Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2016, Article ID 8363507, 19 pages
http://dx.doi.org/10.1155/2016/8363507

Research Article

Score-Informed Source Separation for Multichannel Orchestral Recordings

Marius Miron, Julio J. Carabias-Orti, Juan J. Bosch, Emilia Gómez, and Jordi Janer

Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

Correspondence should be addressed to Marius Miron; marius.miron@upf.edu

Received 24 June 2016; Accepted 3 November 2016

Academic Editor: Alexander Petrovsky

This paper proposes a system for score-informed audio source separation for multichannel orchestral recordings. The orchestral music repertoire relies on the existence of scores. Thus, a reliable separation requires a good alignment of the score with the audio of the performance. To that extent, automatic score alignment methods are reliable when allowing a tolerance window around the actual onset and offset. Moreover, several factors increase the difficulty of our task: a high reverberant image, large ensembles having rich polyphony, and a large variety of instruments recorded within a distant-microphone setup. To solve these problems, we design context-specific methods such as the refinement of score-following output in order to obtain a more precise alignment. Moreover, we extend a close-microphone separation framework to deal with the distant-microphone orchestral recordings. Then, we propose the first open evaluation dataset in this musical context, including annotations of the notes played by multiple instruments from an orchestral ensemble. The evaluation aims at analyzing the interactions of important parts of the separation framework on the quality of separation. Results show that we are able to align the original score with the audio of the performance and separate the sources corresponding to the instrument sections.

1. Introduction

Western classical music is a centuries-old heritage traditionally driven by well-established practices. For instance, large orchestral ensembles are commonly tied to a physically closed place, the concert hall. In addition, Western classical music is bounded by established customs related to the types of instruments played, the presence of a score, the aesthetic guidance of a conductor, and compositions spanning a large time frame. Our work is conducted within the PHENICX project [1], which aims at enriching the concert experience through technology. Specifically, this paper aims at adapting and extending score-informed audio source separation to the inherent complexity of orchestral music. This scenario involves challenges like changes in dynamics and tempo, a large variety of instruments, high reverberance, and simultaneous melodic lines but also opportunities as multichannel recordings.

Score-informed source separation systems depend on the accuracy of different parts which are not necessarily integrated in the same parametric model. For instance, they rely on a score alignment framework that yields a coarsely aligned score [2–5] or, in a multichannel scenario, they compute a panning matrix to assess the weight of each instrument in each channel [6, 7]. To account for that, we adapt and improve the parts of the system within the complex scenario of orchestral music. Furthermore, we are interested in establishing a methodology for this task for future research and we propose a dataset in order to objectively assess the contribution of each part of the separation framework to the quality of separation.

1.1. Relation to Previous Work. Audio source separation is a challenging task when sources corresponding to different instrument sections are strongly correlated in time and frequency [8]. Without any previous knowledge, it is difficult to separate two sections which play, for instance, consonant notes simultaneously. One way to approach this problem is to introduce into the separation framework information about the characteristics of the signals such as a well-aligned score [2–5, 9, 10]. Furthermore, previous research relates the accuracy of the alignment to the quality of source separation [4, 11]. For Western classical music, a correctly aligned score yields the exact time where each instrument is playing. Thus, an important step in score-informed source separation is obtaining a correctly aligned score, which can be done automatically with an audio-to-score alignment system.

Audio-to-score alignment deals with the alignment of a symbolic representation such as the score with the audio of the rendition. In a live scenario, this task deals with following the musical score while listening to the live performance, and it is known as score-following. To our knowledge, with the exception of [12], audio-to-score alignment systems have not been rigorously tested in the context of orchestral music. However, with respect to classical music, limited experimental scenarios comprising Bach chorales played by a four instruments have been discussed in [4, 13]. Furthermore, in [14], a subset of RWC classical music database [15] is used for training and testing, though no details are given regarding the instrumental complexity of the pieces. Moreover, a Beethoven orchestral piece from the same database is tested in [16], obtaining lower accuracy than the other evaluated pieces. These results point out the complexity of an orchestral scenario, underlined in [12]. Particularly, a larger number of instruments, many instruments playing concurrently different melody lines [17], prove to be a more difficult problem than tracking a limited number of instruments as in pop music or, for instance, string quartets and piano pieces. Although, in this paper, we do not propose a new system for orchestral music audio-to-score alignment, the task being a complex and extensive itself, we are interested in analyzing the relation between the task and the quality of score-informed source separation in such a complex scenario.

Besides the score, state-of-the-art source separation methods take into account characteristics of the audio signal which can be integrated into the system, thus achieving better results. For instance, the system can learn timbre models for each of the instruments [7, 18]. Moreover, it can rely on the assumption that the family of featured instruments is known, and their spectral characteristics are useful to discriminate between different sections playing simultaneously, when the harmonics of the notes overlap. In a more difficult case, when neither the score or the timbre of the instrument is available, temporal continuity and frequency sparsity [19] help in distributing the energy between sources in a musically meaningful way. Furthermore, an initial pitch detection can improve the results [6, 20–22], if the method assumes a predominant source. However, our scenario assumes multichannel recordings with distant microphones, in contrast to the close-microphone approach in [6], and we cannot assume that a source is predominant. As a matter a fact, previous methods deal with a limited case: the separation between small number of harmonic instruments or piano [3, 4, 6, 18], leaving the case of orchestral music as an open issue. In this paper, we investigate a scenario characterized by large reverberation halls [23], a large number of musicians in each section, a large diversity of instruments played, abrupt tempo changes, and many concurrent melody lines, often within the same instrument section.

Regarding the techniques used for source separation, matrix decomposition has been increasingly popular for source separation during the recent years [2, 3, 6, 18–20]. Nonnegative matrix factorization (NMF) is a particular case of decomposition which restricts the values of the factor matrices to be nonnegative. The first-factor matrix can be seen as a dictionary representing spectral templates. For audio signals, a dictionary is learned for each of the sources and stored into the basis matrix as a set of spectral templates. The second-factor matrix holds the temporal activation or the weights of the templates. Then, the resulting factorized spectrogram is calculated as a linear combination of the template vectors with a set of weight vectors forming the activation matrix. This representation allows for parametric models such as the source-filter model [3, 18, 20] or the multiexcitation model [9], which can easily capture important traits of harmonic instruments and help separate between them, as it is the case with orchestral music. The multiexcitation model has been evaluated in a restricted scenario of Bach chorales played by a quartet [4] and for this particular database has been extended in the scope of close-microphone recordings [6] and score-informed source separation [11]. From a source separation point of view, in this article, we extend and evaluate the work in [6, 11, 18] for orchestral music.

In order to obtain a better separation with any NMF parametric model, the sparseness of the gains matrix is increased by initializing it with time and frequency information obtained from the score [3–5, 24]. The values between the time frames where a note template is not activated are set to zero and will remain this way during factorization, allowing for the energy from the spectrogram to be redistributed between the notes and the instruments which actually play during that interval. A better alignment leads to better gains initialization and better separation. Nonetheless, audio-to-score alignment mainly fixes global misalignments, which are due to tempo variations, and does not deal with local misalignments [21]. To account for local misalignments, score-informed source separation systems include onset and offset information into the parametric model [3, 5, 24] or use image processing in order to refine the gains matrix so that it closely matches the actual time and frequency boundaries of the played notes [11]. Conversely, local misalignments can be fixed explicitly [25–27]. To our knowledge, none of these techniques have been explored for orchestral music, although there is a scope for testing their usefulness, if we take into account several factors as the synchronization of musicians in large ensembles, concurrent melody lines, and reverberation. Furthermore, the alignment systems are monaural. However, in our case, the separation is done on multichannel recordings and the delays between the sources and microphones might yield local misalignments. Hence, towards a better separation and more precise alignment, we propose a robust method to refine the output of a score alignment system with respect to each audio channel of the multichannel audio.

In the proposed distant-microphone scenario, we normally do not have microphones close to a particular instrument or soloist and, moreover, in an underdetermined case,

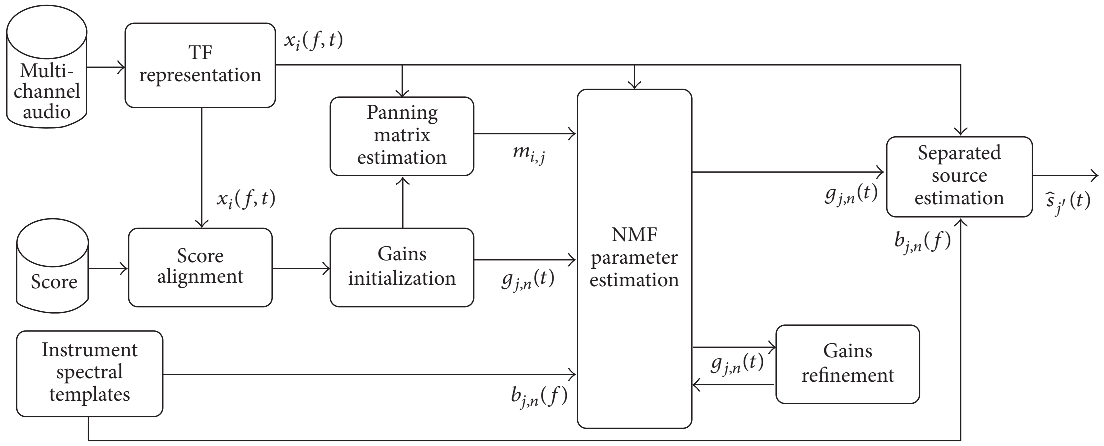


Figure 1: The diagram representing the flow of operations in the system.

the number of sources surpasses the number of microphones. Therefore, recording the sound of an entire section also captures interference from other instruments and the reverberation of the concert hall. To that extent, our task is different from interference reduction in close-microphone recordings [6, 7, 23], these approaches being evaluated for pop concerts [23] or quartets [6]. Additionally, we do not target a blind case source separation as the previous systems [6, 7, 23]. Subsequently, we adapt and improve the systems in [6, 11], by using information from all the channels, similar to parallel factor analysis (PARAFAC) in [28, 29].

With respect to the evaluation, to our knowledge, this is the first time score-informed source separation is objectively evaluated on such a complex scenario. An objective evaluation provides a more precise estimation of the contribution of each part of the framework and their influence on the separation. Additionally, it establishes a methodology for future research and eases the research reproducibility. We annotated the database proposed in [30], comprising four pieces of orchestral music recorded in an anechoic room, in order to obtain a score which is perfectly aligned with the anechoic recordings. Then, using the Roomsim software in [31], we simulate a concert hall in order to obtain realistic multitrack recordings.

1.2. Applications. The proposed framework for score-informed source separation has been used to separate recordings by various orchestras. The recordings are processed automatically and stored in the multimodal repository Repovizz [32]. The repository serves the data through its API for several applications. The first application is called instrument emphasis or orchestra focus and allows for emphasizing a particular instrument over the full orchestra. The second application relates to spatialization of the separated musical sources in the case of virtual reality scenarios and it is commonly known as Acoustic Rendering. Third, we propose an application to estimating the spatial locations of the instruments on the stage. All the three applications are detailed in Section 7.

1.3. Outline. We introduce the architecture of the framework with its main parts in Section 2. Then, in Section 3, we give an outline of the baseline source separation system. Furthermore, we present the extension of the baseline system: the initialization of the gains with score information (Section 4) and the note refinement (Section 4.1). Additionally, the proposed extension to the multichannel case is introduced in Section 5. We present the dataset and the evaluation procedures and discuss the results in Section 6. The demos and applications are described in Section 7.

2. Proposed Approach Overview

The diagram of the proposed framework is presented in Figure 1. The baseline system relies on training spectral templates for the instruments we aim to separate (Section 3.3). Then, we compute the spectrograms associated with the multichannel audio. The spectrograms along with the score of the piece are used to align the score to the audio. From the aligned score, we derive gains matrix that serves as an input for the NMF parameter estimation stage (Section 4), along with the learned spectral templates. Furthermore, the gains and the spectrogram are used to calculate a panning matrix (Section 3.1) which yields the contribution of each instrument in each channel. After the parameter estimation stage (Section 3.2), the gains are refined in order to improve the separation (Section 4.1). Then, the spectrograms of the separated sources are estimated using Wiener filtering (Section 3.5).

For the score alignment step, we use the system in [13] which aligns the scores to a chosen microphone and achieved the best results in MIREX score-following challenge (http://www.music-ir.org/mirex/wiki/2015:Real-time_Audio_to_Score_Alignment_(a.k.a._Score_Following)_Results). However, other state-of-the-art alignment systems can be used at this step, since our final goal is to refine a given score with respect to each channel, in order to minimize the errors in separation (Section 5.2). Accounting for that, we extend the model proposed by [6] and the gains refinement for monaural recordings in [11] to the case of score-informed multichannel source separation in the more complex scenario of orchestral music.

3. Baseline Method for Multichannel Source Separation

According to the baseline model in [6], the short-term complex valued Fourier transform (STFT) in time frame t and frequency f for channel $i = 1, \ldots, I$, where I is the total number of channels, is expressed as

$$\underline{x}_i(f,t) \approx \hat{\underline{x}}_i(f,t) = \sum_{j=1}^{J} \underline{m}_{i,j} s_j(f,t), \tag{1}$$

where $s_j(f,t)$ represents the estimation of the complex valued STFT computed for the source $j = 1, \ldots, J$, with J the total number of sources. Note that, in this paper, we consider a source or instrument, one or more instruments of the same kind (e.g., a section of violins). Additionally, $\underline{m}_{i,j}$ is a mixing matrix of size $I \times J$ that accounts for the contribution of source i to channel j. In addition, we denote $x_i(f,t)$ as the magnitude spectrogram and $m_{i,j}$ as the real-valued panning matrix.

Under the NMF model described in [18], each source $s_j(f,t)$ is factored as a product of two matrices: $g_{j,n}(t)$, the matrix which holds the gains or activation of the basis function corresponding to pitch n at frame t, and $b_{j,n}(f)$, $n = 1, \ldots, N$, the matrix which holds bases, where $n = 1, \ldots, N$ is defined as the pitch range for instrument j. Hence, source j is modeled as

$$s_j(f,t) \approx \sum_{n=1}^{N} b_{j,n}(f)\, g_{j,n}(t). \tag{2}$$

The model represents a pitch for each source j as a single template stored in the basis matrix $b_{j,n}(f)$. The temporal activation of a template (e.g., onset and offset times for a note) is modeled using the gains matrix $g_{j,n}(t)$. Under harmonicity constraints [18], the NMF model for the basis matrix is defined as

$$b_{j,n}(f) = \sum_{h=1}^{H} a_{j,n}(h)\, G(f - hf_0(n)), \tag{3}$$

where $h = 1, \ldots, H$ is the number of harmonics, $a_{j,n}(h)$ is the amplitude of harmonic h for note n and instrument j, $f_0(n)$ is the fundamental frequency of note n, $G(f)$ is the magnitude spectrum of the window function, and the spectrum of a harmonic component at frequency $hf_0(n)$ is approximated by $G(f - hf_0(n))$.

Considering the model given in (3), the initial equation (2) for the computation of the magnitude spectrogram for the source j is expressed as

$$s_j(f,t) \approx \sum_{n=1}^{N} g_{j,n}(t) \sum_{h=1}^{H} a_{j,n}(h)\, G(f - hf_0(n)), \tag{4}$$

and (1) for the factorization of magnitude spectrogram for channel i is rewritten as

$$\hat{x}_i(f,t) = \sum_{j=1}^{J} m_{i,j} \sum_{n=1}^{N} g_{j,n}(t) \sum_{h=1}^{H} a_{j,n}(h)\, G(f - hf_0(n)). \tag{5}$$

3.1. Panning Matrix Estimation.

The panning matrix gives the contribution of each instrument in each channel and as seen in (1) influences directly the separation of the sources. The panning matrix is estimated by calculating an overlapping mask which discriminates the time-frequency zones for which the partials of a source are not overlapped with the partials of other sources. Then, using the overlapping mask, a panning coefficient is computed for each pair of sources at each channel. The estimation algorithm in the baseline framework is described in [6].

3.2. Augmented NMF for Parameter Estimation.

According to [33], the parameters of the NMF model are estimated by minimizing a cost function which measures the reconstruction error between the observed $x_i(f,t)$ and the estimated $\hat{x}_i(f,t)$. For flexibility reasons, we use the beta-divergence [34] cost function, which allows for modeling popular cost functions for different values of β, such as Euclidean (EUC) distance ($\beta = 2$), Kullback-Leibler (KL) divergence ($\beta = 1$), and the Itakura-Saito (IS) divergence ($\beta = 0$).

The minimization procedures assure that the distance between $x_i(f,t)$ and $\hat{x}_i(f,t)$ does not increase with each iteration, thus accounting for the nonnegativity of the basis and the gains. By these means, the magnitude spectrogram of a source is explained solely by additive reconstruction.

3.3. Timbre-Informed Signal Model.

An advantage of the harmonic model is that templates can be learned for various instruments, if the appropriate training data is available. The RWC instrument database [15] offers recordings of solo instrument playing isolated notes along all their corresponding pitch range. The method in [6] uses these recordings along with the ground truth annotation to learn instrument spectral templates for each note of each instrument. More details on the training procedure can be found in the original paper [6].

Once the basis functions $b_{j,n}(f)$ corresponding to the spectral templates are learned, they are used at the factorization stage in any orchestral setup which contains the targeted instruments. Thus, after training the basis $b_{j,n}(f)$ are kept fixed, while the gains $g_{j,n}(t)$ are estimated during the factorization procedure.

3.4. Gains Estimation.

The factorization procedure to estimate the gains $g_{j,n}(t)$ considers the previously computed panning matrix $m_{i,j}$ and the learned basis $b_{j,n}(f)$ from the training stage. Consequently, we have the following update rules:

$$g_{j,n}(t) \longleftarrow g_{j,n}(t) \frac{\sum_{f,i} m_{i,j} b_{j,n}(f)\, x_i(f,t)\, \hat{x}_i(f,t)^{\beta-2}}{\sum_{f,i} m_{i,j} b_{j,n}(f)\, \hat{x}(f,t)^{\beta-1}}. \tag{6}$$

3.5. From the Estimated Gains to the Separated Signals.

The reconstruction of the sources is done by estimating the complex amplitude for each time-frequency bin. In the case of binary separation, a cell is entirely associated with a single source. However, when having many instruments as in orchestral music, it is more advantageous to redistribute

```
(1) Initialize $b_{j,n}(f)$ with the values learned in Section 3.3.
(2) Initialize the gains $g_{j,n}(t)$ with score information.
(3) Initialize the mixing matrix $m_{i,j}$ with the values learned in Section 3.1.
(4) Update the gains using equation (6).
(5) Repeat Step (2) until the algorithm converges (or maximum number of iterations is reached).
```

Algorithm 1: Gain estimation method.

energy proportionally over all sources as in the Wiener filtering method [23].

This model allows for estimating each separated source $s_j(t)$ from mixture $x_i(t)$ using a generalized time-frequency Wiener filter over the short-time Fourier transform (STFT) domain as in [3, 34].

Let $\alpha_{j'}$ be the Wiener filter of source j', representing the relative energy contribution of the predominant source with respect to the energy of the multichannel mixed signal $x_i(t)$ at channel i:

$$\alpha_{j'}(t,f) = \frac{\left|A_{i,j'}\right|^2 \left|\underline{s}_j(f,t)\right|^2}{\sum_j \left|A_{i,j}\right|^2 \left|\underline{s}_j(f,t)\right|^2}. \tag{7}$$

Then, the corresponding spectrogram of source j' is estimated as

$$\hat{\underline{s}}_{j'}(f,t) = \frac{\alpha_{j'}(t,f)}{\left|A_{i,j'}\right|^2} \underline{x}_i(f,t). \tag{8}$$

The estimated source $\hat{s}_{j'}(f,t)$ is computed with the inverse overlap-add STFT of $\hat{\underline{s}}_{j'}(f,t)$.

The estimated source magnitude spectrogram is computed using the gains $g_{j,n}(t)$ estimated in Section 3.4 and $b_{j,n}(f)$ the fixed basis functions learned in Section 3.3: $\hat{s}_j(t,f) = g_{n,j}(t)b_{j,n}(f)$. Then, if we replace $s_j(t,f)$ with $\hat{s}_j(t,f)$ and if we consider the mixing matrix coefficients computed in Section 3.1, we can calculate the Wiener mask from (7):

$$\alpha_{j'}(t,f) = \frac{m_{i,j'}^2 \hat{s}_j(f,t)^2}{\sum_j m_{i,j}^2 \hat{s}_j(f,t)^2}. \tag{9}$$

Using (8), we can apply the Wiener mask to the multichannel signal spectrogram, thus obtaining $\hat{s}_{j'}(f,t)$, the estimated predominant source spectrogram. Finally, we use the phase information from the original mixture signal $\underline{x}_i(t)$, and, through inverse overlap-add STFT, we obtain the estimated predominant source $\hat{s}_{j'}(t)$.

4. Gains Initialization with Score Information

In the baseline method [6], the gains are initialized following a transcription stage. In our case, the automatic alignment system yields a score which offers an analogous representation to the one obtained by the transcription. To that extent, the output of the alignment is used to initialize the gains for the NMF based methods for score-informed source separation.

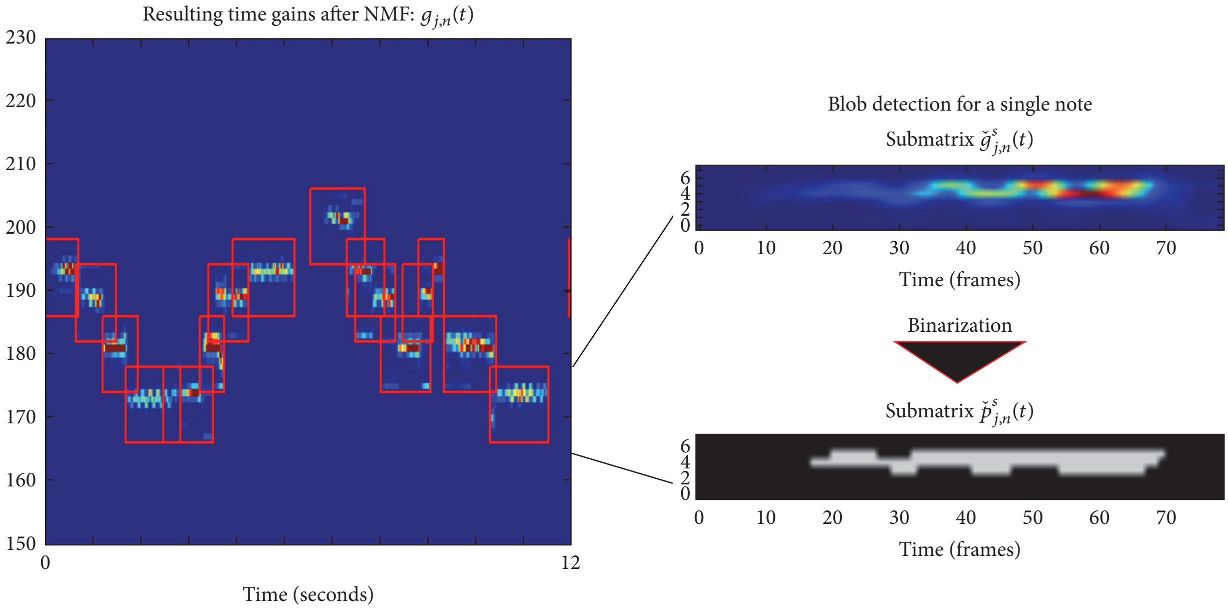
Although the alignment algorithm aims at fixing global misalignments, it does not account for local misalignments. In the case of score-informed source separation, having a better aligned score leads to better separation [11], since it increases the sparseness of the gains matrix by setting to zero the activation for a time frame in which a note is not played (e.g., the corresponding spectral template of the note in the basis matrix is not activated outside this time boundary). However, in a real-case scenario, the initialization of gains derived from the MIDI score must take into account the local misalignments. This has been traditionally done by setting a tolerance window around the onsets and offsets [3, 5, 24] or by refining the gains after a number of NMF iterations and then reestimating the gains [11]. While the former integrates note refinement into the parametric model, the latter detects contours in the gains using image processing heuristics and explicitly associates them with meaningful entities as notes. In this paper, we present two methods for note refinement: in Section 4.1, we detail the method in [11] which is used as a baseline for our framework and, in Section 5.2, we adapt and improve this baseline to the multichannel case.

On these terms, if we account for errors up to d frames in the audio-to-score alignment, we need to increase the time interval around the onset and the offset for a MIDI note when we initialize the gains. Thus, the values in $g_{j,n}(t)$ for instrument j and pitch corresponding to a MIDI note n are set to 1 for the frames where the MIDI note is played, as well as the neighboring d frames. The other values in $g_{j,n}(t)$ are set to 0 and do not change during computation, while the values set to 1 evolve according to the energy distributed between the instruments.

Having initialized the gains, the classical augmented NMF factorization is applied to estimate the gains corresponding to each source j in the mixture. The process is detailed in Algorithm 1.

4.1. Note Refinement. The note refinement method in [11] aims at associating the values in the gains matrix $g_{j,n}(t)$ with notes. Therefore, it is applied after a certain number of iterations of Algorithm 1, when the gains matrix yields a meaningful distribution of the energy between instruments.

The method is applied on each note separately, with the scope of refining the gains associated with the targeted note. The gains matrix $g_{j,n}(t)$ can be understood as a greyscale image with each element in the matrix representing a pixel in the image. A set of image processing heuristics are deployed to detect shapes and contours in this image commonly known



(a)

(b)

Figure 2: After NMF, the resulting gains (a) are split in submatrices (b) and used to detect blobs [11].

as blobs [35, p. 248]. As a result, each blob is associated with a single note, giving the onset and offset times for the note and its frequency contour. This representation further increases the sparsity of the gains $g_{j,n}(t)$, yielding less interference and better separation.

As seen in Figure 2, the method considers an image patch around the pixels corresponding to the pitch of the note and the onset and offset of the note given by the alignment stage, plus the additional d frames accounting for the local misalignments. In fact, the method works with the same submatrices of $g_{j,n}(t)$ which are set to 1 according to their corresponding MIDI note during the gains initialization stage, as explained in Section 3.4. Therefore, for a given note $k = 1, \ldots, K_j$, we process submatrix $\breve{g}^k_{j,n}(t)$ of the gains matrix $g_{j,n}(t)$, where K_j is the total number of notes for instrument j.

The steps of the method in [11], image preprocessing, binarization, and blob selection, are explained in Sections 4.1.2, 4.1.1, and 4.1.3.

4.1.1. Image Preprocessing. The preprocessing stage ensures through smoothing that there are no energy discontinuities within an image patch. Furthermore, it gives more weight to the pixels situated closer to the central bin in the blob in order to eliminate interference from the neighboring notes (close in time and frequency), but still preserving vibratos or transitions between notes.

First, we convolve with a smoothing Gaussian filter [35, p. 86] each row of the submatrix $\breve{g}^k_{j,n}(t)$. We choose a one-dimension Gaussian filter:

$$w(t) = \frac{1}{\sqrt{2\pi}\phi} e^{-(-t^2/2\phi^2)}, \tag{10}$$

where t is the time axis and ϕ is the standard deviation. Thus, each row vector of $\breve{g}^k_{j,n}(t)$ is convolved with $w(t)$, and the result is truncated in order to preserve the dimensions of the initial matrix by removing the mirrored frames.

Second, we penalize values in $\breve{g}^k_{j,n}(t)$ which are further away from the central bin by multiplying each column vector of this matrix with a 1-dimensional Gaussian centered in the central frequency bin, represented by vector $\nu(n)$:

$$\nu(n) = \frac{1}{\sqrt{2\pi}\nu} e^{-(n-\kappa)^2/2\nu^2}, \tag{11}$$

where n is the frequency axis, κ is the position of the central frequency bin, and ν is the standard deviation. The values of the parameters above are given in Section 6.4.2 as a part of the evaluation setup.

4.1.2. Image Binarization. Image binarization sets to zero the elements of the matrix $\breve{g}^k_{j,n}(t)$ which are lower than a threshold and to one the elements larger than the threshold. This involves deriving a submatrix $\breve{p}^k_{j,n}(t)$, associated with note k:

$$\breve{p}^k_{j,n}(t) = \begin{cases} 0, & \text{if } \breve{g}^k_{j,n}(t) < \text{mean}\left(\breve{g}^k_{j,n}(t)\right), \\ 1, & \text{if } \breve{g}^k_{j,n}(t) \geq \text{mean}\left(\breve{g}^k_{j,n}(t)\right). \end{cases} \tag{12}$$

4.1.3. Blob Selection. First, we detect blobs in each binary submatrix $\breve{p}^k_{j,n}(t)$, using the connectivity rules described in [35, p. 248] and [27]. Second, from the detected blob candidates, we determine the best blob for each note in a similar way to [11]. We assign a value to each blob, depending on its area and the overlap with the blobs corresponding to adjacent notes, which will help us penalize the overlap between blobs of adjacent notes.

As a first step, we penalize parts of the blobs which overlap in time with other blobs from different notes $k-1, k, k+1$. This is done by weighting each element in $\breve{g}_{j,n}^{k}(t)$ with factor γ, depending on the amount of overlapping with blobs from adjacent notes. The resulting score matrix has the following expression:

$$\breve{q}_{j,n}^{k}(t) = \begin{cases} \gamma * \breve{g}_{j,n}^{k}(t), & \text{if } \breve{p}_{j,n}^{k}(t) \wedge \breve{p}_{j,n}^{k-1}(t) = 1, \\ \gamma * \breve{g}_{j,n}^{k}(t), & \text{if } \breve{p}_{j,n}^{k}(t) \wedge \breve{p}_{j,n}^{k+1}(t) = 1, \\ \breve{g}_{j,n}^{k}(t), & \text{otherwise,} \end{cases} \quad (13)$$

where γ is a value in the interval $[0, 1]$.

Then, we compute a score for each note l and for each blob associated with the note, by summing up the elements in the score matrix $\breve{q}_{j,n}^{k}(t)$ which are considered to be part of a blob. The best blob candidate is the one with the highest score and further on; it is associated with the note, its boundaries giving the note onset and offsets.

4.2. Gains Reinitialization and Recomputation. Having associated a blob with each note (Section 4.1.3), we discard obtrusive energy from the gains matrix $g_{j,n}(t)$, by eliminating the pixels corresponding to the blobs which were not selected, making the matrix sparser. Furthermore, the energy excluded from instrument's gains is redistributed to other instruments, contributing to better source separation. Thus, the gains are reinitialized with the information obtained from the corresponding blobs and we can repeat the factorization Algorithm 1 to recompute $g_{j,n}(t)$. Note that the energy which is excluded by note refinement is set to zero $g_{j,n}(t)$ and will remain zero during the factorization.

In order to refine the gains $g_{j,n}(t)$, we define a set of matrices $p_{j,n}^{k}(t)$ derived from the matrices corresponding to the best blobs $\breve{p}_{j,n}^{k}(t)$ which contain 1 only for the elements associated with the best blob and 0 otherwise. We rebuild the gains matrix $g_{j,n}(t)$ with the set of submatrices $p_{j,n}^{k}(t)$. For the corresponding bins n and time frames t of note k, we initialize the values in $g_{j,n}(t)$ with the values in $p_{j,n}^{k}(t)$. Then, we reiterate the gains estimation with Algorithm 1. Furthermore, we obtain the spectrogram of the separated sources with the method described in Section 3.5.

5. PARAFAC Model for Multichannel Gains Estimation

Parallel factor analysis methods (PARAFAC) [28, 29] are mostly used under the nonnegative tensor factorization paradigm. By these means, the NMF model is extended to work with 3-valence tensors, where each slice of the sensor represents the spectrogram for a channel. Another approach is to stack up spectrograms for each channel in a single matrix [36] and perform a joint estimation of the spectrograms of sources in all channels. Hence, we extend the NMF model in Section 3 to jointly estimate the gains matrices in all the channels.

5.1. Multichannel Gains Estimation. The algorithm described in Section 3 estimates gains $g_{n,j}$ for source j with respect to single channel i determined as the corresponding row in column j of the panning matrix where element $m_{i,j}$ has the maximum value. However, we argue that a better estimation can benefit from the information in all channels. To this extent, we can further include update rules for other parameters such as mixing matrix $m_{i,j}$ which were otherwise kept fixed in Section 3, because the factorization algorithm estimates the parameters jointly for all the channels.

We propose to integrate information from all channels by concatenating their corresponding spectrogram matrices on the time axis, as in

$$x(f,t) = \begin{bmatrix} x_1(f,t) & x_1(f,t) & \cdots & x_I(f,t) \end{bmatrix}. \quad (14)$$

We are interested in jointly estimating the gains $g_{n,j}$ of the source j in all the channels. Consequently, we concatenate in time the gains corresponding to each channel i for $i = 1, \ldots, I$, where I is the total number of channels, as seen in (15). The new gains are initialized with identical score information obtained from the alignment stage. However, during the estimation of the gains for channel i, the new gains $g_{n,j}^{i}(t)$ evolve accordingly, taking into account the corresponding spectrogram $x_i(f,t)$. Moreover, during the gains refinement stage, each gain is refined separately with respect to each channel:

$$g_{n,j}(t) = \begin{bmatrix} g_{n,j}^{1}(t) & g_{n,j}^{2}(t) & \cdots & g_{n,j}^{I}(t) \end{bmatrix}. \quad (15)$$

In (5), we describe the factorization model for the estimated spectrogram, considering the mixing matrix, the basis, and the gains. Since we estimate a set of I gains for each source $j = 1, \ldots, J$, this will result in J estimations of the spectrograms corresponding to all the channels $i = 1, \ldots, I$, as seen in

$$\hat{x}_i^{j}(f,t) = \sum_{j=1}^{J} m_{i,j} \sum_{n=1}^{N} g_{j,n}^{i}(t) \sum_{h=1}^{H} a_{j,n}(h)\, G(f - h f_0(n)). \quad (16)$$

Each iteration of the factorization algorithm yields additional information regarding the distribution of energy between each instrument and each channel. Therefore, we can include in the factorization update rules for mixing matrix $m_{i,j}$ as in (17). By updating the mixing parameters at each factorization step, we can obtain a better estimation for $\hat{x}_i^{j}(f,t)$:

$$m_{i,j} \longleftarrow m_{i,j} \frac{\sum_{f,t} b_{j,n}(f)\, g_{n,j}(t)\, x_i(f,t)\, \hat{x}_i(f,t)^{\beta-2}}{\sum_{f,t} b_{j,n}(f)\, g_{n,j}(t)\, \hat{x}(f,t)^{\beta-1}}. \quad (17)$$

Considering the above, the new rules to estimate the parameters are described in Algorithm 2.

```
(1) Initialize $b_{j,n}(f)$ with the values learned in Section 3.3.
(2) Initialize the gains $g_{j,n}(t)$ with score information.
(3) Initialize the panning matrix $m_{i,j}$ with the values learned in Section 3.1.
(4) Update the gains using equation (6).
(5) Update the panning matrix using equation (17).
(6) Repeat Step (2) until the algorithm converges (or maximum number of iterations is reached).
```

Algorithm 2: Gain estimation method.

Note that the current model does not estimate the phases for each channel. In order to reconstruct source j, the model in Section 3 uses the phase of the signal corresponding to channel i where it has the maximum value in the panning matrix, as described in Section 3.5. Thus, in order to reconstruct the original signals, we can solely rely on the gains estimated in single channel i, in a similar way to the baseline method.

5.2. Multichannel Gains Refinement. As presented in Section 5.1, for a given source, we obtain an estimation of the gains corresponding to each channel. Therefore, we can apply note refinement heuristics in a similar manner to Section 4.1 for each of the gains $[g^1_{n,j}(t), \ldots, g^I_{n,j}(t)]$. Then, we can average out the estimations for all the channel, making the blob detection more robust to the variances between the channels:

$$g'_{n,j}(t) = \frac{\sum_{i=1}^{I} g^i_{n,j}(t)}{I}. \qquad (18)$$

Having computed the mean over all channels as in (18), for each note $k = 1, \ldots, K_j$, we process submatrix $\overline{g}^k_{j,n}(t)$ of the new gains matrix $g'_{j,n}(t)$, where K_j is the total number of notes for an instrument j. Specifically, we apply the same steps: preprocessing (Section 4.1.1), binarization (Section 4.1.2), and blob selection (Section 4.1.3), to each matrix $\overline{g}^k_{j,n}(t)$ and we obtain a binary matrix $\overline{p}^k_{j,n}(t)$ having 1 s for the elements corresponding to the best blob and 0 s for the rest.

Our hypothesis is that averaging out the gains between all channels makes blob detection more robust. However, when performing the averaging, we do not account for the delays between the channels. In order to compute the delay for a given channel, we can compute the best blob separately with the method in Section 4.1 (matrix $\check{p}^k_{j,n}(t)$) and compare it with the one calculated with the averaged estimation ($\overline{p}^k_{j,n}(t)$). This step is equivalent to comparing the onset times of the two best blobs for the two estimations. Subtracting these onset times, we get the delay between the averaged estimation and the one obtained for a channel and we can correct this in matrix $\overline{p}^k_{j,n}(t)$. Accordingly, we zero-pad the beginning of $\overline{p}^k_{j,n}(t)$ with the amount of zeros corresponding to the delay, or we remove the trailing zeros for a negative delay.

6. Materials and Evaluation

6.1. Dataset. The audio material used for evaluation was presented by Pätynen et al. [30] and consists of four passages of symphonic music from the Classical and Romantic periods. This work presented a set of anechoic recordings for each of the instruments, which were then synchronized between them so that they could later be combined to a mix of the orchestra. Musicians played in an anechoic chamber, and, in order to be synchronous with the rest of the instruments, they followed a video featuring a conductor and a pianist playing each of the four pieces. Note that the benefits of having isolated recordings comes at the expense of ignoring the interactions between musicians which commonly affect intonation and time-synchronization [37].

The four pieces differ in terms of number of instruments per instrument class, style, dynamics, and size. The first passage is a soprano aria of Donna Elvira from the opera Don Giovanni by W. A. Mozart (1756–1791), corresponding to the Classical period, and traditionally played by a small group of musicians. The second passage is from L. van Beethoven's (1770–1827) Symphony no. 7, featuring big chords and string crescendo. The chords and pauses make the reverberation tail of a concert hall clearly audible. The third passage is from Bruckner's (1824–1896) Symphony no. 8, and represents the late Romantic period. It features large dynamics and size of the orchestra. Finally, G. Mahler's Symphony no. 1, also featuring a large orchestra, is another example of late romanticism. The piece has a more complex texture than the one by Bruckner. Furthermore, according to the musicians which recorded the dataset, the last two pieces were also more difficult to play and record [30].

In order to keep the evaluation setup consistent between the four pieces, we focus in the following instruments: violin, viola, cello, double bass, oboe, flute, clarinet, horn, trumpet, and bassoon. All tracks from a single instrument were joined into a single track for each of the pieces.

For the selected instruments, we list the differences between the four pieces in Table 1. Note that in the original dataset the violins are separated into two groups. However, for brevity of evaluation and because in our separation framework we do not consider sources sharing the same instrument templates, we decided to merge the violins into a single group. Note that the pieces by Mahler and Bruckner have a divisi in the groups of violins, which implies a larger number of instruments playing different melody lines simultaneously. This results in a scenario which is more challenging for source separation.

We created a ground truth score, by hand annotating the notes played by the instruments. In order to facilitate this process, we first gathered the scores in MIDI format and automatically computed a global audio-score alignment, using the method from [13] which has won the MIREX

Table 1: Anechoic dataset [30] characteristics.

Piece	Duration	Period	Instrument sections	Number of tracks	Max. tracks/instrument
Mozart	3 min 47 s	Classical	8	10	2
Beethoven	3 min 11 s	Classical	10	20	4
Mahler	2 min 12 s	Romantic	10	30	4
Bruckner	1 min 27 s	Romantic	10	39	12

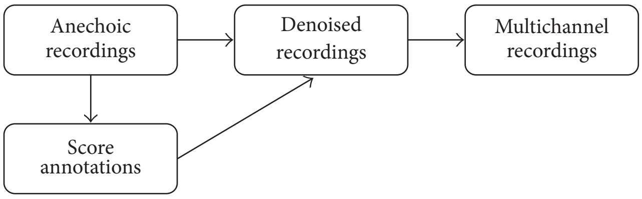


Figure 3: The steps to create the multichannel recordings dataset.

score-following challenge for the past years. Then, we locally aligned the notes of each instrument by manually correcting the onsets and offsets to fit the audio. This was performed using Sonic Visualiser, with the guidance of the spectrogram and the monophonic pitch estimation [38] computed for each of the isolated instruments. The annotation was performed by two of the authors, which cross-checked the work of their peer. Note that this dataset and the proposed annotation are useful not only for our particular task but also for the evaluation of multiple pitch estimation and automatic transcription algorithms in large orchestral settings, a context which has not been considered so far in the literature. The annotations can be found at the associated page (http://mtg.upf.edu/download/datasets/phenicx-anechoic).

During the recording process detailed in [30], the gain of the microphone amplifiers was fixed to the same value for the whole process, which reduced the dynamic range of the recordings of the quieter instruments. This led to noisier recordings for most of the instruments. In Section 6.2, we describe the score-informed denoising procedure we applied to each track. From the denoised isolated recordings, we then used Roomsim to create a multichannel image, as detailed in Section 6.3. The steps necessary to pass from the anechoic recordings to the multichannel dataset are represented in Figure 3. The original files can be obtained from the Acoustic Group at Aalto Univeristy (http://research.cs.aalto.fi/acoustics/). For the denoising algorithm, please refer to http://mtg.upf.edu/download/datasets/phenicx-anechoic.

6.2. Dataset Denoising. The noise related problems in the dataset were presented in [30]. We remove the noise in the recordings with the score-informed method in [39], which relies on learned noise spectral patters. The main difference is that we rely on a manually annotated score, while in [39] the score is assumed to be misaligned, so further regularization is included to ensure that only certain note combinations in the score occur.

The annotated score yields the time interval where an instrument is not playing. Thus, the noise pattern is learned only within that interval. In this way, the method assures that the desired noise, which is a part of the actual sound of the instrument, is preserved in the denoised recording.

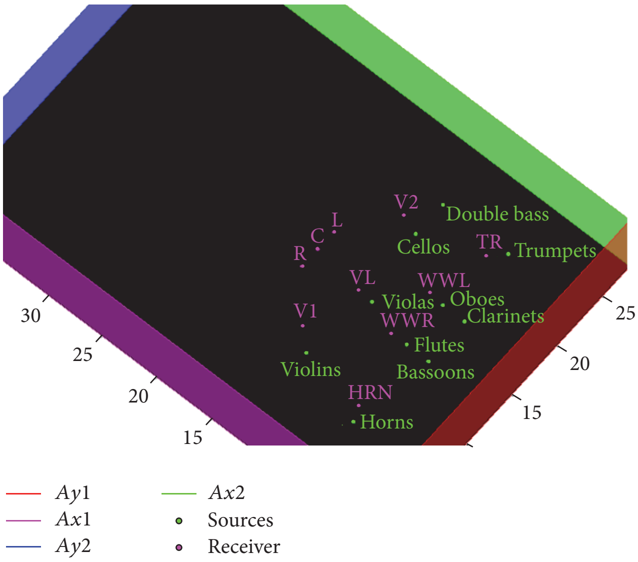


Figure 4: The sources and the receivers (microphones in the simulated room).

The algorithm takes each anechoic recording of a given instrument and removes the noise for the time interval where an instrument is playing, while setting to zero the frames where an instrument is not playing.

6.3. Dataset Spatialization. To simulate a large reverberant hall, we use the software Roomsim [31]. We define a configuration file which specifies the characteristics of the hall and, for each of the microphones, their position relative to each of the sources. The simulated room has similar dimensions to the Royal Concertgebouw in Amsterdam, one of the partners in the PHENICX project, and represents a setup in which we tested our framework. The simulated room's width, length, and height are 28 m, 40 m, and 12 m. The absorption coefficients are specified in Table 2.

The positions of the sources and microphones in the room are common for orchestral concerts (Figure 4). A configuration file is created for each microphone which contains its coordinates (e.g., $(14, 17, 4)$ for the center microphone). Then, each source is defined through polar coordinates relative to the microphone (e.g., $(11.4455, -95.1944, -15.1954)$ radius, azimuth, and elevation for the bassoon relative to the center microphone). We selected all the microphones to be cardioid, in order to match the realistic setup of Concertgebouw Hall.

Table 2: Room surface absorption coefficients.

Standard measurement frequencies (Hz)	125	250	500	1000	2000	4000
Absorption of wall in $x = 0$ plane	0.4	0.3	0.3	0.3	0.2	0.1
Absorption of wall in $x = Lx$ plane	0.4	0.45	0.35	0.35	0.45	0.3
Absorption of wall in $y = 0$ plane	0.4	0.45	0.35	0.35	0.45	0.3
Absorption of wall in $y = Ly$ plane	0.4	0.45	0.35	0.35	0.45	0.3
Absorption of floor, that is, $z = 0$ plane	0.5	0.6	0.7	0.8	0.8	0.9
Absorption of ceiling, that is, $z = Lz$ plane	0.4	0.45	0.35	0.35	0.45	0.3

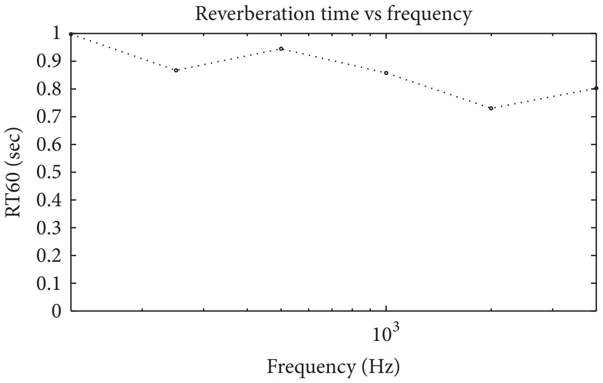


Figure 5: The reverberation time versus frequency for the simulated room.

Using the configuration file and the anechoic audio files corresponding to the isolated sources, Roomsim generates the audio files for each of the microphones along with the impulse responses for each pair of instruments and microphones. The impulse responses and the anechoic signals are used during the evaluation to obtain the ground truth spatial image of the sources in the corresponding microphone. Additionally, we plot Roomsim the reverberation time RT60 [31] across the frequencies in Figure 5.

We need to adapt ground truth annotations to the audio generated with Roomsim, as the original annotations were done on the isolated audio files. Roomsim creates an audio for a given microphone by convolving each of the sources with the corresponding impulse response and then summing up the results of the convolution. We compute a delay for each pair of microphones and instruments by taking the position of the maximum value in the associated impulse response vector. Then, we generate a score for each of the pairs by adding the corresponding delay to the note onsets. Additionally, since the offset time depends on the reverberation and the frequencies of the notes, we add 0.8 s to each note offset to account for the reverberation, besides the added delay.

6.4. Evaluation Methodology

6.4.1. Parameter Selection. In this paper, we use a low-level spectral representation of the audio data which is generated from a windowed FFT of the signal. We use a Hanning window with the size of 92 ms and a hop size of 11 ms. Here, a logarithmic frequency discretization is adopted. Furthermore, two time-frequency resolutions are used. First, to estimate the instrument models and the panning matrix, a single semitone resolution is proposed. In particular, we implement the time-frequency representation by integrating the STFT bins corresponding to the same semitone. Second, for the separation task, a higher resolution of 1/4 of semitone is used, which has proven to achieve better separation results [6]. The time-frequency representation is obtained by integrating the STFT bins corresponding to 1/4 of semitone. Note that in the separation stage, the learned basis functions $b_{j,n}(f)$ are adapted to the 1/4 of semitone resolution by replicating 4 times the basis at each semitone to the 4 samples of the 1/4 of semitone resolution that belong to this semitone. For image binarization, we pick for the first Gaussian $\phi = 3$ as the standard deviation and for the second Gaussian $\kappa = 4$ as the position of the central frequency bin and $\nu = 4$ as the standard deviation, corresponding to one semitone.

We picked 10 iterations for the NMF, and we set the beta-divergence distortion, $\beta = 1.3$, as in [6, 11].

6.4.2. Evaluation Setup. We perform three different kind of evaluations: audio-to-score alignment, panning matrix estimation, and score-informed source separation. Note that, for the alignment, we evaluate the state-of-the-art system in [13]. This method does not align notes but combinations of notes in the score (a.k.a states). Here, the alignment is performed with respect to a single audio channel, corresponding to the microphone situated in the center of the stage. On the other hand, the offsets are estimated by shifting the original duration for each note in the score [13] or by assigning the offset time as the onset for the next state. We denote these two cases as INT or NEX.

Regarding the initialization of the separation framework, we can use the raw output of the alignment system. However, as stated in Section 4 and [3, 5, 24], a better option is to extend the onsets and offsets along a tolerance window to account for the errors of the alignment system and for the delays between center channel (on which the alignment is performed) and the other channels and for the possible errors in the alignment itself. Thus, we test two hypotheses regarding the tolerance window for the possible errors. In the first case, we extend the boundaries with 0.3 s for onsets and 0.6 s for offsets (T1) and in the second with 0.6 s for onsets and 1 s for offsets (T2). Note that the value for the onset times of 0.3 s is not arbitrary but the usual threshold for onsets in the score-following in

Table 3: Score information used for the initialization of score-informed source separation.

Tolerance window size	Offset estimation
T1: onsets, 0.3 s; offsets, 0.6 s	INT: interpolation of the offset time
T2: onsets, 0.6 s; offsets, 0.9 s	NEX: the offset is the onset of the next note

MIREX evaluation of real-time score-following [40]. Two different tolerance windows were tested to account for the complexity of this novel scenario. The tolerance window is slightly larger for offsets due to the reverberation time and because the ending of the note is not as clear as its onset. A summary of the score information used to initialize the source separation framework is found in Table 3.

We label the test case corresponding to the initialization with the raw output of the alignment system as Ali. Conversely, the test case corresponding to the tolerance window initialization is labeled as Ext. Furthermore, within the tolerance window, we can refine the note onsets and offsets with the methods in Section 4.1 (Ref1) and Section 5.2 (Ref2), resulting in other two test cases. Since method Ref1 can only refine the score to a single channel, the results are solely computed with respect to that channel. For the multichannel refinement Ref2, we report the results of the alignment of each instrument with respect to each microphone. A graphic of the initialization of the framework with the four test cases listed above (Ali, Ext, Ref1, and Ref2), along with the ground truth score initialization (GT), is found in Figure 7, where we present the results for these cases in terms of source separation.

In order to evaluate the panning matrix estimation stage, we compute an ideal panning matrix based on the impulse responses generated by Roomsim during the creation of the multichannel audio (see Section 6.3). The ideal panning matrix gives the ideal contribution of each instrument in each channel and it is computed by searching the maximum in the impulse response vector corresponding to each instrument-channel pair, as in

$$m_{\text{ideal}}(i,j) = \max(\text{IR}(i,j)(t)), \tag{19}$$

where $\text{IR}(i,j)(t)$ is the impulse response of source i in channel j. By comparing the estimated matrix $m(i,j)$ with the ideal one $m_{\text{ideal}}(i,j)$, we can determine if the algorithm picked a wrong channel for separation.

6.4.3. Evaluation Metrics. For score alignment, we are interested in a measure which relates to source separation and accounts for the audio frames which are correctly detected, rather than an alignment rate computed per note onset, as found in [13]. Thus, we evaluate the alignment at the frame level rather than at a note level. A similar reasoning on the evaluation of score alignment is found in [4].

We consider 0.011 s the temporal granularity for this measure and the size of a frame. Then, a frame of a musical note is considered a true positive (tp) if it is found in the ground truth score and in the aligned score in the exact time boundaries. The same frame is labeled as a false positive (fp) if it is found only in the aligned score and a false negative (fn) if it is found only in the ground truth score. Since the gains initialization is done with score information (see Section 4), lost frames (recall), and incorrectly detected frames (precision) impact the performance of the source separation algorithm, precision is defined as $p = \text{tp}/(\text{tp}+\text{fp})$ and recall as $r = \text{tp}/(\text{tp}+\text{fn})$. Additionally, we compute the harmonic mean of precision and recall to obtain F-measure as $F = 2 \cdot ((p \cdot r)/(p+r))$.

The source separation evaluation framework and metrics employed are described in [41, 42]. Correspondingly, we use *Source to Distortion Ratio* (SDR), *Source to Interference Ratio* (SIR), and *Source to Artifacts Ratio* (SAR). While SDR measures the overall quality of the separation and ISR the spatial reconstruction of the source, SIR is related to rejection of the interferences and SAR to the absence of forbidden distortions and artifacts.

The evaluation of source separation is a computationally intensive process. Additionally, to process the long audio files in the dataset would require large memory to perform the matrix calculations. To reduce the memory requirements, the evaluation is performed for blocks of 30 s with 1 s overlap to allow for continuation.

6.5. Results

6.5.1. Score Alignment. We evaluate the output of the alignment Ali, along the estimation of the note offsets: INT and NEX in terms of F-measure (see Section 6.4.3), precision, and recall, ranging from 0 to 1. Additionally, we evaluate the optimal size for extending the note boundaries along the onsets and offsets, T1 and T2, for the refinement methods, Ref1 and Ref2, and the baseline, Ext. Since there are lots of differences between pieces, we report the results individually per song in Table 4.

Methods Ref1 and Ref2 depend on a binarization threshold which determines how much energy is set to zero. A lower threshold will result in the consolidation of larger blobs in blob detection. In [11], this threshold is set to 0.5 for a dataset of monaural recordings of Bach chorales played by four instruments. However, we are facing a multichannel scenario where capturing the reverberation is important, especially when we consider that offsets were annotated with a low energy threshold. Thus, we are interested in losing the least energy possible and we set lower values for the threshold: 0.3 and 0.1. Consequently, when analyzing the results, a lower threshold achieves better performance in terms of F-measure for Ref1 (0.67 and 0.72) and for Ref2 (0.71 and 0.75).

According to Table 4, extending note offsets (NEX), rather than interpolating them (INT), gives lower recall in all pieces, and the method leads to losing more frames which cannot be recovered even by extending the offset times in T2: NEX T2 yields always a lower recall when compared to INT T2 (e.g., $r = 0.85$ compared to $r = 0.97$ for Mozart).

The output of the alignment system Ali is not a good option to initialize the gains of source separation system. It has a high precision and a very low recall (e.g., the case INT Ali has $p = 0.95$ and $r = 0.39$ compared to case INT Ext

Table 4: Alignment evaluated in terms of *F*-measure, precision, and recall, ranging from 0 to 1.

			Mozart			Beethoven			Mahler			Bruckner		
			F	*p*	*r*	*F*	*p*	*r*	*F*	*p*	*r*	*F*	*p*	*r*
INT		Ali	0.69	0.93	0.55	0.56	0.95	0.39	0.61	0.85	0.47	0.60	0.94	0.32
	T1	Ref1	0.77	0.77	0.76	0.79	0.81	0.77	0.63	0.74	0.54	0.72	0.73	0.70
		Ref2	**0.83**	0.79	0.88	0.82	0.82	0.81	0.67	0.77	0.60	0.81	0.77	0.85
		Ext	0.82	0.73	0.94	**0.84**	0.84	0.84	**0.77**	0.69	0.87	**0.86**	0.78	0.96
	T2	Ref1	0.77	0.77	0.76	0.76	0.75	0.77	0.63	0.74	0.54	0.72	0.70	0.71
		Ref2	**0.83**	0.78	0.88	**0.82**	0.76	0.87	0.67	0.77	0.59	**0.79**	0.73	0.86
		Ext	0.72	0.57	0.97	0.79	0.70	0.92	**0.69**	0.55	0.93	0.77	0.63	0.98
NEX		Ali	0.49	0.94	0.33	0.51	0.89	0.36	0.42	0.90	0.27	0.48	0.96	0.44
	T1	Ref1	0.70	0.77	0.64	0.72	0.79	0.66	0.63	0.74	0.54	0.68	0.72	0.64
		Ref2	**0.73**	0.79	0.68	**0.71**	0.79	0.65	0.66	0.77	0.58	0.73	0.74	0.72
		Ext	**0.73**	0.77	0.68	**0.71**	0.80	0.65	**0.69**	0.75	0.64	**0.76**	0.79	0.72
	T2	Ref1	0.74	0.78	0.71	0.72	0.74	0.70	0.63	0.74	0.54	0.72	0.79	0.72
		Ref2	**0.80**	0.80	0.80	**0.75**	0.75	0.75	0.67	0.77	0.59	**0.79**	0.73	0.86
		Ext	0.73	0.65	0.85	0.73	0.69	0.77	**0.72**	0.64	0.82	**0.79**	0.69	0.91

Table 5: Instruments for which the closest microphone was incorrectly determined for different score information (GT, Ali, T1, T2, INT, and NEX) and two room setups.

	GT	INT			NEX		
		Ali	T1	T2	Ali	T1	T2
Setup 1	Clarinet	Clarinet, double bass	Clarinet, flute	Clarinet, flute, horn	Clarinet, double bass		Clarinet, flute
Setup 2	Cello	Cello, flute	Cello, flute	Cello, flute	Bassoon	Flute	Cello, flute

which has $p = 0.84$ and $r = 0.84$ for Beethoven). For the case of Beethoven, the output is particularly poor compared to other pieces. However, by extending the boundaries (Ext) and applying note refinement (Ref1 or Ref2), we are able to increase the recall and match the performance on the other pieces.

When comparing the size for the tolerance window for the onsets and offsets, we observe that the alignment is more accurate with detecting the onsets within 0.3 s and offsets within 0.6 s. In Table 4, T1 achieves better results than T2 (e.g., $F = 0.77$ for T1 compared to $F = 0.69$ for T2, Mahler). Relying on a large window retrieves more frames but also significantly damages the precision. However, when considering source separation we might want to lose as less information as possible. It is in this special case that the refinement methods Ref1 and Ref2 show their importance. When facing larger time boundaries as T2, Ref1 and especially Ref2 are able to reduce the errors by achieving better precision with the minimum amount of loss in recall.

The refinement Ref1 has a worse performance than Ref2, the multichannel refinement (e.g., $F = 0.72$ compared to $F = 0.81$ for Bruckner, INT T1). Note that, in the original version [11], Ref1 was assuming monophony within a source as it was tested in the simple case of Bach10 dataset [4]. To that extent, it was relying on a graph computation to determine the best distribution of blobs. However, due to the increased polyphony within an instrument (e.g., violins playing divisi), with simultaneous melodic lines, we disabled this feature and in this case Ref1 has lower recall, it loses more frames. On the other hand, Ref2 is more robust because it computes a blob estimation per channel. Averaging out these estimations yields better results.

The refinement works worse for more complex pieces (Mahler and Bruckner) than for simple pieces (Mozart and Beethoven). Increasing the polyphony within a source and the number of instruments, having many interleaving melodic lines, a less sparse score, also makes the task more difficult.

6.5.2. Panning Matrix. Estimating correctly the panning matrix is an important step in the proposed method, since Wiener filtering is performed on the channel where the instrument has the most energy. If the algorithm picks a different channel for this step, in the separated audio files we can find more interference between instruments.

As described in Section 3.1, the estimation of the panning matrix depends on the number of nonoverlapping partials of the notes found in the score and their alignment with the audio. To that extent, the more nonoverlapping partials we have, the more robust the estimation.

Initially, we experimented with computing the panning matrix separately for each piece. In the case of Bruckner the piece is simply too short, and there are few nonoverlapping partials to yield a good estimation, resulting in errors in the panning matrix. Since the instrument setup is the same for Bruckner, Beethoven, and Mahler pieces (10 sources in the same position on the stage), we decided to jointly estimate the matrix for the concatenated audio pieces and the associated scores. We denote Setup 1 as the Mozart piece played by 8 sources and Setup 2 the Beethoven, Mahler, and Bruckner pieces played by 10 sources.

Since the panning matrix is computed using the score, different score information can yield very different estimation of the panning matrix. To that extent, we evaluate the influence of audio-to-score alignment, namely, the cases INT, NEX, Ali, T1, and T2, and the initialization with the ground truth score information, GT.

In Table 5, we list the instruments for which the algorithm picked the wrong channel. Note that, in the room setup generated with Roomsim, most of the instruments in Table 5 are placed close to other sources from the same family of instruments: for example, cello and double bass, flute with clarinet, bassoon, and oboe. In this case, the algorithm makes more mistakes when selecting the correct channel to perform source separation.

In the column GT of Table 5, we can see that having a perfectly aligned score yields less errors when estimating the panning matrix. Conversely, in a real-life scenario, we cannot rely on hand annotated score. In this case, for all columns of the Table 5 excluding GT, the best estimation is obtained by the combination of NEX and T1: taking the offset time as the onsets of the next note and then extending the score with a smaller window.

Furthermore, we compute the SDR values for the instruments in Table 5, column GT (clarinet and cello), if the separation was to be done in the correct channel or in the estimated channel. For Setup 1, the channel for clarinet is wrongly mistaken to WWL (woodwinds left), the correct one being WWR (woodwinds right), when we have a perfectly aligned score (GT). However, the microphones WWL and WWR are very close (see Figure 4), and they do not capture significant energy from other instrument sections and the SDR difference is less than 0.01 dB. However, in Setup 2, the cello is wrongly separated in the WWL channel, and the SDR difference between this audio and the audio separated in the correct channel is around −11 dB for each of the three pieces.

6.5.3. Source Separation. We use the evaluation metrics described in Section 6.4.3. Since there is a lot of variability

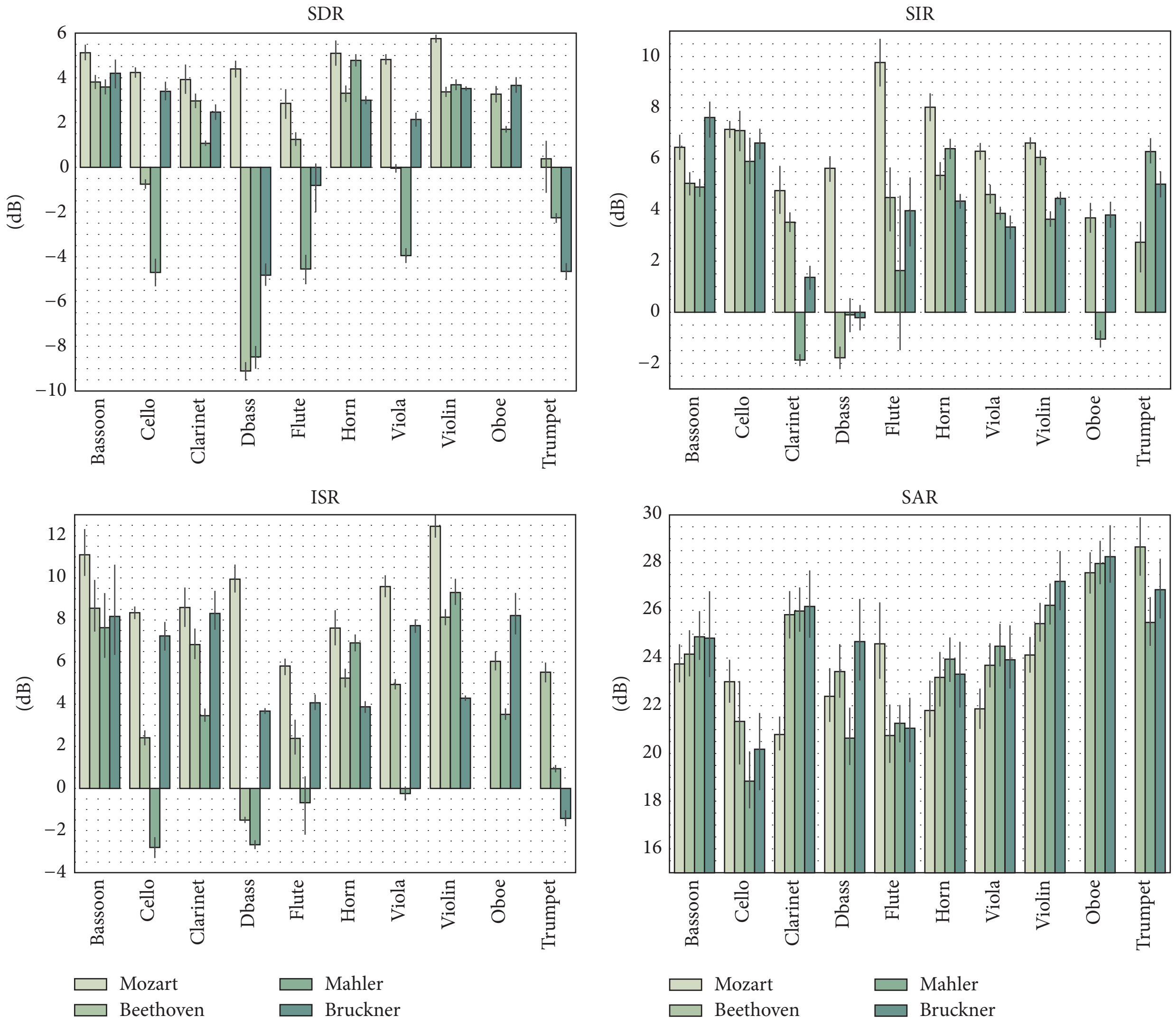


Figure 6: Results in terms of SDR, SIR, SAR, and ISR for the instruments and the songs in the dataset.

between the four pieces, it is more informative to present the results per piece rather than aggregating them.

First, we analyze the separation results per instruments in an ideal case. We assume that the best results for score-informed source separation are obtained in the case of a perfectly aligned score (GT). Furthermore, for this case, we calculate the separation in the correct channel for all the instruments, since, in Section 6.5.2, we could see that picking a wrong channel could be detrimental. We present the results as a bar plot in Figure 6.

As described in Section 6.1 and Table 1, the four pieces had different levels of complexity. In Figure 6, we can see that the more complex the piece is, the more difficult it is to achieve a good separation. For instance, note that cello, clarinet, flute, and double bass achieve good results in terms of SDR on Mozart piece but significantly worse results on other three pieces (e.g., 4.5 dB for cello in Mozart, compared to −5 dB in Mahler). Cello and double bass are close by in both of the setups, similarly for clarinet and flute, and we expect interference between them. Furthermore, these instruments usually share the same frequency range which can result in additional interference. This is seen in lower SIR values for double bass (5.5 dB SIR for Mozart, but −1.8, −0.1, and −0.2 dB SIR for the others) and flute.

An issue for the separation is the spatial reconstruction, measured by the ISR metric. As seen in (9), when applying the Wiener mask, the multichannel spectrogram is multiplied with the panning matrix. Thus, wrong values in this matrix can yield wrong amplitude values of the resulting signals.

This is the case for trumpet, which is allocated a close microphone in the current setup, and for which we expect a good separation. However, trumpet achieves a poor ISR (5.5, 1, and −1 dB) but has a good separation in terms of SIR and SAR. Similarly, other instruments as cello, double bass, flute, and viola face the same problem, particularly for the piece by Mahler. Therefore, a good estimation of the panning matrix is crucial for a good ISR.

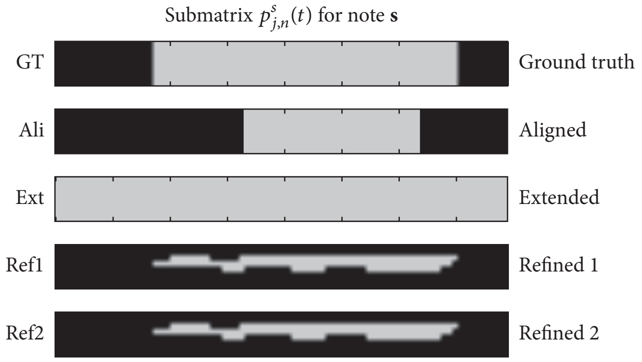


Figure 7: The test cases for initialization of score-informed source separation, for the submatrix $p^s_{j,n}(t)$.

A low SDR is obtained in the case of Mahler that is related to the poor results in alignment obtained for this piece. As seen in Table 4 for INT case, F-measure is almost 8% lower in Mahler than in other pieces, mainly because of the bad precision.

The results are considerably worse for double bass for the more complex pieces of Beethoven (−9.1 dB SDR), Mahler (−8.5 dB SDR), and Bruckner (−4.7 dB SDR), and, for further analysis, we consider it as an outlier, and we exclude it from the analysis.

Second, we want to evaluate the usefulness of note refinement in source separation. As seen in Section 4, the gains for NMF separation are initialized with score information or with a refined score as described in Section 4.2. A summary of the different initialization options and seen in Figure 7. Correspondingly, we evaluate five different initializations of the gains: the perfect initialization with the ground truth annotations (Figure 7 (GT)), the direct output of the score alignment system (Figure 7 (Ali)), the common practice of NMF gains initialization in state-of-the-art score-informed source separation [3, 5, 24] (Figure 7 (Ext)), and the refinement approaches (Figure 7 (Ref1 and Ref2)). Note that Ref1 is the refinement done with the method in Section 4.1 and Ref2 with the multichannel method described in Section 5.2.

We test the difference between the binarization thresholds 0.5 and 0.1, used in the refinement methods Ref1 and Ref2. One-way ANOVA on SDR results gives F-value = 0.0796 and p value = 0.7778, which shows no significant difference between both binarization thresholds.

The results for the five initializations, GT, Ref1, Ref2, Ext, and Ali, are presented in Figure 8, for each of the four pieces. Note that, for Ref1, Ref2, and Ext, we aggregate information across all possible outputs of the alignment: INT, NEX, T1, and T2. Analyzing the results, we note that the more complex the piece, the more difficult to separate between the instruments, the piece by Mahler having the worse results, and the piece by Bruckner a large variance, as seen in the error bars. For these two pieces, other factors as the increased polyphony within a source, the number of instruments (e.g., 12 violins versus 4 violins in a group), and the synchronization issues we described in Section 6.1 can increase the difficulty of separation up to the point that Ref1, Ref2, and Ext have a minimal improvement. To that extent, for the piece by Bruckner, extending the boundaries of the notes (Ext) does not achieve significantly better results than the raw output of the alignment (Ali).

As seen in Figure 8, having a ground truth alignment (GT) helps improving the separation, increasing the SDR with 1–1.5 dB or more for all the test cases. Moreover, the refinement methods Ref1 and Ref2 increase SDR for most of the pieces with the exception of the piece by Mahler. This is due to an increase of SIR and decrease of interferences in the signal. For instance, in the piece by Mozart, Ref1 and Ref2 increase the SDR with 1 dB when compared to Ext. For this piece, the difference in SIR is around 2 dB. Then, for Beethoven, Ref1 and Ref2 increase 0.5 dB in terms of SDR when compared to Ext and 1.5 dB in SIR. For Bruckner, solely Ref2 has a higher SDR; however SIR increases with 1.5 dB in Ref1 and Ref2. Note that not only do Ref1 and Ref2 refine the time boundaries of the notes, but also the refinement happens in frequency, because the initialization is done with the contours of the blobs, as seen in Figure 7. This can also contribute to a higher SIR.

Third, we look at the influence of the estimation of note offsets: INT and NEX, and the tolerance window sizes, T1 and T2, which accounts for errors in the alignment. Note that for this case we do not include the refinement in the results and we evaluate only the case Ext, as we leave out the refinement in order to isolate the influence of T1 and T2. Results are presented in Figure 9 and show that the best results are obtained for the interpolation of the offsets INT. This relates to the results presented in Section 6.5.1. Similarly to the analysis regarding the refinement, the results are worse for the pieces by Mahler and Bruckner, and we are not able to draw a conclusion on which strategy is better for the initialization, as the error bars for the ground truth overlap with the ones of the tested cases.

Fourth, we analyze the difference between the PARAFAC model for multichannel gains estimation as proposed in Section 5.1, compared with the single channel estimation of the gains in Section 3.4. We performed a one-way ANOVA on SDR and we obtain a F-value = 1.712 and a p-value = 0.1908. Hence, there is no significant difference between single channel and multichannel gain estimation, when we are not performing postprocessing of the gains using grain refinement. However, despite the new updates rule do not help, in the multichannel case we are able to better refine the gains. In this case, we aggregate information all over the channels, and blob detection is more robust, even to variations of the binarization threshold. To account for that, for the piece by Bruckner, Ref2 outperforms Ref1 in terms of SDR and SIR. Furthermore, as seen in Table 4 the alignment is always better for Ref2 than Ref1.

The audio excerpts from the dataset used for evaluation, as well as tracks separated with the ground truth annotated score are made available (http://repovizz.upf.edu/phenicx/anechoic_multi/).

7. Applications

7.1. Instrument Emphasis. The first application of our approach for multichannel score-informed source separation

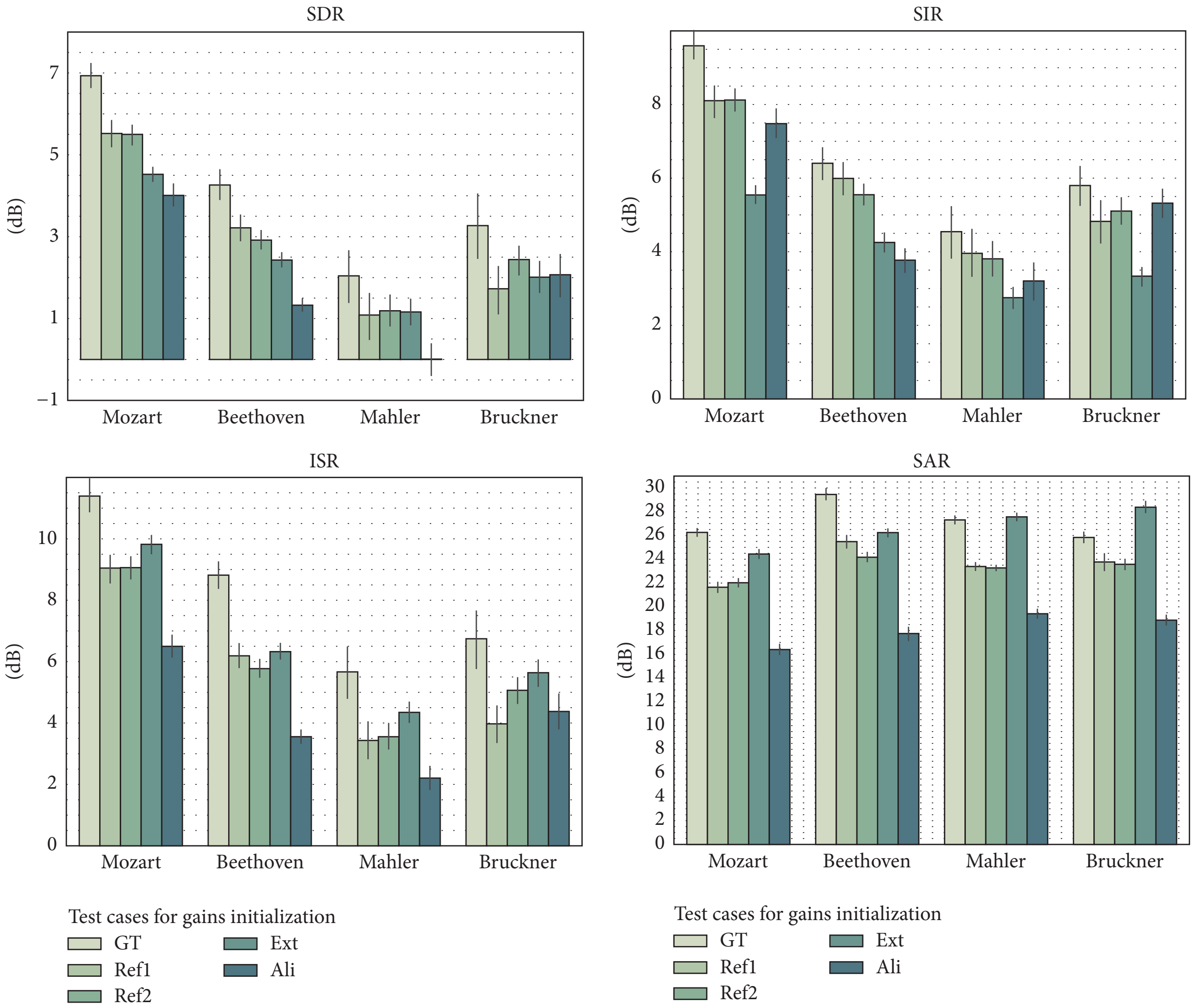


Figure 8: Results in terms of SDR, SIR, AR, and ISR for the NMF gains initialization in different test cases.

is *Instrument Emphasis*, which aims at processing multitrack orchestral recordings. Once we have the separated tracks, it allows emphasizing a particular instrument over the full orchestra downmix recording. Our workflow consists in processing multichannel orchestral recording from which the musical score is available. The multitrack recordings are obtained from a typical on-stage setup in a concert hall, where multiple microphones are placed on stage at certain distance of the sources. The goal is to reduce leakage of other sections, obtaining enhanced signal for the selected instrument.

In terms of system integration, this application has two parts. The front-end is responsible for interacting with the user in the uploading of media content and present the results to the user. The back-end is responsible for managing the audio data workflow between the different signal processing components. We process the audio files in batch estimating the signal decomposition for the full length. For long audio files, as in the case of symphonic recordings, the memory requirements can be too demanding even for a server infrastructure. Therefore, to overcome this limitation, audio files are split into blocks. After the separation has been performed, the blocks associated with each instrument are concatenated, resulting in the separated tracks. The separation quality is not degraded if the blocks have sufficient duration. In our case, we set the block duration to 1 minute. Examples of this application are found online (http://repovizz.upf.edu/phenicx/) and are integrated into the PHENICX prototype (http://phenicx.com/).

7.2. Acoustic Rendering. The second application for the separated material is augmented or virtual reality scenarios, where we apply a spatialization of the separated musical sources. Acoustic Rendering aims at recreating acoustically the recorded performance from a specific listening location and orientation, with a controllable disposition of instruments on the stage and the listener.

We have considered binaural synthesis as the most suitable spatial audio technique for this application. Humans locate the direction of incoming sound based on a number of cues: depending on the angle and distance between listener and source, the sound will arrive with a different intensity

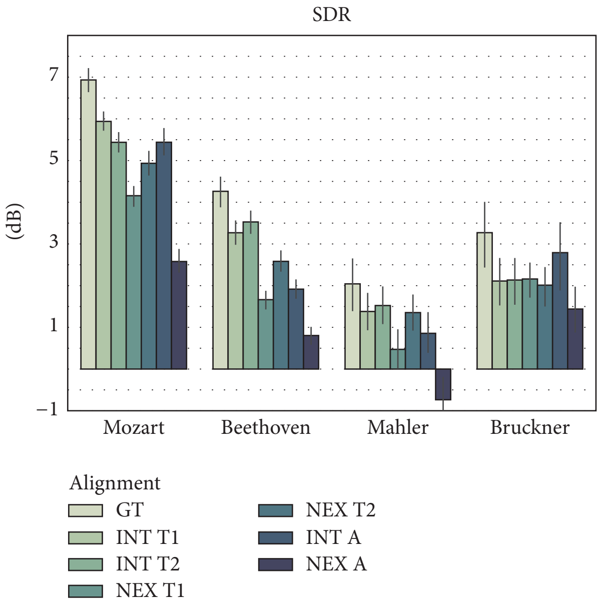


Figure 9: Results in terms of SDR, SIR, SAR, and ISR for the combination between different offset estimation methods (INT and Ext) and different sizes for the tolerance window for note onsets and offsets (T1 and T2). GT is the ground truth alignment and Ali is the output of the score alignment.

and at different time instances at both ears. The idea behind binaural synthesis is to artificially generate these cues to be able to create an illusion of directivity of a sound source when reproduced over headphones [43, 44]. For a rapid integration into the virtual reality prototype, we have used the noncommercial plugin for Unity3D of binaural synthesis provided by 3DCeption (https://twobigears.com/index.php).

Specifically, this application opens new possibilities in the area of virtual reality (VR), where video companies are already producing orchestral performances specifically recorded for VR experiences (e.g., the company WeMakeVR has collaborated with the London Symphony Orchestra and the Berliner Philharmoniker https://www.youtube.com/watch?v=ts4oXFmpacA). Using a VR headset with headphones, the Acoustic Rendering application is able to perform an acoustic zoom effect when pointing at a given instrument or section.

7.3. Source Localization. The third application aims to estimate the spatial location of musical instruments on stage. This application is useful for recordings where the orchestra layout is unknown (e.g., small ensemble performances) for the instrument visualization and Acoustic Rendering use-cases introduced above.

As for inputs for the source localization method we need the multichannel recordings and the approximate position of the microphones on stage. In concert halls, the recording setup consists typically of a grid structure of hanging overhead mics. The position of the overhead mics is therefore kept as metadata of the performance recording.

Automatic Sound Source Localization (SSL) methods make use of microphone arrays and complex signal processing techniques; however, undesired effects such as acoustic reflections and noise make this process difficult, being currently a hot-topic task in acoustic signal processing [45].

Our approach is a novel time difference of arrival (TDOA) method based on note-onset delay estimation. It takes the refined score alignment obtained prior to the signal separation (see Section 4.1). It follows two steps: first, for each instrument source, the relative time delays for the various microphone pairs are evaluated, and, then, the source location is found as the intersection of a pair of a set of half-hyperboloids centered around the different microphone pairs. Each half-hyperboloid determines the possible location of a sound source based on the measure of the time difference of arrival between the two microphones for a specific instrument.

To determine the time delay for each instrument and microphone pair, we evaluate a list of time delay values corresponding to all note onsets in the score and take the maximum of the histogram. In our experiment, we have a time resolution of 2.8 ms, corresponding to the Fourier transform hop size. Note that this method does not require time intervals in which the sources to play isolated as SRP-PHAT [45] and can be used in complex scenarios.

8. Outlook

In this paper, we proposed a framework for score-informed separation of multichannel orchestral recordings in distant-microphone scenarios. Furthermore, we presented a dataset which allows for an objective evaluation of alignment and separation (Section 6.1) and proposed a methodology for future research to understand the contribution of the different steps of the framework (Section 6.5). Then, we introduced several applications of our framework (Section 7). To our knowledge, this is the first time the complex scenario of orchestral multichannel recordings is objectively evaluated for the task of score-informed source separation.

Our framework relies on the accuracy of an audio-to-score alignment system. Thus, we assessed the influence of the alignment on the quality of separation. Moreover, we proposed and evaluated approaches to refining the alignment which improved the separation for three of the four pieces in the dataset, when compared to other two initialization options for our framework: the raw output of the score alignment and the tolerance window which the alignment relies on.

The evaluation shows that the estimation of panning matrix is an important step. Errors in the panning matrix can result into more interference in separated audio, or to problems in recovery of the amplitude of a signal. Since the method relies on finding nonoverlapping partials, an estimation done on a larger time frame is more robust. Further improvements on determining the correct channel for an instrument can take advantage of our approach for source localization in Section 7, provided that the method is reliable enough in localizing a large number of instruments. To that extent, the best microphone to separate a source is the closest one determined by the localization method.

When looking at separation across the instruments, viola, cello, and double bass were more problematic in the more complex pieces. In fact, the quality of the separation in our experiment varies within the pieces and the instruments, and future research could provide more insight on this problem. Note that increasing degree of consonance was related to a more difficult case for source separation [17]. Hence, we could expect a worse separation for instruments which are harmonizing or accompanying other sections, as the case of viola, cello, and double bass in some pieces. Future research could find more insight on the relation between the musical characteristics of the pieces (e.g., tonality and texture) and source separation quality.

The evaluation was conducted on the dataset presented in Section 6.1. The creation of the dataset was a very laborious task, which involved annotating around 12000 pairs of onsets and offsets, denoising the original recordings and testing different room configurations in order to create the multichannel recordings. To that extent, annotations helped us to denoise the audio files, which could then be used in score-informed source separation experiments. Furthermore, the annotations allow for other tasks to be tested within this challenging scenario, such as instrument detection, or transcription.

Finally, we presented several applications of the proposed framework related to Instrument Emphasis or Acoustic Rendering, some of which are already at the stage of functional products.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The authors would like to thank all the partners of the PHENICX consortium for a very fruitful collaboration. Particularly, they would like to thank Agustin Martorell for his insights on the musicology side and correcting the scores and Oscar Mayor for his help regarding Repovizz and his contribution on the applications. This work is partially supported by the Spanish Ministry of Economy and Competitiveness under CASAS Project (TIN2015-70816-R).

References

[1] E. Gómez, M. Grachten, A. Hanjalic et al., "PHENICX: performances as highly enriched aNd Interactive concert experiences," in *Proceedings of the SMAC Stockholm Music Acoustics Conference 2013 and SMC Sound and Music Computing Conference*, 2013.

[2] S. Ewert and M. Müller, "Using score-informed constraints for NMF-based source separation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '12)*, pp. 129–132, Kyoto, Japan, March 2012.

[3] J. Fritsch and M. D. Plumbley, "Score informed audio source separation using constrained nonnegative matrix factorization and score synthesis," in *Proceedings of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '13)*, pp. 888–891, Vancouver, Canada, May 2013.

[4] Z. Duan and B. Pardo, "Soundprism: an online system for score-informed source separation of music audio," *IEEE Journal on Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1205–1215, 2011.

[5] R. Hennequin, B. David, and R. Badeau, "Score informed audio source separation using a parametric model of non-negative spectrogram," in *Proceedings of the 36th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '11)*, pp. 45–48, May 2011.

[6] J. J. Carabias-Orti, M. Cobos, P. Vera-Candeas, and F. J. Rodriguez-Serrano, "Nonnegative signal factorization with learnt instrument models for sound source separation in close-microphone recordings," *EURASIP Journal on Advances in Signal Processing*, vol. 2013, article 184, 2013.

[7] T. Pratzlich, R. M. Bittner, A. Liutkus, and M. Muller, "Kernel Additive Modeling for interference reduction in multi-channel music recordings," in *Proceedings of the 40th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '15)*, pp. 584–588, Brisbane, Australia, April 2014.

[8] S. Ewert, B. Pardo, M. Mueller, and M. D. Plumbley, "Score-informed source separation for musical audio recordings: an overview," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 116–124, 2014.

[9] F. J. Rodriguez-Serrano, Z. Duan, P. Vera-Candeas, B. Pardo, and J. J. Carabias-Orti, "Online score-informed source separation with adaptive instrument models," *Journal of New Music Research*, vol. 44, no. 2, pp. 83–96, 2015.

[10] F. J. Rodriguez-Serrano, S. Ewert, P. Vera-Candeas, and M. Sandler, "A score-informed shift-invariant extension of complex matrix factorization for improving the separation of overlapped partials in music recordings," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '16)*, pp. 61–65, Shanghai, China, 2016.

[11] M. Miron, J. J. Carabias, and J. Janer, "Improving score-informed source separation for classical music through note refinement," in *Proceedings of the 16th International Society for Music Information Retrieval (ISMIR '15)*, Málaga, Spain, October 2015.

[12] A. Arzt, H. Frostel, T. Gadermaier, M. Gasser, M. Grachten, and G. Widmer, "Artificial intelligence in the concertgebouw," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pp. 165–176, Buenos Aires, Argentina, 2015.

[13] J. J. Carabias-Orti, F. J. Rodriguez-Serrano, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Cañadas-Quesada, "An audio to score alignment framework using spectral factorization and dynamic time warping," in *Proceedings of the 16th International Society for Music Information Retrieval (ISMIR '15)*, Malaga, Spain, 2015.

[14] Ö. Izmirli and R. Dannenberg, "Understanding features and distance functions for music sequence alignment," in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR '10)*, pp. 411–416, Utrecht, The Netherlands, 2010.

[15] M. Goto, "Development of the RWC music database," in *Proceedings of the 18th International Congress on Acoustics (ICA '04)*, pp. 553–556, Kyoto, Japan, April 2004.

[16] S. Ewert, M. Müller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 1869–1872, IEEE, Taipei, Taiwan, April 2009.

[17] J. J. Burred, *From sparse models to timbre learning: new methods for musical source separation [Ph.D. thesis]*, 2008.

[18] F. J. Rodriguez-Serrano, J. J. Carabias-Orti, P. Vera-Candeas, T. Virtanen, and N. Ruiz-Reyes, "Multiple instrument mixtures source separation evaluation using instrument-dependent NMF models," in *Proceedings of the 10th international conference on Latent Variable Analysis and Signal Separation (LVA/ICA '12)*, pp. 380–387, Tel Aviv, Israel, March 2012.

[19] A. Klapuri, T. Virtanen, and T. Heittola, "Sound source separation in monaural music signals using excitation-filter model and EM algorithm," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '10)*, pp. 5510–5513, March 2010.

[20] J.-L. Durrieu, A. Ozerov, C. Févotte, G. Richard, and B. David, "Main instrument separation from stereophonic audio signals using a source/filter model," in *Proceedings of the 17th European Signal Processing Conference (EUSIPCO '09)*, pp. 15–19, Glasgow, UK, August 2009.

[21] J. J. Bosch, K. Kondo, R. Marxer, and J. Janer, "Score-informed and timbre independent lead instrument separation in real-world scenarios," in *Proceedings of the 20th European Signal Processing Conference (EUSIPCO '12)*, pp. 2417–2421, August 2012.

[22] P. S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Singing-voice separation from monaural recordings using deep recurrent neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR '14)*, Taipei, Taiwan, October 2014.

[23] E. K. Kokkinis and J. Mourjopoulos, "Unmixing acoustic sources in real reverberant environments for close-microphone applications," *Journal of the Audio Engineering Society*, vol. 58, no. 11, pp. 907–922, 2010.

[24] S. Ewert and M. Müller, "Score-informed voice separation for piano recordings," in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR '11)*, pp. 245–250, Miami, Fla, USA, October 2011.

[25] B. Niedermayer, *Accurate audio-to-score alignment-data acquisition in the context of computational musicology [Ph.D. thesis]*, Johannes Kepler University Linz, Linz, Austria, 2012.

[26] S. Wang, S. Ewert, and S. Dixon, "Compensating for asynchronies between musical voices in score-performance alignment," in *Proceedings of the 40th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '15)*, pp. 589–593, April 2014.

[27] M. Miron, J. Carabias, and J. Janer, "Audio-to-score alignment at the note level for orchestral recordings," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR '14)*, Taipei, Taiwan, 2014.

[28] D. Fitzgerald, M. Cranitch, and E. Coyle, "Non-negative tensor factorisation for sound source separation. acoustics, speech and signal processing," in *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal (ICASSP '06)*, Toulouse, France, 2006.

[29] C. Févotte and A. Ozerov, "Notes on nonnegative tensor factorization of the spectrogram for audio source separation: statistical insights and towards self-clustering of the spatial cues," in *Exploring Music Contents: 7th International Symposium, CMMR 2010, Málaga, Spain, June 21–24, 2010. Revised Papers*, vol. 6684 of *Lecture Notes in Computer Science*, pp. 102–115, Springer, Berlin, Germany, 2011.

[30] J. Pätynen, V. Pulkki, and T. Lokki, "Anechoic recording system for symphony orchestra," *Acta Acustica United with Acustica*, vol. 94, no. 6, pp. 856–865, 2008.

[31] D. Campbell, K. Palomäki, and G. Brown, "A Matlab simulation of shoebox room acoustics for use in research and teaching," *Computing and Information Systems*, vol. 9, no. 3, p. 48, 2005.

[32] O. Mayor, Q. Llimona, M. Marchini, P. Papiotis, and E. Maestre, "RepoVizz: a framework for remote storage, browsing, annotation, and exchange of multi-modal data," in *Proceedings of the 21st ACM International Conference on Multimedia (MM '13)*, pp. 415–416, Barcelona, Spain, October 2013.

[33] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[34] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: with application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.

[35] M. Nixon, *Feature Extraction and Image Processing*, Elsevier Science, 2002.

[36] R. Parry and I. Essa, "Estimating the spatial position of spectral components in audio," in *Independent Component Analysis and Blind Signal Separation: 6th International Conference, ICA 2006, Charleston, SC, USA, March 5–8, 2006. Proceedings*, J. Rosca, D. Erdogmus, J. C. Príncipe, and S. Haykin, Eds., vol. 3889 of *Lecture Notes in Computer Science*, pp. 666–673, Springer, Berlin, Germany, 2006.

[37] P. Papiotis, M. Marchini, A. Perez-Carrillo, and E. Maestre, "Measuring ensemble interdependence in a string quartet through analysis of multidimensional performance data," *Frontiers in Psychology*, vol. 5, article 963, 2014.

[38] M. Mauch and S. Dixon, "PYIN: a fundamental frequency estimator using probabilistic threshold distributions," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '14)*, pp. 659–663, Florence, Italy, May 2014.

[39] F. J. Cañadas-Quesada, P. Vera-Candeas, D. Martínez-Muñoz, N. Ruiz-Reyes, J. J. Carabias-Orti, and P. Cabanas-Molero, "Constrained non-negative matrix factorization for score-informed piano music restoration," *Digital Signal Processing*, vol. 50, pp. 240–257, 2016.

[40] A. Cont, D. Schwarz, N. Schnell, and C. Raphael, "Evaluation of real-time audio-to-score alignment," in *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR '07)*, pp. 315–316, Vienna, Austria, September 2007.

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

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

[43] D. R. Begault and E. M. Wenzel, "Headphone localization of speech," *Human Factors*, vol. 35, no. 2, pp. 361–376, 1993.

[44] G. S. Kendall, "A 3-D sound primer: directional hearing and stereo reproduction," *Computer Music Journal*, vol. 19, no. 4, pp. 23–46, 1995.

[45] A. Martí Guerola, *Multichannel audio processing for speaker localization, separation and enhancement*, Universitat Politècnica de València, 2013.

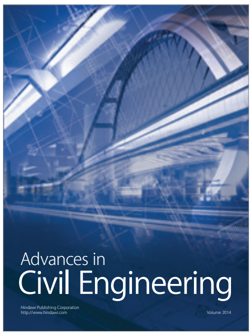
Journal of
Engineering

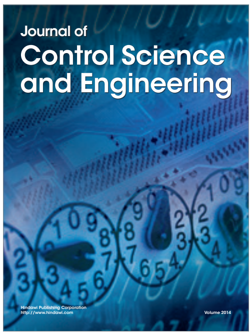
The Scientific
World Journal

International Journal of
Rotating
Machinery

Journal of
Sensors

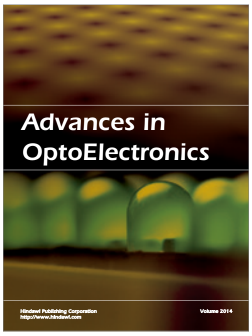
International Journal of
Distributed
Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science
and Engineering

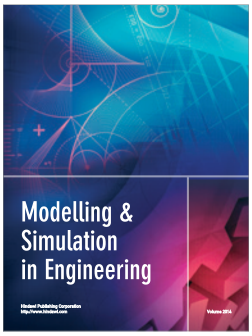
Journal of
Robotics

Journal of
Electrical and Computer
Engineering

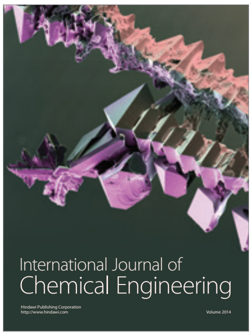

Advances in
OptoElectronics

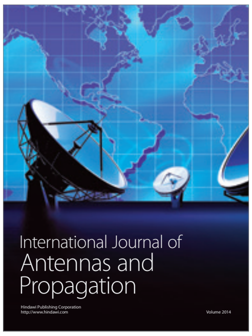
VLSI Design

International Journal of
Navigation and
Observation

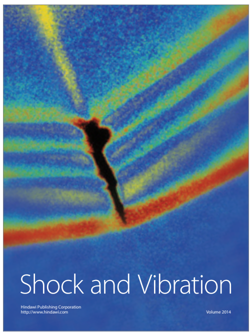

Modelling &
Simulation
in Engineering

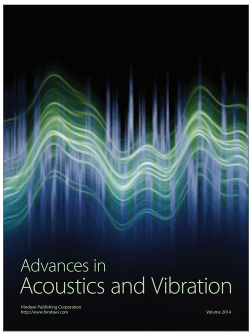
International Journal of
Aerospace
Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and
Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration