which can only separate sources if they occupy different positions in the stereo field [26, 50].

6. Performance Evaluation

The performances of SNTF, SNTF using refiltering, SSNTF, source-filter SSNTF, and source-filter SSNTF with noise basis functions in the context of modelling mixtures of pitched instruments were compared using a set of 40 test mixtures. In the case of source-filter SSNTF with noise basis functions, two noise basis functions were learned in order to aid the elimination of noise and artifacts from the harmonic sources. The 40 test signals were of 4 seconds duration and contained mixtures of melodies played by different instruments and created by using a large library of orchestral samples [51]. Samples from a total of 15 different orchestral instruments were used. A wide range of pitches were covered, from 87 Hz to 1.5 kHz, and the melodies played by the individual instruments in each test signal were in harmony. This was done to ensure that the test signals contained extensive overlapping of harmonics, as this occurs in most real world musical signals. In many cases, the notes played by one instrument overlapped notes played by another instrument to test if the algorithms were capable of discriminating notes of the same pitch played by different instruments.

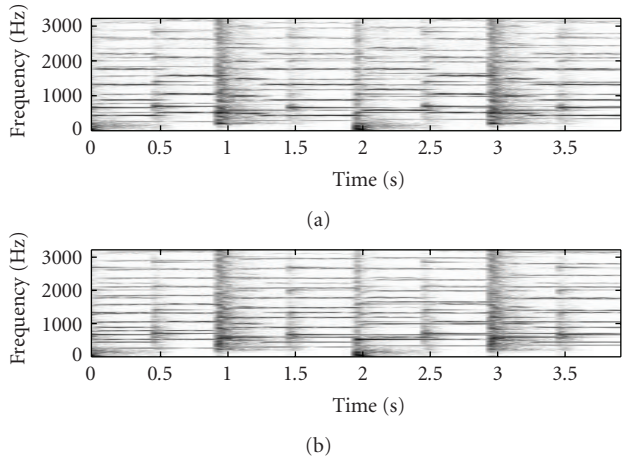

Figure 8: Mixture spectrograms of piano, flute, trumpet, snare, hi-hats, and kick drum.

The 40 test signals consisted of 20 single channel mixtures of 2 instruments and 20 stereo mixtures of 3 instruments, and these mixtures were created by linear mixing of individual single channel instrument signals. In the case of the single channel mixtures, the source signals were mixed with unity gain, and in the case of the stereo mixtures, mixing was done according to

$$\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} = \begin{pmatrix} 0.75 & 0.5 & 0.25 \\ 0.25 & 0.5 & 0.75 \end{pmatrix} \begin{pmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{pmatrix}, \tag{17}$$

where $x_1(t)$ and $x_2(t)$ are the left and right channels of the stereo mixture and $s_1(t)$ represents the first single channel instrument signal and so on.

Spectrograms were obtained for the mixtures, using a short-time Fourier transform with a Hann window of 4096 samples, with a hopsize of 1024 samples between frames. All variables were initialised randomly, with the exception of the frequency basis functions for SNTF-based separation, which were initialised with harmonic basis functions at the

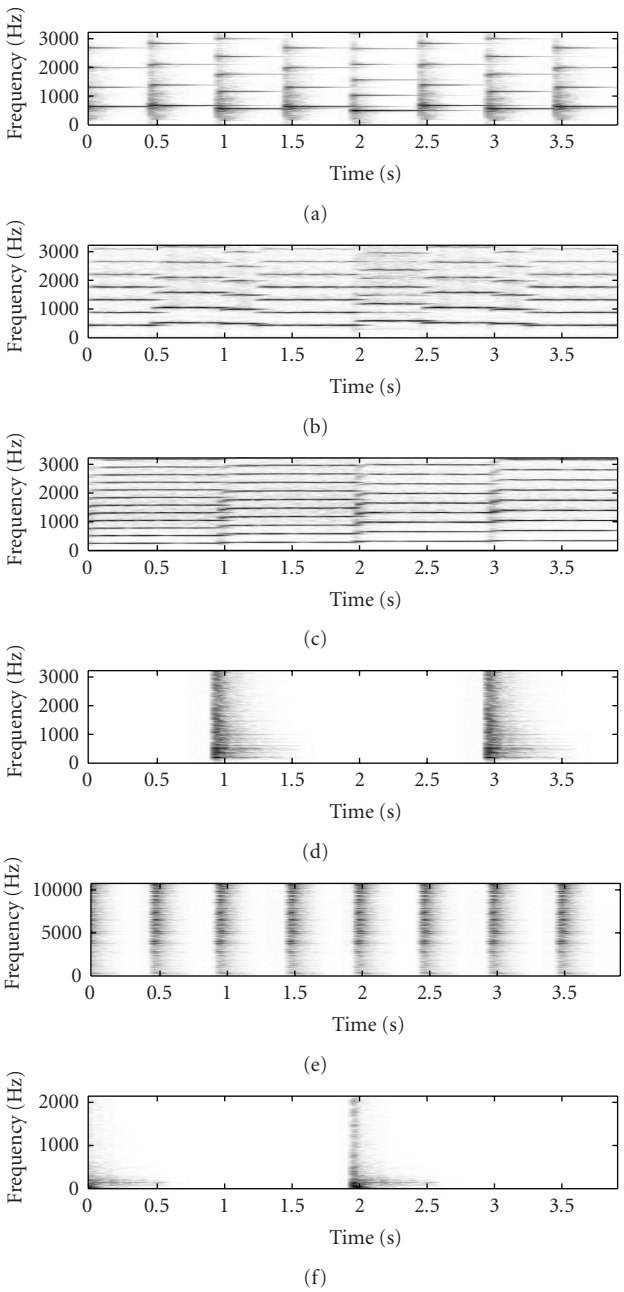

Figure 9: Original spectrograms of (a) piano, (b) flute, (c) trumpet, (f) snare, (g) hi-hats, and (h) kick drum.

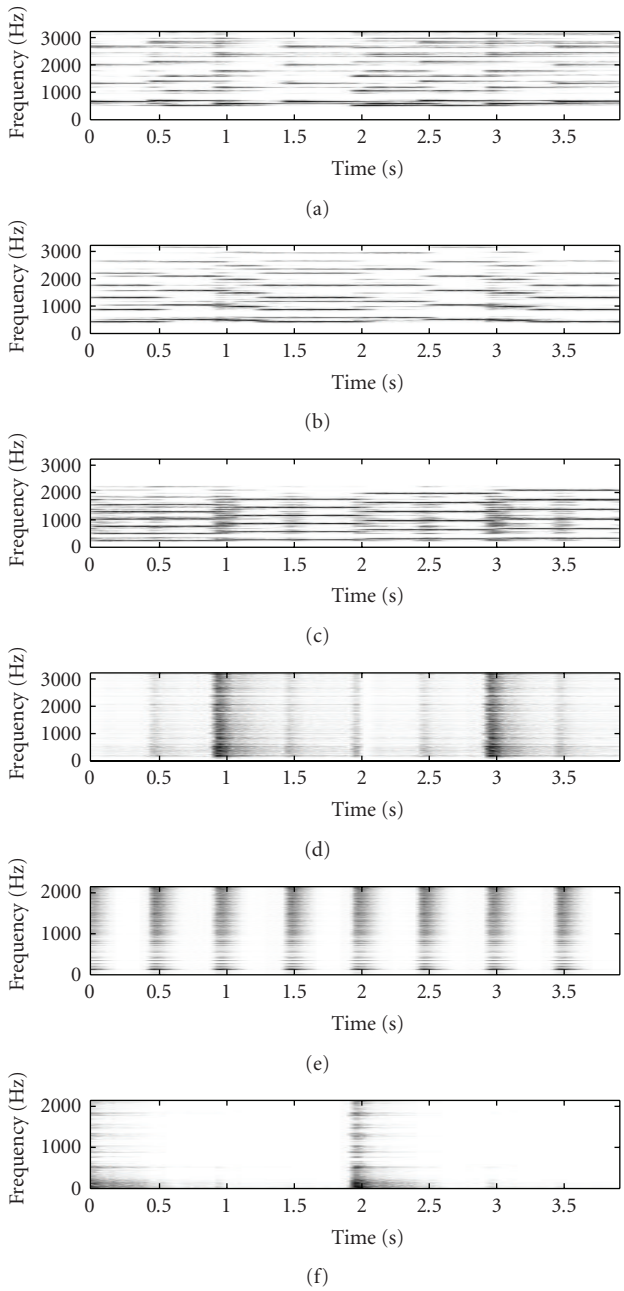

Figure 10: Separated spectrograms of (a) piano, (b) flute, (c) trumpet, (f) snare, (g) hi-hats and (h) kick drum.

frequency of the lowest note played by each instrument in each example. This was done to put SNTF on an equal footing with the SSNTF-based algorithms, where the pitch of the lowest note of each source was provided. The number of allowable notes was set to the largest pitch range covered by an instrument in the test signal and the number of harmonics used in SSNTF was set to 12. The algorithms were run for 300 iterations, and the separated source spectrograms were estimated by carrying out contracted tensor multiplication on the tensor slices associated with an individual source. The recovered source spectrograms were resynthesised using the phase information from the mixture spectrograms. The phase of the channel where the source was strongest was used in the case of the stereo mixtures.

Using the original source signals as a reference, the performance of the different algorithms were evaluated using commonly used metrics, namely the signal-to-distortion ratio (SDR), which provides an overall measure of the sound quality of the source separation, the signal-to-interference ratio (SIR), which measures the presence of other sources in the separated sounds, and the signal-to-artifacts ratio (SAR), which measures the artifacts present in the recovered signal due to separation and resynthesis. Details of these metrics can be found in [52] and a Matlab toolbox to calculate

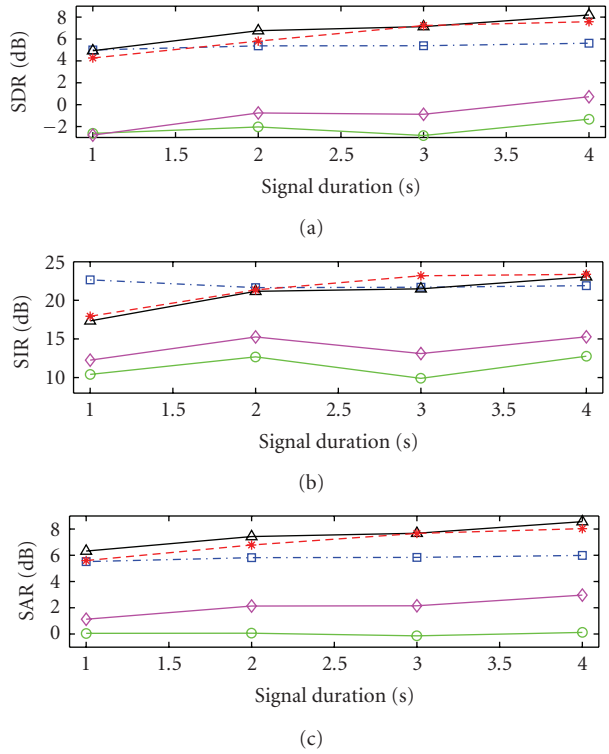

Figure 11: Performance evaluation of SNTF (circle solid), refiltered SNTF (diamond solid), SSNTF (square dash-dotted), source-filter SSNTF (triangle solid), and source-filter SSNTF (star dashed) with noise basis functions for various signal durations.

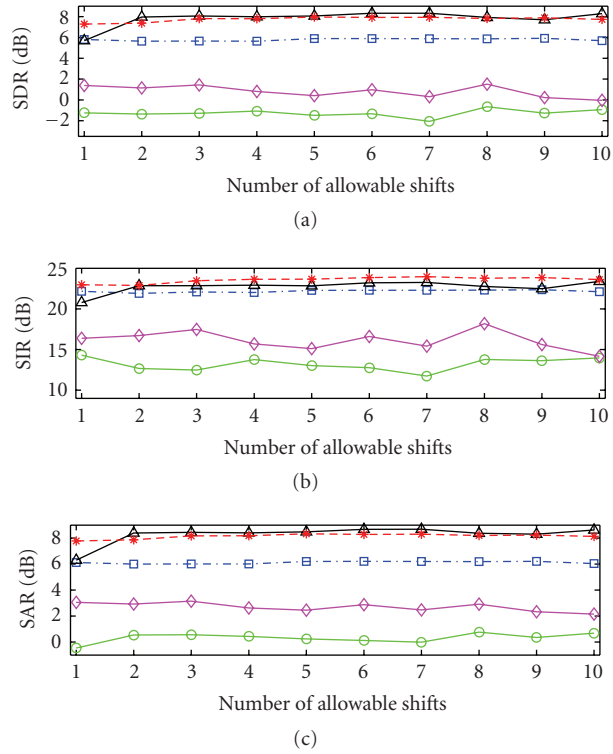

Figure 12: Performance Evaluation of SNTF (circle solid), refiltered SNTF (diamond solid), SSNTF (square dash-dotted), Source-Filter SSNTF (triangle solid) and Source-Filter SSNTF (star dashed) with noise basis functions for various allowable shifts in time.

these measures is available from [53]. As noted previously in Section 3, the provision of the lowest pitch note for each source was sufficient to determine the correct source ordering for all the SSNTF-based algorithms. In the case of the SNTF-based algorithms, the ordering of the sources was determined by associating a separated source with the original source which resulted in the best SIR score. This matching procedure was then checked manually to ensure no errors had occurred.

A number of different tests were run to determine the effect of signal duration on the performance of the algorithms and to determine the effect of using different numbers of allowable shifts in time. For the tests on signal duration, the mixture signals were truncated to lengths of 1, 2, 3, and 4 seconds in length, the number of time shifts was set to 5, and the performance of the algorithms was evaluated. A summary of the results obtained are shown in Figure 11. The results were obtained by averaging the metrics obtained for each separated source to give an overall score for each test mixture. The results for each mixture were then averaged to yield the data shown in the figure. It can be seen that the SSNTF-based algorithms all clearly outperform SNTF-based methods in all cases, though the use of refiltering does improve the performance of SNTF. It can also be seen that signal duration does not have much effect on the results obtained from SSNTF, with the results remaining relatively constant with signal duration, showing that SSNTF can capture harmonic sources even at relatively short signal durations.

In the case of the algorithms incorporating source filtering, performance improved with increased signal duration. This is particularly evident in the case of the SIR metric. This demonstrates that longer signal durations are required to properly capture filters for each instrument. This is to be expected as increased numbers of notes played by each instrument provide more information on which to learn the filter, while the harmonic model with fewer parameters does not require as much information for training. It should be noted that this trend was less evident in the stereo mixtures than in the mono mixtures, suggesting that the spatial positioning of sources in the stereo field may effect the ability to learn the source filters. This can possibly be tested by measuring the separation of the sources while varying the mixing coefficients and is an area for future investigation. Nonetheless, it can be seen that at longer durations the source-filter approaches outperform SSNTF, with the basic source-filter model performing better in terms of SDR and SAR, while the source-filter plus noise approach performs better in terms of SIR.

The results from testing the effect of the number of time shifts on the separation of the sources are shown in Figure 12. These were obtained using the same procedure used for the previous tests. The number of allowable shifts ranged from 1 to 10, which corresponds to a maximum

shift in time of approximately 0.2 second. Once again, the SSNTF-based algorithms clearly outperform SNTF-based approaches, regardless of the shift. However, it can be seen that for both SSNTF and the source-filter plus noise approach, performance is relatively constant with the number of allowable shifts, there is a small improvement in performance up until 7 shifts and beyond this performance degrades slightly. In the case of source-filter SSNTF, there is a noticeable improvement when going from one to two shifts, but beyond this there is little or no variation in performance with increased numbers of shifts. On investigating, this was found to be mainly evident in the stereo mixtures, with the performance of the mono mixtures remaining relatively constant, again highlighting the need to investigate the performance of the algorithms under different mixing coefficients. Overall, it can be seen that the performance of the algorithms is in line with that observed when varying signal duration, with the source-filter plus noise approach performing best in terms of SIR, while source-filter SSNTF performs better in terms of SDR and SAR. Further, the results suggest that in many cases, a single set of harmonic weights can be used to characterise pitched instruments without the need to incorporate timbral change with time.

On listening to the separated sources, the SSNTF-based approaches clearly outperform SNTF. It should be noted that in some cases, SNTF using refiltering resulted in audio quality comparable to the SSNTF-based approaches, however this was only in a small number of examples. In the majority of cases the addition of the source-filter improves on the results obtained by SSNTF. On comparing the source-filter approach to the source-filter plus noise model, it was observed that the results varied from mixture to mixture, with a considerable improvement in resynthesis quality of some sources and a reduction of quality in other cases, while in a large number of tests no major differences could be heard in the results. This shows that in many cases for clean mixture signals of pitched instruments, there is no need to incorporate noise basis functions. Nevertheless, the use of noise basis functions is still useful in the presence of noise or percussion instruments. It should also be noted that in half of the test mixtures SNTF did not manage to correctly separate the sources, which, in conjunction with the distortion due to the smearing of the frequency bins due to the mapping from log to linear frequency, goes a long way towards explaining the negative SDR and SIR scores. While SNTF using refiltering resulted in improved resynthesis in the cases where the sources had been separated correctly, it also suffered from the reliablity issues of the underlying SNTF technique and this is reflected in the poor scores for all metrics. This indicates that the SSNTF-based techniques are considerably more robust than SNTF-based techniques.

The separated sources can also be resynthesised via an additive synthesis approach, and on listening, the results obtained were comparable to those obtained from the spectrogram-based resynthesis. However, as the additive synthesis approach uses different phase information than the spectrogram-based resynthesis, the results are not comparable using the metrics used in this paper. This highlights the need to develop a set of perceptually-based metrics for sound source separation and is an area for future research.

Also investigated was the goodness of fit of the models to the original spectrogram data, as measured by the cost function. It was observed that the results obtained for SSNTF were on average 64% smaller than those for SNTF, despite the fact that SSNTF has a smaller number of free parameters, as the number of harmonics was considerably smaller than the number of frequency bins used in the constant Q spectrogram for SNTF. This highlights the benefits of using an approach solely formulated in the linear frequency domain. Using source-filter SSNTF, with an additional $K \times n$ parameters over SSNTF, resulted in an average reduction in the cost function of 76% in comparison to SNTF, and a reduction of 33% in comparison to SSNTF.

Overall it can be seen that the methods proposed in this paper offer a considerable improvement over previous separation methods using SNTF. Large improvements can be seen in the performance metrics over the previous SNTF method, and it can also be seen that the proposed models result in an improved fit to the original data.

7. Conclusions

The use of shift-invariant tensor factorisations for the purposes of musical sound source separation, with a particular emphasis on pitched instruments, has been discussed, and problems with existing algorithms were highlighted. The problem of grouping notes to sources can be overcome by incorporating shift invariance in frequency into the factorisation framework, but comes at the price of requiring the use of a log-frequency representation. This causes considerable problems when attempting to resynthesise the separated sources as there is no exact mapping available to map from a log-frequency representation back to a linear-frequency representation, which results in considerable degradation in the sound quality of the separated sources. While refiltering can overcome this problem to some extent, there are still problems with resynthesis.

A further problem with existing techniques was also highlighted, in particular the lack of a strict harmonic constraint on the recovered frequency basis functions. Previous attempts to impose harmonicity used an ad hoc constraint where the basis functions were zeroed in regions where no harmonic activity was expected. While this does guarantee that there will be no activity in these regions, it does not guarantee that the basis functions recovered will have the shape that a sinusoid would have if present in these regions.

Sinusoidal shifted 2D nonnegative tensor factorisation was then proposed as a means of overcoming both of these problems simultaneously. It takes advantage of the fact that a closed form solution exists for calculating the spectrum of a sinusoid of known frequency, and uses an additive-synthesis inspired approach for modeling pitched instruments, where each note played by an instrument is modelled as the sum of a fixed number of weighted sinusoids in harmonic relation to each other. These weights are considered to be invariant to changes in the pitch, and so each note is modelled using the same weights regardless of pitch. The frequency

spectrum of the individual harmonics is calculated in the linear frequency domain, eliminating the need to use a log-frequency representation at any point in the algorithm, and harmonicity constraints are imposed explicitly by using a signal dictionary of harmonic sinusoid spectra. Results show that using this signal model results in a better fit to the original mixture spectrogram than algorithms involving the use of a log-frequency representation, thereby demonstrating the benefits of being able to perform the optimisation solely in the linear-frequency domain.

However, it should be noted that the proposed model is not without drawbacks. In particular, best results were obtained if the pitch of the lowest note of each pitched instrument was provided to the algorithm. In most cases this information will not be readily available, and this necessitates the use of the standard shifted 2D nonnegative tensor factorisation algorithm to estimate these pitches before using the sinusoidal model. Research is currently ongoing on other methods to overcome this problem, but despite this, it is felt that the advantages of the new algorithm more than outweigh this drawback.

Using the same harmonic weights or instrument basis function regardless of pitch is only an approximation to the real world situation where the timbre of an instrument does change with pitch. To overcome this limitation, the incorporation of a source-filter model into the tensor factorisation framework had previously been proposed by others. Unfortunately, in the context of sound source separation, it was found that it was difficult to obtain good results using this approach as there were too many parameters to optimise. However, the addition of the strict harmonicity constraint proposed in this paper was found to restrict the range of solutions sufficiently to make the problem tractable.

It had previously been observed that the addition of harmonic constraints was required to create a system which could handle both pitched and percussive instrumentations simultaneously. However, previous attempts at such systems suffered due to the use of log-frequency representations and the lack of a strict harmonic constraint. The combined model presented here extends this earlier work from single channel to multichannel signals, and overcomes these problems by use of sinusoidal constraints applied in the linear-frequency domain, as well as incorporating the source filter model into the system, and so represents a more general model than those previously proposed.

In testing using common source separation performance metrics, the extended algorithms proposed were found to considerably outperform existing tensor factorisation algorithms, with considerably reduced signal distortion and artifacts in the resynthesis. The extended algorithms were also found to be more reliable than SNTF-based approaches.

In conclusion, it has been demonstrated that use of an additive-synthesis based approach for modelling instruments in a factorisation framework overcomes problems associated with previous approaches, as well as allowing extensions to existing models. Future work will concentrate on the improvement of the proposed models, both in terms of increased generality and in improved resynthesis of the separated sources, as well as investigating the effects of the mixing coefficients on the separations obtained. It is also proposed to investigate the use of frequency domain performance metrics as a means of increasing the perceptual relevance of source separation metrics.

Acknowledgments

This research was part of the IMAAS project funded by Enterprise Ireland. The authors wish to thank Mikel Gainza, Matthew Hart, and Dan Barry for their helpful discussions and comments during the preparation of this paper. The authors also wish to thank the reviewers for their helpful comments which resulted in a much improved paper.

References

[1] J. P. Stautner, *Analysis and synthesis of music using the auditory transform*, M.S. thesis, MIT Electrical Engineering and Computer Science Department, Massachusetts Institute of Technology, Cambridge, Mass, USA, 1983.

[2] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[3] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural Computation*, vol. 12, no. 2, pp. 337–365, 2000.

[4] B. A. Olshausen and D. J. Field, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology*, vol. 14, no. 4, pp. 481–487, 2004.

[5] D. Lee and H. Seung, "Learning the parts of objects by nonnegative matrix factorisation," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[6] P. Paatero and U. Tapper, "Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[7] M. Casey and A. Westner, "Separation of mixed audio sources by independent subspace analysis," in *Proceedings of the International Computer Music Conference (ICMC '00)*, pp. 154–161, Berlin, Germany, August-September 2000.

[8] T. Virtanen, "Sound source separation using sparse coding with temporal continuity objective," in *Proceedings of the International Computer Music Conference (ICMC '03)*, pp. 231–234, Singapore, September 2003.

[9] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA '03)*, pp. 177–180, New Paltz, NY, USA, October 2003.

[10] D. FitzGerald, B. Lawlor, and E. Coyle, "Sub-band independent subspace analysis for drum transcription," in *Proceedings of the 5th International Conference on Digital Audio Effects (DAFX '02)*, pp. 65–69, Hamburg, Germany, September 2002.

[11] S. Raczynski, N. Ono, and S. Sagayama, "Multipitch analysis with harmonic nonnegative matrix approximation," in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR '07)*, pp. 381–386, Vienna, Austria, September 2007.

[12] P. Sajda, S. Du, and L. Parra, "Recovery of constituent spectra using non-negative matrix factorization," in *Wavelets: Applications in Signal and Image Processing X*, vol. 5207 of *Proceedings of SPIE*, pp. 321–331, San Diego, Calif, USA, August 2003.

[13] T. Virtanen, *Sound source separation in monaural music signals*, Ph.D. thesis, Tampere University of Technology, Tampere, Finland, 2006.

[14] D. FitzGerald, M. Cranitch, and E. Coyle, "Shifted non-negative matrix factorisation for sound source separation," in *Proceedings of the 13th IEEE/SP Workshop on Statistical Signal Processing*, pp. 1132–1137, Bordeaux, France, July 2005.

[15] M. Mørup, L. K. Hansen, and S. M. Arnfred, "Sparse higher order non-negative matrix factorization," *Technical Report IMM2007-04658, Technical University of Denmark*.

[16] S. A. Abdallah and M. D. Plumbley, "Polyphonic transcription by non-negative sparse coding of power spectra," in *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR '04)*, pp. 318–325, Barcelona, Spain, October 2004.

[17] R. M. Parry and I. Essa, "Incorporating phase information for source separation via spectrogram factorization," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 2, pp. 661–664, Honolulu, Hawaii, USA, April 2007.

[18] R. M. Parry and I. Essa, "Phase-aware non-negative spectrogram factorization," in *Proceedings of the 7th International Conference on Independent Component Analysis and Signal Separation (ICA '07)*, vol. 4666 of *Lecture Notes in Computer Science*, pp. 536–543, London, UK, September 2007.

[19] R. Kompass, "A generalized divergence measure for non-negative matrix factorization," in *Proceedings of the Neuroinformatics Workshop*, Torun, Poland, September 2005.

[20] A. Cichocki, R. Zdunek, and S.-I. Amari, "Csiszár's divergences for non-negative matrix factorization: family of new algorithms," in *Proceedings of the 6th International Conference on Independent Component Analysis and Blind Signal Separation (ICA '06)*, vol. 3889 of *Lecture Notes in Computer Science*, pp. 32–39, Springer, Charleston, SC, USA, March 2006.

[21] P. D. O. Grady, *Sparse separation of under-determined speech mixtures*, Ph.D. thesis, National University of Ireland Maynooth, Kildare, Ireland, 2007.

[22] D. FitzGerald, *Automatic drum transcription and source separation*, Ph.D. thesis, Dublin Institute of Technology, Dublin, Ireland, 2004.

[23] B. W. Bader and T. G. Kolda, "Algorithm 862: MATLAB tensor classes for fast algorithm prototyping," *ACM Transactions on Mathematical Software*, vol. 32, no. 4, pp. 635–653, 2006.

[24] D. FitzGerald, M. Cranitch, and E. Coyle, "Non-negative tensor factorisation for sound source separation," in *Proceedings of the Irish Signals and Systems Conference*, pp. 8–12, Dublin, Ireland, September 2005.

[25] R. M. Parry and I. Essa, "Estimating the spatial position of spectral components in audio," in *Proceedings of the 6th International Conference on Independent Component Analysis and Blind Signal Separation (ICA '06)*, vol. 3889 of *Lecture Notes in Computer Science*, pp. 666–673, Charleston, SC, USA, March 2006.

[26] D. Barry, B. Lawlor, and E. Coyle, "Sound source separation: azimuth discrimination and resynthesis," in *Proceedings of the 7th International Conference on Digital Audio Effects (DAFX '04)*, Naples, Italy, October 2004.

[27] P. Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," in *Proceedings of the 5th International Conference on Independent Component Analysis and Blind Signal Separation*, vol. 3195 of *Lecture Notes in Computer Science*, pp. 494–499, Grenada, Spain, September 2004.

[28] T. Virtanen, "Separation of sound sources by convolutive sparse coding," in *Proceedings of the ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing (SAPA '04)*, Jeju, Korea, October 2004.

[29] M. N. Schmidt and M. Mørup, "Nonnegative matrix factor 2-D deconvolution for blind single channel source separation," in *Proceedings of the 6th International Conference on Independent Component Analysis and Blind Signal Separation (ICA '06)*, vol. 3889 of *Lecture Notes in Computer Science*, pp. 700–707, Charleston, SC, USA, March 2006.

[30] E. Vincent and X. Rodet, "Music transcription with ISA and HMM," in *Proceedings of the 5th International Conference on Independent Component Analysis and Blind Signal Separation (ICA '04)*, vol. 3195 of *Lecture Notes in Computer Science*, pp. 1197–1204, Granada, Spain, September 2004.

[31] A. B. Nielsen, S. Sigurdsson, L. K. Hansen, and J. Arenas-García, "On the relevance of spectral features for instrument classification," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 2, pp. 485–488, Honolulu, Hawaii, USA, April 2007.

[32] J. C. Brown, "Calculation of a constant Q spectral transform," *Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[33] J. Eggert, H. Wersing, and E. Körner, "Transformation-invariant representation and NMF," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN '04)*, vol. 4, pp. 2535–2539, Budapest, Hungary, July 2004.

[34] D. FitzGerald, M. Cranitch, and E. Coyle, "Shifted 2D non-negative tensor factorisation," in *Proceedings of the Irish Signals and Systems Conference*, pp. 509–513, Dublin, Ireland, June 2006.

[35] M. Mørup and M. N. Schmidt, "Sparse non-negative tensor 2D deconvolution (SNTF2D) for multi channel time-frequency analysis," Tech. Rep., Technical University of Denmark, Copenhagen, Denmark, 2006.

[36] D. FitzGerald, M. Cranitch, and E. Coyle, "Sound source separation using shifted non-negative tensor factorisation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, vol. 5, pp. 653–656, Toulouse, France, May 2006.

[37] M. Slaney, "Pattern playback in the 90s," in *Advances in Neural Information Processing Systems 7*, MIT Press, Cambridge, Mass, USA, 1996.

[38] D. FitzGerald, M. Cranitch, and E. Coyle, "Resynthesis methods for sound source separation using non-negative factorisation methods," in *Proceedings of the Irish Signals and Systems Conference*, Derry, Ireland, September 2007.

[39] D. FitzGerald, M. Cranitch, and M. Cychowski, "Towards an inverse constant Q transform," in *Proceedings of the 120th AES Convention*, Paris, France, May 2006.

[40] M. N. Schmidt and M. Mørup, "Nonnegative matrix factor 2-D deconvolution for blind single channel source separation," in *Proceedings of the 6th International Conference on Independent Component Analysis and Blind Signal Separation (ICA '06)*, vol. 3889 of *Lecture Notes in Computer Science*, pp. 700–707, Charleston, SC, USA, March 2006.

[41] D. DeFatta, J. Lucas, and W. Hodgkiss, *Digital Signal Processing: A System Design Approach*, John Wiley & Sons, New York, NY, USA, 1988.

[42] A. Freed, X. Rodet, and P. Depalle, "Performance, synthesis and control of additive synthesis on a desktop computer using FFT-1," in *Proceedings of the 19th International Computer*

Music Conference (ICMC '93), vol. 19, pp. 98–101, Waseda University Center for Scholarly Information, International Computer Music Association, Tokyo, Japan, September 1993.

[43] N. F. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*, Springer, New York, NY, USA, 2nd edition, 1998.

[44] Tensor Toolbox for Matlab, http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/.

[45] J. Woodruff, B. Pardo, and R. Dannenberg, "Remixing stereo music with score-informed source separation," in *Proceedings of the 7th International Symposium on Music Information Retrieval (ISMIR '06)*, Victoria, Canada, October 2006.

[46] T. Virtanen and A. Klapuri, "Analysis of polyphonic audio using source-filter model and non-negative matrix factorization," in *Proceedings of the Advances in Models for Acoustic Processing, Neural Information Processing Systems Workshop*, Whistler, Canada, December 2006.

[47] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen, "Discrete-time modelling of musical instruments," *Reports on Progress in Physics*, vol. 69, no. 1, pp. 1–78, 2006.

[48] M. R. Schroeder and B. S. Atal, "Code-excited linear prediction (CELP): high-quality speech at very low bit rates," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '85)*, vol. 10, pp. 937–940, Tampa, Fla, USA, April 1985.

[49] X. Serra, "Musical sound modeling with sinusoids plus noise," in *Musical Signal Processing*, G. D. Poli, A. Picialli, S. T. Pope, and C. Roads, Eds., Swets & Zeiltlinger, Lisse, The Netherlands, 1997.

[50] Ö. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.

[51] P. Siedlaczek, Advanced Orchestra Library Set, 1997.

[52] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[53] BSS_Eval toolbox, http://bassdb.gforge.inria.fr/bss_eval.