On-the-fly audio source separation - a novel user-friendly framework

Dalia El Badawy, Ngoc Q K Duong, Alexey Ozerov

On-the-fly audio source separation - a novel user-friendly framework

Dalia El Badawy, Ngoc Q. K. Duong, *Member, IEEE* and Alexey Ozerov, *Member, IEEE*

***Abstract*—This article addresses the challenging problem of single-channel audio source separation. We introduce a novel user-guided framework where source models that govern the separation process are learned *on-the-fly* from audio examples retrieved online. The user only provides the search keywords that describe the sources in the mixture. In this framework, the generic spectral characteristics of each source are modeled by a *universal sound class model* learned from the retrieved examples via non-negative matrix factorization. We propose several group sparsity-inducing constraints in order to efficiently exploit a relevant subset of the universal model adapted to the mixture to be separated. We then derive the corresponding multiplicative update rules for parameter estimation. Separation results obtained from automated and user tests on mixtures containing various types of sounds confirm the effectiveness of the proposed framework.**

***Index Terms*—On-the-fly audio source separation, user-guided, non-negative matrix factorization, group sparsity, universal sound class model.**

I. Introduction

AUDIO source separation is a desired processing step within many real-world applications such as sound post-production, robotics, and audio enhancement [1]. However, it has remained a challenging task especially when the input is a single-channel mixture. Indeed, in this case the problem is highly ill-posed and, in contrast to multichannel mixing case, additional spatial information about the sources is not available. Earlier approaches usually assume that the sources are sparse in the short-time Fourier transform (STFT) domain and estimate the predominant source's STFT coefficients via *e.g.* binary masking [2] or ℓ_1-minimization [3], [4]. The separation performance achievable by these techniques is very limited in reverberant environments [5], [6] where the sources' STFT coefficients are quite overlapped. A more recent class of algorithms known as *informed* source separation [7], [8] utilizes prior information about the sources to guide the separation process, and was shown to be successful in many contexts using different types of prior information. For instance, such information may include musical scores of the corresponding music sources [7], [9], [10] or text of the corresponding speech sources [8]. In some approaches this symbolic information is then converted to audio using a MIDI synthesizer for musical scores [9], [10] or a speech synthesizer for text [8]. These synthesized signals (that may also include cover tracks as in [11]) called *deformed references* in [12] can be used to roughly learn the spectral and temporal characteristics of one or more sources in the mixtures so as to guide the separation process [8]–[10], [12]. A subclass of informed source separation approaches is *user-guided* separation methods where the prior information is provided by a user. Such information can be *e.g.*, user-"hummed" sounds that mimic the sources in the mixture [13] or source activity annotation along time [14] or in a time-frequency plane [15]; the annotation information is then used, instead of training data, to guide the separation process. Furthermore, recent publications disclose an interactive strategy [16], [17] where the user can perform annotations on the spectrogram of intermediate separation results to gradually correct the remaining errors. Note however that most of the existing approaches need to use prior information which may not be easy to acquire in advance (*e.g.*, musical score, text transcript), is difficult to produce (*e.g.*, user-hummed examples), or simply requires very experienced users while being very time consuming (*e.g.*, time-frequency annotations).

The main motivation of this work is to introduce a simple framework that enables everyone to easily perform source separation. We hence present the new concept of *on-the-fly* source separation inspired by on-the-fly visual search methods [18], [19] from the computer vision research community. More specifically, the proposed framework only requires the user to listen to the mixture and type some keywords that describe the sources to be separated; in other words, the user interaction is now carried out at a higher semantic level. For instance, a user would request to separate the "wind noise" (source 1 description) from the "bird song" (source 2 description). The given descriptions or keywords are then used to search the internet for similar audio examples that will be employed to govern the separation process. For this purpose, supervised approaches based on *e.g.*, nonnegative matrix factorization (NMF) [20], [21] or its probabilistic formulation known as Probabilistic Latent Component Analysis (PLCA) [13], [22], where retrieved examples can be used to pre-learn the spectral dictionaries of the corresponding sources, are of great interest. Other methods in the prior art that couple the decomposition of the reference signals together with the mixture could also be considered [8], [11], [12], [23]. Regardless of the approach, several challenges, as detailed in Section II, arise in this on-the-fly framework due to (i) the unknown quality of the retrieved examples and (ii) possible lack of source descriptions (*i.e.* some sources may not be described by the user). In our preliminary work [24], we investigated several strategies to handle issues with the quality of the retrieved

D. El Badawy is a student at EPFL, Switzerland, e-mail: (dalia.elbadawy@epfl.ch).

N. Q. K. Duong and A. Ozerov are with Technicolor R&I, France, e-mail: (quang-khanh-ngoc.duong@technicolor.com; alexey.ozerov@technicolor.com).

This work was done while the first author was an intern at Technicolor.

Manuscript received Month xx, 2016; revised Month xx, 2016.

examples and found that the one using a *universal sound class model (USCM)*[1] learned from examples via NMF with a *group sparsity* constraint is generally more efficient than the others. Note that since the USCM is actually an over-complete dictionary, a sparsity constraint is needed to help fit the most relevant spectral patterns to the sources in the mixture.

This article extends our preliminary work [24], [27] by providing the algorithms along with their mathematical derivations in addition to new results from a user test. Altogether, the main contributions of our proposed on-the-fly paradigm work are four-fold:

- We introduce a general framework for on-the-fly audio source separation which greatly simplifies the user interaction.
- We propose a novel so called *relative group sparsity* constraint and show its benefit in the semi-supervised case where training examples for some sources are missing.
- We derive several algorithms for parameter estimation when different group sparsity-inducing penalties and relative group sparsity-inducing penalties are used.
- We perform a range of evaluations, including both supervised and semi-supervised scenarios, and a user-test to validate the benefit of the proposed framework.

The remainder of this paper is organized as follows. Section II gives an overview of the on-the-fly framework and the related challenges. In Section III, we recall some background on supervised source separation based on NMF. We then propose several algorithms for parameter estimation with the use of different sparsity-inducing constraints in Section IV. Evaluation results with a user-test are presented in Section V. Finally, we conclude in Section VI.

II. On-the-fly framework and challenges

A. Overview and challenges

The proposed framework only requires minimal user input enabling inexperienced users to apply source separation to essentially any mixture.It is applicable as well when relevant training examples for some sources are either not readily available offline or not representative enough, which is likely the case for uncommon sounds such as animal or environmental sounds. The general workflow is shown in Fig. 1. The user inputs a few keywords specifying the sources in the mixture (*e.g.*, "dog barking", "wind", etc.,), then a search engine retrieves relevant source examples accordingly. The source spectral models are then learned on-the-fly and used for supervising the separation. This approach is actually analogous to on-the-fly methods in visual search where a user types a persons name (e.g., "Clint Eastwood") [18] or an objects description (e.g., "car") [19] and a classifier is trained using example images retrieved via Google Image Search.

Although the on-the-fly approach simplifies the user interaction and eliminates the need for offline training samples, there are several challenges that need to be addressed as follows:

- (C_1) *Handling irrelevant examples*: Some retrieved examples may contain sounds with entirely different spectral characteristics than those of the source in the mixture, *e.g.*, searching for "bird chirps" and obtaining some "chirp signal" examples too. Those examples should not be used in training.
- (C_2) *Handling noisy examples*: Some retrieved examples are actually mixtures of relevant and irrelevant sounds, *e.g.*, "female speech" with a dog barking in the background. Those examples may still be useful and should not be discarded entirely.
- (C_3) *Handling missing examples*: This may happen when the user describes only the sources of interest and ignores the remaining sources or when the search engines do not return results for some of the provided keywords. We refer to this challenge as the semi-supervised case where all non-described sources that possibly appear in the mixture should be grouped as one background source.

In fact in our previous work [24] to handle the first challenge, we investigated the use of a simple example preselection scheme based on the spectral similarity between the examples and the mixture to discard irrelevant examples. Thus, one can imagine having additional user interaction after specifying the keywords. For instance, the user may screen the list of retrieved examples and subjectively select a more relevant subset for training. This is the "Examples Refinement" step in Fig. 1.

B. Graphical User Interface

We implemented the system along with a graphical user interface (GUI) as shown in Fig. 2 and employed it for our user-tests. It features the ability to listen to a mixture and input one or more keywords describing the different sources. Then, per source, an online search for audio is performed. Next, the user can listen to the list of retrieved audio examples as well as view their waveforms or spectrograms (useful for the more advanced users). The optional example selection is then done by ticking the corresponding checkboxes. USCMs are then learned on-the-fly to guide the separation. The last step is to output the separated sources. A video showing a demo is available online at http://youtu.be/mBmJW7cy710/. On the practical side, the data transferred between the user and the server consists of the keywords and the mixture file as well as the separated sources which are sent back to the user. On the server, each example file requires computing the STFT followed by NMF; the examples are independent and these operations can thus be done in parallel. Then once the USCMs have all been constructed, the separation step is faster as the multiplicative updates are performed only one time. The overall complexity thus mostly depends on the number of training examples and the size of USCMs. Thus on an average PC, it would take from 30 seconds to a few minutes to get the separation results back.

[1]The term "universal speech model" was introduced in [25] for the separation of speech and noise, and was inspired by the term "universal background model" used for speaker verification [26]. We here extend it to "universal sound class model", since our framework deals with the separation of sources belonging to any sound class.

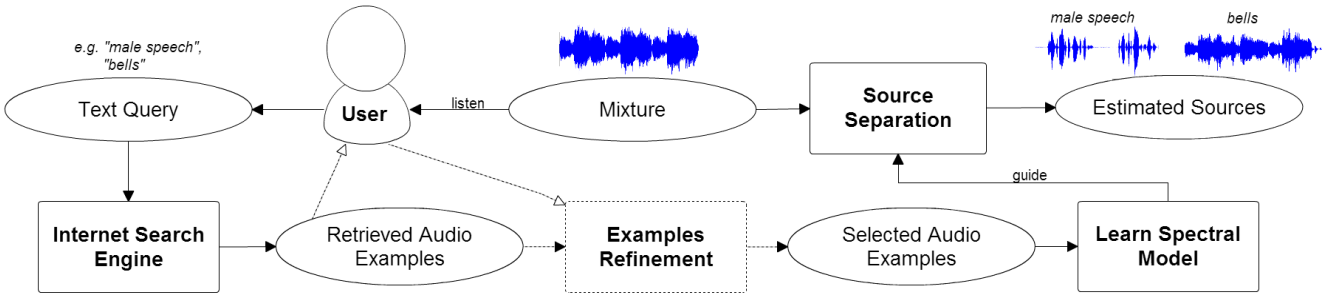
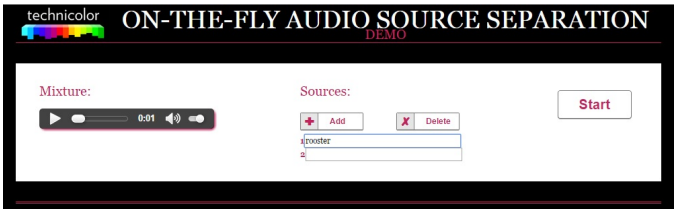
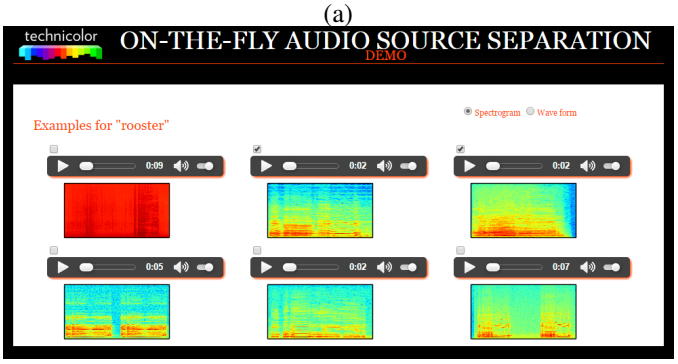


Fig. 1. General workflow of the proposed on-the-fly framework. A user listens to the mixture and types some keywords describing the sources. These keywords are then used to retrieve examples to learn spectral models for the described sources. Optionally, the user may listen to the retrieved examples and discard irrelevant ones.



(a)



(b)

Fig. 2. Screenshots of the proposed GUI. (a) The user can listen to a mixture, and then type keywords describing different sources within it. (b) A set of retrieved examples (waveforms or spectrograms can be displayed) for each source with checkboxes so that the user can select the most appropriate ones to be used for training the source spectral model.

III. Background on NMF-based Source Separation

A. Conventional supervised approach

We discuss in this section a standard supervised source separation approach. We base our framework on NMF since it is one of the most popular and well-suited models in the state of the art on audio source separation. As per *e.g.* [22], [25], first source spectral models are learned on-the-fly from training data retrieved online. Then these models are used to supervise the separation.

Assuming J sources, let $\mathbf{X} \in \mathbb{C}^{F\times N}$ and $\mathbf{S}_j \in \mathbb{C}^{F\times N}$ be the STFT coefficients of the single channel mixture signal and the j-th source signal, respectively, where F is the number of frequency bins and N the number of time frames. Usual additive mixing is assumed as

$$\mathbf{X} = \sum_{j=1}^{J} \mathbf{S}_j. \tag{1}$$

Let $\mathbf{V} = |\mathbf{X}|^{\cdot 2}$ be the power spectrogram of the mixture, where $\mathbf{X}^{\cdot p}$ is the matrix with entries $[\mathbf{X}]_{il}^{p}$, $^{\cdot p}$ denotes an element-wise operation. Then, NMF algorithms construct two non-negative matrices $\mathbf{W} \in \mathbb{R}^{F\times K}$ and $\mathbf{H} \in \mathbb{R}^{K\times N}$ such that $\mathbf{V} \approx \hat{\mathbf{V}} = \mathbf{WH}$. The factorization is usually done by solving the following optimization problem [20], [28]

$$\mathbf{W}^*, \mathbf{H}^* = \underset{\mathbf{H}\geq 0, \mathbf{W}\geq 0}{\arg\min}\ D(\mathbf{V}\|\mathbf{WH}), \tag{2}$$

where

$$D(\mathbf{V}\|\hat{\mathbf{V}}) = \sum_{f,n=1}^{F,N} d(\mathbf{V}_{fn}\|\hat{\mathbf{V}}_{fn}) \tag{3}$$

and $d(\cdot\|\cdot)$ is a scalar divergence measure. We use the Itakura-Saito (IS) divergence defined as

$$d_{IS}(x\|y) = \frac{x}{y} - \log\left(\frac{x}{y}\right) - 1 \tag{4}$$

which is appropriate for audio signals due to its scale invariance [20]. The parameters $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{H}\}$ are usually initialized with random non-negative values and are iteratively updated via multiplicative update (MU) rules [20], [28].

In the training step of the supervised setting, a spectral model for each source j, denoted by $\mathbf{W}_{(j)}$, is first learned from the corresponding training examples concatenated together by optimizing criterion (2). Then the spectral model for all sources $\mathbf{W}$ is obtained by concatenating the source models as:

$$\mathbf{W} = [\mathbf{W}_{(1)}, \ldots, \mathbf{W}_{(J)}]. \tag{5}$$

Then in the separation step, the time activation matrix $\mathbf{H}$ is estimated via the MU rules optimizing (2) [20], while $\mathbf{W}$ is kept fixed. Note that the activation matrix is also partitioned into horizontal blocks as

$$\mathbf{H} = [\mathbf{H}_{(1)}^T, \ldots, \mathbf{H}_{(J)}^T]^T, \tag{6}$$

where $\mathbf{H}_{(j)}$ denotes the block characterizing the time activations for the j-th source.

Once the parameters $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{H}\}$ are obtained, Wiener filtering is applied to compute the source STFT coefficients as

$$\hat{\mathbf{S}}_j = \frac{\mathbf{W}_{(j)}\mathbf{H}_{(j)}}{\mathbf{WH}} \odot \mathbf{X}, \tag{7}$$

where $\odot$ denotes the element-wise Hadamard product and the division is also element-wise. Finally, the inverse STFT is computed to produce the time domain source estimates.

B. USCM-based approach

The conventional supervised approach as described in Section III-A assumes using all retrieved (or user-selected) examples for a given source to learn the source spectral model. This may not be suitable in the current framework due to the challenges mentioned in Section II where the noisy examples may lead to a poor spectral model. Thus, in this section we propose an efficient and flexible approach to better utilize the examples, when available, for guiding the separation, while also handling the case of missing examples. In the following, the training examples refer to either the full list of retrieved examples or the user-selected examples in case of user intervention. We employ a so-called *universal sound class model*, learned in advance from training examples, with sparsity constraints on the activation matrix $\mathbf{H}$ in order to enforce the selection of only few representative spectral patterns during the model fitting. In the following, we first present the USCM construction, and then the optimization criterion for model fitting.

1) USCM construction: Assuming that the j-th source is described by the user and some examples are retrieved for it, we denote by $\mathbf{V}_{jp}$ the spectrogram of the p-th example corresponding to the j-th source. First, $\mathbf{V}_{jp}$ is used to learn the NMF spectral model, denoted by $\mathbf{W}_{jp}$, by optimizing the criterion (similar to (2)):

$$\mathbf{H}^*_{jp}, \mathbf{W}^*_{jp} = \underset{\mathbf{H}_{jp}\geq 0, \mathbf{W}_{jp}\geq 0}{\arg\min}\ D(\mathbf{V}_{jp}\|\mathbf{W}_{jp}\mathbf{H}_{jp}), \tag{8}$$

where $\mathbf{H}_{jp}$ is the corresponding time activation matrix. Given $\mathbf{W}_{jp}$ for all examples, the USCM for the j-th source is constructed as

$$\mathbf{W}_{(j)} = [\mathbf{W}_{j1}, \ldots, \mathbf{W}_{jP_j}] \tag{9}$$

where P_j is the number of retrieved examples for the j-th source.

2) Model fitting for supervised source separation: In the supervised setting, we assume having source models for all the sources in the mixture, that is to say that for every source, the user gave its description and examples were successfully retrieved. It can be seen that the USCM $\mathbf{W}_{(j)}$ constructed in (9) becomes a large matrix when the number of examples increases, and it is often redundant since different examples may share similar spectral patterns. Therefore, in the NMF decomposition of the mixture, the need for a sparsity constraint arises to fit only a subset of each $\mathbf{W}_{(j)}$ to the source in the mixture. In other words, the mixture is decomposed in a supervised manner, given $\mathbf{W}$ constructed from $\mathbf{W}_{(j)}$ as in (5) and fixed, by solving the following optimization problem

$$\mathbf{H}^* = \underset{\mathbf{H}\geq 0}{\arg\min}\ D(\mathbf{V}\|\mathbf{WH}) + \Psi(\mathbf{H}) \tag{10}$$

where $\Psi(\mathbf{H})$ denotes a penalty function imposing sparsity on the activation matrix $\mathbf{H}$. Different penalties can be chosen, as will be discussed in Section IV, resulting in a sparse matrix $\mathbf{H}$ as visualized in Fig. 3b and Fig. 3c.

3) Model fitting for semi-supervised source separation: We describe in this section a so-called semi-supervised setting where not all of the source models can be learned in advance [25]. This occurs either when the user only describes the sources of interest and not all of them or when the search engine fails to retrieve examples for a given query.

We propose to model all the "missing" sources as one background source whose spectrogram can be approximately factorized as $\mathbf{W}_\mathrm{b}\mathbf{H}_\mathrm{b}$, where $\mathbf{W}_\mathrm{b}$ and $\mathbf{H}_\mathrm{b}$ are the corresponding spectral model and activation matrix, respectively. The parameter $\boldsymbol{\theta}_\mathrm{b} = \{\mathbf{W}_\mathrm{b}, \mathbf{H}_\mathrm{b}\}$ can be randomly initialized with a small number of components (*i.e.* number of columns in $\mathbf{W}_\mathrm{b}$) K_b. All the other sources, for which some examples are available, are modeled as in the supervised case by $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{H}\}$ (see Fig. 4e and Fig. 4f). The parameters are estimated altogether by optimizing the following criterion

$$\mathbf{H}^*, \mathbf{W}^*_\mathrm{b}, \mathbf{H}^*_\mathrm{b} = \underset{\mathbf{H}\geq 0, \mathbf{W}_\mathrm{b}\geq 0, \mathbf{H}_\mathrm{b}\geq 0}{\arg\min}\ D(\mathbf{V}\|\mathbf{WH} + \mathbf{W}_\mathrm{b}\mathbf{H}_\mathrm{b}) + \Psi(\mathbf{H}). \tag{11}$$

We see that in contrast to criterion (10) $\mathbf{W}_\mathrm{b}$ is updated as well and there is no group sparsity-inducing penalty on $\mathbf{H}_\mathrm{b}$. The reason is that, as opposed to $\mathbf{W}$, $\mathbf{W}_\mathrm{b}$ is neither an overcomplete dictionary nor has an underlying structure that can be exploited for regularization.

IV. Sparsity constraints and algorithms

In this section we consider two classes of sparsity constraints, namely *group sparsity* and a newly proposed *relative group sparsity* for the optimization problem (10) and (11). In each case, two variations are considered: a *block* sparsity-inducing penalty and a *component* sparsity-inducing penalty. For every constraint, we give the corresponding algorithm for estimating the parameters.

A. Group sparsity constraints and parameter estimation algorithm

We consider a group sparsity-inducing penalty defined as

$$\Psi_\mathrm{gr}(\mathbf{H}) = \sum\nolimits_{j=1}^{J} \lambda_j \sum\nolimits_{g=1}^{G_j} \log\left(\epsilon + \|\mathbf{H}_{(j,g)}\|_1\right), \tag{12}$$

where $\mathbf{H}_{(j,g)}$ $(g = 1, \ldots, G_j)$ are the groups within the activation sub-matrix $\mathbf{H}_{(j)}$ corresponding to the j-th USCM (see equation (6) for the definition of $\mathbf{H}_{(j)}$), G_j the total number of groups for the j-th source, $\|\cdot\|_1$ denotes the ℓ_1 matrix norm, $\epsilon > 0$ and $\lambda_j \geq 0$ are trade-off parameters determining the contribution of the penalty for each source. Note that in the remainder of the paper, $\mathbf{H}_{(j,g)}$ should not be confused with $\mathbf{H}_{jp}$ in (8). We introduce two options for defining the groups $\mathbf{H}_{(j,g)}$ and derive the corresponding MU rules for the parameter estimation as follows.

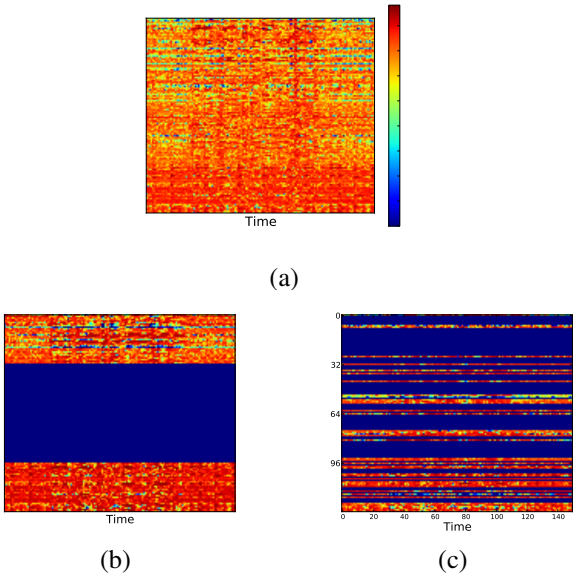


Fig. 3. Estimated activation matrix $\mathbf{H}$ for two sources in a mixture containing a rooster and bird chirps where two retrieved examples for each source were used for training the USCMs: (a) without a sparsity constraint, (b) with a block sparsity-inducing penalty (blocks corresponding to poorly fitting models are zero), and (c) with a component sparsity-inducing penalty (rows corresponding to poorly fitting spectral components from different models are zero).

1) Block sparsity-inducing penalty: As in [25], we restrict the groups to be sub-matrices of $\mathbf{H}_{(j)}$ corresponding to the spectral models $\mathbf{W}_{(j,p)}$ trained using the p-th example (see (8) for the definition of $\mathbf{W}_{(j,p)}$). In that case the indices g and p coincide and $G_j = P_j$. This so-called *block sparsity*-inducing strategy allows filtering out irrelevant spectral models $\mathbf{W}_{(j,l)}$, thus dealing with irrelevant retrieved examples (challenge C_1 in Section II). An illustration for the estimated activation matrix $\mathbf{H}$ for that case is shown in Fig. 3b where blocks corresponding to irrelevant examples for each source are set to zero.

2) Component sparsity-inducing penalty: As an alternative solution to fitting the universal model, we restrict the groups to be rows of $\mathbf{H}_{(j)}$ corresponding to different spectral components (in that case the number of groups G_j simply equals to the number of rows in $\mathbf{H}_{(j)}$). This so-called *component sparsity*-inducing strategy allows filtering out irrelevant spectral components, thus dealing with noisy retrieved examples (challenge C_2 in Section II). Fig. 3c shows an estimated activation matrix $\mathbf{H}$ where rows corresponding to irrelevant spectral components for each source are set to zero.

3) MU rules for parameter estimation: MU rules for the optimization of criterion (10) (supervised case) and (11) (semi-supervised case) are summarized in Algorithms 1 and 2, respectively, where $\eta > 0$ is a constant parameter, $\mathbf{P}_{(j,g)}$ is a matrix of the same size as $\mathbf{H}_{(j,g)}$ whose entries have the same value, and $\mathbf{P}$ is a matrix concatenating all $\mathbf{P}_{(j,g)}$. This algorithm is almost identical to the one proposed in [21], except that the groups are defined differently and $\mathbf{W}$ is not updated. It is proven in [21] using a majorization-minimization [29] formulation that these updates with $\eta = 1/2$ are *monotonic*, i.e., the cost function is non-increasing after each iteration.

Algorithm 1 MU rules for NMF with group sparsity in supervised case

Input: $\mathbf{V}$, $\mathbf{W}$, $\boldsymbol{\lambda}$
Output: $\mathbf{H}$
  Initialize $\mathbf{H}$ randomly
  $\hat{\mathbf{V}} = \mathbf{W}\mathbf{H}$
  repeat
    for $j = 1, \ldots, J, g = 1, \ldots, G_j$ **do**
      $\mathbf{P}_{(j,g)} \leftarrow \frac{\lambda_j}{\epsilon + \|\mathbf{H}_{(j,g)}\|_1}$
    end for
    $\mathbf{P} = [\mathbf{P}_{(1,1)}^T, \ldots, \mathbf{P}_{(1,G_1)}^T, \ldots, \mathbf{P}_{(J,1)}^T, \ldots, \mathbf{P}_{(J,G_J)}^T]^T$
    $\mathbf{H} \leftarrow \mathbf{H} \odot \left(\frac{\mathbf{W}^T(\mathbf{V} \odot \hat{\mathbf{V}}^{\cdot -2})}{\mathbf{W}^T \hat{\mathbf{V}}^{\cdot -1} + \mathbf{P}} \right)^{\cdot \eta}$
    $\hat{\mathbf{V}} \leftarrow \mathbf{W}\mathbf{H}$
  until convergence

Algorithm 2 MU rules for NMF with group sparsity in semi-supervised case

Input: $\mathbf{V}$, $\mathbf{W}$, $\boldsymbol{\lambda}$
Output: $\mathbf{H}, \mathbf{H}_b, \mathbf{W}_b$
  Initialize $\mathbf{H}$, $\mathbf{H}_b$, and $\mathbf{W}_b$ randomly
  $\hat{\mathbf{V}} = \mathbf{W}\mathbf{H} + \mathbf{W}_b\mathbf{H}_b$
  repeat
    for $j = 1, \ldots, J, g = 1, \ldots, G_j$ **do**
      $\mathbf{P}_{(j,g)} \leftarrow \frac{\lambda_j}{\epsilon + \|\mathbf{H}_{(j,g)}\|_1}$
    end for
    $\mathbf{P} = [\mathbf{P}_{(1,1)}^T, \ldots, \mathbf{P}_{(1,G_1)}^T, \ldots, \mathbf{P}_{(J,1)}^T, \ldots, \mathbf{P}_{(J,G_J)}^T]^T$
    $\mathbf{H} \leftarrow \mathbf{H} \odot \left(\frac{\mathbf{W}^T(\mathbf{V} \odot \hat{\mathbf{V}}^{\cdot -2})}{\mathbf{W}^T(\hat{\mathbf{V}}^{\cdot -1}) + \mathbf{P}} \right)^{\cdot \eta}$
    $\mathbf{H}_b \leftarrow \mathbf{H}_b \odot \left(\frac{\mathbf{W}_b^T(\mathbf{V} \odot \hat{\mathbf{V}}^{\cdot -2})}{\mathbf{W}_b^T \hat{\mathbf{V}}^{\cdot -1}} \right)^{\cdot \eta}$
    $\mathbf{W}_b \leftarrow \mathbf{W}_b \odot \left(\frac{(\mathbf{V} \odot \hat{\mathbf{V}}^{\cdot -2})\mathbf{H}_b^T}{\hat{\mathbf{V}}^{\cdot -1}\mathbf{H}_b^T} \right)^{\cdot \eta}$
    Normalize $\mathbf{W}_b$ and $\mathbf{H}_b$ component-wise (see, e.g., [20])
    $\hat{\mathbf{V}} \leftarrow \mathbf{W}\mathbf{H} + \mathbf{W}_b\mathbf{H}_b$
  until convergence

Note that the updates of $\mathbf{H}$ are identical in both the supervised and semi-supervised cases. Additionally, in the semi-supervised case, since the derivations of (11) with respect to $\mathbf{W}_\mathrm{b}$ and $\mathbf{H}_\mathrm{b}$ are not affected by the sparsity constraint $\Psi(\mathbf{H})$, the updates of $\mathbf{W}_\mathrm{b}$ and $\mathbf{H}_\mathrm{b}$ are straightforwardly derived as in [30].

B. Relative group sparsity constraints and parameter estimation algorithm

For the separation to be feasible, we require that every learned source model has a corresponding non-zero activation; however, this constraint is not enforced by the group sparsity penalty in (12) where it can happen that a group of different sources are fit together using the same source model, instead of separately using their designated models, rendering their separation impossible. We observed this "source vanishing" phenomenon in practice as illustrated in Fig. 4a (in case of

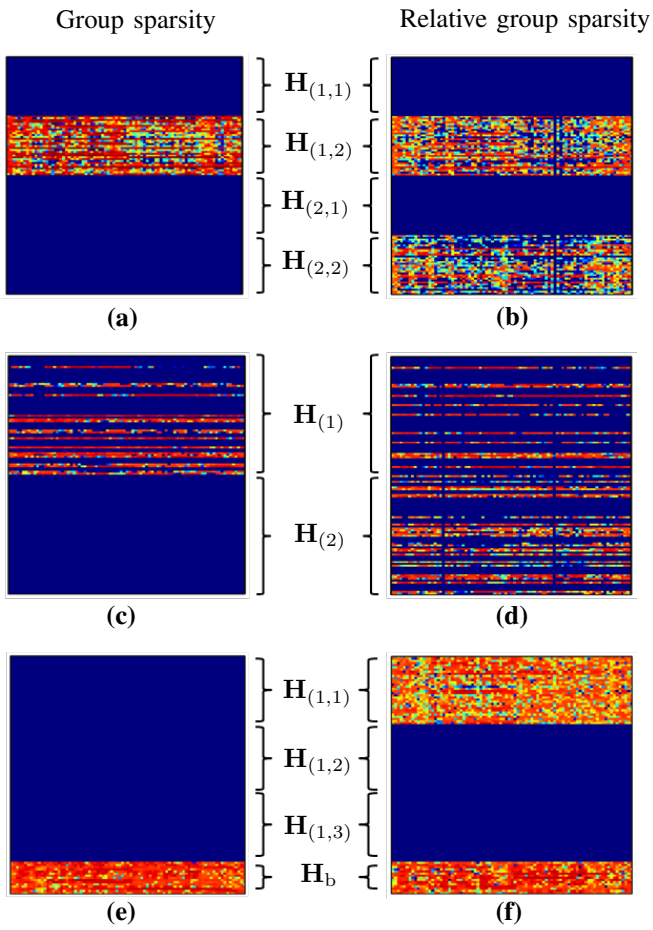


Fig. 4. Examples of estimated activation matrices $\mathbf{H}$ for two sources in a mixture containing a rooster and bird chirps where two retrieved examples for each source were used for training the USCMs. Left column: **(a)** block sparsity in the supervised case, **(c)** component sparsity in the supervised case, and **(e)** block sparsity in the semi-supervised case. Right column: same settings as in the left column, but for the proposed relative block/component sparsity.

using the block sparsity-inducing penalty) and Fig. 4c (in case of using the component sparsity-inducing penalty) in the supervised case. Moreover, the problem worsens in the semi-supervised case, depicted in Fig. 4e for the block sparsity case, where the entire mixture is fit by the estimated background model only (the same effect occurs for the component sparsity case). This is due to the fact that $\mathbf{W}_{\mathrm{b}}$ and $\mathbf{H}_{\mathrm{b}}$ are now fully unconstrained in (11), whereas $\mathbf{W}$ is fixed and $\mathbf{H}$ is constrained by the group sparsity-inducing penalty. It can also be seen that increasing the trade-off parameters λ_j in the penalty (12) (thus decreasing the number of active groups) increases the chances of source vanishing in both the supervised and semi-supervised cases. In this section, we present a novel sparsity-inducing penalty which helps prevent this problem completely.

1) Relative group sparsity-inducing penalties: This observation motivates us to introduce a general solution based on a new notion, namely *relative group sparsity*. While we present it here within the context of NMF, the idea extends to other dictionary decomposition schemes. We assume that the groups are organized into so-called *supergroups* (*i.e.* $\mathbf{H}_{(j)}$ corresponding to a USCM is considered as a supergroup), and we characterize the relative group sparsity constraint $\Psi(\mathbf{H})$ by the following properties

- It induces the sparsity of the groups (as in group sparsity), and at the same time
- It induces the anti-sparsity of the supergroups (*i.e.* prevents them from vanishing entirely).

In other words, the group sparsity property is now considered *relative* to the corresponding supergroup $\mathbf{H}_{(j)}$ and not within the full set of coefficients in $\mathbf{H}$. It is formulated as [27]

$$\Psi_{\mathrm{rel}}(\mathbf{H}) = \sum_{j=1}^{J} \lambda_j \sum_{g=1}^{G_j} \log\left(\frac{\epsilon + \|\mathbf{H}_{(j,g)}\|_1}{\|\mathbf{H}_{(j)}\|_1^{\gamma_j}}\right), \quad (13)$$

where γ_j are some non-negative constants. Penalty (13) can be also rewritten as

$$\Psi_{\mathrm{rel}}(\mathbf{H}) = \Psi_{\mathrm{gr}}(\mathbf{H}) - \sum_{j=1}^{J} \lambda_j \gamma_j G_j \log\left(\|\mathbf{H}_{(j)}\|_1\right), \quad (14)$$

and one can easily see that, while the new penalty keeps the group sparsity property thanks to $\Psi_{\mathrm{gr}}(\mathbf{H})$ defined in (12), it prevents (when $\gamma_j > 0$) the supergroups from vanishing since if $\|\mathbf{H}_{(j)}\|_1$ tends to zero, then $-\log\left(\|\mathbf{H}_{(j)}\|_1\right)$ tends to $+\infty$. This formulation generalizes the group sparsity constraint in the sense that (13) reduces to (12) for $\gamma_j = 0$. So while we only require that γ_j is non-zero for the relative group sparsity to be active, in our experiments we show results for $\gamma_j = 1$ and $\gamma_j = \frac{1}{G_j}$. The latter was chosen to act as a normalization such that the effect of the penalty is even across the USCMs regardless of their size.

Note also that one can introduce either the *relative block sparsity-inducing penalty* or the *relative component sparsity-inducing penalty* by defining a group $\mathbf{H}_{(j,g)}$ to be either a block or a row in $\mathbf{H}$ similar to what has been presented in Section IV-A.

2) MU rules for parameter estimation: MU rules for parameter estimation when using the new penalty $\Psi_{\mathrm{rel}}(\mathbf{H})$ are derived in the same way as the rules for group sparsity in Section IV-A. The resulting algorithms for both supervised and semi-supervised cases are summarized in Algorithm 3 and Algorithm 4, respectively. Details on the derivation of Algorithm 3 are given in the Appendix, and that of Algorithm 4 is very similar. Note that $\mathbf{P}_{(j,g)}$ and $\mathbf{Q}_{(j,g)}$ are matrices of the same size as $\mathbf{H}_{(j,g)}$ whose entries have the same value, and $\mathbf{P}$ and $\mathbf{Q}$ are concatenations of $\mathbf{P}_{(j,g)}$ and $\mathbf{Q}_{(j,g)}$, respectively.

V. Experiments

We start by describing the data set, parameter settings, and evaluation metrics in Section V-A. We then evaluate the performance of the proposed supervised and semi-supervised on-the-fly audio source separation algorithms in Section V-B. The sensitivity of the different algorithms with respect to the choice of the trade-off parameter λ_j which determines the contribution of the sparsity penalty is presented in Section V-C. We finally present user-test results in Section V-D.

A. Data, parameter settings, and evaluation metrics

We evaluated the performance of the proposed on-the-fly algorithms on a data set of 15 single-channel mixtures of

Algorithm 3 MU rules for NMF with relative group sparsity in the supervised case

Input: $\mathbf{V}$, $\mathbf{W}$, $\boldsymbol{\lambda}$
Output: $\mathbf{H}$
  Initialize $\mathbf{H}$ randomly
  $\hat{\mathbf{V}} = \mathbf{WH}$
  repeat
    for $j = 1, \ldots, J, g = 1, \ldots, G_j$ **do**
      $\mathbf{P}_{(j,g)} \leftarrow \frac{\lambda_j}{\epsilon + \|\mathbf{H}_{(j,g)}\|_1}$
      $\mathbf{Q}_{(j,g)} \leftarrow \frac{\lambda_j G_j \gamma_j}{\|\mathbf{H}_{(j)}\|_1}$
    end for
    $\mathbf{P} = [\mathbf{P}_{(1,1)}^T, \ldots, \mathbf{P}_{(1,G_1)}^T, \ldots, \mathbf{P}_{(J,1)}^T, \ldots, \mathbf{P}_{(J,G_J)}^T]^T$
    $\mathbf{Q} = [\mathbf{Q}_{(1,1)}^T, \ldots, \mathbf{Q}_{(1,G_1)}^T, \ldots, \mathbf{Q}_{(J,1)}^T, \ldots, \mathbf{Q}_{(J,G_J)}^T]^T$
    $\mathbf{H} \leftarrow \mathbf{H} \odot \left(\frac{\mathbf{W}^T(\mathbf{V} \odot \hat{\mathbf{V}}^{\cdot -2}) + \mathbf{Q}}{\mathbf{W}^T(\hat{\mathbf{V}}^{\cdot -1}) + \mathbf{P}} \right)^{\cdot \eta}$
    $\hat{\mathbf{V}} \leftarrow \mathbf{WH}$
  until convergence

Algorithm 4 MU rules for NMF with relative group sparsity in the semi-supervised case

Input: $\mathbf{V}$, $\mathbf{W}$, $\boldsymbol{\lambda}$
Output: $\mathbf{H}$
  Initialize $\mathbf{H}$, $\mathbf{H}_b$, and $\mathbf{W}_b$ randomly
  $\hat{\mathbf{V}} \leftarrow \mathbf{WH} + \mathbf{W}_b\mathbf{H}_b$
  repeat
    for $j = 1, \ldots, J, g = 1, \ldots, G_j$ **do**
      $\mathbf{P}_{(j,g)} \leftarrow \frac{\lambda_j}{\epsilon + \|\mathbf{H}_{(j,g)}\|_1}$
      $\mathbf{Q}_{(j,g)} \leftarrow \frac{\lambda_j G_j \gamma_j}{\|\mathbf{H}_{(j)}\|_1}$
    end for
    $\mathbf{P} = [\mathbf{P}_{(1,1)}^T, \ldots, \mathbf{P}_{(1,G_1)}^T, \ldots, \mathbf{P}_{(J,1)}^T, \ldots, \mathbf{P}_{(J,G_J)}^T]^T$
    $\mathbf{Q} = [\mathbf{Q}_{(1,1)}^T, \ldots, \mathbf{Q}_{(1,G_1)}^T, \ldots, \mathbf{Q}_{(J,1)}^T, \ldots, \mathbf{Q}_{(J,G_J)}^T]^T$
    $\mathbf{H} \leftarrow \mathbf{H} \odot \left(\frac{\mathbf{W}^T(\mathbf{V} \odot \hat{\mathbf{V}}^{\cdot -2}) + \mathbf{Q}}{\mathbf{W}^T(\hat{\mathbf{V}}^{\cdot -1}) + \mathbf{P}} \right)^{\cdot \eta}$
    $\mathbf{H}_b \leftarrow \mathbf{H}_b \odot \left(\frac{\mathbf{W}_b^T(\mathbf{V} \odot \hat{\mathbf{V}}^{\cdot -2})}{\mathbf{W}_b^T \hat{\mathbf{V}}^{\cdot -1}} \right)^{\cdot \eta}$
    $\mathbf{W}_b \leftarrow \mathbf{W}_b \odot \left(\frac{(\mathbf{V} \odot \hat{\mathbf{V}}^{\cdot -2})\mathbf{H}_b^T}{\hat{\mathbf{V}}^{\cdot -1}\mathbf{H}_b^T} \right)^{\cdot \eta}$
    Normalize $\mathbf{W}_b$ and $\mathbf{H}_b$ component-wise (see, e.g., [20])
    $\hat{\mathbf{V}} \leftarrow \mathbf{WH} + \mathbf{W}_b\mathbf{H}_b$
  until convergence

two sources artificially mixed at 0 dB signal to noise ratio (SNR). Note that during the mixing, we made sure that two sources had more or less the same duration so that in all the mixtures both sources appear most of the time. The mixtures were sampled at either 16000 Hz or 11025 Hz and their duration varies between 1 and 13 seconds. The sources in the mixtures were selected as follows: (*female speech, traffic*), (*female speech, cafe*), (*male speech, bells*), (*male speech, car*), (*woman singing, restaurant*), (*drums, guitar*), (*applause, electric guitar*), (*piano, ringtone*), (*violin, cough*), (*bat, owl*), (*chirps, rooster*), (*chirps, river*), (*siren, dog*), (*cat, dog*), and (*ocean, cricket*). The speech samples (*female speech*, *male speech*) were obtained from the "American English" ITU-T P.501[2] dataset. The following sources *cafe*, *car*, and *restaurant* were obtained from DEMAND[3] from one channel out of the 16 channels. The music instruments (*drums*, *electric guitar*, *guitar*, *piano*, *violin*, *woman singing*) were obtained from QUASI[4]. The remainder were from various websites, mostly www.grsites.com/archive/sounds/ (*bells*, *cat*, *chirps*, *dog*, *rooster*, *river*, *traffic*), but also www.sounddogs.com (*bat*, *owl*) and www.wavlist.com (*cricket*), among others. The diversity in the types of sources should demonstrate the advantage of the proposed on-the-fly strategy since, as opposed to speech where pre-trained models are fairly common, having a pre-trained model for every possible sound class is not viable. In the implementation of the framework, sound examples for training were retrieved from www.findsounds.com, a search engine for audio, as well as from www.freesound.org, a database of user-uploaded sounds. Note that these two websites are different from the ones used to get the sources in the mixtures; thus the possibility that the training set contains a source from the mixtures is very small. The retrieved files were restricted to those with sampling rates at least as high as that of the mixture, and the ones with higher sampling rates were down-sampled accordingly. For retrieval in our experiments, we differentiate between two types of search keywords: i) *reference* keywords given by an expert (the first author) who prepared the dataset and thus had also listened to the separate sources and not only the mixtures and ii) *user* keywords given by non-expert users in our user test. It is important to note that the reference keywords are not the only "correct" keywords since other synonyms can be used. Table I lists the reference keywords and the corresponding user keywords along with the number of times a keyword was given by the users. Note that some reference keywords like "male speech" or "female speech" are repeated in more than one mixture, thus the count of their corresponding user keywords is more than the number of users.

Other parameters were set as follows. The STFT was calculated using a sine window and a frame length of 47 ms with 50% overlap, the number of iterations for MU updates was 200 for learning the USCM $\mathbf{W}_{(j)}$ and 100 for separating the mixture, and the number of NMF components for each spectral model learned from one example $\mathbf{W}_{(j,p)}$ was set to 32. In the semi-supervised case, the number of NMF components for the background source was $K_b = 10$ to avoid overfitting since $\mathbf{W}_b$ is unconstrained. Additionally, since the number of training examples P_j per source was different depending on the availability of the data (search results), the trade-off parameter λ_j determining the contribution of the sparsity-inducing penalty was set to $\lambda_0 FNP_j$ (where λ_0 is a constant) so that λ_j is greater when more examples are available. The intuition here is that the smaller the USCM $\mathbf{W}_{(j)}$ is, which happens when few examples are available, the lower the level of sparsity that should be imposed in the decomposition. We found these settings to generally result in a good separation performance.

[2]http://www.itu.int/rec/T-REC-P.501
[3]http://parole.loria.fr/DEMAND/
[4]http://www.tsi.telecom-paristech.fr/aao/en/2012/03/12/quasi/

TABLE I
REFERENCE KEYWORDS AND THE CORRESPONDING USER KEYWORDS.

Reference keywords	User keywords
applause	background noise, cheers, concert, concert crowd, crowd, crowd cheering (2), crowd concert, people cheering
bat	bird (2), bird cackling, bird chirping, birds sound, monkey (2), jungle, night animal
bells	bell tone, bells, bells church, church bell (3), church bells (3)
cafe	chattering, crowd, crowd speech, crowd talking (2), many people talking, party, people, people talking
car	aeroplane noise, ambient noise, boat motor, calm noise, car, drive, nothing, thunder storm, wind
cat	cat (7), cat meow, cat meowing
chirps	bird (4), bird chirping (6), birds (2), birds chirping (2), birds sing (2), night creatures, sparrows
cough	caugh, cough (2), coughing (5), man caughing
cricket	bird, birds, birds sing, cricket (2), night, night animal, night creatures, tweet-tweet
dog	dog (5), dog bark, dog barking (2), dogs
drums	bass drum, drum (3), drum beats, drums, percussion, rythmic, tap beats
electric guitar	electric guitar, guitar (2), guitar concert, music (3), music playing, riff guitar
female speech	female speech (3), female voice (2), female voice english, girl read, girl talking (5), woman, woman read, woman speak, woman speaking, woman speech, woman talking (2), woman voice
guitar	acoustic guitar, electronic organ, guitar (7)
male speech	male english speech (3), male speech english, man reading, man speak, man speaking (2), man speech, man talking (4), man voice, men speech, poetry recitation, read (2)
ocean	car, driving a car, road traffic, sea waves, storm, street, traffic noice, waterfall, waves
owl	dog, dog moaning, owl (4), owl hooting, pigeon, woodpecker
piano	pianist, piano (4), piano music (3), soft piano strings
restaurant	chattering ambiance, crowd (2), crowd noise with photo clicks, crowd speech, crowd talking, people noise, people talking (2)
ringtone	jingle phone ringing, mobile ringtone (2), phone ringing, phone ringtone, ring, ringing, ringtone, smartphone ring
river	motor engine noise, river, river flowing, sea, stream, water (2), water boiling, water flowing
rooster	cock (2), cock cluck, cock-a-doodle-do, hen (2), rooster (3)
siren	ambulance, police, police car, police siren (5), siren
traffic	car, car passing, car running, road traffic, road with cars, street traffic, traffic noice, traffic noise, traffic sound
violin	cello strings orchestra, music album, music (2), piano, soundtrack, violin (3)
woman singing	brasilian woman singing, brazilian song, girl singing (3), singing woman, woman singing (3)

The source separation performance was evaluated in terms of the normalized signal-to-distortion ratio (NSDR), which measures the overall signal distortion, and the normalized signal-to-interference ratio (NSIR) which measures the leakage of the other sources [31], [32]. Recall that the normalized values are computed by subtracting the SDR and SIR of the original mixture signal from those of the separated sources [32]. The normalization serves to show the gain of using the proposed source separation system as opposed to a naive method that simply assigns the mixture as a source estimate. These metrics are measured in dB and are averaged over all sources and all mixtures for the different algorithms.

B. Separation results using reference keywords

In this experiment, we use the reference keywords given by the expert for retrieval. The goal is to evaluate and compare the performance of the different algorithms. For the supervised case (*i.e.*, Algorithm 1 and Algorithm 3), two keywords were used to retrieve examples for both sources in the mixture, while only one keyword was used for the semi-supervised case (*i.e.*, Algorithm 2 and Algorithm 4). Note that, in the semi-supervised scenario, we tested two cases as follows (i) a keyword was provided for source 1 only and (ii) a keyword was provided for source 2 only; we then averaged the obtained separation results.

We compare the average separation performance obtained using the four sparsity-inducing penalties presented in the paper: *block sparsity* as the baseline [25], the proposed *component sparsity*, *relative block sparsity*, and *relative component sparsity*. Results for the supervised case (*i.e.,* Algorithm 1 for *block sparsity* and *component sparsity*, and Algorithm 3 for *relative block sparsity*, and *relative component sparsity*) are shown in Table II, while those for the semi-supervised case (*i.e.,* Algorithm 2 and Algorithm 4) are shown in Table III. In each case, we run the algorithms with different values of the trade-off parameter λ_0, and the value resulting in the highest average NDSR is chosen and shown in the tables along with the corresponding NSDR and NSIR. Note that the result shown in Table II is 1.8 dB NDSR higher than that reported in our previous work [24]. The reason is that: (1) the dataset (training and testing set) is enlarged by the size and the variation of the sound sources; and (2) the parameter λ_j is here adapted per mixture and not constant for the whole dataset. Also, for easier reading here, we do not compare again the separation performance with the standard supervised NMF setting without using USCM model nor with some other baselines as it has been investigated in our previous study [24]. For the relative sparsity cases, we tested two values for the hyper-parameter γ_j: a fixed $\gamma_j = 1$ as a natural choice, and $\gamma_j = \frac{1}{G_j}$ such that the denominator term $\|\mathbf{H}_{(j)}\|_1^{\gamma_j}$ in the penalty (13) is adaptively normalized with respect to the size of the group G_j.

First, as expected, the results obtained in the supervised case are much better than those achieved in the semi-supervised

TABLE II
SUPERVISED CASE: AVERAGE SOURCE SEPARATION PERFORMANCE.

Method	NSDR	NSIR
Block sparsity [25] ($\lambda_0 = 1 \times 10^{-4}$)	5.14	9.80
Component sparsity ($\lambda_0 = 1 \times 10^{-6}$)	5.91	10.67
Rel. block sparsity ($\gamma_j = 1, \lambda_0 = 1 \times 10^{-4}$)	4.78	9.27
Rel. component sparsity ($\gamma_j = 1, \lambda_0 = 1 \times 10^{-6}$)	**6.15**	10.70
Rel. block sparsity ($\gamma_j = \frac{1}{G_j}, \lambda_0 = 1 \times 10^{-4}$)	5.03	9.44
Rel. component sparsity ($\gamma_j = \frac{1}{G_j}, \lambda_0 = 1 \times 10^{-6}$)	5.96	**10.72**

TABLE III
SEMI-SUPERVISED CASE: AVERAGE SOURCE SEPARATION PERFORMANCE.

Method	NSDR	NSIR
Block sparsity ($\lambda_0 = 1 \times 10^{-4}$)	0.74	4.66
Component sparsity ($\lambda_0 = 2 \times 10^{-8}$)	1.98	6.22
Rel. block sparsity ($\gamma_j = 1, \lambda_0 = 4 \times 10^{-4}$)	1.68	6.03
Rel. component sparsity ($\gamma_j = 1, \lambda_0 = 5 \times 10^{-7}$)	**2.31**	**6.64**
Rel. block sparsity ($\gamma_j = \frac{1}{G_j}, \lambda_0 = 1 \times 10^{-4}$)	1.09	5.74

case where examples for one source are missing. Second, using an adaptive $\gamma_j = \frac{1}{G_j}$ in the supervised case improved the NSDR for the relative block sparsity by 0.25 dB but had no significant effect on the relative component sparsity; in contrast it negatively affected the performance in the semi-supervised case. Third, we note that the proposed component sparsity-inducing penalty achieves a better separation performance than the block sparsity-inducing penalty which was exploited in [25], in both supervised and semi-supervised cases. A possible explanation is that the former offers more flexibility by exploiting the most representative spectral patterns from different spectral models that match the mixture. Last, it is worth noting that the proposed relative component sparsity-inducing penalty performs the best in both supervised and semi-supervised cases in terms of both NSDR and NSIR, the advantage being more significant in the semi-supervised case likely because the source vanishing problem is more severe. We note that the corresponding average signal-to-artifact ratio (SAR) for the different algorithms was on the order of 11 dB. In particular, the SAR corresponding to the relative component sparsity-inducing penalty was 10.98 dB and 11.35 dB for the adaptive γ_j.

In general, the methods would fail if the retrieved examples are quite dissimilar from the actual sources in the mixture. As an example, a mixture of of an electric guitar and applause (cheers and whistles) had low NSDR for both source estimates (0.54 dB and -0.49 dB respectively). In this case, we observed that the retrieved training files for the applause contained mostly just clapping sounds and as such the learned USCM did not capture the cheers; similarly for the guitar where most retrieved examples were not close to the chords in the mixture.

C. Separation results with different choices of λ_j

One of the most important parameters in the presented algorithms in the on-the-fly framework is the trade-off parameter λ_j determining the contribution of the sparsity-inducing penalty. We propose to set $\lambda_j = \lambda_0 FNP_j$ so that it is normalized with respect to the size of the USCM and is larger when more examples are used. In this experiment, we varied λ_0 and assessed the sensitivity of the different algorithms described in Section V-B to this choice in the semi-supervised scenario. The dataset and other parameter settings are the same as described before. The results are shown in Fig. 5 where $\lambda_0 = \{10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$ for the block/relative block sparsity algorithms and $\lambda_0 = \{10^{-7}, 5 \times 10^{-7}, 10^{-6}, 5 \times 10^{-6}, 10^{-5}\}$ for the component/relative component sparsity algorithms. Note that the range of λ_0 is different for the (relative) block and component sparsity algorithms as they are different types of penalties so their optimal range is different.

As can be seen, the relative block sparsity and relative component sparsity algorithms are generally more stable than the block sparsity and component sparsity algorithms over a large range of λ_0 where the results obtained by the former algorithms drop sharply for the last point. Within a good range, *i.e.* the first four points, the relative block sparsity with $\gamma_j = \frac{1}{G_j}$ is the most stable one as its NSDR varies at most 0.2 dB. The relative component sparsity algorithm, which offers the highest performance in general, is not very sensitive to the considered parameter though it has more than 1 dB NSDR difference within the considered range.

D. Separation results for the user test

In the second experiment, our goal was to evaluate the performance of the proposed on-the-fly framework when practically used by non-expert users. We also test the effect of the examples refinement step on learning the USCM. The algorithms based on the proposed *relative* block/component sparsity-inducing penalties, which perform better than those using the block/component sparsity-inducing penalties as shown in Section V-B, were tested using the input from 9 different users who were of different age groups, technical backgrounds, and were all not native English speakers. The best parameter settings as determined from Section V-B were used. Using the GUI described in Section II-B, the users were asked to process each of the 15 mixtures as follows. First, they were asked to listen to the mixture and accordingly type keywords describing the two sources. They were instructed to change the keywords in case the search engine did not return results. Then, they were required to listen to the retrieved examples and select those that sound more similar to the sources in the mixture; at least one example was required to be selected. Given the recorded user input (keywords and selected examples), we examine two possibilities of using the examples in guiding the separation process as follows:

- All retrieved examples are used (All).
- Only the subset of examples selected by the user is used (Subset).

The source separation performance, averaged over all 9 users and over all mixtures per method, is shown in Table IV and Table V for the supervised and semi-supervised cases, respectively. We note that the results for the average user are mostly lower than those for the expert in the supervised case due to the following issues. As can be seen from the keywords in Table I, some sounds like *bat* and *owl* were sometimes

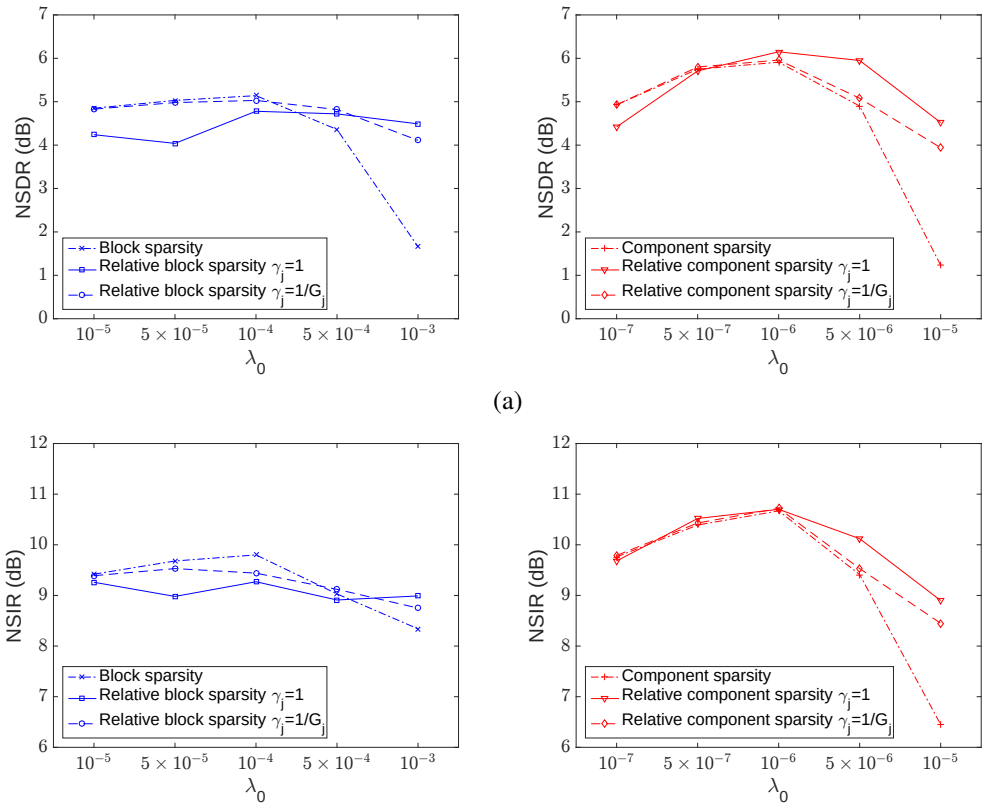


Fig. 5. Separation performance of the different algorithms, in terms of NSDR (a) and NSIR (b), as a function of λ_0.

not recognized by the users and were confused with other sounds (e.g., bird sounds). Also, some spelling mistakes can be found (e.g, *caugh* instead of *cough*). This may have negatively affected the results. Additionally, one of the mixtures included a popular ringtone composed of marimba notes; however, the retrieved examples mainly included classical telephone rings, perhaps "marimba" would have been a better choice for searching. In the semi-supervised case, the expert results are not better than the average user. The reason is likely that the guidance is reduced in this setting and the overall performance is quite lower compared to the supervised case.

Nevertheless, the performance globally follows the same trend as presented in Section V-B: relative component sparsity generally outperforms relative block sparsity, especially in the semi-supervised case, with the SAR on the order of 9 dB. It is interesting to observe the effect of selecting a subset of examples. As can be seen in Table IV, using a subset of examples selected by the users only improves the performance in the supervised case. However, in the semi-supervised case, such a pre-selection even negatively affects the results as can be observed in Table V. This is likely due to the fact that having few selected examples (only one in the extreme case) leads to having fewer components in the learned spectral model for which a sparse decomposition is not optimal. Thus, it seems to be better to keep all retrieved examples for the known source and let the relative component sparsity penalty induce the appropriate selection.

TABLE IV
USER TEST IN THE SUPERVISED CASE: AVERAGE SOURCE SEPARATION PERFORMANCE.

Method	NSDR	NSIR
Relative block sparsity (All)	2.42	7.53
Relative block sparsity (Subset)	**3.16**	**8.24**
Relative component sparsity (All)	2.91	7.75
Relative component sparsity (Subset)	2.98	8.19

TABLE V
USER TEST IN THE SEMI-SUPERVISED CASE: AVERAGE SOURCE SEPARATION PERFORMANCE.

Method	NSDR	NSIR
Relative block sparsity (All)	1.88	7.50
Relative block sparsity (Subset)	1.24	7.34
Relative component sparsity (All)	**2.78**	**8.04**
Relative component sparsity (Subset)	1.53	7.60

VI. CONCLUSION

In this paper, we presented the novel concept of on-the-fly audio source separation and described several algorithms to implement it. Specifically, we proposed using a universal sound class model learned by NMF from retrieved examples and imposing group sparsity-inducing constraints to efficiently handle the selection of the most representative spectral patterns. Additionally, we introduced the notion of relative group sparsity to overcome a so-called *source vanishing* problem that occurs in the considered on-the-fly paradigm. In contrast to

other state-of-the-art user-guided approaches, the considered framework greatly simplifies the user interaction with the system such that everyone, not necessarily an expert, can do source separation by just typing keywords describing the audio sources in the mixture. Experiments on mixtures containing various types of sounds confirm the potential of the proposed framework as well as the corresponding algorithms. Future work may be devoted to running real-world experiments, studying the use of a different group sparsity model that induces dynamic relationships between atoms or groups [33], as well as extending the framework to multichannel mixtures where *spatial* source models (*e.g.* those from [34] or [35]) may also be learned. Additionally, investigating the optimal USCM model size for different types of sound sources would be an interesting direction.

APPENDIX
DERIVATION OF MU RULES IN ALGORITHM 3

Let $C(\mathbf{H})$ denote the right part of criterion (10) with relative group sparsity penalty $\Psi(\mathbf{H}) = \Psi_{\text{rel}}(\mathbf{H})$ defined as in (13) and $D(\cdot\|\cdot)$ being IS divergence specified as in equations (3) and (4). The partial derivative of $C(\mathbf{H})$ with respect to h_{kn} writes

$$\nabla_{h_{kn}} C(\mathbf{H}) = \sum_{f=1}^{F} w_{fk} \left(\frac{1}{[\mathbf{WH}]_{fn}} - \frac{v_{fn}}{[\mathbf{WH}]_{fn}^2} \right) + \frac{\lambda_j}{\epsilon + \|\mathbf{H}_{(j,g)}\|_1} - \frac{\lambda_j G_j \gamma_j}{\|\mathbf{H}_{(j)}\|_1} \quad (15)$$

Following a standard approach for MU rules derivation (see *e.g.*, [20], [28]), we represent $\nabla_{h_{kn}} C(\mathbf{H})$ as

$$\nabla_{h_{kn}} C(\mathbf{H}) = \nabla^+_{h_{kn}} C(\mathbf{H}) - \nabla^-_{h_{kn}} C(\mathbf{H}) \quad (16)$$

with $\nabla^+_{h_{kn}} C(\mathbf{H}), \nabla^-_{h_{kn}} C(\mathbf{H}) \geq 0$ defined as

$$\nabla^+_{h_{kn}} C(\mathbf{H}) \triangleq \sum_{f=1}^{F} w_{fk} \frac{1}{[\mathbf{WH}]_{fn}} + \frac{\lambda_j}{\epsilon + \|\mathbf{H}_{(j,g)}\|_1}, \quad (17)$$

$$\nabla^-_{h_{kn}} C(\mathbf{H}) \triangleq \sum_{f=1}^{F} w_{fk} \frac{v_{fn}}{[\mathbf{WH}]_{fn}^2} + \frac{\lambda_j G_j \gamma_j}{\|\mathbf{H}_{(j)}\|_1}, \quad (18)$$

and we update each parameter h_{kn} as

$$h_{kn} \leftarrow h_{kn} \left(\frac{\nabla^-_{h_{kn}} C(\mathbf{H})}{\nabla^+_{h_{kn}} C(\mathbf{H})} \right)^{.\eta}, \quad (19)$$

where $\eta = 0.5$ following the derivation in [21]. Rewritten in a matrix form, we obtain the updates of the activation matrix $\mathbf{H}$ in Algorithm 3.

ACKNOWLEDGMENT

The authors would like to thank all the colleagues at Technicolor's Rennes research center who participated in the experiments.

REFERENCES

[1] S. Makino, T.-W. Lee, and H. Sawada, *Blind Speech Separation*, Springer, 2007.
[2] O. Yılmaz and S. T. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Trans. on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.
[3] S. Winter, W. Kellermann, H. Sawada, and S. Makino, "MAP-based underdetermined blind source separation of convolutive mixtures by hierarchical clustering and ℓ_1-norm minimization," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, 2007, article ID 24717.
[4] P. Bofill and M. Zibulevsky, "Underdetermined blind source separation using sparse representations," *Signal Processing*, vol. 81, pp. 2353–2362, 2001.
[5] E. Vincent, S. Araki, F. Theis, G. Nolte, P. Bofill, H. Sawada, A. Ozerov, V. Gowreesunker, D. Lutter, and N. Q. K. Duong, "The Signal Separation Campaign (2007-2010): Achievements and remaining challenges," *Signal Processing*, vol. 92, pp. 1928–1936, 2012.
[6] N. Ono, Z. Rafii, D. Kitamura, N. Ito, and A. Liutkus, "The 2015 signal separation evaluation campaign," in *Proc. Latent Variable Analysis and Signal Separation*, 2015, pp. 387–395.
[7] 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.
[8] L. L. Magoarou, A. Ozerov, and N. Q. K. Duong, "Text-informed audio source separation. example-based approach using non-negative matrix partial co-factorization," *Journal of Signal Processing Systems*, pp. 1–5, 2014.
[9] J. Ganseman, P. Scheunders, G. J. Mysore, and J. S. Abel, "Source separation by score synthesis," in *Proc. ICMC*, 2010, pp. 462–465.
[10] J. Fritsch and M. Plumbley, "Score informed audio source separation using constrained nonnegative matrix factorization and score synthesis," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 888–891.
[11] N. Souviraá-Labastie, E. Vincent, and F. Bimbot, "Music separation guided by cover tracks: designing the joint nmf model," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP) 2015*, 2015.
[12] N. Souviraá-Labastie, A. Olivero, E. Vincent, and F. Bimbot, "Multi-channel audio source separation using multiple deformed references," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 11, no. 23, pp. 1775–1787, 2015.
[13] P. Smaragdis and G. J. Mysore, "Separation by humming: User-guided sound extraction from monophonic mixtures," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2009, pp. 69–72.
[14] N. Q. K. Duong, A. Ozerov, and L. Chevallier, "Temporal annotation-based audio source separation using weighted nonnegative matrix factorization," in *IEEE Int. Conf. on Consumer Electronics (ICCE-Berlin)*, 2014.
[15] A. Lefèvre, F. Bach, and C. Févotte, "Semi-supervised NMF with time-frequency annotations for single-channel source separation," in *Int. Conf. on Music Information Retrieval (ISMIR)*, 2012, pp. 115–120.
[16] N. J. Bryan and G. J. Mysore, "Interactive refinement of supervised and semi-supervised sound source separation estimates," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 883–887.
[17] N. Q. K. Duong, A. Ozerov, L. Chevallier, and J. Sirot, "An interactive audio source separation framework based on nonnegative matrix factorization," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 1586–1590.
[18] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "On-the-fly specific person retrieval," in *13th Int. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, 2012, pp. 1–4.
[19] K. Chatfield and A. Zisserman, "Visor: Towards on-the-fly large-scale object category retrieval," in *Asian Conference on Computer Vision*, ser. Lecture Notes in Computer Science. Springer, 2012, pp. 432–446.
[20] C. Févotte, N. Bertin, and J. Durrieu, "Non-negative matrix factorization with the Itakura-Saito divergence. with application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.
[21] A. Lefèvre, F. Bach, and C. Févotte, "Itakura-Saito non-negative matrix factorization with group sparsity," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 21–24.
[22] P. Smaragdis, B. Raj, and M. Shashanka, "Supervised and semi-supervised separation of sounds from single-channel mixtures," in *Int. Conf. on Independent Component Analysis and Signal Separation (ICA)*, 2007, pp. 414–421.

[23] T. Gerber, M. Dutasta, L. Girin, and C. Févotte, "Audio source separation using multiple deformed references," in *International Society for Music Information Retrieval Conf. (ISMIR)*, 2012.
[24] D. El Badawy, N. Q. K. Duong, and A. Ozerov, "On-the-fly audio source separation," in *IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP)*, 2014, pp. 1–6.
[25] D. L. Sun and G. J. Mysore, "Universal speech models for speaker independent single channel source separation," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 141–145.
[26] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1, pp. 19–41, 2000.
[27] D. El Badawy, A. Ozerov, and N. Q. K. Duong, "Relative group sparsity for non-negative matrix factorization with application to on-the-fly audio source separation," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015, accepted.
[28] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural and Information Processing Systems 13*, 2001, pp. 556–562.
[29] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Stat.*, vol. 58, no. 1, pp. 30–37, Feb. 2004.
[30] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the beta-divergence," *Neural Computation*, vol. 23, no. 9, pp. 2421–2456, Sep. 2011.
[31] 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.
[32] A. Ozerov, P. Philippe, R. Gribonval, and F. Bimbot, "One microphone singing voice separation using source-adapted models," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005, pp. 90–93.
[33] A. Hurmalainen, R. Saeidi, and T. Virtanen, "Similarity induced group sparsity for non-negative matrix factorisation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4425–4429.
[34] N. Q. K. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1830–1840, 2010.
[35] J. Nikunen and T. Virtanen, "Direction of arrival based spatial covariance model for blind sound source separation," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 22, no. 3, pp. 727–739, 2014.

Dalia El Badawy is currently a doctoral student at Ecole polytechnique fédérale de Lausanne (EPFL). She received the M.Sc. in communication systems from EPFL in 2015 and the B.Sc. in digital media engineering and technology from the German University in Cairo in 2012. Her research interests include applications of audio signal processing and acoustics as well as machine learning.

Ngoc Q. K. Duong received the B.S. degree from Posts and Telecommunications Institute of Technology (PTIT), Vietnam, in 2004, and the M.S. degree in electronic engineering from Paichai University, Korea, in 2008. He obtained the Ph.D. degree at the French National Institute for Research in Computer Science and Control (INRIA), Rennes, France in 2011.

From 2004 to 2006, he was with Visco JSC as a System Engineer. He was also a Research Engineer for the acoustic echo/noise cancelation system at Emersys Company, Korea in 2008. He is currently a Senior Scientist at Technicolor R&D France where he has worked since Nov. 2011. His research interest concerns signal processing (audio, image, and video), machine learning, and affective computing. He has received several research awards, including the IEEE Signal Processing Society Young Author Best Paper Award in 2012 and the Bretagne Young Researcher Award in 2015. He is the co-author of more than 30 scientific papers and about 25 pending patents.

Alexey Ozerov holds a Ph.D. in Signal Processing from the University of Rennes 1 (France). He worked towards this degree from 2003 to 2006 in the labs of France Telecom R&D and in collaboration with the IRISA institute. Earlier, he received an M.Sc. degree in Mathematics from the Saint-Petersburg State University (Russia) in 1999 and an M.Sc. degree in Applied Mathematics from the University of Bordeaux 1 (France) in 2003. From 1999 to 2002, Alexey worked at Terayon Communicational Systems (USA) as a R&D software engineer, first in Saint-Petersburg and then in Prague (Czech Republic). He was for one year (2007) in Sound and Image Processing Lab at KTH (Royal Institute of Technology), Stockholm, Sweden, for one year and half (2008-2009) in TELECOM ParisTech / CNRS LTCI - Signal and Image Processing (TSI) Department, and for two years (2009 - 2011) with METISS team of IRISA / INRIA - Rennes. Now he is a Senior Scientist in Technicolor Research & Innovation at Rennes, France. Since 2016 he is a Distinguished member of the Technicolor Fellowship Network and currently he is a member of IEEE Signal Processing Society Audio and Acoustic Signal Processing Technical Committee. He received the IEEE Signal Processing Society Best Paper Award in 2014. His research interests include image processing, audio restauration, audio source separation, source coding, audio classification and automatic speech recognition.