IMPROVING SCORE-INFORMED SOURCE SEPARATION FOR CLASSICAL MUSIC THROUGH NOTE REFINEMENT

Marius Miron **Julio José Carabias-Orti** **Jordi Janer**
Music Technology Group, Universitat Pompeu Fabra
marius.miron,julio.carabias,jordi.janer@upf.edu

ABSTRACT

Signal decomposition methods such as Non-negative Matrix Factorization (NMF) demonstrated to be a suitable approach for music signal processing applications, including sound source separation. To better control this decomposition, NMF has been extended using prior knowledge and parametric models. In fact, using score information considerably improved separation results. Nevertheless, one of the main problems of using score information is the misalignment between the score and the actual performance. A potential solution to this problem is the use of audio to score alignment systems. However, most of them rely on a tolerance window that clearly affects the separation results. To overcome this problem, we propose a novel method to refine the aligned score at note level by detecting both, onset and offset for each note present in the score. Note refinement is achieved by detecting shapes and contours in the estimated instrument-wise time activation (gains) matrix. Decomposition is performed in a supervised way, using training instrument models and coarsely-aligned score information. The detected contours define time-frequency note boundaries, and they increase the sparsity. Finally, we have evaluated our method for informed source separation using a dataset of Bach chorales obtaining satisfactory results, especially in terms of SIR.

1. INTRODUCTION

Sound source separation has been actively addressed during the recent years with various applications ranging from predominant melody transcription [10], to interference removal in close microphone recordings [4]. State of the art systems particularly target the separation of the predominant harmonic instrument from the accompaniment [3, 4, 10, 18], and less often the separation of various instruments in classical music [9, 15].

Besides [18](recurrent neural networks), and [9](particle filters), the aforementioned systems are based on non-negative matrix factorization (NMF) [19], a technique that efficiently decomposes a magnitude spectrogram into a set of template (basis) and activation (gains) vectors. However, when dealing with a non-convex problem, the NMF can converge to a local minima solution for which the sources are not well separated. Towards a better separation, the system can benefit from prior knowledge. On this account, a set of musical meaningful variables are introduced into the parametric model and estimated jointly.

Furthermore, important improvements are reported when score information is added to guide the decomposition process [3, 9, 12, 15, 17]. In this case, the best performance is achieved when the audio is perfectly aligned with the score [23]. However, in a real-case scenario, a perfect aligned score is not available, and a score-alignment algorithm is needed [5, 8, 9, 13].

Conversely, as enounced in [3], besides the global misalignments, fixed by score-alignment systems, we can also encounter local misalignments. With respect to this problem, source separation systems propose to estimate the onset implicitly into the parametric NMF model, by increasing the time boundaries for the onsets in the gains matrix at the initialization stage [12, 15, 17]. However, an interesting question is whether such an initialization results in a better separation than refining the gains and correcting the local misalignments prior to the source separation.

Several methods deal with explicitly correcting local misalignments [21, p. 103], [20,27]. The latter finds shapes and contours (blobs) in a pitch salience function, obtained by pre-processing the spectrogram of the signal and then filtering the spectral peaks for each instrument. However, this method does not use any information regarding the timbre, which is more desirable when distributing energy between different instruments.

The goal of this paper is to use the note refinement information in order to improve score-informed source separation of harmonic instruments. Specifically, we have two contributions: we adapt the source separation framework in [24] to the score-informed case, and, notably, we correct the local misalignments in the score and refine the time-frequency zones of the gains used in source separation. First, we compute the initial gains by distributing the energy among instruments with the source separation NMF algorithm proposed in [24]. The computed gains offer a more robust representation than the pitch salience used in [20], because timbre information is used to deal with the problem of overlapping partials between the instruments, and because the gains are represented on log-frequency scale and are less noisy than the pitch salience in [20]. As a result, detecting and assigning blobs to notes in the gains matrix can be done more robustly. Second, we can use the processed gains to reiterate the NMF source separation. Consequently, instead of initializing the NMF with the MIDI information, we can use the blobs associated with each note. On this account, we restrict the potential interferences not only in time but also in frequency, and achieve better separation.

We evaluate the note refinement and the source separation on the Bach10 dataset [9]. Accordingly, note refinement is performed on an artificially generated score with local misalignments, and on the output the DTW based score alignment algorithm [5]. Furthermore, we evaluate the score-informed source separation, as we want more insight on which initialization method yields better source separation.

The remainder of this paper is structured as follows. First, we describe the existing source separation framework and then, in Section 3, the note refinement method and its application to monaural score informed source separation. Then, we evaluate score alignment and source separation. Finally, we discuss the results and restate the contributions to prior work.

2. NMF FOR SOURCE SEPARATION

In this section we explain the Source Separation Framework used for sound source separation. Further information can be found in [24].

2.1 Signal Model

Techniques based on Non-negative Matrix Factorization (NMF) can be used to efficiently decompose an audio spectrogram as a linear combination of spectral basis functions. In such a model, the short-term magnitude (or power) spectrum of the signal $x(f,t)$ in time-frame t and frequency f is modeled as a weighted sum of basis functions as:

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

where $g_n(t)$ is the gain of the basis function n at frame t, and $b_n(f)$, $n = 1, ..., N$ are the bases. Note that model in eq. (1) only holds under the assumption of a) strong sparsity (only one source active per time-frequency(TF) bin) or b) local stationarity (only for power spectrogram) [2].

When dealing with musical instrument sounds, it is natural to assume that each basis function represents a single pitch, and the corresponding gains contain information about the onset and offset times of notes having that pitch [4]. Besides, restricting the model in (1) to be harmonic is particularly useful for the analysis and separation of musical audio signals since each basis can define a single fundamental frequency and instrument. Harmonicity constrained basis functions are defined as:

$$b_{j,n}(f) = \sum_{h=1}^{H} a_{j,n}(h) G(f - h f_0(n)), \tag{2}$$

where $b_{j,n}(f)$, are the bases for each note n of instrument j, $n = 1, ..., N$ is defined as the pitch range for instrument $j = 1, ..., J$, where J is the total number of instruments present in the mixture, $h = 1, ..., H$ 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 $G(f - hf_0(n))$. Therefore, the harmonic constrained model for the magnitude spectra of a music signal is defined as:

$$\hat{x}(f,t) = \sum_{j=1}^{J} \sum_{n=1}^{N} \sum_{h=1}^{H} g_{j,n}(t) a_{j,n}(h) G(f - h f_0(n)), \tag{3}$$

where the time gains $g_{j,n}(t)$ and the harmonic amplitudes $a_{j,n}(h)$ are the parameters to be estimated.

2.2 Augmented NMF for Parameter Estimation

Non-negativity of the parameters is a common restriction imposed to the signal decomposition method for music signal processing applications. Furthermore, the factorization parameters of equation (3) are estimated by minimizing the reconstruction error between the observed $x(f,t)$ and the modeled $\hat{x}(f,t)$ spectrograms, using a cost function, which is this case the Beta-divergence [14]:

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

For particular values of β, Beta-divergence includes in its definition the most popular cost functions, Euclidean (EUC) distance ($\beta = 2$), Kullback-Leibler (KL) divergence ($\beta = 1$) and the Itakura-Saito (IS) divergence ($\beta = 0$). The parameters in (1) are estimated with an iterative cost minimization algorithm based on multiplicative update (MU) rules, as discussed in [19]. Under these rules, $D(x(f,t)|\hat{x}(f,t))$ does not increase with each iteration while ensuring the non-negativity of the bases and the gains. These MU rules are obtained applying diagonal rescaling to the step size of the gradient descent algorithm (see [19] for further details).

Lets denote as θ_l the parameter to be estimated. Then, the MU rule for θ_l is obtained by computing the derivative $\nabla_{\theta_l} D$ of the cost function with respect to θ_l. This derivative can be expressed as a difference between two positive terms $\nabla_{\theta_l}^+ D$ and $\nabla_{\theta_l}^- D$ [25] and thus, the update rule for parameter θ_l can be expressed as:

$$\theta_l \leftarrow \theta_l \frac{\nabla_{\theta_l}^- D(x(f,t)|\hat{x}(f,t))}{\nabla_{\theta_l}^+ D(x(f,t)|\hat{x}(f,t))}. \tag{5}$$

2.3 Timbral Informed Signal Model

As showed in [6], when appropriate training data are available, it is useful to learn the instrument-dependent bases in advance and keep them fixed during the analysis of the signals. In the commented work, the amplitudes of each note of each musical instrument $a_{j,n}(h)$ are learnt by using the RWC database [16] of solo instruments playing isolated notes together with their ground-truth transcription. Thus, gains are set to unity for each pitch at those time frames where the instrument is active while the rest of the gains are set to zero. Note that gains initialized to zero remain zero because of the multiplicative update rules, and therefore the frame is represented only with the correct pitch.

The MU rules are computed from equation (5) for the amplitude coefficients and the gains as

$$a_{j,n}(h) \leftarrow a_{j,n}(h) \frac{\sum_{f,t} x(f,t)\hat{x}(f,t)^{\beta-2} g_{j,n}(t) G(f-hf_0(n))}{\sum_{f,t} \hat{x}(f,t)^{\beta-1} g_{j,n}(t) G(f-hf_0(n))} \tag{6}$$

$$g_{j,n}(t) \leftarrow g_{j,n}(t) \frac{\sum_{f,m} x(f,t)\hat{x}(f,t)^{\beta-2} a_{j,n}(h) G(f-hf_0(n))}{\sum_{f,m} \hat{x}(f,t)^{\beta-1} a_{j,n}(h) G(f-hf_0(n))} \tag{7}$$

Finally, the training procedure is summarized in Algorithm 1.

Algorithm 1 Instrument modeling algorithm

```
1 Compute $x(f,t)$ from a solo performance for each instrument in the training database
2 Initialize gains $g_{j,n}(t)$ with the ground truth transcription $R_{j,n}(t)$ and $a_{j,n}(h)$ with random positive values.
3 Update the gains using eq. (6).
4 Update the bases using eq. (7).
5 Repeat steps 2-3 until the algorithm converges (or maximum number of iterations is reached).
6 Compute basis functions $b_{j,n}(f)$ for each instrument $j$ using eq. (2).
```

The training algorithm obtains an estimation of the basis functions $b_{j,n}(f)$ required at the factorization stage for each instrument. Since the instrument dependent basis functions $b_{j,n}(f)$ are held fixed, the factorization can be reduced to the estimation of the gains $g_{j,n}(t)$ for each of the trained instruments j.

2.4 Gains estimation

Here, the classical augmented NMF factorization with MU rules is applied to estimate the gains corresponding to each source j in the mixture. The process is detailed in Algorithm 2.

Algorithm 2 Gain Estimation Method

```
1 Initialize $b_{j,n}(f)$ with the values learned in section 2.3. Use random positive values to initialize $g_{j,n}(t)$.
2 Update the gains using eq. (7).
3 Repeat step 2 until the algorithm converges (or maximum number of iterations is reached)
```

2.5 From the estimated gains to the separated signals

In this work, we assume that the individual sources $y_j(t), j = 1...J$ that compose the mixed signal $x(t)$ are linearly mixed, so $x(t) = \sum_{j=1}^{J} y_j(t)$. Lets denote the power spectral density of source j at TF bin (f,t) as $|X_j(t,f)|^2, j = 1...J$, then, each ideally separated source $y_j(t)$ can be estimated from the mixture $x(t)$ using a generalized time-frequency Wiener filter over the Short-Time Fourier Transform (STFT) domain as in [14, 15].

Here we use the Wiener filter soft-masking strategy as in [24]. In particular, the soft-mask α_j of source j represents the relative energy contribution of each source to the total energy of the mixed signal $x(t)$ and is obtained as:

$$\alpha_j(t,f) = \frac{\hat{Y}_j(t,f)^2}{\sum_j \hat{Y}_j(t,f)^2} \tag{8}$$

where $\hat{Y}_j(t,f)$ is the estimated source magnitude spectrogram computed as $\hat{Y}_j(t,f) = g_{n,j}(t) b_{j,n}(f)$, $g_{n,j}$ are the gains estimated in Section 2.4 and $b_{j,n}(f)$ are the fixed basis functions learnt in Section 2.3.

Then, the magnitude spectrogram $\hat{X}_j(t,f)$ is estimated for each source j as:

$$\hat{X}_j(t,f) = \alpha_j(t,f) \cdot X(t,f) \tag{9}$$

where $X(t,f)$ is the complex-valued STFT of the mixture at TF bin (t,f).

Finally, the estimated source $\hat{y}_j(t)$ is computed with the inverse overlap-add STFT over $\hat{X}_j(f,t)$, with the phase spectrogram of the original mixture.

3. PROPOSED METHOD

We adapt the source separation framework described in Section 2 to the score-informed scenario. The framework is initialized with the gains $g_{j,n}^{init}(t)$ derived from a MIDI score having alignment errors. Next, the resulting gains after the NMF separation $g_{j,n}(t)$ are refined with a set of image processing heuristics which we describe in the Section 3.2. Finally, the refined gains $p_{j,n}(t)$ are used to reinitialize the framework and reiterate the separation, towards a better result.

3.1 Score-informed gains computation

We use as input a coarsely aligned score and the associated audio recording. The MIDI score has local misalignments up to d frames for the onset and the offset times. Thus, we initialize the source separation system in Section 2 with the MIDI notes by adding d frames before the onset and after the offset. Consequently, for an instrument j, and all the bins in a semitone n associated with a MIDI note (Figure 1B), we set the matrix $g_{j,n}^{init}(t)$ to 1 for the frames where the MIDI note is played as well as for the d frames around the onset and the offset of the MIDI note. The other values in $g_{j,n}^{init}(t)$ are set to 0 do not change during computation, while the values set to 1 evolve according to the energy distributed between the instruments. The final gains are computed with the algorithm described in Section 2.4, obtaining $g_{j,n}(t)$, the gains which will be used during the note refinement stage (Figure 1C).

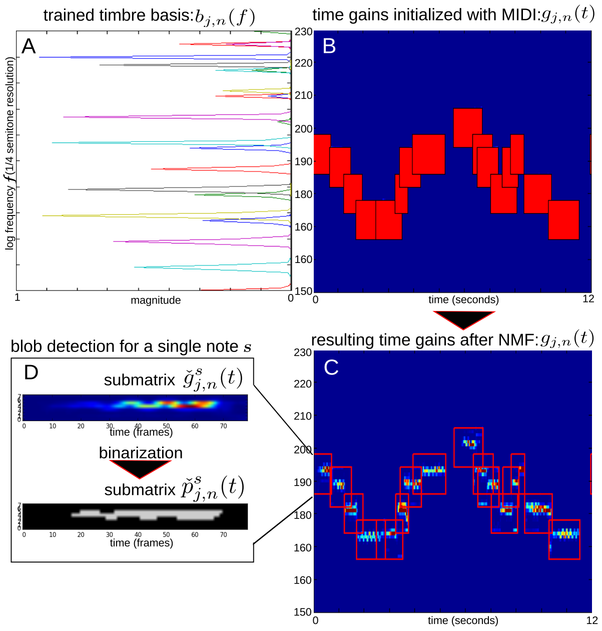

Figure 1. *A. The reconstructed signal can be seen as the product between the several harmonic components (A) and the gains (B). After NMF, the resulting gains (C) are split in submatrices and used to detect blobs (D).*

3.2 Note refinement

The shape and contours detected in an image, and associated with meaningful objects, are commonly known as blobs [22, p. 248]. Additionally, if we consider the matrix associated with a grayscale image, an image patch is any submatrix of the corresponding matrix.

During the note refinement stage we apply image processing on the gains matrix $g_{j,n}(t)$ in order to associate the entities in an image, namely the blobs, with notes. The chosen blobs give the onset and offset times. Additionally, the areas of the blobs are used to reiterate the separation.

The refinement of the gains occurs for each note separately. Hence, for each note s from the input score we choose an image patch centered at the semitone n corresponding to its associated MIDI note value. Precisely, we process a submatrix of $g_{j,n}(t)$, namely $\breve{g}^s_{j,n}(t)$, for $s = 1...S$, where S is the total number of notes in the score for an instrument j. The size of submatrix $\breve{g}^s_{j,n}(t)$, as seen in Figure 1D, is equal to the one of the submatrices which has been set to 1 at the initialization for the corresponding note s. Thus, $\breve{g}^s_{j,n}(t)$ has a width of two semitones and a length corresponding to the prolonged duration of the note s.

3.2.1 Image binarization

Each image patch is preprocessed in two steps before binarization. Initially, each row vector of the submatrix $\breve{g}^s_{j,n}(t)$ is convolved with a smoothing gaussian filter to remove noise and discontinuities. Then each column of the same submatrix is multiplied with a gaussian centered at the central frequency bin, in order to penalize the values far from the central bin, but still to preserve vibratos or transitions between notes.

First, we apply a smoothing filter [22, p. 86] on the image patch. We choose a one dimension Gaussian filter:

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

where t is the time axis and $\phi = 3$ is the standard deviation . The first and the last σ elements of each row vector n of the matrix $\breve{g}^s_{j,n}(t)$ are mirrored at the beginning, respectively at the end of the vector. Then each row vector of $\breve{g}^s_{j,n}(t)$ 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 multiply $\breve{g}^s_{j,n}(t)$ with a 1-dimensional gaussian centered in the central frequency bin:

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

where n is the frequency axis, $\kappa = 4$ is the position of the central frequency bin and the standard deviation $\nu = 4$(one semitone). Then, each column vector of $\breve{g}^s_{j,n}(t)$ is multiplied with $v(n)$.

Image binarization assumes calculating a submatrix $\breve{p}^s_{j,n}(t)$, associated with note s:

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

3.2.2 Blob selection

For a note s we detect blobs the corresponding binary submatrix $\breve{p}^s_{j,n}(t)$, using the connectivity rules described in [22, p. 248] and [20].

Furthermore, we need to determine the best blob for each note. A simple solution is to compute a score for each blob by summing all the values in $\breve{g}^s_{j,n}(t)$ included in the area associated with the blob. However, we want to penalize parts of the blobs which overlap in time with other blobs from different notes $s-1, s, s+1$. Basically, we want to avoid picking the same blobs for two adjacent notes. Thus, we weight each element in $\breve{g}^s_{j,n}(t)$ with a factor γ, depending on the amount of overlapping with blobs from adjacent notes, and we build a score matrix:

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

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

Note that we do not use the dynamic programming method in [20] because the images patches are small, thus we have to choose between very few blobs and, to that respect, the Dijkstra algorithm is superfluous.

Furthermore, we compute a score for each note s and for each blob associated with the note, by summing up the elements in the score matrix $\breve{q}^s_{j,n}(t)$ which are a part of a blob. Furthermore, the selected blob for a note s is the one having the maximum score. The boundaries of the selected blob give the note onset and offset. Additonally, the area of the blob can be used to reiterate source separation.

3.3 Extension to score informed source separation

Our assumption is that better alignment gives a more sparse initialization of the gains $g_{j,n}(t)$, which limits the way energy distributes along instruments during the NMF, and yields better source separation. Additionally, we can further increase sparsity by knowing the frequency boundaries of the notes and by initializing the gains with the detected blob contours. However, by limiting the areas in the activations to the area of the chosen blobs, we discard energy from the unchosen blobs. This energy which is discarded from an instrument can be redistributed between the other instruments by reiterating the factorization.

Let $p^s_{j,n}(t)$ be the matrices derived from the submatrices $\tilde{p}^s_{j,n}(t)$, containing 1 for the elements associated with the selected blob for the note s and 0 otherwise. Then, the new matrix $g_{j,n}(t)$ can be formed with the submatrices $p^s_{j,n}(t)$. For the corresponding bins n and time frames f of a note s, we initialize the values in $g_{j,n}(t)$ with the values in $p^s_{j,n}(t)$. Subsequently, we repeat the factorization using the timbre-informed algorithm described in Section 2.4, this time initializing it with the refined gains. Moreover, the calculate the spectrogram of the separated sources with the method described in Section 2.5.

4. EXPERIMENTAL SETUP

a) Time-Frequency representation: In this paper we use a low-level spectral representation of the audio data which is generated from a windowed FFT of the signal. A Hanning window with the size of 92 ms, and a hop size of 11 ms are used (for synthetic and real-world signals). 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 [4]. These time-frequency representations are obtained by integrating the short-term Fourier transform (STFT) bins corresponding to the same semitone, or $1/4$ semitone, interval. Note that in the separation stage, the learnt basis functions $b_{j,n}(f)$ are adapted to the $1/4$ semitone resolution by replicating 4 times the basis at each semitone to the 4 samples of the $1/4$ semitone resolution that belong to this semitone.

b) Dataset: We evaluate the note refinement and the source separation on the Bach10 dataset presented in [9] and comprising ten J.S. Bach chorales played by a quartet (violin, clarinet, tenor saxophone and bassoon), each piece having the duration $\approx$ 30 seconds. The instruments were recorded separately, then mixed to create a monaural audio sampled at 44.1 kHz. Moreover, the Bach10 dataset has certain traits which influence the note refinement and source separation. For instance, the chorales present a homophonic texture which makes it more difficult when performing source separation. Additionally, the results are directly related to the tempo of the recordings [9]. For this dataset, the tempo is slower than other classical music pieces, there are very few notes below the quarter note level, and we have prolonged notes, known as fermata.

The audio files are accompanied by two MIDI scores: the perfectly aligned ground truth, and a score which has global and local misalignments. Moreover, in order to test the note refinement we use two datasets. The dataset *disA*, proposed in [20], introduces errors for the ground truth onsets and offsets in the interval [100...200] ms. Additionally, we plan to refine the alignment at the note level for the score alignment method described in [5], denoted as dataset *dtwJ*. The method offers solely note onset information, therefore we use the onset of the next note as the note offset for the current note.

c) Evaluation metrics: For score aligment, we evaluate note onsets and offsets in terms of alignment rate, similarly to [7], ranging from 0 to 1, defined as the proportion of correctly aligned notes in the score within a given threshold. For source separation, the evaluation framework and the metrics are described in [26] and [11]. Correspondingly, we use *Source to Distortion Ratio* (SDR), *Source to Interference Ratio* (SIR), and *Source to Artifacts Ratio* (SAR).

d) Parameters tuning: We picked 50 number of iterations for the NMF, and we experimentally determined value for the beta-divergence distortion, $\beta = 1.3$.

5. RESULTS

5.1 Score alignment

We measure the aligment rate of the input score presenting misalignments (B), the alignment method described in Section 3.2 (E), and the one in [20] (D), on the two datasets ”disA” and ”dtwJ”. We vary the threshold within the interval [15..200]. Subsequently, in Figure 2 we present the results for the datasets ”disA” and ”dtwJ”. The errors of the original scores are presented with dotted and straight black lines. For the aligned onsets, aligned rates are drawn with dashed lines and for offsets with straight lines.

We observe that both refinement methods improve the align rate of the scores with local misalignments (black line). For lower threshold, the proposed method (red) improves the method in [20] (blue). Moreover, considering that offsets are more difficult to align, the proposed alignment outperforms the one in [20] when it comes to detecting offsets, within a larger threshold.

5.2 Source separation

We use the evaluation metrics described in Section 4c. We initialize the gains of the separation framework with different note information, as seen in Figure 3. Specifically, we evaluate the perfect initialization with the ground-truth MIDI, Figure 3(A), with the score having local misalignments (*disA*) or the output of a score aligment system *dtwJ*, Figure 3(B), the common practice of NMF gains initialization in state of the art score-informed source separation [12,15,17], Figure 3(C), and the refinement aproaches: Figure 3(D,E,F). Note that in D and E we initialize the

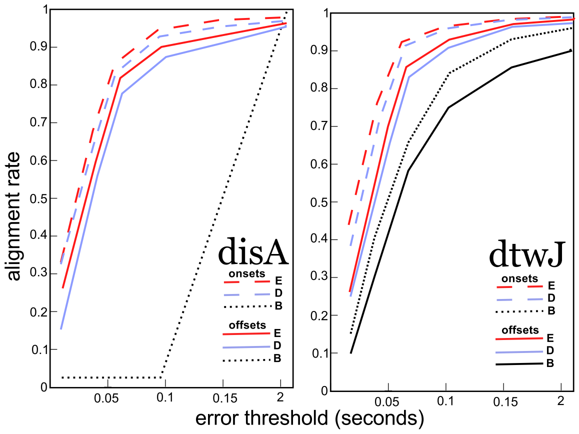

Figure 2. *Alignment rate for the two datasets; "B" denotes the score to be refined; "E" and "D" are the scores refined with the methods in Section 3.2 and [20].*

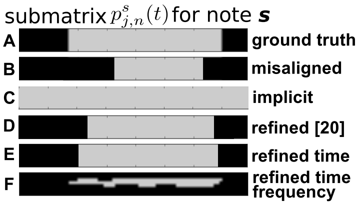

Figure 3. *The test cases for initialization of score-informed source separation, for the submatrix $p^s_{j,n}(t)$*

gains prior to a note refinement stage with the methods described in [20] (refined [20]) and in the Section 3.2 (refined time), and in F we further refine the gains as proposed in Section 3.3 (refined time frequency).

The results for the test cases A-F, for the two datasets *disA* and *dtwJ* are presented in Table 1 in terms of means of SDR, SIR, SAR. Additionally, audio examples of the separation can be listened online [1].

The proposed system F improves over the other cases in terms of SDR, for all the input scores. Particularly, when we refine the gains in frequency we obtain higher SIR values, hence less interference. Consequently, F yields better results than A, the initialization with ground-truth MIDI annotations, and than E, which is note refinement in time, without tracking the shape of the blob. On the other hand, the ground-truth A has better SAR values, less artifacts, but has more interference, since F sets to zero some parts of the gains matrix for which the energy does not get redistributed. Additionally, F improves over C, the implicit initialization which extends the time span for the gains, which is the most used approach by the state of the art score-informed source separation algorithms when dealing with local misalignments. On the other hand, the worse decision is not to do any refinement, as in case B.

Moreover, F achieves better results than A-E refining the alignment of [5] (dataset *dtwJ*). However, as this dataset does not have large local misalignments, the difference between F and C, and even B, is not as high as for dataset *disA*, and the improvement is not remarkable. Note that F is better than A in this case as well, suggesting that our proposed method is robust with regards to different kinds of inputs: significant local misalignments as the dataset *disA*, or smaller as dataset *dtwJ*. Additionally, ground truth offsets are close to the next note onsets, thus *dtwJ* achieves better separation compared to *disA*.

	dataset *disA*			dataset *dtwJ*		
	SDR	SIR	SAR	SDR	SIR	SAR
A	6.31	7.10	**25.26**	6.31	7.10	25.26
B	3.72	4.04	15.20	6.19	6.99	24.59
C	5.18	5.67	19.62	6.25	6.97	25.31
D	5.89	6.80	22.41	5.79	6.67	23.69
E	6.24	7.08	24.43	6.07	6.99	24.58
F	**6.35**	**7.37**	24.18	**6.37**	**7.23**	**25.45**

Table 1. Means of SDR, SIR, ISR for the datasets $disA$ and $dtwJ$ for test cases A-F, for all the instruments

Furthermore, with respect to the performance achieved by other source separation frameworks, tested on the same dataset [9], the results in terms of SDR are similar. The method we propose in this paper is used with the source separation framework [24], but can be adapted to other NMF based frameworks. However, due to the TF representation used in the method, even for ideal TF masks, the separated examples might exhibit cross-talk at high frequencies. This fact is reflected in the measures by lower SIR values.

6. CONCLUSIONS

We proposed a timbre-informed note refinement method to correct local misalignments and to refine the output of state of the art audio-to-score alignment systems, for monaural classical music recordings. We extended the source separation framework proposed in [24] for the case of monoaural score informed source separation by refining the gains. The approach increases the sparseness of the gains initialization, achieving better performance than the implicit approach of estimating the onset with a parametric model, as [12,15,17], especially for input scores having large local misalignments. Particularly, the proposed system reduces the interference, resulting in higher SIR values. Additionally, the method improves the alignment rate over the one in [20], and is more robust because it uses meaningful timbre information.

As future work, the selection of the best blob and the binarization threshold could be included into the factorization framework through the cost function. Moreover, we plan to test our method with more complex orchestral recordings, and for multi-channel source separation.

7. ACKNOWLEDGEMENTS

This work was supported by the European Commission, FP7 (Seventh Framework Programme), STREP project, ICT-2011.8.2 ICT for access to cultural resources, grant agreement No 601166. Phenicx Project

8. REFERENCES

[1] Bach10 dataset source separation demo. `https://dl.dropboxusercontent.com/u/80928189/demos/index.html`.

[2] L. Benaroya, F. Bimbot, and R. Gribonval. Audio source separation with a single sensor. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(1):191–199, 2006.

[3] J.J. Bosch, K. Kondo, R. Marxer, and J. Janer. Score-informed and timbre independent lead instrument separation in real-world scenarios. In *Signal Processing Conference (EUSIPCO)*, pages 2417–2421, Aug 2012.

[4] 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 J. Adv. Sig. Proc.*, 2013:184, 2013.

[5] 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. *ISMIR*, 2015.

[6] J. J. Carabias-Orti, T. Virtanen, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Canadas-Quesada. Musical Instrument Sound Multi-Excitation Model for Non-Negative Spectrogram Factorization. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1144–1158, October 2011.

[7] A. Cont, D. Schwarz, N. Schnell, and C. Raphael. Evaluation of real-time audio-to-score alignment. In *ISMIR*, 2007.

[8] S. Dixon. Match: A music alignment tool chest. In *ISMIR*, 2005.

[9] Z. Duan and B. Pardo. Soundprism: An online system for score-informed source separation of music audio. *IEEE Journal of Selected Topics in Signal Processing*, pages 1–12, 2011.

[10] J.L. Durrieu, A. Ozerov, and C. Févotte. Main instrument separation from stereophonic audio signals using a source/filter model. *EUSIPCO*, (1), 2009.

[11] 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*, 19(7):2046–2057, September 2011.

[12] S. Ewert and M. Müller. Score-Informed Voice Separation For Piano Recordings. *ISMIR*, pages 245–250, 2011.

[13] S. Ewert, M. Müller, and P. Grosche. High resolution audio synchronization using chroma onset features. In *ICASSP*, pages 1869–1872. IEEE, 2009.

[14] C. Févotte, N. Bertin, and JL. Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830, 2009.

[15] J. Fritsch and M. Plumbley. Score informed audio source separation using constrained non-negative matrix factorization andscore synthesis. *ICASSP*, pages 888–891, 2013.

[16] M. Goto. Development of the rwc music database. In *ICA*, pages 553–556, 2004.

[17] R. Hennequin, B. David, and R. Badeau. Score informed audio source separation using a parametric model of non-negative spectrogram. *ICASSP*, (1), 2011.

[18] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis. Singing-voice separation from monaural recordings using deep recurrent neural networks. In *ISMIR*, 2014.

[19] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[20] M. Miron, J.J. Carabias, and J. Janer. Audio-to-score alignment at the note level for orchestral recordings. In *ISMIR*, 2014.

[21] B. Niedermayer. *Accurate Audio-to-Score Alignment Data Acquisition in the Context of Computational Musicology*. PhD thesis, Johannes Kepler Universität, 2012.

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

[23] C. Raphael. A classifier-based approach to score-guided source separation of musical audio. *Comput. Music J.*, 32(1):51–59, March 2008.

[24] 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 *LVA/ICA*, pages 380–387, 2012.

[25] D.L. Sun and C. Fevotte. Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6201–6205, May 2014.

[26] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(4):1462–1469, July 2006.

[27] S. Wang, S. Ewert, and S. Dixon. Compensating for asynchronies between musical voices in score-performance alignment. In *ICASSP*, pages 589–593, Brisbane, Australia, 2015.