GENERALISED PRIOR SUBSPACE ANALYSIS FOR POLYPHONIC PITCH TRANSCRIPTION

Derry FitzGerald, Matt Cranitch

Cork Institute of Technology
Rossa Avenue, Bishopstown, Cork, Ireland
derry.fitzgerald@cit.ie

Eugene Coyle

Dublin Institute of Technology
Kevin St., Dublin, Ireland
eugene.coyle@dit.ie

ABSTRACT

A reformulation of Prior Subspace Analysis (PSA) is presented, which restates the problem as that of fitting an undercomplete signal dictionary to a spectrogram. Further, a generalization of PSA is derived which allows the transcription of polyphonic pitched instruments. This involves the translation of a single frequency prior subspace of a note to approximate other notes, overcoming the problem of needing a separate basis function for each note played by an instrument. Examples are then demonstrated which show the utility of the generalised PSA algorithm for the purposes of polyphonic pitch transcription.

1. INTRODUCTION

Prior Subspace Analysis (PSA) was first proposed as a technique for transcription and sound source separation of drum sounds [1], and was found to be successful at tackling the transcription of certain types of drum sounds. However, the method was not suitable for the transcription of pitched instruments, as it required an individual prior subspace for each note of a pitched instrument. The remainder of this paper describes a reformulation and extension of the original PSA algorithm to allow the transcription of polyphonic music. Section 2 describes the reformulation of PSA in terms of fitting an undercomplete signal dictionary to a time-frequency representation of a signal, and Section 3 describes an extension to this model to allow polyphonic pitched instrument transcription. Section 4 shows preliminary results obtained using this generalised PSA algorithm.

2. PRIOR SUBSPACE ANALYSIS - A REFORMULATION

Given an input signal, PSA assumes that a magnitude spectrogram of the signal **Y** results from the superposition of l unknown spectrograms Y_j. Further, it is assumed that each of these spectrograms can be represented as the outer product of an invariant frequency basis function, and an invariant amplitude basis function, in the manner of Independent Subspace Analysis [2]. This yields:

$$\mathbf{Y} = \sum_{j=1}^{l} Y_j = \sum_{j=1}^{l} a_j s_j \qquad (1)$$

It is then assumed that there are known frequency basis functions or prior subspaces a_{pr} that are good approximations to the actual subspaces. Substituting for the a_j with these prior subspaces yields:

$$\mathbf{Y} \approx \sum_{j=1}^{l} a_{pr} s_j \qquad (2)$$

In matrix notation this becomes:

$$\mathbf{Y} \approx \mathbf{A_{pr}s} \qquad (3)$$

As originally formulated, PSA obtained estimates of **s** by multiplying the overall spectrogram by the pseudo-inverse of the frequency basis functions to obtain an intial estimate of **s**. Independent Component Analysis (ICA) [3] then performed to yield an improved estimate of **s**.

In recent years, it has been proposed that sound source separation can be achieved by means of sparse decomposition in a signal dictionary [4]. The signal dictionary used in this research consisted of a wavelet packet dictionary. More recently, it has been proposed that sound source separation in single channel signals can be carried out by fitting an overcomplete signal dictionary to the signal, in conjunction with knowledge of spectral cues such as the head related transfer function [5].

The above research suggests a different view of the prior subspaces used in PSA, namely that the prior subspaces are a signal dictionary, albeit a very undercomplete signal dictionary. The PSA problem can then be stated as follows: given a signal dictionary, $\mathbf{A_{pr}}$, and a spectrogram **Y**, find an estimate of **s** given some suitable criteria. In this case, a suitable criteria would be to assume that the data is sparse in nature. Using the pseudo-inverse is not suitable as it assumes the data is gaussian in nature. While following the use of the pseudo-inverse with ICA goes some way to solving this problem, the PSA problem is closer in formulation to Non-negative matrix factorisation (NMF) [6] and Non-negative Sparse Coding (NNSC) [7].

Both NMF and NNSC attempt to approximate a non-negative matrix **x** of size n x m, such as a spectrogram, by decomposing it into a mixing matrix **A** of size n x r, and a set of feature vectors **s**, of size r x m:

$$\mathbf{x} \approx \mathbf{As} = \sum_{a=1}^{r} A_r s_r \qquad (4)$$

where r is the number of basis functions chosen to represent the original data. Both methods assume that the input data is sparse in nature [7], and have been used for source separation and transcription of polyphonic audio [8,9,10]. However, both suffer from the problem of choosing a suitable r to give the best interpretation of the data, though this is less of a problem for NMF. Further, both suffer from permutation ambiguities. Both algorithms start by randomly initialising **A** and **s**, ensuring that the initialisations are non-negative. Both methods use a multiplicative update rule for **s**, while NNSC updates **A** via gradient descent, and NMF via a multiplicative update. Both **A** and **s** are then updated sequentially until convergence.

In the case of PSA, initial estimates of **A** are available, and so it is only necessary to update for **s**. This results in a new formulation for PSA, described below in pseudo-code:

```
1.  Obtain a magnitude spectrogram Y of the input signal.
2.  Randomly initialise s, where s contains the amplitude envelopes of the sources, ensuring that the data is non-negative.
3.  Update s using either of these update rules
        $\mathbf{s} = \mathbf{s} .* \left(\mathbf{A}_{\mathbf{pr}}^{\mathbf{T}} * \mathbf{Y}\right) ./ \left(\mathbf{A}_{\mathbf{pr}}^{\mathbf{T}} * \mathbf{A}_{\mathbf{pr}} * \mathbf{s} + \lambda\right)$ (NNSC)
        $\mathbf{s} = \mathbf{s} .* \left(\mathbf{A}_{\mathbf{pr}}^{\mathbf{T}} * \left(\mathbf{Y} ./ \left(\mathbf{A}_{\mathbf{pr}} * \mathbf{s}\right)\right)\right) ./ \mathbf{A}_{\mathbf{pr}}^{\mathbf{T}} * \mathbf{O}$ (NMF)
    where $\mathbf{A}_{\mathbf{pr}}$ are the prior subspaces, $^{\mathbf{T}}$ denotes matrix transpose, .* denotes elementwise multiplication, and ./ denotes elementwise division. $\lambda$ is a non-negative scalar , and O is an all-ones matrix the same size as Y.
4.  Iterate step 3. to convergence.
```

In tests the NMF-based update rule was found to give better performance than the NNSC-based rule, suggesting that the NMF-based rule provides a better fit to the underlying data. Also, in most cases 50 iterations was found to give sufficient convergence. Another useful update rule for **s** is that proposed by Abdallah [11]. It should be noted that the use of that update rule requires the use of a power spectrogram as opposed to a magnitude spectrogram, and that the prior subspaces have to be modified in accordance with this.

It is important to point out at this stage that the new PSA algorithm offers significant improvements over the original PSA algorithm. Firstly, the use of non-negativity means that the results obtained will be more consistent with real world situations, where negative amplitudes of sources cannot occur. This was a problem with the original PSA algorithm in that the amplitude envelopes obtained were sometimes physically implausible. Secondly, because we are only updating **s**, and the prior subspaces are held constant, there is no longer any permutation and scaling ambiguities in the algorithm. This means that the recovered sources are directly associated with the prior subspaces, and there is no longer any need to identify the sources after processing.

In the case of using the new PSA algorithm for drum transcription of snare, kick drum and hi-hats, this means that the assumptions used to identify the sources in the original PSA transcription algorithm are no longer required. These assumptions were that the kick drum had a lower spectral centroid than the snare, and that the hi-hats occurred more frequently than the snare drum. The elimination of these assumptions allows the new algorithm to function in a wider range of circumstances. When tested on the same data set as the original PSA algorithm, the performance improved to a 94.7% success rate, as opposed to the 92.5% success rate achieved with the original algorithm. Table 1, below, shows the results obtained using the original PSA algorithm, while Table 2 shows the results obtained using the reformulated PSA algorithm. It can be seen that the performance in transcription of snares and hi-hats has improved, while there has been a small degradation in the recovery of the kick drum. Nevertheless, it can be seen that the new algorithm has outperformed the original PSA algorithm.

It should be noted that a similar reformulation of the PSA algorithm described above was arrived at independently by Paulus et al [12]. However, for the purposes of pitched instrument transcription, both reformulations suffer from the need for an individual prior subspace for each note present. Methods to overcome this problem are presented in Section 3.

Type	Total	Missing	Incorrect	%
Snare	21	0	2	90.5
Kick	33	0	0	100
Hats	79	2	6	89.9
Overall	133	2	8	92.5

Table 1: Drum Transcription Results using the original PSA algorithm

Type	Total	Missing	Incorrect	%
Snare	21	0	0	100
Kick	33	1	1	93.9
Hats	79	0	5	93.7
Overall	133	1	6	94.7

Table 2: Drum Transcription Results using the reformulated PSA algorithm.

3. GENERALISED PRIOR SUBSPACE ANALYSIS

It can be seen from the above that an extended model is needed to reflect the situation where various notes from the same instrument occur over the course of a spectrogram. Previous work attempting to deal with this includes the non-linear Independent Subspace Analysis model proposed by Vincent et al [13]. In this model, chord spectra are represented as sums of note power spectra, and note spectra are represented as sums of instrument dependant log-power spectra. Note durations are then modeled using Hidden Markov Models. Time-Frequency analysis was carried out using a log-frequency scale and successful transcription was obtained for two duo recordings.

A potential way of overcoming the problem of dealing with multiple notes belonging to a single source is to assume that the notes belonging to a single source consist of translated versions of a single frequency basis function. This single frequency basis function is then taken to represent the typical frequency spectrum of any note played on the instrument in question. This is a simplified approximation of the real situation, where the frequency spectrum of the note does vary with pitch. Despite this, the assumption does represent a valid approximation over a limited pitch range. A version of this assumption is used in commercial music samplers and synthesisers, where a recorded note of a given pitch is used to generate other notes in proximity to the original note. It should be noted that the use of this assumption also places a further restriction on the type of spectrogram being analysed, namely that the frequency resolution of the spectrogram must be logarithmic in scale.

Figure 1, below, shows the frequency spectra of two different notes played on a French horn. It can be clearly seen that the spectra of the two notes are very similar, and so the spectrum of either note can be approximated by a translation of the other note. It is also assumed that no significant information is contained in the extremes of the translated frequency basis function. It can be seen that this assumption holds for the spectra shown in Figure 1. This assumption also sets limits on how far a given basis function can be translated. For example, translating the first of the two spectra shown by more than 20 bins to the left will result in part of the first partial to be moved to the end of the frequency spectrum, where it is clearly not supposed to occur.

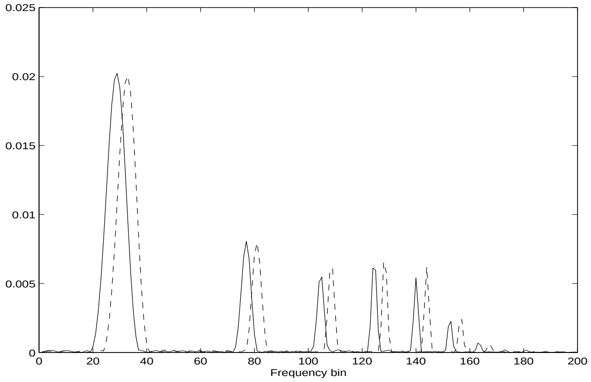

Figure 1. *Spectra of two notes of a French horn*

To translate a given *n* x 1 vector, an *n* x *n* translation matrix can be used. Such a translation matrix can be generated by rearranging the columns of the identity matrix. For example, to achieve a shift up of one, the translation matrix would be obtained from **I**(:,[n, 1:n-1]) where **I** denotes the identity matrix, and where the ordering of the columns is contained in the square brackets. A simple example of a shift up of one, which is obtained by a shift to the left of the ones in the identity matrix is given below for a 5 x 1 matrix.

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 1 \end{pmatrix} \qquad (5)$$

For the transcription of a single instrument playing multiple notes, the signal model now becomes:

$$\mathbf{Y} = \sum_{j=1}^{r} \mathbf{T_j} A s_j \qquad (6)$$

where **Y** is a log-frequency spectrogram of size *n* x *m*, *A* is an *n* x 1 vector containing a typical harmonic profile and s_j contains the amplitude basis function, of size 1 x *m* associated with translation matrix $\mathbf{T_j}$ of size *n* x *n*. An algorithm which attempts to learn both *A* and s_j from an input spectrogram is described in [14].

The utility of this signal model can be seen in that a single basis function can now be used to model a pitched instrument and can be seen as a means of generalising the PSA model to deal with pitched instruments, as a single prior subspace of an instrument note can be used to generate other notes from the instrument. Suitable prior subspaces for a given instrument can then be obtained via a number of methods. NMF can be performed on a single note of an instrument to obtain a harmonic profile, or a single frame of a log-frequency spectrogram with a well established harmonic profile can be chosen as a basis function.

For a predefined set of translations and a given instrument prior subspace, the signal model can be rewritten as:

$$\mathbf{Y} = \sum_{j=1}^{r} A_{Tj} s_j = \mathbf{A_T s} \qquad (7)$$

where $A_{Tj} = \mathbf{T_j}A$, and $\mathbf{A_T}$ is a matrix of size *n* x *r*, and **s** is a matrix of size *r* x *m*. In other words, the signal model can be collapsed to the standard PSA model. Therefore, the same optimisation techniques can be used to obtain **s** as in the reformulated PSA model.

The resulting generalised PSA algorithm for the transcription of a pitched instrument can then be summarised as follows:

1. Obtain a spectrogram with log-frequency resolution of the input signal.
2. Determine transformation matrices $\mathbf{T_j}$ for a given range of translations.
3. Obtain translated versions A_{Tj} of prior subspace *A* from $A_{Tj} = \mathbf{T_j}A$.
4. Randomly initalise **s**, ensuring non-negativity.
5. Update **s** using the update rule:
   $$\mathbf{s} = \mathbf{s}.*\left(\mathbf{A}_\mathbf{T}^\mathbf{T} * \left(\mathbf{Y}./\left(\mathbf{A_T} * \mathbf{s}\right)\right)\right)./\,\mathbf{A}_\mathbf{T}^\mathbf{T} * \mathbf{O}$$
   where $^\mathbf{T}$ denotes matrix transpose, .* denotes elementwise multiplication, and ./ denotes elementwise division, and **O** is an all-ones matrix the same size as **Y**
6. Iterate step 5 to convergence.

The NNSC-based update rule was found to be unsuited for the purposes of pitched instrument transcription and so is not included in the algorithm. Preliminary results obtained using the algorithm are detailed in Section 4.

4. TRANSCRIPTION USING GENERALISED PRIOR SUBSPACE ANALYSIS

To test the effectiveness of the generalised PSA algorithm, a number of simple tests were carried out. Firstly, a recording was made of a sampled piano playing a C major scale from note C5 to note C6. This sampled piano made use of 4 separate piano note samples per octave, A prior subspace was obtained for piano note G5 from a completely different sampled piano. The results obtained for **s** are shown in Figure 2 below, in which a shift of one corresponds to a pitch change of a semitone. The range of translations was set to +/- 10, though a greater number could have been used without affecting the result.

It can be seen that the algorithm has successfully captured the notes in the input waveform. As the prior subspace was a G5 piano note, this note has a shift of 0 in the above plot, and it can be seen that the notes played do indeed follow the pattern of a major scale. As the translated priors are fixed, the algorithm does not suffer from the source ordering problem inherent in blind source separation algorithms, and so the recovered amplitude basis functions will be presented in the correct order. This means that the basis functions can be plotted in a manner similar to a piano roll, as shown below in Figure 2.

The second test carried out on the algorithm involved the transcription of a series of three-note piano chords. The same sampled piano was used as in the previous example, and the same prior, a piano-note of pitch G5 was used. A pianoroll plot of the midifile used to generate the audio signal is shown in Figure 3. This was generated using the Miditoolbox [15]. Figure 4 then shows the output of the generalised PSA algorithm. Remembering that a translation of 0 corresponds to note G5, it can be seen that the algorithm has successfully transcribed the piano chords, with the notes clearly distinguishable from any noise in the basis functions. This shows that the algorithm is capable of transcribing polyphonic music.

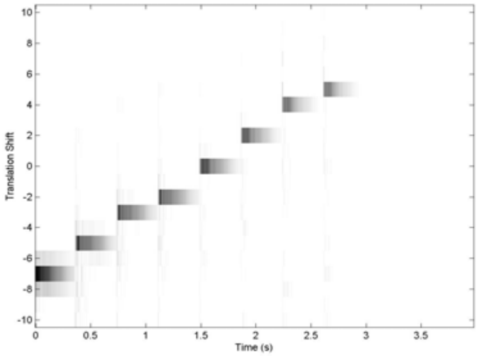

Figure 2. *Outputof algorithm from a piano scale*

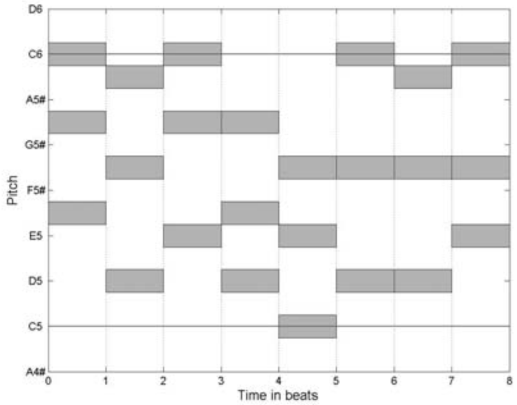

Figure 3. *Pianoroll of 3 note piano chords*

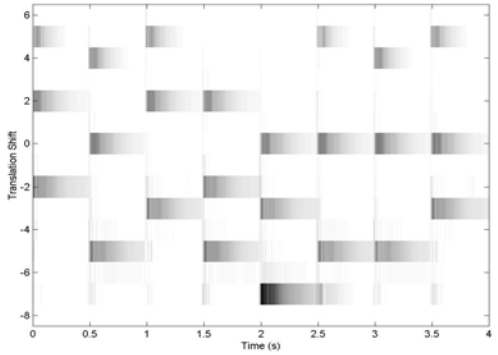

Figure 4. *Output of algorithm fromaudio signal of piano*

The third test performed on the algorithm was an audio signal which contained a trumpet and a french horn playing separate melody lines. Both were created from an orchestral samples library, with 4 samples per octave. The translation range was set to +/- 25, and the pitch range of the actual signal was from D5 to E4. The piano prior from the previous examples was used in an attempt to see if a single harmonic prior was capable of transcribing more than a single instrument. The midi-file used to create the audio signal is shown in Figure 5. In this figure the trumpet, is represented by black, and the french horn, on channel 2, is represented by grey.

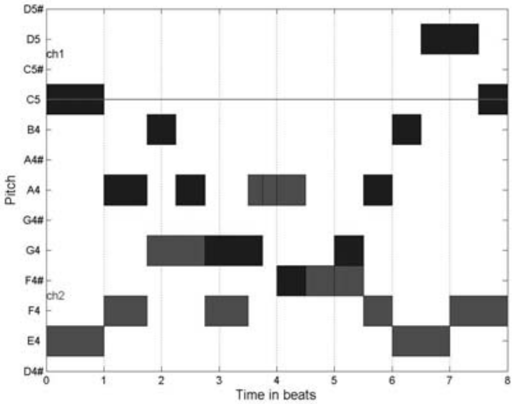

Figure 5. *Pianoroll of trumpet and French horn.*

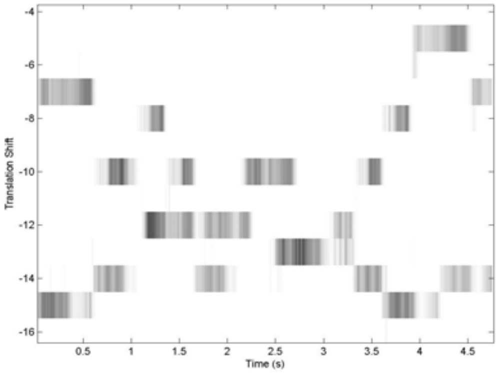

Figure 6. *Output of algorithm from trumpet and french horn example*

Figure 6 then shows the output of the generalised PSA algorithm. It can be seen that the notes have been successfully recovered by the algorithm. This suggests that a generalised harmonic profile has the potential to be able to successfully transcribe a wide range of instruments. For transcription purposes, this can be considered a strength, in that it would not be necessary to have a different prior for each instrument in order for successful transcription to occur. However, for the purposes of sound source separation, this would not be useful.

Finally, the algorithm was applied to a recording of a grand piano, recorded in a small theater in reverberant conditions to see how the algorithm would deal with real-world audio. Further, to see if a synthetic harmonic prior could be used for transcription, a harmonic prior was generated from a sum of 7 sinusoids, with each partial having half the amplitude of the previous partial. The prior used corresponded to the pitch of middle C (261.63 Hz). The output of the generalised PSA algorithm is shown in Figure 7, below.

When compared by hand to the recording, the algorithm was found to successfully recover the vast majority (93%) of the notes played, with the exception of a small number of low amplitude notes. This was after thresholding and elimination of short duration activations of **s.** It is possible that some form of perceptual weighting as described in [16] on the input spectrogram may be of use in recovering these notes. The success of this test shows that the algorithm can function on real-world signals, and that a synthetic prior can be used to attempt transcription of a real instrument. It also demonstrates that a single prior can function over a wide pitch range, in this case dealing successfully with a range of two octaves.

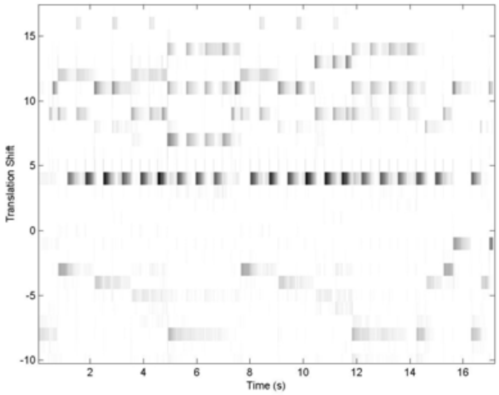

Figure 7. *Output of algorithm from grand piano recording*

5. CONCLUSIONS

A reformulation of PSA as a signal dictionary fitting problem has been presented. Following on from this, a generalisation of the PSA algorithm was derived which uses translations of a single freuquency basis functions to represent different notes. The effectiveness of the generalised PSA algorithm as a method for polyphonic music transcription was then demonstated using both midi-generated and real-world signals. It also demonstrates that a well chosen undercomplete signal dictionary, in this case a single dictionary element and translations thereof, can be used to extract much meaningful information from audio signals. Future work will concentrate on identifying accurately the performance of the algorithm for transcription of polyphonic music in a wide range of situations, and on attempting to improve performance through the use of perceptual weighting. It is also intendend to attempt to transcribe both pitched instruments and percussive instruments simultaneously by appending a set of percussion priors to the set of translated pitch priors.

6. ACKNOWLEDGEMENTS

This research was supported by funding from the Irish Research Council for Science, Engineering and Technology.

7. REFERENCES

[1] D. FitzGerald, “Automatic Drum Transcription and Source Separation”, PhD. Thesis, Dublin Institute of Technology, 2004

[2] M.A. Casey and A. Westner, “Separation of Mixed Audio Sources By Independent Subspace Analysis” in Proc. Of ICMC 2000, pp. 154-161, Berlin, Germany.

[3] P. Comon, “Independent component analysis - a new concept?”, Signal Processing, 36(3): 287-314, April 1994

[4] M. Zibulevsky and B. A. Pearlmutter , “Blind Source Separation by Sparse Decomposition in a Signal Dictionary”, Neural Computation, 13(4):863-882. 2001.

[5] B. A. Pearlmutter and A. M. Zador. “Monaural Source Separation Using Spectral Cues.” In Proceedings of ICA2004, September 22-24, 2004, Granada, Spain, pages 478-485.

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

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

[8] H. Asari,. “Non-negative Matrix Factorization: A possible way to learn sound dictionaries”, http://zadorlab.cshl.edu/asari/nmf.html 2004.

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

[10] P. Smaragdis, J.C. Brown, "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

[11] S. A. Abdallah and M. D. Plumbley, “Polyphonic transcription by non-negative sparse coding of power spectra.”, Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004), Barcelona, Spain, October 10-14, 2004.

[12] J. Paulus and T. Virtanen, “Drum transcription with non-negative spectrogram factorization”, submitted to the submitted to European Signal Processing Conference 2005.

[13] E. Vincent and X. Rodet, “Music transcription with ISA and HMM.” In Proceedings of ICA , 2004. September 22-24, 2004, Granada, Spain, pages 478-485.

[14] D. FitzGerald, M. Cranitch, E. Coyle, “Shifted Non-negative Matrix Factorisation for Sound Source Separation”, Statistics in Signal Processing conference, Bordeaux, France, July 2005.

[15] T. Eerola, and P. Toiviainen, “MIR in Matlab: The Midi Toolbox.” In Proceedings of 5th International Conference on Music Information Retrieval (ISMIR 2004), pp. 22-27, Barcelona, 2004.

[16] T. Virtanen, “Separation of sound sources by convolutive sparse coding,” in Proc. of ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio Processing, 2004.