Extended Non-negative Tensor Factorisation models for Musical Sound Source Separation

Derry Fitzgerald, Dept. of Electronic Engineering, Cork Institute of Technology, Cork, Ireland
Matt Cranitch, Dept. of Electronic Engineering, Cork Institute of Technology, Cork, Ireland
Eugene Coyle, Dept. of Control Engineering, Dublin Institute of Technology, Kevin Street, Dublin, Ireland

Abstract

Recently, shift invariant tensor factorisation algorithms have been proposed for the purposes of sound source separation of pitched musical instruments. However, 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 [3]. However, it was not until the development of Independent Component Analysis (ICA) [2], and techniques such as Sparse Coding [4] and Non-negative Matrix Factorisation (NMF) [32] that factorisation-based approaches received much attention for the analysis and separation of musical audio signals.

Factorisation-based approaches were initially applied to single channel separation of musical sources, where the input signal was used to generate a spectrogram $X$ of size $n \times m$. This spectrogram was then factorised to yield a reduced rank approximation

$$X \approx \hat{X} = AS$$

where $A$ is of size $n \times r$ and $S$ is of size $r \times m$, with $r$ less than $n$ and $m$. In this case, the columns of $A$ contains frequency basis functions, while the corresponding rows of $S$ contains 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 sources sum together to generate the spectrogram. This does not take into account the effects of phase when the sources 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 [34] 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 non-negative constraints in conjunction with a suitable cost function [35, 36].

A commonly used cost function, proposed by Hoyer [4] is:

$$D_{spar}(A, S) = \frac{1}{2} \|X - AS\|^2 + \lambda \sum_{i,j} f(S_{ij})$$

which attempts to balance the reconstruction of the spectrogram with the sparseness of the source activations, where $\lambda$ determines the trade-off between sparseness and accurate reconstruction and $f$ defines how sparseness is measured, with $f(s) = |s|$ being a commonly used choice of measure. A problem with this cost function is determining the optimum value for $\lambda$. Recently a method for determining this value has been proposed by Morup [5], using an $L$-curve approach where the reconstruction error is plotted against the $L_0$-norm of the sparse code matrix. The point of maximum curvature of this plot is then taken as the optimal value of $\lambda$.

Another commonly used cost function is the generalised Kullback-Liebler divergence proposed by Lee and Seung
\[
D(X|\hat{X}) = \sum_{ij} \left( X_{ij} \log \frac{X_{ij}}{\hat{X}_{ij}} - X_{ij} + \hat{X}_{ij} \right)
\]  
(3)

which is equivalent to assuming a Poisson noise model for the data. 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 [1],[27]. A sparseness constraint can also be added to this cost function, and multiplicative update equations which ensure non-negativity can be derived for these cost functions [6]. Other cost functions have been developed for factorisation of audio spectrograms such as that of Abdallah which assumes multiplicative gamma-distributed noise in power spectrograms [31]. A similar cost function recently proposed by Parry attempts to incorporate phase into the factorisation by using a probabilistic phase model [7]. Families of parameterised cost functions have been proposed, such as the Beta divergence [8], and Csiszar’s divergences [9]. The use of the Beta divergence for the separation of speech signals has been investigated by O’Grady [10], 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 [34, 35], but in practice it is difficult to obtain the correct clustering for reasons discussed in [33].

More recently, the above techniques have been extended to tensor factorisation models to deal with stereo or multi-channel signals by FitzGerald [28] and Parry [11]. Adopting the convention that tensors are denoted using calligraphic uppercase letters such as \(\mathcal{A}\), the signal model can be expressed as:

\[
\hat{X} \approx \sum_{k=1}^{K} \mathcal{G}_k \odot \mathcal{A}_k \odot \mathcal{S}_k
\]  
(4)

where \(\mathcal{X}\) is an \(s \times n \times m\) tensor containing the spectrograms of the \(s\) channels, \(\mathcal{G}\) is an \(s \times K\) matrix containing the gains of the \(K\) basis functions in each channel, \(\mathcal{A}\) is a matrix of size \(n \times K\) containing a set of frequency basis functions, and \(\mathcal{S}\) is a matrix of size \(m \times K\) containing the amplitude basis functions. Outer product multiplication is denoted by \(\odot\) and \(k\) denotes the \(k\)th column of a given matrix.

Many commercial stereo recordings 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. 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 [13] and Virtanen [12]. 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. This shortcoming had been addressed by Vincent using a non-linear ISA approach [37], but this technique required pre-training 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 frequency spectra of notes played by a given instrument do change with pitch. 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 synthesizers, where a prerecorded note of a given pitch is used to generate other notes close in pitch to the original note.

Up until 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) [26]. 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:

\[
X = CY
\]  
(5)
where \( Y \) is a linear-frequency spectrogram with \( f \) frequency bins and \( t \) time frames. \( C \) 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 \( X \) is a log-frequency spectrogram of size \( cf \times t \). It can be seen that \( C \) 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 \( f_k = f_0 \beta^k \) where \( f_0 \) denotes the center frequency of the \( k^{th} \) band, \( \beta = 2^{1/12} \) and \( f_0 \) is a reference frequency, then the spacing of the bands will match that of the even-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 [13] 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 \( I(:,[n,1:n-1]) \) where the identity matrix is denoted by \( I \) and the ordering of the columns is contained in the square brackets. For \( K \) allowable translations, these translation matrices are then grouped into a translation tensor of size \( n \times K \times n \).

When dealing with tensor notation, we use the conventions described by Bader and Kolda in [38]. As noted previously, 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 \( W \) is a tensor of size \( I_1 \times \cdots \times I_N \times J_1 \times \cdots \times J_M \) and \( Y \) 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:

\[
\langle WY \rangle_{1:N,1:N}(j_1, \ldots, j_M, k_1, \ldots, k_P) = \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} W(i_1, \ldots, i_N, j_1, \ldots, j_M) Y(i_1, \ldots, i_N, k_1, \ldots, k_P)
\]

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. Further, for simplicity of notation, we use the convention that \( k \) denotes the tensor slice associated with the \( k^{th} \) source, with the singleton dimension included in the size of the slice.

### 2.1 Shifted 2D Non-negative Tensor Factorisation

All of the algorithms incorporating shift invariance can be seen as special cases of a more general model, shifted 2D Non-negative Tensor Factorisation (SNTF), proposed by FitzGerald [30], and separately by [14]. The SNTF model can then be described as:

\[
\mathcal{X} \approx \sum_{k=1}^{K} \langle \mathcal{G}_k \langle \langle T \mathcal{A}_k \rangle_{3,1} \langle \mathcal{S}_k \mathcal{P} \rangle_{3,1} \rangle_{2,4,1,3} \rangle_{2,2}
\]

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. \( T \) 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. If the number of channels is set to \( r = 0 \), and the allowable frequency translations \( z \) is also set to zero, then the model collapses to that proposed by Virtanen in [12]. Similarly setting \( p = 0 \) results in the model proposed in [29], while setting both \( r \) and \( p \) to zero results in the model described in eqn. (4). In [30], the generalised Kullback-Liebler 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 \( k^{th} \) separated source can then be estimated from eqn. (6) using only the tensor slices associated with the \( k^{th} \) 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 [17]. 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, though this approach can often result in the increased presence of other sources in the separated signals.

As noted previously, the mapping from log-frequency to linear frequency domain 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.
[15, 16]. However, a simpler method of overcoming this problem is to incorporate the mapping into the model. This can be done by replacing $T$ in eqn. (6) with $(CT)^{(2,1)}$ where $C$ is an approximate map from log to linear domain. This mapping can simply be the transpose of $C$, the mapping used in eqn. (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 [10]. 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.

3 Sinusoidal Shifted 2D Non-negative Tensor Factorisation

While SNTF has been shown to be capable of separating mixtures of harmonic pitched instruments, 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 non-zero values at regions which correspond to a perfectly harmonic sound has been proposed by Virtanen [1] and later by Raczynski [18], who used it for the purposes of multi-pitch 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.

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 [20]. For example, using a hanning 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:

$$X(k) = |0.5D(g) + 0.25(D_1(g) + D_2(g))|$$

where $g = f_k - f_0$ with $f_k = \frac{k2\pi}{N}$ being the centre frequency of the $k^{th}$ FFT bin and where $D$ is defined as:

$$D(g) = \frac{\sin\left(\frac{2\pi D g}{N}\right)}{\sin\left(\frac{\pi g}{N}\right)}$$

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 efficient additive synthesis, in particular the inverse FFT method of Rodet and Depalle [19].

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 $n$ by $h$ 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 $n$ by $z$ by $h$. 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 hanning window as an example, the tensor can then be defined as:

$$T(k, i, j) = |0.5D(g_{ki}) + 0.25(D_1(g_{ki}) + D_2(g_{ki}))|$$

where $g_{ki} = f_k - f_{i,j}$ 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 $\beta$ is as previously defined in section 2. This assumes equal-tempered tuning, but other tuning systems can be used if required.

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

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

where $\alpha$ is the inharmonicity factor for the instrument in question [21]. 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 $T(k, i, j)$ 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.

$T$ 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:

$$X \approx \hat{X} = \sum_{k=1}^{K} \langle G_k \langle T \mathcal{W}_k \rangle_{[3,1]} \langle S_k \mathcal{P} \rangle_{[3,1]} \rangle_{[2,4,1,3]} \langle 2,2 \rangle$$

(11)

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

$$G_k = G_k \otimes \frac{\langle [D \langle T \rangle \mathcal{W}_k \rangle_{[1,3,1,2]} \langle S_k \mathcal{P} \rangle_{[3,1]} \rangle_{[1,2,4]} \{1,2,4\}}{\langle [G_k \langle T \rangle \mathcal{O} \rangle_{[1,3,1,2]} \langle S_k \mathcal{P} \rangle_{[3,1]} \rangle_{[1,2,4]} \{1,2,4\}}$$

(12)

$$W_k = W_k \otimes \frac{\langle (G_k \langle T \rangle \mathcal{A} \mathcal{k} \rangle_{[2,5,1]} \langle D \rangle_{[2,1,2]} \{P \} \rangle_{[2,3,2,1]}}{\langle [G_k \langle T \rangle \mathcal{A} \mathcal{k} \rangle_{[2,5,1]} \langle O \rangle_{[2,1,2]} \{P \} \rangle_{[2,3,2,1]}}$$

(13)

$$S_k = S_k \otimes \frac{\langle (G_k \langle T \rangle \mathcal{A} \mathcal{k} \rangle_{[2,5,1]} \langle D \rangle_{[2,1,2]} \{P \} \rangle_{[2,3,2,1]}}{\langle [G_k \langle T \rangle \mathcal{A} \mathcal{k} \rangle_{[2,5,1]} \langle O \rangle_{[2,1,2]} \{P \} \rangle_{[2,3,2,1]}}$$

(14)

where $D = X \otimes \hat{X}$ and $O$ is an all-ones tensor with the same dimensions as $X$, and all division is taken as elementwise.

These update equations are similar to those of SNTF, with $\mathcal{W}$ substituted for $\mathcal{A}$. However, as opposed to the original SNTF algorithm, where $T$ described a set of translations, it now contains a sinusoidal signal dictionary, and $\mathcal{W}$ contains sets of harmonic weights describing the instrument as opposed to the instrument basis functions contained in $\mathcal{A}$. It is proposed to call this new algorithm Sinsoidal Shifted 2D Non-negative 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, which avoids the problem of generating phase information for the separated signal.

The SSNTF algorithm was implemented in Matlab using the Tensor Toolbox available from [39], as were all subsequent algorithms described in this paper. The algorithm was found to converge reliably. 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 [22], 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.

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

As an example of the improved reconstruction that SSNTF can provide, figure 1 shows the frequency spectra of a flute note separated from a single channel mixture of flute and piano. 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, and which takes into account the expected shape of a sinusoid of a given frequency.

Figure 2 shows the original mixture signal of piano and flute, piano signals (the original waveform, SSNTF separated waveform, SSNTF additive-resynthesised waveform and SNTF separated waveform respectively) and flute signals (again the original waveform, SSNTF separated waveform, SSNTF additive-resynthesised waveform and SNTF separated waveform respectively). It can be seen that the signals recovered using SSNTF, using either spectrogram-based reconstruction or additive resynthesised reconstruction are considerably closer to the original signals than those recovered by SNTF, with much less amplitude modulation evident in the SSNTF signals in particular.
improved recovery of the sources was also noted on playback of the separated SSNTF signals in comparison to those obtained using SNTF.

3.1 Test Results

In order to quantify the performance of SSNTF in comparison to SNTF in the context of modelling mixtures of pitched instruments, a number of tests were carried out. Fifteen test signals of 4 seconds duration containing mixtures of melodies played by two different instruments were created using a large library of orchestral samples [40]. Samples from a total of 15 different orchestral instruments were used. A wide range of pitches were covered, from 87.31 Hz to 1567.98 Hz, 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.

Spectrograms were generated for each test signal and these were analysed ten times by both SNTF and SSNTF, with random initialisation of the variables for each run, with the exception of the frequency basis functions for SNTF, which were initialised with harmonic basis functions at the frequency of the lowest note played by each instrument in each example. This was done to put SNTF on an equal footing with SSNTF, 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, the number of translations in time set to five, and the number of harmonics used in SSNTF was set to 12. The algorithms were ran for 100 iterations, and the average cost scores from the 10 runs for the fifteen signals are presented in Table 1.

In all cases the algorithms converged reliably, and for each test example the SSNTF scores were always considerably lower than the lowest SNTF score, with the SSNTF score being on average 64% smaller than the SNTF score, thus indicating that the model provides a much better fit to the input data. While the fact that SSNTF provides a much better fit to the original data does not say anything about whether the sources have been separated correctly, it does suggest that when separation is successful, the resultant resynthesis will be considerably better than using SNTF as the original data will have been reconstructed much more accurately.

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 proposed the incorporation of a source-filter model approach in the factorisation method as a means of overcoming this problem [23]. In the source-filter framework for sound production, the source is typically a vibrating object, such as a violin string, and the filter 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 has been in use for some time.
in both sound synthesis and speech coding [24, 25], but had not been previously incorporated into 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 \( F \) of size \( n \times K \times n \). In this case the \( k \)th slice of \( F \) 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:

\[
\mathcal{X} \approx \hat{\mathcal{X}} = \sum_{k=1}^{K} (G_k \langle R_k W_k \rangle_{\{2,4,[2,1]\}} \langle V_k \rangle_{\{2,4,[2,1,3]\}})_{\{2,2\}}
\]

(15)

where \( R_k = \langle F_k T \rangle_{\{3,1\}} \) and \( V_k = \langle S_k P \rangle_{\{3,1\}} \).

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

\[
G_k = G_k \otimes \langle D \langle R_k W_k \rangle_{\{2,4,[2,1]\}} \langle V_k \rangle_{\{2,5,[2,1,3]\}} \rangle_{\{2,1,2\}}
\]

(16)

\[
F_k = \langle T W_k \rangle_{\{3,1\}} \langle V_k \rangle_{\{2,4,1,3\}} \rangle_{\{1,3,2,3\}}
\]

(17)

\[
V_k = W_k \otimes \langle \langle D \langle R_k W_k \rangle_{\{2,2\}} \langle V_k \rangle_{\{1,2,4,1,4\}} \rangle_{\{1,2,4\}} \rangle_{\{2,2\}}
\]

(18)

\[
S_k = S_k \otimes \langle \langle D \langle R_k W_k \rangle_{\{2,2\}} \langle V_k \rangle_{\{1,2,4,1,4\}} \rangle_{\{1,2,4\}} \rangle_{\{2,2\}}
\]

(19)

The resultant algorithm was tested using the same procedures and test data as SSNTF and was found to converge reliably. The addition of the fixed instrument filters was found to further reduce the cost score by on average 76% in comparison with SNTF and 33% in comparison with SSNTF. This demonstrates that the source-filter model is a better fit to the original data than the previous two methods. Table 1 also gives the average cost function scores for the fifteen test examples.

Figure 3 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 as a result of the harmonic constraints imposed by SSNTF which means that 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 that of the flute.

Figure 4 shows the recovered waveforms for the piano and flute respectively. It can be observed that the flute signal is closer to the original given in figure 2, with the onsets and offsets of the flute more clearly visible. This demonstrates the utility of using the source-filter approach as a means of improving the accuracy of the SSNTF model.

5 Combined model

Musical signals, especially popular music, typically contain unpitched instruments such as drum sounds in addi-
tation 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 [1], 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 plus residual sinusoidal analysis techniques described by Serra [41] in that it allows the pitched or sinusoidal part of the signal to be resynthesised separately from the noise part of the signal.

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:

\[
\mathcal{X} \approx \hat{\mathcal{X}} = \sum_{k=1}^{K} (g_{k} \langle \mathcal{R}_{k}, \mathcal{W}_{k} \rangle \{[2,4],[2,1]\} V_{k}) \{[2,4],[2,1,3]\} \{2,2\} \\
+ \sum_{l=1}^{L} \langle \mathcal{M}_{l} A_{l} \rangle \langle \mathcal{H}_{l} Q \rangle \{[2,1]\} \{[2,3,1,2]\} \{2,2\}
\]

(20)

where \( \mathcal{M} \) is a tensor of size \( r \times L \), which contains the gains of each of the \( L \) percussive sources, \( \mathcal{A} \) is now a tensor of size \( n \times L \times q \) where \( q \) is the number of allowable time shifts for the percussive sources, \( \mathcal{H} \) is a tensor of size \( L \times m \) and \( Q \) is a translation tensor of size \( m \times q \times m \). Multiplicative update equations, based on the generalised Kullback-Liebler 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:

\[
\mathcal{M}_{l} = \mathcal{M}_{l} \otimes \frac{\langle \mathcal{D} (\mathcal{A}_{l} (\mathcal{H}_{l} Q) \{[2,1]\} \{[2,3,1,2]\} \{[2,3,1,3]\}) \rangle}{\langle \mathcal{O} (\mathcal{A}_{l} (\mathcal{H}_{l} Q) \{[2,1]\} \{[2,3,1,2]\} \{[2,3,1,3]\}) \rangle} \\
\mathcal{A}_{l} = \mathcal{A}_{l} \otimes \frac{\langle (\mathcal{M}_{l} \mathcal{D}) \{[1,1]\} (\mathcal{H}_{l} Q) \{[2,1]\} \{[1,3,1,3]\} \rangle}{\langle \mathcal{M}_{l} Q \{[1,1]\} (\mathcal{H}_{l} Q) \{[2,1]\} \{[1,3,1,3]\} \rangle} \\
\mathcal{H}_{l} = \mathcal{H}_{l} \otimes \frac{\langle (\mathcal{M}_{l} \mathcal{A}_{l}) \{[2,2]\} (\mathcal{D} Q) \{[3,3]\} \{[1,3,4],[1,2,4]\} \rangle}{\langle (\mathcal{M}_{l} \mathcal{A}_{l}) \{[2,2]\} (\mathcal{O} Q) \{[3,3]\} \{[1,3,4],[1,2,4]\} \rangle}
\]

(21) (22) (23)

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 [41] in that it allows the pitched or sinusoidal part of the signal to be resynthesised separately from the noise part of the signal.

Figure 5: Waveforms of (a,b) original stereo mixture, (c) flute, (d) piano, (e) trumpet, (f) hi-hats, (g) snare and (h) kick drum

Figure 6: Separated Waveforms of (a) flute, (b) piano, (c) trumpet, (d) hi-hats, (e) snare and (f) kick drum

As an example of the use of the combined model, figure 5 shows a stereo mixture containing three pitched instruments, flute, piano and trumpet, and three percussion instruments, hi-hats, snare and kick drum, as well as the
original unmixed waveforms of these instruments. The piano, snare and kick drum were all panned to the center, with the hi-hats and flute panned mid-left and the trumpet mid-right. Figure 5 shows the separated signals obtained using the combined model. It can be seen that the sources have been recovered well, with each individual instrument identifiable. Some trace of the hi-hats is visible in the snare signal, 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 the combined model and 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 [42, 43].

6 Performance Evaluation

The performance of the algorithms was evaluated on the same set of test signals as previously described in section 3.1. The evaluation metrics used were 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 [44] and a Matlab toolbox to calculate these measures is available from [45].

Table 2 shows the results obtained, measured in dB, using these metrics for all the test signals using SNFT, SSNTF with spectrogram reconstruction-based resynthesis, SSNTF using additive synthesis reconstruction (denoted by A-SSNTF), Source-Filter SSNTF, and Source-Filter SSNTF with the addition of noise basis functions (denoted by SF-SSNTF + N) respectively. In this last case, two noise basis functions were learned in order to aid the elimination of noise and artifacts from the harmonic sources. It can be seen that the results for SDR and SAR are in line with those obtained from measuring the cost function scores, where source-filter SSNTF outperforms SSNTF and SNFT, with SNFT providing significantly worse results than either of the methods proposed in this paper. In the case of SIR, both SSNTF and source-filter SSNTF clearly outperform SNFT, and in general, the source-filter approach gives better SIR scores than SNFT, but for some sources, SNFT obtains a better SIR. In general, regardless of the method used, the SIR scores are considerably better than those of SAR and SDR, indicating that tensor factorisation approaches in general give good rejection of other sources present in the mixture signals. It was also observed that a negative SIR score for either source in a mixture when using SNTF indicated that the sources had not been separated correctly. However, this does not hold for other methods tested such as SSNTF using additive synthesis reconstruction.

The addition of the noise basis functions can be seen to have improved the performance of half of the sources, while slightly degrading it in other cases. In many cases, improvements were obtained in the scores for one source at the expense of the other source in the mixture, and on average the addition of the noise basis functions results in slightly lower SDR and SAR, and a slightly improved SIR.

The additive synthesis reconstruction using SSNTF performed poorly in terms of both SDR and SAR, and in some cases, SIR as well. However, on listening to the resynthesis of the various sources using this technique and those obtained using SSNTF with spectrogram-based resynthesis, it was found that the sources were of comparable audio quality. This is not surprising as both methods use the same information to reconstruct the signals, the principal difference between the two lying in the phases of the sinusoids used to generate the reconstruction. This results in large differences to the time-domain waveforms, with the spectrogram-based reconstruction considerably closer to the original instrument signals. Nevertheless, regardless of method, the reconstructions should, and do sound quite similar. This serves to highlight that measuring separation performance in the time domain is not an adequate way to obtain perceptually reliable results. It remains an open issue as to how best measure the separation performance from a perceptual viewpoint, and the authors are currently investigating a frequency domain based approach.

Despite the problems with the time domain metrics for additive synthesis reconstruction, the results obtained by the other techniques for both SDR and SAR are higher by some distance than those obtained via SNFT. This strongly suggests that these techniques represent a considerable improvement over previous shift-invariant factorisation approaches which required the use of a log-frequency representation.

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 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.

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 non-negative 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 to modeling pitched instruments, where each note played by an instrument is modeled 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 modeled using the same weights regardless of pitch. The frequency spectra of the harmonics are 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. Test results show that using this signal models results in a considerably better fit to the original mixture spectrogram than algorithms involving the use of a log-frequency representation, thereby demonstrating both the benefits of using a harmonic constraint and 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 non-negative 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. The resulting source-filter factorisation model was also demonstrated to be a better fit to the original data than both standard and sinusoidal shifted 2D non-negative tensor factorisation techniques.

It had previously been observed that the addition of harmonic constraints was required to create a system which could handle both pitched and percussive instrumentation 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 multi-channel 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 sound 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. However, a problem with the use of time domain metrics, in particular when used with additive synthesis signal reconstruction, has been highlighted.

In conclusion, it has been demonstrated that use of an additive-synthesis based approach to modeling 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. 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.

8 Acknowledgements

This research was carried out as part of the IMAAS project funded by Enterprise Ireland under the Commercialisation Fund. The authors wish to thank Mikel Gainza, Matthew Hart and Dan Barry for their helpful discussions and comments during the preparation of this paper.

References

[1] Virtanen, T., "Sound Source Separation in Monaural Music Signals", Tampere University of Technology, 2006.

[2] Comon, P., "Independent Component Analysis - a new concept?", Signal Processing, 36, pp. 287-314, 1994

[3] Stautner, J.P., "Analysis and Synthesis of Music using the Auditory Transform", Masters Thesis, MIT EECS Department, 1983

[4] Hoyer, P.O. (2002) "Non-negative sparse coding" in Neural Networks for Signal Processing XII (Proc. IEEE Workshop on Neural Networks for Signal Processing), pp. 557-565, Martigny, Switzerland.

[5] M. Mørup, M. N. Schmidt, L. K. Hansen, Shift Invariant Sparse Coding of Image and Music Data, submitted, JMLR, 2007

[6] M. Mørup, L. K. Hansen, S. M. Arnfred, Sparse Higher Order Non-negative Matrix Factorization, submitted to Neural Computation, 2006

[7] R. Mitchell Parry and Irfan Essa, "Phase-aware Non-negative Spectrogram Factorization" In Proc. Int’l Conf. on Independent Component Analysis and Source Separation, 2007.

[8] R. Kompass. A generalized divergence measure for non-negative matrix factorization. In Neuroinformatics workshop, Torun, Poland, Sept. 2005.

[9] A. Cichocki, R. Zdunek, and S.-I. Amari, "Csiszar’s Divergences for Non-Negative Matrix Factorization: Family of New Algorithms", In: Independent Component Analysis and Blind Signal Separation, ICA 2006, Charleston, SC, USA, March 5-8, 2006, Lecture Notes in Computer Science, Vol. 3889, Springer, pp. 32 - 39, 2006

[10] P. D. O Grady. Sparse Separation of Under-Determined Speech Mixtures. PhD thesis, National University of Ireland Maynooth, 2007

[11] R. Mitchell Parry and Irfan Essa, “Estimating the Spatial Position of Spectral Components in Audio”, In Proc. Int’l Conf. on Independent Component Analysis and Blind Source Separation, 2006.

[12] Virtanen, Tuomas, Separation of Sound Sources by Convolutive Sparse Coding, ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing

[13] Smaragdis, P. 2004. Non-Negative Matrix Factor Deconvolution; Extraction of Multiple Sound Sources from Monophonic Inputs. In 5th International Conference on Independent Component Analysis and Blind Signal Separation, Grenada, Spain, September 2004.

[14] M. Mørup, M. N. Schmidt, Sparse Non-negative Tensor 2D Deconvolution (SNTF2D) for multi channel time-frequency analysis, Technical Report, Technical University of Denmark, 2006.

[15] D. FitzGerald, M Cranitch, E. Coyle, “Resynthesis methods for Sound Source Separation using Non-negative Factorisation methods”, Proceedings of the Irish Signals and Systems Conference, Derry 2007.

[16] D. FitzGerald, M. Cranitch, M. Cychowski, “Towards an Inverse Constant Q Transform”, 120th AES Convention, Paris, France, 2006
[17] M. Slaney, “Pattern Playback in the 90s”, Advances in Neural Information Processing Systems 7, MIT Press, 1996.

[18] Raczynski, S., Ono, N., and Sagayama, S., “Multipitch Analysis with Harmonic Nonnegative Matrix Approximation”. Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007), Vienna, Austria, 2007.

[19] Freed, A., Rodet, X., Depalle, P. "Performance, Synthesis and Control of Additive Synthesis on a Desktop Computer Using FFT-1." Proceedings of the 19th International Computer Music Conference, Waseda University Center for Scholarly Information 1993, International Computer Music Association.

[20] DeFatta, D., Lucas, J., Hodgkiss, W., "Digital Signal Processing: A System Design Approach", Wiley, 1988

[21] N. F. Fletcher and T. D. Rossing, The Physics of Musical Instruments, 2nd ed. New York: Springer-Verlag, 1998.

[22] J. Woodruff, B. Pardo and R. Dannenberg, "Remixing Stereo Music with Score-informed Source Separation," in Proceedings of the 7th International Conference on Music Information Retrieval, Victoria, Canada, October 8-12, 2006

[23] Virtanen, T., Klapuri, A., Analysis of polyphonic audio using source-filter model and non-negative matrix factorization, in Advances in Models for Acoustic Processing, Neural Information Processing Systems Workshop, 2006 (extended abstract).

[24] V. Valimaki, J. Pakarinen, C. Erkut, and M. Karjalainen. Discrete-time modelling of musical instruments. Reports on progress in physics, 69:1-78, 2006.

[25] M.R. Schroeder and B.S. Atal. Code-excited linear prediction (CELP): High-quality speech at very low bit rates. In IEEE ICASSP, pages 937-940, Tampa, Florida, 1985.

[26] Brown, J.C., (1991). “Calculation of a Constant Q Spectral Transform” J. Acoust. Soc. Am. 89 425-434.

[27] FitzGerald, D., Cranitch, M., and Coyle, E., "Shifted Non-negative Matrix Factorisation for Sound Source Separation", Proceedings of the IEEE conference on Statistics in Signal Processing, Bordeaux, France, July 2005.

[28] FitzGerald, D., Cranitch, M., and Coyle, E., "Non-negative Tensor Factorisation for Sound Source Separation", Proceedings of the Irish Signals and Systems Conference, Dublin, September 2005.

[29] FitzGerald, D., Cranitch, M., and Coyle, E., "Sound Source Separation using shifted Non-negative Tensor Factorisation", IEEE International Conference on Acoustics, Speech and Signal Processing, 2006 (ICASSP2006), Toulouse France.

[30] FitzGerald, D., Cranitch, M., and Coyle, E., "Shifted 2D Non-negative Tensor Factorisation", Proceedings of the Irish Signals and Systems Conference, Dublin, June 2006.

[31] Abdallah, S.A. 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 2004), Barcelona, Spain, October 10-14, 2004.

[32] Lee, D. and Seung, H., "Algorithms for non-negative matrix factorization", Adv. Neural Info. Proc. Syst. 13, 556-562 (2001).

[33] FitzGerald D (2004), “Automatic Drum Transcription and Source Separation”, Ph.D. Thesis, Dublin Institute of Technology, Dublin, Ireland.

[34] Casey M. and Westner A. (2000) "Separation of Mixed Audio Sources By Independent Subspace Analysis" in Proc. of ICMC 2000, pp. 154-161, Berlin, Germany.

[35] Virtanen T. (2003) "Sound Source Separation Using Sparse Coding with Temporal Continuity Objective", Proc. of International Computer Music Conference (ICMC2003), Singapore, 2003.

[36] Smaragdis, P., Brown, J.C., "Non-negative Matrix Factorization for Polyphonic Music Transcription", IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), pp. 177-180, October 2003

[37] Vincent, E and Rodet, X., "Music transcription with ISA and HMM". In Proc. ICA , 2004.

[38] Brett W. Bader and Tamara G. Kolda, Algorithm 862: MATLAB Tensor Classes for Fast Algorithm Prototyping, ACM Transactions on Mathematical Software, 32(4), December 2006.

[39] Tensor Toolbox for Matlab, available at http://csmr.ca.sandia.gov/ tgkolda/TensorToolbox/

[40] Peter Siedlaczek, Advanced Orchestra Library Set, 1997.

[41] Serra, X., “Musical Sound Modeling with Sinusoids plus Noise”, G.D. Poli, A. Piccialli, S.T. Pope and C. Roads, Eds, Musical Signal Processing, Swets & Zeitlinger Publishers, 1997.

[42] D. Barry and B. Lawlor and E. Coyle, “Sound Source Separation: Azimuth Discrimination and Resynthesis”, Proc.of 7th International Conference on Digital Audio Effects, (DAFX 04), Naples, Italy, 2004.

[43] O. 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, July 2004.

[44] E. Vincent, R. Gibonval and C. Fevotte. “Performance Measurement in Blind Audio Source Separation”, IEEE Transactions on Speech and Audio Processing, vol. 14 no. 4 pp 1462-1469, July, 2006.

[45] BSS_Eval toolbox available at http://bass-db.gforge.inria.fr/bss_eval/