Multimed Tools Appl (2014) 72:925–949
DOI 10.1007/s11042-013-1398-8

Monophonic constrained non-negative sparse coding using instrument models for audio separation and transcription of monophonic source-based polyphonic mixtures

Francisco José Rodríguez-Serrano · Julio José Carabias-Orti · Pedro Vera-Candeas · Francisco Jesús Canadas-Quesada · Nicolás Ruiz-Reyes

Published online: 8 March 2013

Abstract In this paper we propose a monophonic constrained signal decomposition model applied to polyphonic signals composed of several monophonic sources from different musical instruments. The harmonic constraint is particularly effective for tonal instruments because each note is associated with a unique basis. The monophonic constraint is implemented by enforcing single-non-zero gains per instrument in the factorization process. The proposed method uses previously trained instrument models with a supervised procedure. Both constraints (harmonic and monophonic) are implemented in a deterministic manner. The proposed method has been tested for two audio signal applications, Sound Source Separation and Automatic Music Transcription. Comparison with other state-of-the-art methods using a dataset of polyphonic mixtures composed of monophonic sources has produced competitive and promising results.

Keywords Non-negative sparse coding (NNSC) · Sparse representations · Non-negative matrix factorization (NMF) · Spectral analysis · Harmonicity · Sparsity · Monophony · Music transcription · Source separation

1 Introduction

Sound Source Separation (SSS) and Automatic Music Transcription (AMT) are two different signal processing tasks but share certain processes in common. In fact, some authors claim that AMT is a prerequisite for music SSS [14], while others think that music SSS is prerequisite for AMT [15].

F. J. Rodríguez-Serrano (✉) · J. J. Carabias-Orti · P. Vera-Candeas · F. J. Canadas-Quesada · N. Ruiz-Reyes
Telecommunication Engineering Department, University of Jaen, Alfonso X El Sabio, 28, 23700 Linares, Jaen, Spain
e-mail: fjrodrig@ujaen.es

On the one hand, SSS can be applied to many real-world audio signals that are composed of mixtures of several sound sources. SSS is the process by which individual sources are decomposed from the signal mixture. Depending on the number of sources and sensors used in the experiments, SSS can be classified into three cases. *Overdetermined* cases are those which the number of sensors is higher than the number of sources [22, 34, 36, 44]. For *determined* cases, the number of sources and the number of sensors are the same. Finally, *underdetermined* cases are those for which the number of sensors is lower than the number of sources. In this paper, we discuss SSS for a single sensor (channel in our case) [27, 35], which is the most critical case within the underdetermined class.

On the other hand, AMT is the process of generating a score (i.e. a symbolic representation of played notes) from a piece of audio. Music transcription is a very complicated task for polyphonic signals, because the signals from individual notes overlap in time and frequency. A common type of music transcription is the pitched transcription [23], where the onset times, offset times, and pitches of each note are estimated from a recording. However, current transcription systems do not provide individual transcriptions for each instrument that contributes to the mixture.

In this work, we present a method that may be applied to both AMT and SSS at the same time. The method has been designed for the particular case of monaural polyphonic signals composed of several monophonic and harmonic sources.

The proposed method may be classified as a signal decomposition method. In fact, similar methods have been intensively used for audio applications such as SSS and AMT, with reliable results [20, 34, 43]. These methods try to decompose the audio spectrogram into a linear combination of spectral basis functions. The short-term magnitude (or power) spectrum of the signal $x(f, t)$ in the frame t and frequency f is modelled as a weighted sum of basis functions as

$$\hat{x}(f, t) = \sum_{n=1}^{N} g_n(t) b_n(f) \tag{1}$$

where $g_n(t)$ is the gain of the basis function n at frame t, and $b_n(f)$, $n = 1, ..., N$ are the bases. When dealing with harmonic sounds in the context of automatic music transcription, each basis function should ideally represent a single pitch, so that the corresponding gains contain information about the onset and offset times of notes having that pitch.

The process of learning basis functions can be *Supervised* or *Unsupervised* depending on whether prior information about the musical composition (such as instruments actually being played) is used or not. In the supervised case, the basis can be fixed or adapted to the actual music scene of the analysed signal. In this work, we use a supervised learning process with fixed basis that have been shown [6] to provide a good generalization of the model parameters.

There are several methods in the literature for performing signal decomposition such as Atomic Decomposition [19], Independent Component Analysis (ICA) [32], Non-Negative Matrix Factorization [24], and Sparse Coding [1].

Sparsity constraints can be applied to the signal decomposition process. Sparse representations have received increased attention for audio applications such as polyphonic audio transcription [1, 2] and audio source separation [29, 40]. Sparse coding attempts to produce a sparse spectral decomposition in regions where the probability densities of the gains are centred around zero and have long tails [21],

such that most of the energy is grouped by only a few basis with non-zero gains. This assumption fits well with the concept that only a relatively small fraction of the available notes in music are sounded at each frame. For power or magnitude spectrograms NMF and sparse coding leading can be combined into a non-negative sparse coding (NNSC) [2, 21] method for signal decomposition.

The method proposed here (using monophonic constraints for each instrument), enforces sparseness such that only one gain is active at each frame. This extreme sparseness constraint has been previously used in other signal decomposition methods in the literature. For example, within a statistical framework, this kind of restriction is introduced into Gaussian Scaled Mixture Model (GSMM) [3] or Factorial Scaled Hidden Markov Model (FS-HMM) [30] under Gaussianity and Itakura Saito (IS) divergence assumptions.

In this paper, we propose a deterministic factorization method that is used to process monoaural signals from polyphonic mixtures of several monophonic instruments. Several works have utilised these kinds of signals (such as GSMM [3] and FS-HMM [30]), but within a probabilistic framework. The method proposed here is novel because single-pitch and harmonic constraints are enforced deterministically. Each instrument contributing to the signal is explicitly assumed to be monophonic, i.e., there is only one possible state (note) per instrument at each frame. These kinds of signal are very typical for some wind and rubbed string instruments (in some cases). Some instrumental chorals have been composed for such kind of instruments, (e.g. the Bach chorals used in this work as a test database). The source separation and the transcription identifying the gains are computed for each instrument being played, the computation can be run at real time in some cases. In AMT, individual transcription for each instrument in the mixture is estimated. To the best of our knowledge, no other work in the literature evaluates AMT by obtaining the transcription per instrument in polyphonic mixtures. In this work, the instrument models are learned in a training stage and held fixed during the testing stage, as proposed in [6]. The proposed methods are tested for SSS and AMT and compared to other state-of-the-art methods with promising results.

The paper is structured as follows: Section 2 reviews the harmonic and sparsity constrained signal models from previous studies, as well as theoretical background on NMF and instrument modelling; Section 3 explains the proposed method for constraining a polyphonic signal model to have a single non-zero gain per instrument at each frame and provides the algorithm for signal spectral decomposition; the proposed approach is applied in Section 4 for SSS and AMT using polyphonic mixtures composed of several monophonic single-instrument sources, the results are compared with those obtained by other state-of-the-art methods; finally, we draw some conclusions and discuss future work in Section 5.

2 Theoretical background

2.1 Basic Harmonic Constrained (BHC) model

Musical notes (excluding transients) played on tonal instruments are pseudo-periodic, with a spectra of regularly spaced frequency peaks [6]. In fact, models are commonly constrained to be harmonic [4, 6, 33, 39]. The harmonic constraint

improves the modelling because each basis function is associated, in advance, with a pitch n by means of its fundamental frequency $f_0(n)$. This constraint is introduced in the model presented in (1) as

$$b_{n,j}(f) = \sum_{m=1}^{M} a_{n,j}[m]G(f - mf_0(n)) \tag{2}$$

where $b_{n,j}(f)$ are the bases for each note n and instrument j, m is the selected harmonic, M is the number of harmonics, $a_{n,j}[m]$ is the amplitude of harmonic m for note n and instrument j, $G(f)$ is the magnitude spectrum of the window function, and the spectrum of a harmonic component at frequency $mf_0(n)$ is approximated by $G(f - mf_0(n))$.

The model for the magnitude spectra of a music signal is then obtained as (see (1))

$$\hat{x}(f,t) = \sum_{j=1}^{J} \sum_{n=1}^{N(j)} \sum_{m=1}^{M} g_{n,j}(t) a_{n,j}[m] G(f - mf_0(n)) \tag{3}$$

where J is the number of instruments and $N(j)$ is the total number of possible notes for the instrument j. Here the time gains $g_{n,j}(t)$ and the harmonic amplitudes $a_{n,j}[m]$ are the model parameters to be estimated. These parameters are ussually estimated by minimizing the reconstruction error between the observed spectrogram $x(f,t)$ and the modelled one $\hat{x}(f,t)$.

The most popular cost functions are the Euclidean (EUC) distance, the generalised Kullback–Leibner (KL) and the Itakura–Saito (IS) divergences. The β-divergence (see (4)) is another commonly used cost function that encompasses the three previously mentioned cost functions in its definition, i.e., EUC ($\beta = 2$), KL ($\beta = 1$) and IS ($\beta = 0$), and is defined as follows,

$$D_\beta(x|\hat{x}) = \begin{cases} \dfrac{1}{\beta(\beta-1)}\left(x^\beta + (\beta-1)\hat{x}^\beta - \beta x \hat{x}^{\beta-1}\right) & \beta \in (0,1) \cup (1,2] \\ x \log \dfrac{x}{\hat{x}} - x + \hat{x} & \beta = 1 \\ \dfrac{x}{\hat{x}} + \log \dfrac{x}{\hat{x}} - 1 & \beta = 0 \end{cases} \tag{4}$$

Several systems using the β-divergence cost function can be found in [12, 13, 39].

2.2 BHC with sparse constraint model

Sparsity is a natural restriction applied to gains that forces the signal model to have only a few non-zero gains $g_{n,j}(t)$ at each frame t. The assumption of sparsity conforms to the notion that only a relatively small fraction of the available musical notes are sounded at each frame [1]. Signal processing studies with constrained sparsity in signal models can be found in [1, 6, 16, 21, 40].

A typical way of introducing sparsity into signal models for minimizing a divergence is to use a regularization penalty term [16]. This penalty term discards the

solutions where most of the gains takes non-zero values. The global distortion can be formulated as:

$$D(x(f,t)|\hat{x}(f,t)) = D_\beta(x(f,t)|\hat{x}(f,t)) + \lambda \sum_{f,t} \phi(g_{n,j}(t)) \tag{5}$$

where D_β is the reconstruction distortion defined in (4), λ is a parameter controlling the importance of the regularised term, and ϕ is a function that penalises non-zero gains. Several definitions for the penalty term can be found in the literature. For example, Olshausen and Field [28] have suggested the functions $\phi(x) = -\exp(-x^2)$, $\phi(x) = \log(x^2 - 1)$ and $\phi(x) = |x|$, as possible penalty terms. For practical purposes we have used the third function in the experimental section, as it has been shown to be less sensitive to variations in the parameter λ [40] and provides an effective means of finding sparse solutions [5, 7].

2.3 Monophonic constrained models

For polyphonic signals composed of monophonic sources, the sparseness should be enforced such that only one gain per instrument is active at each frame. This extreme sparsity constraint has been previously used in other probabilistic signal decomposition methods. For example, Benaroya et al. [3] proposed a method for SSS in which each source STFT is modelled by a Gaussian Mixture Model (GMM); the GMM is modulated by a frame-dependent amplitude parameter accounting for nonstationarity, resulting in the Gaussian Scaled Mixture Model (GSMM) where the source is implicity assumed to be monophonic with many possible states. Ozerov et al. [30] proposed a method called the Factorial Scaled Hidden Markov Model (FS-HMM) that generalised GSMM and NMF using the Itakura Saito divergence (IS-NMF) and incorporates temporal continuity through Markov Modeling.

2.4 Augmented NMF for parameter estimation

Constraining parameters to be non-negative has been efficient in learning the spectrogram factorization models [41]. In fact, this constraint has been widely used in music transcription [4, 6, 39] and source separation [31, 41].

When the parameters are restricted to be non-negative, as in the case of magnitude spectra, a common way to compute the factorization is to minimize the reconstruction error between the observed spectrogram $x(f,t)$ and the modelled one $\hat{x}(f,t)$.

To obtain the model parameters that minimize the cost function, Lee et al. [25] proposes an iterative algorithm based on multiplicative update rules. Under these rules, $D_\beta(x(f,t)|\hat{x}(f,t))$ is shown to be non-increasing at each iteration while ensuring non-negativity of the bases and the gains. These multiplicative update rules are obtained by applying diagonal rescaling to the step size of the gradient descent algorithm (see [25] for further details). The multiplicative update rule for each scalar parameter θ_l is given by expressing the partial derivatives of the cost function $\nabla_{\theta_l} D_\beta$ as the quotient of two positive terms $\nabla^-_{\theta_l} D_\beta$ and $\nabla^+_{\theta_l} D_\beta$:

$$\theta_l \leftarrow \theta_l \frac{\nabla^-_{\theta_l} D_\beta(x(f,t)|\hat{x}(f,t))}{\nabla^+_{\theta_l} D_\beta(x(f,t)|\hat{x}(f,t))} \tag{6}$$

The main advantage of the multiplicative update rule in (6) is that non-negativity of the bases and the gains is ensured, resulting in an augmented non-negative matrix factorization (NMF) algorithm. For the harmonic-constrained model of (3), multiplicative updates that minimize the β-divergence for the amplitudes of the model are computed by [12],

$$a_{n,j}[m] \leftarrow a_{n,j}[m] \frac{\sum_{f,t} x(f,t)\hat{x}(f,t)^{\beta-2} g_{n,j}(t) G(f - mf_0(n))}{\sum_{f,t} \hat{x}(f,t)^{\beta-1} g_{n,j}(t) G(f - mf_0(n))} \tag{7}$$

Furthermore, when using the regularised penalty term of (5) with $\phi(x) = |x|$, the gains are estimated with the following multiplicative updates [16],

$$g_{n,j}(t) \leftarrow g_{n,j}(t) \frac{\sum_{f,m} x(f,t)\hat{x}(f,t)^{\beta-2} a_{n,j}[m] G(f - mf_0(n))}{\lambda + \sum_{f,m} \hat{x}(f,t)^{\beta-1} a_{n,j}[m] G(f - mf_0(n))} \tag{8}$$

where λ is the regularization term. The sparsity constraint is not imposed for $\lambda = 0$.

2.5 Instrument modeling

All the revised models of this section require that the basis functions $b_{n,j}(f)$ to be estimated for each note n and instrument j. As given in (2), the basis functions can be derived from the peak amplitudes $a_{n,j}[m]$, m being the considered partial when using the harmonic restriction. The amplitudes $a_{n,j}[m]$ are estimated in advance by using the RWC database [17, 18] as a training database of solo instruments (more details on the training database can be found in the experimental setup section). Let $R_{n,j}(t)$ denote a binary time/frequency matrix that represents the ground-truth transcription of the training data. The time dimension t represents frames and the frequency dimension represents the MIDI scale. As $R_{n,j}(t)$ is known in advance for the training database, gains in the training stage are initialised such that only the gain value associated with the active pitch n at frame t and played by instrument j is set to unity, whereas the rest of the gains are set to zero. Gains initialised to zero remain at zero, and therefore the frame is represented with the correct pitch. With this initialization, the application of sparse constraints is not necessary at the training stage. The training procedure is summarised in Algorithm 1.

Algorithm 1 Training algorithm description

1. Compute $x(t, f)$ from a solo performance for each instrument in the training database
2. Initialise gains $g_{n,j}(t)$ with the ground truth transcription $R_{n,j}(t)$ and amplitudes $a_{n,j}[m]$ with random positive values.
3. Update amplitudes $a_{n,j}[m]$ using (7).
4. Update gains $g_{n,j}(t)$ using (8) with $\lambda = 0$.
5. Repeat steps 2–3 until the algorithm converges (or the maximum number of iterations is reached).
6. Compute basis functions $b_{n,j}(f)$ for each instrument j using (2).

The training algorithm computes the basis functions $b_{n,j}(f)$ required at the factorization stage for each instrument. The instrument-dependent basis functions $b_{n,j}(f)$

are known and held fixed, and therefore, the factorization of new signals of the same instrument can be reduced to estimate the gains $g_{n,j}(t)$. The training procedure summarised in Algorithm 1 is suitable for all revised spectral decomposition models.

3 Proposed factorization method

3.1 Monophonic Basic Harmonic Constrained Model for Monophonic Signals (MBHC-MS)

First, we introduce the monophonic restriction for the simpler case of monophonic signals (the j index is removed from the equations). As stated above, the gains can be computed once the instrument's models have been estimated. The magnitude spectrogram can be reconstructed with (9), using the fixed basis functions derived from the training stage. The basis functions $b_{n_{\text{opt}}}(f)$ and the gain $g_{n_{\text{opt}},t}$ are chosen to minimise the β-divergence function at frame t, under the assumption that only one gain is non-zero at each frame. Thus, the signal model with the monophonic constraint (which is implemented deterministically) is defined for monophonic signals as follows.

$$\hat{x}_{n,t}(f) = g_{n_{\text{opt}},t} b_{n_{\text{opt}}}(f) \tag{9}$$

where $\hat{x}_{n,t}(f)$ is the modelled signal for the optimum note n_{opt} at frame t.

$$n_{\text{opt}}(t) = \arg \min_{n=1,\ldots,N} D_\beta \left(x_t(f) | g_{n,t} b_n(f)\right) \tag{10}$$

3.1.1 Gain estimation using sparse coding for monophonic signals

The MBHC-MS model of (10) allows the gains to be directly computed from the input data $x(f,t)$ and the amplitudes $a_n[m]$ without the need of an iterative NMF algorithm for monophonic signals. In this method, the optimum non-zero gain at each frame $g_{n_{\text{opt}},t}$ is the gain that minimises the cost function. The gain is estimated using an exhaustive search, without any iterative algorithm, over the set of distortion values generated for each note at each frame. In practical terms, the note that achieves the minimum distortion is the optimum note at each frame.

For β-divergence, the cost function for note n and frame t can be formulated as

$$\begin{aligned} &D_\beta(x_t(f)|g_{n,t} b_n(f)) \\ &= \sum_f \frac{1}{\beta(\beta-1)} \left(x_t(f)^\beta + (\beta-1)(g_{n,t} b_n(f))^\beta - \beta x_t(f)(g_{n,t} b_n(f))^{\beta-1}\right) \end{aligned} \tag{11}$$

The value of the gain for note n and frame t is then computed by minimizing (11). Conveniently, this minimization has a unique non-zero solution due to the scalar nature of the gain for note n and frame t.

$$g_{n,t} = \frac{\sum_f x_t(f) b_n(f)^{(\beta-1)}}{\sum_f b_n(f)^\beta} \tag{12}$$

Finally, the note that minimises the β-divergence at each frame is selected as the optimum note.

$$n_{\text{opt}}(t) = \arg \min_{n=1,\ldots,N} D_\beta \left(x_t(f) | \frac{\sum_f x_t(f) b_n(f)^{(\beta-1)}}{\sum_f b_n(f)^\beta} b_n(f) \right) \tag{13}$$

The proposed solution is valid for $\beta \in [0, 2]$ and for monophonic signals.

Equation (13) describes the selection of the optimum note at frame t for the MBHC-MS model. It represents the note that minimizes the distortion between the original signal and the reconstruction with the estimated gains and the selected basis for each note.

In summary, a novel method is presented that enforces single-pitch and harmonic constraints in a deterministic manner, performs the NNSC-based decomposition with β-divergence [13], and uses instrument specific information that is learned in a supervised way (i.e. using a training stage).

3.2 Monophonic Basic Harmonic Constrained Model for Polyphonic Mixtures (MBHC-PM)

Polyphonic signals occur when mixtures of multiple monophonic instruments are played at the same time. Polyphonic signals are very common in Western music, especially with wind instruments. The monophonic constraint can be extended to model polyphonic signals. The signal model is now defined as

$$\hat{x}(f, t) = \sum_{j=1}^{J} g_{n_j(t), j} b_{n, j}(f) \tag{14}$$

where $j = 1, \ldots, J$ is the instrument index and $n_j(t)$ is the note played by instrument j at time t. The signal model now includes different basis functions $b_{n,j}(f)$ for each instrument. It must be stressed that such a model is monophonic constrained because only one note $n_j(t)$ can be active at each frame t for each instrument j.

Equation (14) describes the signal decomposition model for the MBHC-PM model. Here, in contrast with (9) (where only one note was present at the signal), there are more than one note played at the same time (one note per instrument). Then, the signal is composed by the sum of the J instrument notes contributions. Each contribution can be described as the multiplication of the estimated gain for the selected note and the corresponding basis.

As in MBHC-MS method, the basis functions $b_{n,j}(f)$ for each instrument j are learned in advance and then held fixed. Each basis function models the spectrum of unique note for a given instrument (see (2)).

In this method, information about the instruments being played is required to select the appropriate basis functions. The audio applications then only have to estimate the gains $g_{n_j(t), j}$ for the different instruments at each frame.

In the monophonic constrained model for polyphonic mixtures, the distortion at frame t using β-divergence can be expressed as

$$D_\beta(x_t(f)|\sum_{j=1}^{J} g_{n_j(t),j} b_{n,j}(f)) = \sum_f \frac{1}{\beta(\beta-1)} \tag{15}$$

$$\cdot \left(x_t(f)^\beta + (\beta-1)\left(\sum_{j=1}^{J} g_{n_j(t),j} b_{n,j}(f)\right)^\beta \right.$$

$$\left. - \beta x_t(f)\left(\sum_{j=1}^{J} g_{n_j(t),j} b_{n,j}(f)\right)^{\beta-1}\right) \tag{16}$$

Equation (15) represents the same as (11) for the MBHC-MS model. This is the distortion caused by the reconstructed signal with the selected note for each instrument. In the case of MBHC-MS (only one note is active at each frame) it has a unique non-zero solution (12). However in the case of MBHC-PM (more than one note is active at each frame, one per instrument), the solution can be reached by two methods one by NMF (Section 3.2.1) and other with sparse coding (Section 3.2.2).

The optimum note for each instrument j at frame t is computed as the combination of notes for all the instruments that minimises the distortion at frame t. Once the gains $g_{n_j(t),j}$ are obtained, each distortion is computed and the optimum combination of notes (one per instrument) is selected.

3.2.1 Gain estimation using NMF for polyphonic mixtures of monophonic sources

The monophonic constraint for polyphonic mixtures of monophonic sources is enforced, within a deterministic framework by requiring the gains $g_{n_j(t),j}$ to be single-non-zero at each frame and instrument. Thus only J notes (one per instrument) can be active at a given frame. The J active notes (a maximum of one per instrument) are those that minimises the distortion between the original signal spectrogram and the estimated one. This optimum combination of notes is searched for over the dynamic range of notes for each instrument. The combinatorial search space is represented as follows,

$$\Psi = \{M_k, 1 \leqslant k \leqslant S\} \tag{17}$$

where M_k is the k-th combination composed of a single note candidate for each instrument and S is the total number of possible combinations. Each combination M_k can be formulated as

$$M_k = \left\{n_j^k,\ j = 1, ..., J\right\} \tag{18}$$

where n_j^k is the note played by instrument j at the k-th combination from Ψ.

For polyphonic signals, the gains can not be computed directly as in the MBHC-MS method. The gains must now be estimated using a gradient-based algorithm. This procedure is based on the minimization of the distortion between the estimated spectrogram and the target one using augmented NMF with multiplicative update (MU) rules as described in (6), following [25]. Here, the distortion to be minimised is shown in (15) and should be computed for each combination M_k from Ψ. In practice, the minimization is performed by computing the partial derivative of the distortion for note $n_i^k(t)$ and instrument i of gain $g_{n_i^k(t),i}$ can be formulated as

$$\frac{dD_\beta}{dg_{n_i^k(t),i}} = \sum_f \left(\sum_{j=1}^{J} g_{n_j^k(t),j} b_{n_j^k(t),j}(f) \right)^{\beta-1} b_{n_i^k(t),i}(f) \tag{19}$$

$$- \sum_f x_t(f) \left(\sum_{j=1}^{J} g_{n_j^k(t),j} b_{n_j^k(t),j}(f) \right)^{\beta-2} b_{n_i^k(t),i}(f) \tag{20}$$

where $n_i^k(t)$ and i indicate the selected note and instrument respectively, that must be minimised for the combination M_k. Thus, the MU rule for each gain $g_{n_i^k(t),i}$ can be formulated, as

$$g_{n_i^k(t),i} \leftarrow g_{n_i^k(t),i} \frac{\sum_f x_t(f) \left(\sum_{j=1}^{J} g_{n_j^k(t),j} b_{n_j^k(t),j}(f) \right)^{\beta-2} b_{n_i^k(t),i}(f)}{\sum_f \left(\sum_{j=1}^{J} g_{n_j^k(t),j} b_{n_j^k(t),j}(f) \right)^{\beta-1} b_{n_i^k(t),i}(f)} \tag{21}$$

The gain of note $n_i^k(t)$ and the selected instrument i for the combination M_k at each frame t is estimated using the gradient algorithm and applying (21) with only a few iterations. In fact, only $\alpha = 5$ iterations were used. Performing more iterations did not produce better results in our preliminary tests. This NMF computation is used in order to factorize the analysed frame with only these notes and evaluate the distortion that it causes. As the maximum number of selected notes is 4 (when there are four instruments) and only the gain of these 4 notes must be estimated, a low number of iterations is needed. Besides, the gains are initialized by using the direct gain estimation of MBHC-MS, supposing that there is only one active note. Then the NMF iterative code must only refine the initialization.

To justify the use of only 5 iterations Table 1 shows the distortion caused by the factorization of a four instruments file with 5, 10, 15 and 20 iterations. The 0 iteration column represents the distortion caused only by the initialization gains.

After estimating the gains $g_{n_i^k(t),j}$ for all the combinations M_k, (15) is applied to compute the associated distortion. The optimum solution at each frame is obtained

Table 1 Distortion caused when applying MBHC-PS with [0, 5, 10, 15, 20] iterations over a file with four instruments

No. of iterations	0	5	10	15	20
Distortion	$2.8548 * 10^6$	$2.5949 * 10^6$	$2.5948 * 10^6$	$2.5948 * 10^6$	$2.5948 * 10^6$

by selecting the combination M_k that generates the minimum distortion, as indicated in (22).

$$M_{k_{\text{opt}}} = \arg \min_{M_k \in \Psi} D_\beta \left(x_t(f) | \sum_{j=1}^{J} g_{n_j^k(t),j} b_{n_j^k(t),j}(f) \right) \tag{22}$$

In summary, the method for decomposing polyphonic signals from monophonic instruments using β-divergence is shown in Algorithm 2. The performance of this algorithm for SSS and AMT is shown in Tables 4 and 6 at Section 4.4 in comparison with other state-of-the art methods.

Algorithm 2 MBHC-PM gain estimation algorithm

```
Initialise b_{n,j}(f) with the trained instrument models
for t = 1 to number of frames do
  for k = 1 to S do
    Initialise gains g_{n_j^k(t),j} with MBHC-MS values (assuming that only one
    instrument is present on the signal) for notes n_j^k of the combination M_k and
    zero for the rest
    for α iterations do
      for i = 1 to J do
        Update the gains g_{n_i^k(t),i} using (21).
      end for
    end for
    Compute the β-divergence with (15)
  end for
  Select the combination of notes M_k that generates the lowest β-divergence
  using (19).
end for
```

3.2.2 Gain estimation using non negative sparse coding (NNSC) for polyphonic mixtures of monophonic sources

Despite the reduced number of NMF iterations needed when using the factorization algorithm described in Section 3.2.1, the process must be repeated for each combination M_k from Ψ. As is well-known, the iterative nature of NMF factorization makes it unsuitable for real-time applications.

MBHC-PM can be adapted for sparse coding to produce a direct solution (as in MBHC-MS), and avoiding an iterative procedure. This option allows MBHC-PM to be used in real-time applications for a low polyphony level, as we will demonstrate in Section 5. For $\beta = 2$ (Euclidean distance), (19) can be simplified to compute the gains directly using Non Negative Sparse Coding (NNSC), i.e., an iterative algorithm is not needed. The global minimum of the distortion function in (19) is found for $\beta = 2$ by assuming $D_\beta = 0$. The resulting expression can be modified for the combination M_k at frame t as follows,

$$\sum_{j=1}^{J} g_{n_j^k(t),j} \sum_f b_{n_i^k(t),i}(f) b_{n_j^k(t),j}(f) = \sum_f b_{n_i^k(t),i}(f) x_t(f) \tag{23}$$

Equation (23) can be rewritten using matrix notation as,

$$\mathbf{gB} = \mathbf{c} \tag{24}$$

where $\mathbf{g}$ is a $1 \times J$ gains vector, $\mathbf{B}$ is a $J \times J$ matrix depending on the basis and $\mathbf{c}$ is a $1 \times J$ vector dependent both on the gains and the audio signal. $g(j) = g_{n_j^k(t),j}$ is the unknown gain vector for the selected combination M_k at frame t, $B(j,i) = \sum_f b_{n_i^k(t),i}(f) b_{n_j^k(t),j}(f)$ and $c(i) = \sum_f b_{n_i^k(t),i}(f) x_t(f)$. $\mathbf{B}$ can be already computed because it contains the cross correlation matrix of the basis $b_{n_j^k(t),j}$. $\mathbf{B}$ takes high values when the notes are harmonically related and low values otherwise. $\mathbf{c}$ should be computed online because it depends on the audio signal spectrogram.

Then the gains can be estimated in just one step by

$$\mathbf{g} = \mathbf{cB}^{-1} \tag{25}$$

where $g(j) = g_{n_j^k(t),j}$. Equation (25) can generate negative values that are set to zero as in [26].

After estimating of the gains for all the combinations from Ψ, (22) is used to select the optimum combination $M_{k_{\text{opt}}}$ that generates the minimum distortion at each frame.

3.3 Candidates selection for polyphonic mixtures of monophonic sources

An exhaustive search over Ψ is highly computationally intensive, because there is a large number of combinations, which increase dramatically with the number of instruments (with the level of polyphony).

A general expression for calculating the number of combinations of elements from a group with repeated notes per instrument is

$$S = \left(\frac{N_t}{J}\right) = \frac{N_t!}{J!(N_t - J)!} \tag{26}$$

where S is the total number of combinations $N_t = \sum_{j=1}^{J} N(j)$ is the total number of notes from all the instruments (with repeated notes per instrument), $N(j)$ is the number of notes for the instrument j, and J is the number of notes in a combination, which reduces to the number of instruments in the case of monophonic instruments. This expression should be modified to subtract the combinations that contain more than one note by the same instrument (without repeating notes for each instrument) as follows,

$$S = \frac{N_t!}{J!(N_t - J)!} - \sum_{j=1}^{J} \frac{N(j)!}{J!(N(j) - J)!} \tag{27}$$

where J is the number of instruments and $N(j)$ is the number of possible notes for instrument j.

For example, a duet for the violin (46 possible notes) and clarinet (40 possible notes) produces 1,840 combinations according to (27). Moreover for polyphony level 4 (with bassoon, clarinet, violin and saxophone) the number of combinations is over 23 million. This large number of combinations has a correspondingly large computational cost, and thus the space Ψ to be searched should be reduced. This

reduction is facilitated by limiting the possible notes for each instrument. In (27), the number of possible notes per instrument $N(j)$ is the whole range of notes for instrument j. Instead, the exhaustive search is limited to only C note candidates per instrument, which were previously selected using a fast transcription algorithm.

Note candidates are selected using information about the instrument models and the mixed signal. The candidates selection must be fast to serve as a good alternative for saving computational cost and time.

In this work, we obtain a list of candidates list using the MBHC-MS model from Section 3.1. Although the model is designed for monophonic signals, it is adapted to polyphonic signals by assuming that only one instrument is being played. The distortion caused by this monophonic solution is then computed using (11) and (12). The C notes that causes a lower distortion rate are the selected candidates for the instrument in the reduced exhaustive search at the next stage. This factorization has a very low computational cost, resulting in a fast selection of candidates.

Algorithm 3 describes the computational procedure for the selection of note candidates.

Algorithm 3 Description of the candidates selection algorithm

```
Initialise b_{n,j}(f) to the trained instrument models
for j = 1 to J do
    for t = 1 to number of frames do
        Compute MBHC-MS with (12) and (11)
        Select the C notes that causes the lowest β-divergence for the instrument j
        at frame t, C being the number of note candidates
    end for
end for
```

The key here is to determine the optimal number of candidates C that reduces the computational cost while not being so restrictive such that the correct note is lost. The performance of the candidates selector has been tested using the Bach Chorals database [9] to determine the number of candidates per instrument. The results are shown in Table 2. Fifteen candidates per instrument are needed to maintain an accuracy at least 5 % in selecting the correct note from the note candidates.

Table 3 compares the number of combinations with and without the proposed candidate selection algorithm, showing that the number of combinations is greatly reduced by selecting 15 candidates for each instrument. It must be stressed the number of combinations for the candidate selection algorithm is computed using (27) where C replaces $N(j)$ as the number of possible notes per instrument. The effect of applying this candidate selection algorithm will be next tested with the AMT and SSS applications.

Table 2 Percentage of notes lost by candidates selection

% of lost notes	No. of candidates per instrument				
Polyphony	5 (%)	10 (%)	15 (%)	20 (%)	25 (%)
2	9	5	1.6	0.3	0.08
3	16	7	2.2	0.4	0.1
4	24	10	2.8	0.4	0.1

Table 3 Number of combinations S for candidate selection (15 candidates) using the entire dynamic range of each instrument. Polyphony 2 is computed using a bassoon and a clarinet, Polyphony 3 is computed using a bassoon, a clarinet and a saxophone, and Polyphony 4 is computed using a bassoon, a clarinet, a saxophone and a violin

	Polyphony 2	Polyphony 3	Polyphony 4
Candidate selection ($C = 15$)	225	12.825	483.000
Entire dynamic range	1,560	197.000	23,987.000

4 Evaluation

In this section, the algorithms proposed in Section 3 are evaluated for applying both SSS and AMT to polyphonic mixtures composed of monophonic sources. These algorithms are compared to other state-of-the-art algorithms to assess their performance.

For AMT, individual the transcription is computed for each instrument present in the mixture. To the best of our knowledge, no other work in the literature simultaneously performs AMT and SSS for polyphonic mixtures. For comparison, we have adapted other state-of-the-art signal decomposition methods specifically designed for monophonic instruments.

4.1 Training and testing data

At the training stage (see Section 2.5), the basis functions are estimated using the RWC musical instrument sound database [17, 18] and the full pitch range for each instrument. Four instruments are studied in the experiments (violin, clarinet, tenor saxophone and bassoon). Individual sounds are available with a semitone frequency resolution over the entire range of notes for each instrument. Files from the RWC database have different playing styles. Files with a normal playing style and mezzo dynamic level are selected as in the literature. Training with different playing styles leads to different models. However, as demonstrated in [6], the selected configuration (normal playing style and mezzo dynamic level) is representative of the different models.

The database proposed in [9] is used for the testing stage. This database consists of 10 J.S. Bach four-part chorales [9, 10] with the corresponding aligned MIDI data. The audio files are approximately 30 s long and are sampled at 44.1 KHz from real performances. Each music excerpt consist of an instrumental quartet (violin, clarinet, tenor saxophone and bassoon), and each instrument is given in an isolated track. Individual lines were mixed to create a total of 10 performances with four-part polyphony from, 60 duets and 40 trios.

4.2 Experimental setup

4.2.1 Time-frequency representation

Many NMF-based signal processing applications usually adopt frequency logarithmic discretization. For example, uniformly spaced subbands on the Equivalent Rectangular Bandwidth (ERB) scale are assumed in [4, 39]. Here, we use the resolution of a single semitone as in [6]. Additionally, the training database and the ground truth

score information are composed of notes that are separated by one semitone in frequency. In this work, we implement a time-frequency representation by integrating the STFT bins corresponding to the same semitone interval.

The frame size and the hop size for the STFT are set to 128 ms and 32 ms respectively. Other values for the experimental parameters are the following: (1) 20 partials per basis function for the harmonic constraint models ($M = 20$); and (2) 50 iterations for the NMF-based algorithms, except for the MBHC-PM algorithm where this value is set to 5, as justified in Section 3.2.1.

4.2.2 Music separation: method and metrics

- Source separation consists of estimating the corresponding amplitude of each time-frequency cell for each source. Some systems utilises binary separation, which means that the entire energy of a bin is assigned to a single source. However, it has been demonstrated that better results can be obtained with a non-binary decision, i.e., distributing the energy proportionately over all the sources. Practically, this method is more suitable for harmonic polyphonic signals due to partial overlapping. The use of separation Wiener masks is common in the source separation literature [11]. In the present work, instrument models are used as separation method, providing reliable amplitude estimation for the overlapped partials.
- For an objective evaluation of the performance of the separation method we use the metrics implemented in [37, 38]. These metrics are commonly accepted by the specialised scientific community, and therefore facilitate a fair evaluation of the method. Each separated signal is assumed to produces a distortion model that can be expressed as follows,

$$\hat{s}_j(t) - s_j(t) = e_j^{\text{target}}(t) + e_j^{\text{interf}}(t) + e_j^{\text{artif}}(t) \tag{28}$$

where $\hat{s}_j$ is the estimated source signal for instrument j, s_j is the original signal of the instrument j, e^{target} is the error term associated with the target distortion component, e^{interf} is the error term due to interference of the other sources and e^{artif} is the error term attributed to the numerical artifacts of the separation algorithm. The metrics for each separated signal are the *Source to Distortion Ratio* (SDR), the *Source to Interference Ratio* (SIR), and the *Source to Artifacts Ratio* (SAR) [37, 38].

$$SDR_j = 10\log_{10}\frac{\sum_t \left|s_j(t)\right|^2}{\sum_t \left|\hat{s}_j(t) - s_j(t)\right|^2} \tag{29}$$

$$SIR_j = 10\log_{10}\frac{\sum_t \left|s_i(t) + e_j^{\text{target}}(t)\right|^2}{\sum_t \left|e_j^{\text{interf}}(t)\right|^2} \tag{30}$$

$$SAR_j = 10\log_{10}\frac{\sum_t \left|s_i(t) + e_j^{\text{target}}(t) + e_j^{\text{interf}}(t)\right|^2}{\sum_t \left|e_j^{\text{artif}}(t)\right|^2} \tag{31}$$

4.2.3 Music transcription: method and metrics

- Given the time-varying amplitudes of all the basis functions $g_{n,j}(t)$, our method for music transcription is the same as in [4, 6, 39], i.e., we determine whether a note is active or not on a frame-by-frame basis using the following equation:

$$\Omega(n,\ j,\ t) = g_{n,j}(t) \geq \left(10^{T/20} \max_{nt} g_{n,j}(t)\right) \tag{32}$$

  where $\Omega(n,\ j,\ t)$ is the resulting binary transcription and T is the fixed detection threshold in decibels (dB) which is learned from the training data.
  A threshold is required in BHC-based methods to decide which notes are activated at each frame. In contrast, MBHC-based methods do not need a threshold for activating notes, because only one note per instrument is active at each frame. However, a threshold is necessary so that no notes are activated during intervals of silence.
- Transcription methods can be tested by two groups of metrics: note-wise and frame-wise metrics. Frame-wise metrics are used in this work, as in [6]. Practically, we use the frame-level version of the metric proposed in [8] to objectively evaluate transcription performance. The overall accuracy Acc(%) is defined as follows:

$$Acc = \frac{TP}{FP + FN + TP} \tag{33}$$

  where *TP* (true positives) is the number of correctly transcribed note-frames (over all notes), *FP* (false positives) is the number of inactive note-frames transcribed as active, and *FN* (false negatives) is the number of active note-frames transcribed as inactive. Acc ranges from 0 to 1, where Acc = 1 corresponds to perfect transcription.

4.3 Algorithms for comparison

The advantages of the methods proposed here are highlighted by comparing the approach in Section 3 to the methods described in Section 2 (BHC and BHC with sparse constraint). The proposed methods were compared to two state-of-the-art monophonic restricted methods: Gaussian Scaled Mixture Models (GSMM) [3] and Factorial Scaled Hidden Markov Models (FS-HMM) [30], which were both implemented using the Flexible Audio Source Separation Toolbox (FASST) [29]. The last two models are constrained to have a single non-zero entry for each instrument at each frame.

Although FASST was originally designed for sound source separation, we have adapted it for automatic music transcription. FASST gives a gains matrix as output of the signal factorization for each source. Then, a threshold is applied to each matrix in order to obtain a binary transcription of the source. This thresholding is also applied to the gains matrixes obtained from the proposed method from Section 3 and it is explained at Section 4.2.3.

Different FASST configurations have been tested, but the results are not provided here due to space consideration. FASST allow to use the classical FFT time-frequency representation or the QERB one, which is more suitable for musical instrument because it uses a logarithmic frequency resolution scale instead of the

FFT which uses a linear scale. At a linear scale, small variations of the fundamental frequency can produce variations larger than the main lobe of the window transform at high frequencies. The best performance was obtained using the QERB time-frequency representation and by computing the decompositions with the Generalised Expectation Maximization (GEM) algorithm where the generative model was modified to use a Poisson distribution (in its original form, FASST utilises a Gaussian distribution with IS divergence). Using the Poisson distribution is equivalent to performing the factorization with the Kullback-Leibler divergence ($\beta = 1$) [42]. The number of bases K was set to 114 (i.e. the MIDI notes ranged from 24 to 137), which is independent of the modelled instrument, all the modeled instruments have its dynamic ranges between this MIDI notes.

4.4 Results

As just stated, we have tested the reliability of our method for SSS and AMT tasks using polyphonic mixtures of monophonic sources from the database proposed in [9]. We have analysed the performance of the BHC, BHC with sparse constraints and MBHC-PM methods as functions of the parameter β. Practically, a value for the divergence $\beta = 1.5$ produces the most reliable results, but the optimization of β is omitted here due to space considerations. Therefore, the proposed MBHC-PM method uses this optimum β value. $\beta = 2$ will be used to evaluate the MBHC-PM method using Sparse Coding. As will be explained later, the results obtained using MBHC with NNSC do not differ much from the iterative version (MBHC-PM with NMF), and because a very low runtime is required to perform the factorization, the method is a suitable alternative for real-time applications.

The results are averaged between all the files and are presented separately for each method and application. Following [6], the NMF free parameters are randomly initialised and the measures for each file are computed after 30 executions. In our experiments, the 95 % confidence intervals for the accuracy (Acc) are less than 1.6 % for all the algorithms, which means that the differences between most algorithms are statistically significant. A similar result is observed for the source separation metrics, where the 95 % confidence intervals for the SDR are less than 1.4 dB for all algorithms.

4.4.1 Source separation results

The numerical results for SSS in terms of SDR, SIR and SAR (in dB) are displayed for all the tested methods in Table 4.

The MBHC-PM and MBHC-PM methods with candidate selection show very similar results for all polyphony levels, demonstrating that using 15 candidates per instrument is a good choice. In Section 3.3, we justified the use of candidates selection based on the large reduction in computational cost. Table 2 showed that less than 5 % of the correct notes are lost for 15 note candidates. Table 4 also shows that the candidates selection procedure has no effect on the separation results.

The NNSC MBHC-PM ($\beta = 2$) method is slightly outperformed by the NMF MBHC-PM ($\beta = 1.5$) method for all polyphony levels. Thus, MBHC-PM method with NNSC is a reliable and fast method that can be used for real-time applications.

Taking all these considerations into account, all the MBHC-PM algorithms perform better that the other tested methods , attaining SDR values of 7.94 dB at

Table 4 Source separation results (dB) for the methods using polyphony 2, 3 and 4: MBHC-PM with the NMF approach (NMF MBHC-PM $\beta = 1.5$), MBHC-PM with the NMF approach and candidates selection (NMF MBHC-PM with candidates selection $\beta = 1.5$), MBHC-PM with the NNSC approach (NNSC MBHC-PM $\beta = 2$) and MBHC-PM with the NNSC approach with candidates selection (NNSC MBHC-PM with candidates selection $\beta = 2$). A comparison with state-of-the-art methods (BHC, BHC with sparse constraints ($\lambda = 1$), GSMM and FS-HMM) is also shown

Method	J = 2			J = 3			J = 4		
	SDR	SIR	SAR	SDR	SIR	SAR	SDR	SIR	SAR
NMF MBHC-PM	**7.94**	20.03	11.95	**3.14**	15.63	4.84	–	–	–
NMF MBHC-PM with candidates selection	**7.94**	20.01	11.95	**3.14**	15.56	4.82	**1.81**	14.33	1.43
NNSC MBHC-PM	6.15	17.83	9.38	2.51	14.77	3.6	–	–	–
NNSC MBHC-PM with candidates selection	6.31	18.32	9.6	2.51	14.77	3.6	1.38	13.64	0.51
BHC with sparse constraint	4.32	17.34	4.72	2.04	15.64	3.62	1.26	14.13	−2.01
BHC	4.19	18.24	4.56	1.9	14.41	1.5	1.13	23.9	0.77
FS-HMM	3.6	14.54	5.5	1.4	12.32	4.35	0.94	10.27	3.05
GSMM	3.72	14.7	5.9	1.56	13.01	4.56	0.96	10.58	3.12

polyphony level 2. The next best method (BHC with sparse constraint) produces a SDR approximately 2.5 dB below that of the MBHC-PM methods. MBHC-PM algorithms produce better results than BHC and BHC with sparse constraints due to the use of the monophonic constraint. Additionally, the monophonic constrained models avoid interferences between different instruments and artifacts as can be seen from the SIR and SAR values of Table 4 at all polyphony levels. BHC (when the sparse constraint is not enforced), yields a similar SDR value to that obtained by using BHC with sparse constraints, while FS-HMM and GSMM methods produce lower values. This under-performance of the FASST-based methods may be caused by the smaller number of parameters that are estimated with MBHC-PM, BHC and BHC with sparse constraints methods than with the FS-HMM and GSMM methods. The harmonic constrained methods have a smaller number of parameter to be estimated because each basis function is only defined by M amplitudes as expressed in (2), while the FASST-based methods require all the points in the frequency range to be estimated.

Table 5 shows measures from a runtime test for 30 s excerpts of a duet and a tercet. The candidate selection stage considerably reduces the computation time. BHC with sparse constraints, BHC, FS-HMM, and GSMM are not feasible for real-time implementation. NNSC MBHC-PM method with candidate selection and $\beta = 2$ reduces the runtime approximately 40 %, but the results in Table 4 are

Table 5 Runtime test for a 30 s excerpt at polyphony levels 2 and 3

Method	J = 2 (s)	J = 3 (s)
NMF MBHC-PM	18.074	1,026.342
NNSC MBHC-PM	3.157	179.949
NMF MBHC-PM with candidates selection	21	1.228
NNSC MBHC-PM with candidates selection	13	747
BHC with sparse constraint	356	20.295
BHC	356	20.295
FS-HMM	23.425	1,335.225
GSMM	24.362	1,388.634

worse. However, the strongest runtime reduction is achieved using the candidate selection algorithm. Selecting $C = 15$ note candidates per instrument produces the same separation results while reducing the runtime by more than 99 % for the examples shown in Table 5. MBHC-PM without candidate selection is not run at polyphony level 4 because of the large number of combinations involved (an example using the same number of combinations is given in Table 3).

The MBHC-PM method (for the two NMF MBHC-PM and NNSC MBHC-PM algorithms) with and without candidate selection produces very little differences in the results as shown in Table 4. The AMT results will therefore be computed only for the candidate selection version.

Finally, real-time implementation is only possible for the NMF MBHC-PM and NNSC MBHC-PM methods, both with candidate selection and when $J = 2$, as shown in Table 5.

All experiments were performed using Matlab on a 2.00 GHz Intel Xeon processor. Examples of source separation results at different polyphony levels are available at http://anclas3.ujaen.es/monosourceseparation.

4.4.2 Automatic music transcription results

Table 6 shows the AMT results using the same methods as SSS, although the method without candidate selection is not included. The AMT results agree with the SSS results.

The MBHC-PM method clearly outperforms the other methods as in the SSS testing, demonstrating the reliability of the monophonic constrained method for polyphonic signals composed of monophonic sources. Better results are obtained once more for NMF MBHC-PM ($\beta = 1.5$) than for NNSC MBHC-PM ($\beta = 2$). Thus, we conclude, as in [6], that the Euclidean distance ($\beta = 2$) is not the optimal value for β. However, the NNSC algorithm that can only be used with this β value is less complex than the NMF based algorithms.

The main difference between the AMT and SSS results comes from the BHC and BHC with sparse constraints methods. A significant gap is seen when comparing the results of both methods in Tables 4 and 6. Thus, the sparse constraint is more effective in the AMT task, probably due to the difficult decision that was taken to select a threshold to obtain the transcription (see (32)). In contrast, the SSS task with Wiener masks and the sparse constraint favours the concentration of energy in some of the time-frequency cells, but as the energy is proportionately distributed between instruments, all the instruments possess some energy at each time-frequency cell.

In general, the sparse and monophonic constrained models are observed to fit monophonic sources better than the methods without these constraints (such as BHC). The monophonic constraint also appears to be a better choice for polyphonic signals composed of monophonic sources than the sparse constraint given by (5).

All the methods decrease in accuracy as the polyphony level increases because it is more difficult to distinguish each note that arises, with the instrument, as the polyphony level goes up. This is because as the number of instruments increases, it is not easy to fit the basis function associated with each note derived from the corresponding instrument model to the spectral shape of the signal. It must be stressed that the proposed method allows an independent transcription to be obtained for each instrument. Other transcription methods for polyphonic signals, such as those proposed in [23, 39], compute the general transcription without

Table 6 Automatic Music Transcription (Acc) results for the following methods at polyphony levels 2, 3 and 4: NMF MBHC-PM with candidate selection and $\beta = 1.5$ and NNSC MBHC-PM with candidate selection and $\beta = 2$). Comparison with state-of-the-art methods (BHC, BHC with sparse constraints ($\lambda = 1$), GSMM and FS-HMM) is also shown. From [6], the Euclidean distance ($\beta = 2$) is not the optimum value for the β parameter. However, the NNSC-based algorithm (which uses $\beta = 2$) is less complex than the NMF-based algorithm

Method	Instrument	J = 2	J = 3	J = 4
NMF MBHC-PM with candidates selection	Bassoon	0.55	0.4	0.32
	Clarinet	0.65	0.44	0.39
	Saxophone	0.64	0.43	0.38
	Violin	0.55	0.39	0.33
	Mean	0.60	0.42	0.35
NNSC MBHC-PM with candidates selection	Bassoon	0.38	0.3	0.23
	Clarinet	0.6	0.41	0.28
	Saxophone	0.56	0.36	0.27
	Violin	0.43	0.31	0.22
	Mean	0.49	0.35	0.25
BHC with sparse constraint	Bassoon	0.33	0.26	0.20
	Clarinet	0.53	0.41	0.34
	Saxophone	0.5	0.33	0.19
	Violin	0.32	0.21	0.16
	Mean	0.42	0.3	0.22
BHC	Bassoon	0.33	0.23	0.19
	Clarinet	0.41	0.26	0.20
	Saxophone	0.36	0.18	0.12
	Violin	0.3	0.16	0.14
	Mean	0.35	0.21	0.16
FS-HMM	Bassoon	0.27	0.15	0.1
	Clarinet	0.33	0.16	0.12
	Saxophone	0.22	0.09	0.09
	Violin	0.25	0.14	0.11
	Mean	0.27	0.14	0.11
GSMM	Bassoon	0.24	0.14	0.09
	Clarinet	0.35	0.17	0.12
	Saxophone	0.22	0.15	0.1
	Violin	0.3	0.16	0.12
	Mean	0.28	0.15	0.1

distinguishing between instruments. Thus, these methods do not show the same decrease in accuracy as the polyphony level increases. The same underperformance is observed for the SSS results (Table 4) for increasing polyphony levels.

FS-HMM and GSMM suffer from the same difficulties in SSS: more free parameters must be estimated than in the other methods, as there is no harmonic restriction, resulting in under-performance compared to the other methods. The FASST-based methods must estimate the entire frequency bin range from the QERB transform. However harmonic constrained methods must only estimate one set of M amplitudes per note, as described in (2).

Examining the results for each instrument, the saxophone and clarinet outperform the bassoon and violin by 10 %. The difference in performance can be attributed to the fit of the trained model to the actual instrument being played. This mismatch between the actual instrument and the associated instrument model can be caused by the way the musician plays the instrument, such as how a violin string is rubbed, or by physical differences between the model and the actual instrument, as in the case of bassoon. It must be stressed that the instrument models are obtained from a music database, so that the learned instrument models have significant differences with respect to the instrument signals used for testing, that are from a different database.

5 Conclusions

In this paper, a monophonic restricted factorization method (MBHC-PM) is proposed to model polyphonic mixtures of monophonic sources, where harmonic and single-non-zero gain constraints are enforced in a deterministic manner. We present two different algorithms to perform the factorization: an NMF-based algorithm (suitable for $\beta = [0, 2]$) and a less complex NNSC based algorithm (which is only valid for $\beta = 2$). The MBHC-PM method method and other state-of-the-art methods have been tested using a database containing 40 solo files of bassoon, clarinet, tenor saxophone and violin performances (10 per instrument). SSS and AMT results have been computed for all methods; the best results were obtained using the MBHC-PM method.

An independent transcription per instrument from each file is obtained in the AMT tests, facilitated by the use of instrument models to distinguish the timbre of notes between different instruments.

BHC and BHC with sparse constraints methods does not use a monophonic constraint, and therefore more suitable for polyphonic signals because they suffer from the activation of more than one pitch at each frame. Using the MBHC-PM method, the single-non-zero constraint mitigates this problem, as demonstrated by the results.

The FS-HMM and GSMM methods suffer from the large number of parameters that need to be estimated due to the lack of harmonic restrictions in these methods.

The SSS and AMT results show that increases in polyphony seriously affect the results. However, promising results are obtained for low levels of polyphony by using instrument-dependent basis functions, which have been trained in advance.

Finally, this paper highlights the advantages of the proposed MBHC-PM methods over other state-of-the-art methods. Additionally, the proposed approach can be implemented in real-time for a polyphony level of 2.

In future work, we will combine information from the instrument models and the score to reduce the high computational cost associated with polyphony levels above 2. We will also update the instrument models during testing to achieve a better fit between the modelled instruments and the instruments being played.

Acknowledgements This work was supported by the Andalusian Business, Science and Innovation Council under project P10- TIC-6762, (FEDER) the Spanish Ministry of Science and Innovation under Project TEC2009-14414-C03-02, and the University of Jaen under Project R1/12/2010/64.

The authors would like to thank Z. Duan for kindly sharing his annotated real world music database with them.

References

1. Abdallah S, Plumbley M (2004) Polyphonic music transcription by non-negative sparse coding of power spectra. In: Proc. 5th Int. Society for Music Information Retrieval conf. (ISMIR), Barcelona, Spain
2. Abdallah S, Plumbley M (2006) Unsupervised analysis of polyphonic music by sparse coding. IEEE Trans Neural Netw 17(1):179–196
3. Benaroya L, Bimbot F, Gribonval R (2006) Audio source separation with a single sensor. IEEE Trans Audio Speech Lang Process 14(1):191–199
4. Bertin N, Badeau R, Vincent E (2010) Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription. IEEE Trans Audio Speech Lang Process 18(3):538–549

5. Candés EJ, Wakin MB (2008) An introduction to compressive sampling. IEEE Signal Process Mag 25(2):21–30
6. Carabias-Orti JJ, Virtanen T, Vera-Candeas P, Ruiz-Reyes N, Cañadas-Quesada FJ (2011) Musical instrument sound multi-excitation model for non-negative spectrogram factorization. IEEE J Sel Topics Signal Process 5(6):1144–1158
7. Chen SS, Donoho DL, Saunders MA (1998) Atomic decomposition by basis pursuit. SIAM J Sci Comput 20:33–61
8. Dixon S (2000) On the computer recognition of solo piano music. In: Proceedings of Australasian computer music conference
9. Duan Z, Pardo B (2011) Soundprism: an online system for score-informed source separation of music audio. IEEE J Sel Topics Signal Process 5(6):1205–1215
10. Duan Z, Pardo B, Zhang C (2010) Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. IEEE Trans Audio Speech Lang Process 18(8):2121–2133
11. Every MR, Szymanski JE (2006) Separation of synchronous pitched notes by spectral filtering of harmonics. IEEE Trans Audio Speech Lang Process 14(5):1845–1856
12. Févotte C, Idier J (2011) Algorithms for nonnegative matrix factorization with the beta-divergence. Neural Comput 23(9):2421–2456
13. Févotte C, Bertin N, Durrieu JL (2009) Nonnegative matrix factorization with the Itakura–Saito divergence. With application to music analysis. Neural Comput 21(3):793–830
14. FitzGerald D, Cranitch M, Coyle E (2009) On the use of the beta divergence for musical source separation. In: Signals and systems conference (ISSC 2009), IET Irish, 10–11 June 2009, pp 1–6
15. Gainza M, Coyle E (2007) Automating ornamentation transcription. In: IEEE international conference on acoustics, speech and signal processing, 2007. ICASSP 2007, vol 1, 15–20 April 2007, pp I-69–I-72
16. Gemmeke JF, Virtanen T, Hurmalainen A (2011) Exemplar-based sparse representations for noise robust automatic speech recognition. IEEE Trans Audio Speech Lang Process 19(7): 2067–2080
17. Goto M (2004) Development of the RWC music database. In: Proc. of the 18th international congress on acoustics (ICA 2004), pp I-553–I-556 (invited paper)
18. Goto M, Hashiguchi H, Nishimura T, Oka R (2002) RWC music database: popular, classical, and jazz music databases. In: Proc. of the 3rd Int. Society for Music Information Retrieval conf. (ISMIR), Paris, France
19. Gribonval R, Bacry E (2003) Harmonic decomposition of audio signals with matching pursuit. IEEE Trans Signal Process 51(1):101–111
20. Helen M, Virtanen T (2005) Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In: Proc. EUSIPCO
21. Hoyer P (2004) Non-negative matrix factorization with sparseness constraints. J Mach Learn Res 5:1457–1469
22. Hyvarinen A, Oja E (2000) Independent component analysis: algorithms and applications. Neural Netw 13:411–430
23. Klapuri A (2004) Signal processing methods for the automatic transcription of music. PhD thesis, Tampere University of Technology
24. Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401:788–791
25. Lee DD, Seung HS (2000) Algorithms for non-negative matrix factorization. In: Proc. of neural information processing systems, Denver, USA
26. Marxer R, Jordi J, Bonada J (2012) Low-latency instrument separation in polyphonic audio using timbre models. In: Proc. LVA/ICA
27. Namgook C, Kuo C-CJ (2009) Underdetermined audio source separation from anechoic mixtures with long time delay. In: IEEE international conference on acoustics, speech and signal processing, 2009. ICASSP 2009, 19–24 April 2009, pp 1557–1560
28. Olshausen BA, Field DF (1997) Sparse coding with an overcomplete basis set: a strategy employed by V1? Vis Res 37:3311–3325
29. Ozerov A, Févotte C (2010) Multichannel non-negative matrix factorization in convolutive mixtures for audio source separation. IEEE Trans Audio Speech Lang Process 18(3):550–563
30. Ozerov A, Févotte C, Charbit M (2009) Factorial scaled hidden Markov model for polyphonic audio representation and source separation. In: IEEE workshop on applications of signal processing to audio and acoustics, WASPAA'09, pp 121–124
31. Ozerov A, Vincent E, Bimbot F (2012) A general flexible framework for the handling of prior information in audio source separation. IEEE Trans Audio Speech Lang Process 20(4): 1118–1133

32. Plumbley M (2003) Algorithms for nonnegative independent component analysis. IEEE Trans Neural Netw 14(3):534–543
33. Raczyński SA, Ono N, Sagayama S (2007) Multipitch analysis with harmonic nonnegative matrix approximation. In: Proc. int. conf. music inf. retrieval (ISMIR), pp 381–386
34. Reyes-Gomez MJ, Raj B, Ellis D (2003) Multi-channel source separation by factorial HMMs. In: Proc. ICASSP, vol I, pp 664–667
35. Sawada H, Araki S, Makino S (2011) Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment. IEEE Trans Audio Speech Lang Process 19(3):516–527
36. Smaragdis P (1998) Blind separation of convolved mixtures in the frequency domain. Neurocomputing 22:21–34
37. Valentin E, Vincent E, Harlander N, Hohmann V (2011) Subjective and objective quality assessment of audio source separation. IEEE Trans Audio Speech Lang Process 19(7):2046–2057
38. Vincent E (2012) Improved perceptual metrics for the evaluation of audio source separation. In: 10th int. conf. on latent variable analysis and signal separation (LVA/ICA 2012)
39. Vincent E, Bertin N, Badeau R (2010) Adaptive harmonic spectral decomposition for multiple pitch estimation. IEEE Trans Audio Speech Lang Process 18(3):528–537
40. Virtanen T (2007) Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria. IEEE Trans Audio Speech Lang Process 15(3): 1066–1074
41. Virtanen T, Klapuri A (2006) 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
42. Virtanen T, Cemgil AT, Godsill S (2008) Bayesian extensions to non-negative matrix factorisation for audio signal modeling. In: Proc. int. conf. acoust., speech, signal process. (ICASSP), Las Vegas, USA
43. Wang B, Plumbley MD (2005) Musical audio stream separation by non-negative matrix factorization. In: Proc. DMRN summer conference, Glasgow
44. Zibulevsky M, Kisilev P, Zeevi YY, Pearlmutter B (2002) Blind source separation via multinode sparse representation. In: NIPS

Francisco José Rodríguez-Serrano was born in Linares, Spain, in 1986. He received the M.Sc. degree in telecommunication engineering from the University of Jaén, Spain, in 2009, where he has been working as a researcher at the Telecommunication Engineering Department of the University of Jaen since then, and he is currently pursuing the Ph.D. degree. His Ph.D. dissertation is focused on sound source separation and synchronization techniques applied to musical signals. Currently, his areas of research interest include sound source separation, automatic sound synchronization and signal modeling.

Julio José Carabias-Orti received the M.Sc. degree in computer science and the Doctor of Science degree from the University of Jaen, Jaen, Spain, in 2006 and 2011, respectively. He is currently working as a research fellow at the Telecommunication Engineering Department, University of Jaen. His research topics include automatic music transcription, sound source separation, factorization algorithms and machine learning.

Pedro Vera-Candeas was born in Madrid, Spain, in 1976. He received the M.Sc. degree in telecommunication engineering from the University of Málaga (UMA), Málaga, Spain, in 2000 and the Ph.D. degree from the University of Alcala, Alcalá de Henares, Spain, in 2006. Since 2000, he has been with the Telecommunication Engineering Department, University of Jaén. Currently, he is an associate professor in signal processing and communications area. His areas of research interest are signal processing and its applications to audio analysis and ultrasonic NDT. He has been involved in research projects of the Spanish Ministry of Science and Education (MEC) and private companies.

Francisco Jesús Canadas-Quesada was born in Linares (Jaén), Spain, in 1977. He received the M.Sc. degree in telecommunication engineering from the University of Malaga, Spain, in 2004 and the Ph.D. degree from the University of Jaén, Spain, in 2009. From 2004 to 2006, he was an engineer on a Europe Research Project (INTUITION Network Excellence) in Electronic Technology Department, University of Malaga. Since 2006, he has been with the Telecommunication Engineering Department, University of Jaén. Currently, he is an assistant professor in the signal processing and communications area. His research interests include audio signal processing and sound source separation.

Nicolás Ruiz-Reyes was born in Linares (Jaén), Spain, in 1967. He received the M.Sc. degree in telecommunication engineering from the Technical University of Madrid (UPM), Madrid, Spain, in 1993 and Ph.D. degree in telecommunication engineering from the University of Alcala, Alcalá de Henares, Spain, in 2001. Since 2010, he has been with the Telecommunication Engineering Department, University of Jaén. Currently, he is a professor in the signal processing and communications area. His areas of research interest are signal processing and its applications to communications, speech and audio analysis, electrical and biomedical engineering, and ultrasonic NDT. He is coauthor of about 150 papers, and is involved in research projects of the Spanish Ministry of Science and Education, European Commission, and private companies.