An ADMM-based algorithm with minimum dispersion constraint for on-line unmixing of hyperspectral images

Ludivine Nus, Sébastian Miron, David Brie

An ADMM-based algorithm with minimum dispersion constraint for on-line unmixing of hyperspectral images

Ludivine NUS, Sebastian MIRON, David BRIE
CRAN, Université de Lorraine, CNRS, Vandœuvre-lès-Nancy, France
firstname.lastname@univ-lorraine.fr

***Abstract*—Pushbroom imaging systems are emerging techniques for real-time acquisition of hyperspectral images. These systems are frequently used in industrial applications to control and sort products on-the-fly. In this paper, the on-line hyperspectral image blind unmixing is addressed. We propose a new on-line method based on Alternating Direction Method of Multipliers approach, particularly well-adapted to pushbroom imaging systems. The proposed algorithm presents faster convergence and lower computational complexity compared to the algorithms based on multiplicative update rules. Because of the generally ill-posed nature of the unmixing problem, we impose a minimum endmembers dispersion constraint to regularize the unmixing problem; this constraint can be interpreted as a convex relaxation of the minimum volume constraint and, therefore, presents interesting optimization properties. Experimental results on synthetic and real data sets demonstrate the effectiveness of our method in terms of rapidity and accuracy.**

***Index Terms*—Hyperspectral imaging, Pushbroom acquisition system, On-line unmixing, ADMM, Minimum dispersion constraint.**

I. Introduction

Hyperspectral imaging is a powerful tool which combines the power of digital imaging and spectroscopy. Each pixel in a hyperspectral image provides local spectral information about a scene of interest across a large number of contiguous bands. Because of the limited spatial resolution of the sensor, mixed pixels (pixels containing the contributions of several components) are often encountered in hyperspectral data. Thus, spectral unmixing is an important technique for hyperspectral data interpretation, as it allows to decompose a mixed pixel into a collection of spectral signatures (also called *endmembers*) and their relative proportions (also called *abundances*).

This paper addresses the problem of sequential (or on-line) spectral unmixing of hyperspectral Near InfraRed (NIR) images acquired by a *pushbroom* imager [26], by means of Non-negative Matrix Factorization (NMF)-like approaches. This problem is encountered *e.g.,* in real-time industrial systems, for product quality control applications. On-line unmixing methods present several advantages over the off-line methods: *i)* they allow to alleviate computational burden and reduce memory requirements for big hyperspectral data cubes; *ii)* they are well-adapted to real-time data processing for on-line industrial acquisition systems; *iii)* they allow to highlight (track) the spatial/time variability of the endmembers in a hyperpectral imaging application.

We benefit from the support of the ANR-OPTIFIN (Agence Nationale de la Recherche-OPTimisation des FINitions) project.

A. Hyperspectral image unmixing

Spectral unmixing methods are based on a mixture model describing how the endmembers are combined in the acquired image. Depending on the application and/or the data, this model can be linear or non-linear [4]. In this article, we focus on the Linear Mixing Model (LMM) [19]; it is the most commonly used model in hyperspectral unmixing because it is simple to understand and represents a good approximation of the physical reality in most applications. The LMM model can be generally expressed in the following form:

$$\mathbf{X} \approx \mathbf{SA}, \tag{1}$$

with $\mathbf{X} \in \mathbb{R}_+^{L\times P}$, $\mathbf{S} = [\mathbf{s}_1, ..., \mathbf{s}_R] \in \mathbb{R}_+^{L\times R}$ and $\mathbf{A} \in \mathbb{R}_+^{R\times P}$, where $\mathbb{R}_+$ denotes the set of non-negative real numbers. In hyperspectral imaging, the P columns of $\mathbf{X}$ represent the data samples (pixels) recorded at L wavelengths. $\mathbf{S}$ is a matrix containing on its columns the R normalized endmembers and $\mathbf{A}$ is a matrix containing on its columns the abundances for the recorded samples. Under the non-negativity and the sum-to-one constraint *i.e.*, $\mathbf{S} \geq \mathbf{0}$, $\mathbf{A} \geq \mathbf{0}$ and $\mathbf{A}^T\mathbf{1} = \mathbf{1}$ (where $\mathbf{1}$ is an all-ones vector), the LMM admits a geometric interpretation: all the pixels belongs to the $(R-1)$-dimensional simplex whose vertices are the endmembers $\mathbf{s}_1, ..., \mathbf{s}_R$. Fig.1 illustrates this geometrical representation for a mixture of $R = 3$ endmembers. When the sum-to-one constraint does not hold, the observations belong to the positive cone defined by the endmembers.

There are many methods in the literature for hyperspectral unmixing. Most of them are based on the *pure pixel assumption*, *i.e.,* the existence of pixels containing a single source, see *e.g.* [29], [38]. However, the pure pixel assumption is a strong requirement that generally does not hold for highly mixed data, and finding the endmembers is therefore a more challenging task. For this case, other family of methods has been developed, based on the volume minimization idea introduced by Craig in 1994 [9]. These approaches consist in minimizing the volume of the simplex containing the data by "creating" virtual endmembers such as in [2], [24].

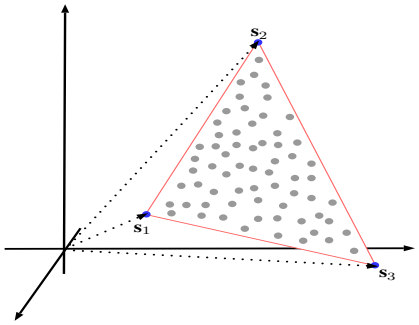


Fig. 1: Geometrical representation of the LMM for $R = 3$ endmembers. All the observed pixels (gray points) belong to a 2-simplex (red lines) whose vertices are the endmembers (blue points)

However the performance and computational efficiency of these methods is often limited because of complicated simplex volume calculations, sensitivity to initialization and lack of rigorous performance analysis. A more detailed analysis can be found in [4].

To overcome the above problems, Non-negative Matrix Factorization (NMF) [21] has been applied to hyperspectral data unmixing. For a non-negative matrix $\mathbf{X}$, the NMF consists in estimating two matrices, $\mathbf{S} \geq \mathbf{0}$ and $\mathbf{A} \geq \mathbf{0}$, satisfying (1). In general, the NMF is not unique and, therefore, to reduce the size of the set of admissible solutions, it is necessary to add regularization terms. The most effective constraint approaches are *volume-regularized* NMF such as [1], [13], [14], [23], [28], [33], [43], [41], which can be considered as state-of-the-art methods in blind hyperspectral unmixing.

B. Pushbroom acquisition scheme

In pushbroom imaging systems, hyperspectral data cubes are acquired slice by slice, sequentially in time (which represents the so-called along track spatial dimension). Fig. 2 illustrates the acquisition of a hyperspectral image by a pushbroom imager; the image is acquired by moving the sensor across the scene. The stream of spatial-spectral data arrays is then stacked to form the hyperspectral data cube. For each acquisition time k ($k = 1, ..., K$), the new slice (represented by a dotted line in Fig. 3) is a matrix of dimensions $N_x \times N_\lambda$, where N_x denotes the across track spatial dimension (one line of the scene) and N_λ the spectral dimension (wavelengths).

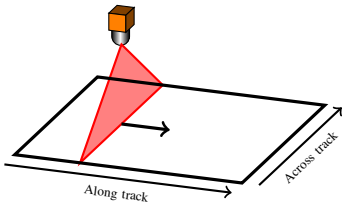


Fig. 2: Data acquisition with pushbroom imager

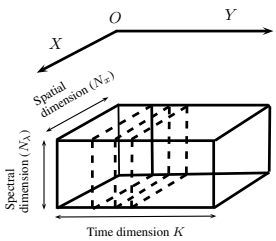


Fig. 3: Data structure for pushbroom acquisition

The goal of the on-line blind hyperspectral unmixing is to produce real-time estimates of the endmember and abundance matrices for each new incoming slice, at the pushbroom system acquisition rate.

C. On-line NMF methods

On-line NMF algorithms sequentially update the endmembers and abundances as the data size increases. In that respect, they are perfectly adapted to the processing of hyperspectral data streaming as they allow to maintain a low and controlled computational complexity. These algorithms can be gathered into two main categories, depending on the considered assumptions on the endmembers. In [27], [37], [15], [42], it is assumed that the endmembers do not vary from one sample to another, while in [22], [7], [44], [36], [39], [30], the endmembers may evolve between successive samples. In particular, the Incremental NMF (INMF) [7] considers that the endmembers evolve slowly between two consecutive acquisitions; this is now the most widely used assumption adopted in on-line NMF algorithms. In the context of hyperspectral unmixing, it allows to account for the so-called spectral variability. However, as in the off-line case, the uniqueness of the solution is not guaranteed, which led to the development of on-line regularized NMF such as [44], [39], [36]. Recently, the on-line Minimum Volume Constraint-NMF algorithm (OMVC-NMF) was introduced in [30]. It is a straightforward adaptation of [44] specially designed for pushbroom hyperspectral imaging system. We did not find in the literature another on-line NMF algorithm, except OMVC-NMF, adapted to on-line processing of hyperspectral images. Thus, OMVC-NMF will serve as benchmark for the on-line volume regularized NMF method proposed in this paper. Most of the algorithms mentioned above are based on multiplicative update rules which are known to be highly sensitive to initialization and often suffer from slow convergence rate [25]. An alternative to multiplicative updates is to make use of the Alternating Direction Method of Multipliers (ADMM) as presented in [42], which proved its superiority over multiplicative updates with respect to both reconstruction accuracy and efficiency [35], [40], [17].

D. Main contributions

In this paper, we introduce a new algorithm for on-line blind unmixing of hyperspectral images, specially devised for pushbroom acquisition systems. The original contributions of this algorithm compared to our previous work [30] and to related state-of-the-art on-line approaches (see *e.g.* [42]) are:

1) The addition of the Minimum Dispersion Constraint (MDC) [2], to regularize the problem. MDC can be interpreted as a convex relaxation of the minimum volume simplex constraint. Thanks to its convexity, this constraint offers interesting optimization properties and enables explicit updates of the parameters.
2) The integration of a tracking capability to the algorithm which allows modeling dynamic content changes.
3) The use of ADMM approach in the context of on-line volume regularized NMF.

The remainder of this paper is organized as follows: Section 2 is dedicated to the formulation of the on-line NMF problem

for the acquisition scheme of a pushbroom imager. Section 3 provides a comparative study of different volume regularizers. In particular, we highlight the differences between the minimum dispersion and the determinant-based constraint. Section 4 presents the derivation of the proposed on-line ADMM approach; convergence and computational complexity are also discussed. Finally, in Section 5, we give extensive experimental results on both synthetic and real data. The results are compared to those of the off-line (or *batch*) counterpart of the proposed approach and to another benchmark on-line matrix factorization algorithm. We conclude the paper in Section 6.

II. On-line hyperspectral unmixing for pushbroom acquisition scheme

A. Data model

The principle of the proposed on-line method is to alternatively update the endmember and abundance matrices estimated at time instant k when a new sample (slice) arrives at time instant $(k+1)$. One way to handle the problem is to unfold the hyperspectral image as shown in Fig. 4 where $\tilde{\mathbf{X}}^{(1)} = \mathbf{X}^{(1)}$ is the first slice of the hyperspectral image and $\tilde{\mathbf{X}}^{(k)}$ is the k^{th} slice. The entire dataset at time instant $(k+1)$, *i.e.,* $\mathbf{X}^{(k+1)}$, can be represented as the concatenation of the first k samples with the new incoming sample *i.e.*, $\mathbf{X}^{(k+1)} = \left[\mathbf{X}^{(k)}\ \tilde{\mathbf{X}}^{(k+1)}\right]$. Similarly, we define $\mathbf{S}^{(k+1)} = \left[\mathbf{S}^{(k)}\ \tilde{\mathbf{S}}^{(k+1)}\right]$ and $\mathbf{A}^{(k+1)} = \left[\mathbf{A}^{(k)}\ \tilde{\mathbf{A}}^{(k+1)}\right]$.

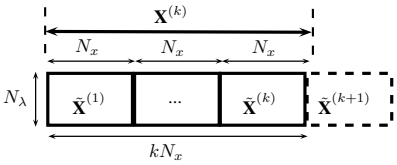


Fig. 4: Unfolded pushbroom hyperspectral image

Then, the on-line NMF model is given by:

$$\tilde{\mathbf{X}}^{(k+1)} \approx \tilde{\mathbf{S}}^{(k+1)}\tilde{\mathbf{A}}^{(k+1)}. \tag{2}$$

B. Cost function

The simplest way to fit the NMF model to the data is to minimize the least square distance between the data and the model. Let $\mathcal{J}^{(k)}$ denote the cost function corresponding to the first k samples:

$$\mathcal{J}^{(k)}\left(\mathbf{S}^{(k)},\mathbf{A}^{(k)}\right) = \frac{1}{2}\sum_{\ell=1}^{k}\left\|\tilde{\mathbf{X}}^{(\ell)} - \tilde{\mathbf{S}}^{(\ell)}\tilde{\mathbf{A}}^{(\ell)}\right\|_F^2.$$

When the $(k+1)^{th}$ sample, $\tilde{\mathbf{X}}^{(k+1)}$ arrives, the corresponding cost function can be decomposed as follows:

$$\begin{aligned}\mathcal{J}^{(k+1)}\left(\mathbf{S}^{(k+1)},\mathbf{A}^{(k+1)}\right) &= \frac{1}{2}\sum_{\ell=1}^{k}\left\|\tilde{\mathbf{X}}^{(\ell)} - \tilde{\mathbf{S}}^{(\ell)}\tilde{\mathbf{A}}^{(\ell)}\right\|_F^2 \\ &+\frac{1}{2}\left\|\tilde{\mathbf{X}}^{(k+1)} - \tilde{\mathbf{S}}^{(k+1)}\tilde{\mathbf{A}}^{(k+1)}\right\|_F^2.\end{aligned} \tag{3}$$

Without further assumptions, (3) is just a set of $(k+1)$ independent least squares problems $\left\|\tilde{\mathbf{X}}^{(l)} - \tilde{\mathbf{S}}^{(l)}\tilde{\mathbf{A}}^{(l)}\right\|_F^2$ and in this case, the on-line setup has no particular interest. However, a natural assumption is that the endmembers vary only slightly between consecutive samples *i.e.,* $\tilde{\mathbf{S}}^{(k+1)} \approx \tilde{\mathbf{S}}^{(k)},\ \forall k$. Thus, the cost function (3) can be expressed as:

$$\begin{aligned}\mathcal{J}^{(k+1)}\left(\tilde{\mathbf{S}}^{(k+1)},\tilde{\mathbf{A}}^{(k+1)}\middle|\mathbf{A}^{(k)}\right) &= \mathcal{J}^{(k)}\left(\tilde{\mathbf{S}}^{(k+1)}\middle|\mathbf{A}^{(k)}\right) \\ &+\tilde{\mathcal{J}}^{(k+1)}\left(\tilde{\mathbf{S}}^{(k+1)},\tilde{\mathbf{A}}^{(k+1)}\right),\end{aligned}$$

with

$$\mathcal{J}^{(k)}\left(\tilde{\mathbf{S}}^{(k+1)}\middle|\mathbf{A}^{(k)}\right) = \frac{1}{2}\sum_{\ell=1}^{k}\left\|\tilde{\mathbf{X}}^{(\ell)} - \tilde{\mathbf{S}}^{(k+1)}\tilde{\mathbf{A}}^{(\ell)}\right\|_F^2,$$

$$\tilde{\mathcal{J}}^{(k+1)}\left(\tilde{\mathbf{S}}^{(k+1)},\tilde{\mathbf{A}}^{(k+1)}\right) = \frac{1}{2}\left\|\tilde{\mathbf{X}}^{(k+1)} - \tilde{\mathbf{S}}^{(k+1)}\tilde{\mathbf{A}}^{(k+1)}\right\|_F^2.$$

In order to add some tracking capability to the algorithm, a weighting coefficient α $(0 \leq \alpha \leq 1)$ is incorporated into the cost function as:

$$\mathcal{J}^{(k+1)}\left(\tilde{\mathbf{S}}^{(k+1)},\tilde{\mathbf{A}}^{(k+1)}\right) = \alpha\mathcal{J}^{(k)} + (1-\alpha)\tilde{\mathcal{J}}^{(k+1)}. \tag{4}$$

A version of the cost function (4) is used by the INMF algorithm proposed in [7]. Nevertheless, without additional constraint, the solution of (4) is not unique. The uniqueness of the NMF relies on the sparsity of the underlying latent variables. In particular, if either $\mathbf{S}$ and/or $\mathbf{A}$ has only non-zero entries, the NMF factorization is not unique. To reduce the size of the set of admissible solutions, we propose to impose on the endmembers matrix $\tilde{\mathbf{S}}^{(k+1)}$, a geometric constraint which forces the simplex bounded by the endmembers to circumscribe the data as closely as possible. For now, we denote this constraint by $\mathrm{Vol}\left(\tilde{\mathbf{S}}^{(k+1)}\right)$ and integrate it into the cost function as follows:

$$\begin{aligned}\mathcal{J}_{vol}^{(k+1)}\left(\tilde{\mathbf{S}}^{(k+1)},\tilde{\mathbf{A}}^{(k+1)}\right) &= \alpha\mathcal{J}^{(k)} + (1-\alpha)\tilde{\mathcal{J}}^{(k+1)} \\ &+\mu\ \mathrm{Vol}\left(\tilde{\mathbf{S}}^{(k+1)}\right),\end{aligned} \tag{5}$$

where $\mathcal{J}_{vol}$ becomes the volume regularized criterion and $\mu \geq 0$ controls the trade-off between the data fitting term and the volume regularizer. The choice of $\mathrm{Vol}\left(\tilde{\mathbf{S}}^{(k+1)}\right)$ is discussed in the next section. Thus, the proposed approach in this paper aims at solving the following optimization problem:

$$\underset{\tilde{\mathbf{S}}^{(k+1)}\geq\mathbf{0},\tilde{\mathbf{A}}^{(k+1)}\geq\mathbf{0}}{\text{minimize}}\ \mathcal{J}_{vol}^{(k+1)}\left(\tilde{\mathbf{S}}^{(k+1)},\tilde{\mathbf{A}}^{(k+1)}\right), \tag{6}$$

for a particular choice of the volume regularizer term.

III. Volume regularization

The classical measure for the minimum volume constraint is the determinant, *i.e.* $\mathrm{Vol}\left(\tilde{\mathbf{S}}^{(k+1)}\right) = \det\left(\tilde{\mathbf{S}}^{(k+1)}\right)$ [34]. However, this determinant is defined only if $\tilde{\mathbf{S}}^{(k+1)}$ is a square matrix. In the case where $\tilde{\mathbf{S}}^{(k+1)}$ is a tall matrix, a pertinent choice for $\mathrm{Vol}\left(\tilde{\mathbf{S}}^{(k+1)}\right)$ is [33], [43]:

$$\mathrm{Vol}\left(\tilde{\mathbf{S}}^{(k+1)}\right) = \det\left(\tilde{\mathbf{S}}^{(k+1)^T}\tilde{\mathbf{S}}^{(k+1)}\right), \tag{7}$$

or its variants presented in [8], [28]. The expression (7) is mathematically justified because it can be interpreted as the square volume of the simplex bounded by the endmembers [31, Theorem 7].

A. Identifiability

A lot of work has been done in the last decade to understand the identifiability of the NMF model. The results in [10], [20], [18] have shown that if the matrices $\tilde{\mathbf{S}}^{(k+1)}$ and $\tilde{\mathbf{A}}^{(k+1)}$ both contain null elements, according to a certain pattern, then the model is unique. However, in hyperspectral imaging, endmembers $\tilde{\mathbf{S}}^{(k+1)}$ are likely to be strictly positive and thus, the model is not indentifiable. This is the case where volume minimization approaches can be successfully used. Based on the recent work of [14], we give the following identifiability sufficient conditions for our on-line model:

Sufficient identifiability conditions: if $\forall\ k$, $\text{rank}\left(\tilde{\mathbf{S}}^{(k)}\right) = \text{rank}\left(\tilde{\mathbf{A}}^{(k)}\right) = R$ and $\tilde{\mathbf{A}}^{(k)}$ is sufficiently scattered (see [14] for the exact definition of sufficiently scattered), then model (2) is identifiable.

This condition stipulates that there is no rank loss over the different slices of the hyperspectral image.

B. Volume regularizers

In this section, to simplify the notations, the $(k+1)$ upper indices are omitted. The use of the minimum volume constraint in (7) makes the $\tilde{\mathbf{S}}$ subproblem non convex and therefore more difficult to solve. To tackle the determinant minimization problems, others volume regularizer surrogates can be considered. An alternative formulation for $\text{Vol}\left(\tilde{\mathbf{S}}\right)$, proposed in [3] is:

$$\log\left(\det\left(\tilde{\mathbf{S}}^T\tilde{\mathbf{S}}\right)\right). \tag{8}$$

The choice of the logarithm of the determinant rather than the determinant itself is mainly motivated by algorithmic reasons, since it drastically simplifies the update rule for $\tilde{\mathbf{S}}$. To avoid strong negative values of (8) when $\tilde{\mathbf{S}}$ becomes close to singularity, *i.e.* $\det\left(\tilde{\mathbf{S}}^T\tilde{\mathbf{S}}\right) \to 0$, a modified version was proposed in [14], [23]:

$$\log\left(\det\left(\tilde{\mathbf{S}}^T\tilde{\mathbf{S}} + \epsilon\mathbf{I}\right)\right), \tag{9}$$

where $\epsilon > 0$ is a specified small value and $\mathbf{I}$ is the identity matrix. While (8) is a concave function, it appears that (9) has a stationary point in $\mathbf{0}$ and is convex in a neighborhood of $\mathbf{0}$. Indeed, let $\left\{\lambda_i^2, i = 1, ..., R\right\}$ be the ordered eigenvalues of $\tilde{\mathbf{S}}^T\tilde{\mathbf{S}}$, that is the λ_i's are the singular values of $\tilde{\mathbf{S}}$, then:

$$\log\left(\det\left(\tilde{\mathbf{S}}^T\tilde{\mathbf{S}} + \epsilon\mathbf{I}\right)\right) = \sum_{i=1}^{R}\log\left(\lambda_i^2 + \epsilon\right) = f(\lambda_i),$$
$$\frac{\partial f(\lambda_i)}{\partial\lambda_i} = \frac{2\lambda_i}{\lambda_i^2 + \epsilon}.$$

Clearly, $\lim_{\lambda_i\to 0}\frac{\partial f(\lambda_i)}{\partial\lambda_i} = 0$. In addition $\frac{\partial^2 f(\lambda_i)}{\partial\lambda_i^2} = \frac{2\epsilon - 2\lambda_i^2}{(\lambda_i^2+\epsilon)^2} \geq 0$ if $\lambda_i^2 \leq \epsilon$. In fact, in the neighborhood of $\lambda_i = 0$, we can write:

$$\begin{aligned}\log\left(\det\left(\tilde{\mathbf{S}}^T\tilde{\mathbf{S}} + \epsilon\mathbf{I}\right)\right) &= R\log(\epsilon) + \sum_{i=1}^{R}\log\left(\frac{\lambda_i^2}{\epsilon} + 1\right)\\ &\approx R\log(\epsilon) + \frac{1}{\epsilon}\sum_{i=1}^{R}\lambda_i^2\\ &= R\log(\epsilon) + \frac{1}{\epsilon}\text{trace}\left(\tilde{\mathbf{S}}^T\tilde{\mathbf{S}}\right).\end{aligned} \tag{10}$$

In other words, in the neighborhood of $\lambda = 0$, $\log\left(\det\left(\tilde{\mathbf{S}}^T\tilde{\mathbf{S}} + \epsilon\mathbf{I}\right)\right)$ essentially behaves as $\text{trace}\left(\tilde{\mathbf{S}}^T\tilde{\mathbf{S}}\right)$, the extent of the neighborhood being controlled by the value of ϵ. This has an important consequence: while $\log\left(\det\left(\tilde{\mathbf{S}}^T\tilde{\mathbf{S}}\right)\right)$ will favor rank deficiency as the regularization parameter μ increases [12], the use of $\log\left(\det\left(\tilde{\mathbf{S}}^T\tilde{\mathbf{S}} + \epsilon\mathbf{I}\right)\right)$ will preserve the full column rank of the solution $\tilde{\mathbf{S}}$ even for large values of the regularization parameter.

Finally, a convex surrogate for $\text{Vol}\left(\tilde{\mathbf{S}}\right)$ was proposed in [2] as:

$$\text{trace}\left(\tilde{\mathbf{S}}\mathbf{P}\tilde{\mathbf{S}}^T\right), \tag{11}$$

where $\mathbf{P} = \mathbf{I} - \frac{1}{r}\mathbf{1}_R\mathbf{1}_R^T$ ($\mathbf{1}_R$ is an all-ones column vector of size R). Expression (11) can be interpreted as a measure of the dispersion of the endmembers around their centroid. In the following, we will denote the constraint $\log\left(\det\left(\tilde{\mathbf{S}}^T\tilde{\mathbf{S}} + \epsilon\mathbf{I}\right)\right)$ by MVC (for Minimum Volume Constraint), and $\text{trace}\left(\tilde{\mathbf{S}}\mathbf{P}\tilde{\mathbf{S}}^T\right)$ by MDC (for Minimum Dispersion Constraint).

In order to illustrate the difference between these two regularization functions, consider a matrix $\tilde{\mathbf{S}}$ composed of two unit column vectors:

$$\tilde{\mathbf{S}} = \begin{bmatrix}1 & \cos(\beta)\\ 0 & \sin(\beta)\end{bmatrix},$$

with β an angle varying between 0 et $\frac{\pi}{2}$. If $\beta = 0$, the vectors are collinear (the volume is zero); *a contrario*, if $\beta = \frac{\pi}{2}$, the vectors become orthogonal (the volume is maximal and equal to 1). We plotted in Fig. 5 the two functions: MVC with varying ϵ and MDC, for increasing values of β; the curves were normalized to set their minimum value to 0 and their maximum value to 1. For small values of ϵ, MVC promotes rank-deficient solutions but ϵ should no be chosen too small to avoid a bad conditioning of $\tilde{\mathbf{S}}^T\tilde{\mathbf{S}} + \epsilon\mathbf{I}$. For large values of ϵ, rank deficiency is no longer promoted and both MDC and MVC preserve the full column rank of $\tilde{\mathbf{S}}$. This rank preserving property is essential to ensure that the sufficient identifiability condition of section III-A is satisfied.

IV. OMDC-ADMM ALGORITHM

To implement the proposed approach, in this paper we chose MDC over MVC because of its convexity; it favors full column rank solutions and allows an efficient implementation of on-line ADMM. The resulting algorithm will be termed as *On-line MDC-ADMM* (OMDC-ADMM).

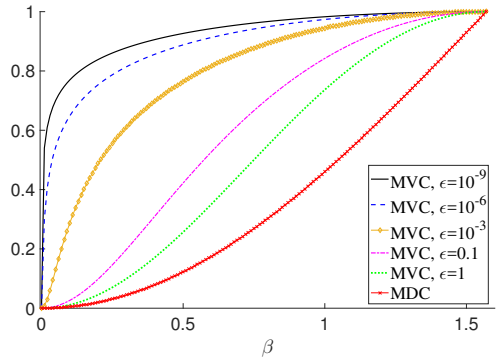


Fig. 5: MVC and MDC as functions of the angle β

A. Algorithm derivation

Defining

$$\mathcal{J}_{vol}^{(k+1)}\left(\tilde{\mathbf{S}}^{(k+1)},\tilde{\mathbf{A}}^{(k+1)}\right) = \alpha\mathcal{J}^{(k)} + (1-\alpha)\tilde{\mathcal{J}}^{(k+1)} + \mu \text{ trace}\left(\tilde{\mathbf{S}}^{(k+1)}\mathbf{P}\tilde{\mathbf{S}}^{(k+1)^T}\right) \quad (12)$$

and introducing the auxiliary variables $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$, problem (6) is equivalent to:

$$\begin{aligned}\underset{\tilde{\mathbf{S}}^{(k+1)},\tilde{\mathbf{A}}^{(k+1)},\tilde{\mathbf{V}}^{(k+1)},\tilde{\mathbf{U}}^{(k+1)}}{\text{minimize}} & \ \mathcal{J}_{vol}^{(k+1)}\left(\tilde{\mathbf{S}}^{(k+1)},\tilde{\mathbf{A}}^{(k+1)}\right) \\ & +\mathbb{I}_{\mathbb{R}_+}\left(\tilde{\mathbf{V}}^{(k+1)}\right) + \mathbb{I}_{\mathbb{R}_+}\left(\tilde{\mathbf{U}}^{(k+1)}\right), \\ \text{subject to } \tilde{\mathbf{S}}^{(k+1)} &= \tilde{\mathbf{U}}^{(k+1)} \text{ and } \tilde{\mathbf{A}}^{(k+1)} = \tilde{\mathbf{V}}^{(k+1)}, \end{aligned} \quad (13)$$

where $\mathbb{I}_{\mathbb{R}_+}$ is the indicator function of $\mathbb{R}_+$ ensuring the non-negativity of endmembers and abundances. For algorithmic convenience, we use the scaled form of ADMM [5] in which the linear and quadratic terms are combined in the augmented Lagrangian and the dual variables are scaled. The augmented Lagrangian $\mathcal{L}$ for the problem (13) is given by:

$$\begin{aligned}&\mathcal{L}\left(\tilde{\mathbf{A}}^{(k+1)},\tilde{\mathbf{S}}^{(k+1)},\tilde{\mathbf{V}}^{(k+1)},\tilde{\mathbf{U}}^{(k+1)},\tilde{\mathbf{\Pi}}^{(k+1)},\tilde{\mathbf{\Lambda}}^{(k+1)}\right) \\ &= \mathcal{J}_{vol}^{(k+1)}\left(\tilde{\mathbf{S}}^{(k+1)},\tilde{\mathbf{A}}^{(k+1)}\right) \\ &+ \frac{\rho}{2}\left\|\tilde{\mathbf{A}}^{(k+1)} - \tilde{\mathbf{V}}^{(k+1)} + \tilde{\mathbf{\Pi}}^{(k+1)}\right\|_F^2 - \frac{\rho}{2}\left\|\tilde{\mathbf{\Pi}}^{(k+1)}\right\|_F^2 \\ &+ \frac{\rho}{2}\left\|\tilde{\mathbf{S}}^{(k+1)} - \tilde{\mathbf{U}}^{(k+1)} + \tilde{\mathbf{\Lambda}}^{(k+1)}\right\|_F^2 - \frac{\rho}{2}\left\|\tilde{\mathbf{\Lambda}}^{(k+1)}\right\|_F^2 \\ &+ \mathbb{I}_{\mathbb{R}_+}\left(\tilde{\mathbf{V}}^{(k+1)}\right) + \mathbb{I}_{\mathbb{R}_+}\left(\tilde{\mathbf{U}}^{(k+1)}\right). \end{aligned} \quad (14)$$

The parameter $\rho > 0$ controls the convergence of the method. $\tilde{\mathbf{\Lambda}}^{(k+1)}$ and $\tilde{\mathbf{\Pi}}^{(k+1)}$ are the scaled versions of the dual variables corresponding to the equality constraints $\tilde{\mathbf{S}}^{(k+1)} = \tilde{\mathbf{U}}^{(k+1)}$ and $\tilde{\mathbf{A}}^{(k+1)} = \tilde{\mathbf{V}}^{(k+1)}$, respectively. ADMM optimization alternately minimizes the augmented Lagrangian (14) with respect to $\left(\tilde{\mathbf{A}}^{(k+1)},\tilde{\mathbf{V}}^{(k+1)}\right)$ and $\left(\tilde{\mathbf{S}}^{(k+1)},\tilde{\mathbf{U}}^{(k+1)}\right)$ and then, updates the dual variables $\tilde{\mathbf{\Pi}}^{(k+1)}$ and $\tilde{\mathbf{\Lambda}}^{(k+1)}$. The updates of $\tilde{\mathbf{A}}^{(k+1)}, \tilde{\mathbf{V}}^{(k+1)}$ and $\tilde{\mathbf{\Pi}}^{(k+1)}$ at data slice $(k+1)$ can be expressed as:

$$\begin{aligned}\tilde{\mathbf{A}}^{(k+1)} &= \left((1-\alpha)\tilde{\mathbf{S}}^{(k)^T}\tilde{\mathbf{S}}^{(k)} + \rho\mathbf{I}\right)^{-1} \\ &\quad \left((1-\alpha)\tilde{\mathbf{S}}^{(k)^T}\tilde{\mathbf{X}}^{(k+1)} + \rho\left(\tilde{\mathbf{V}}^{(k)} - \tilde{\mathbf{\Pi}}^{(k)}\right)\right), \quad (15a)\\ \tilde{\mathbf{V}}^{(k+1)} &= \max\left(\mathbf{0}, \tilde{\mathbf{A}}^{(k+1)} + \tilde{\mathbf{\Pi}}^{(k)}\right), \quad (15b)\\ \tilde{\mathbf{\Pi}}^{(k+1)} &= \tilde{\mathbf{\Pi}}^{(k)} + \tilde{\mathbf{A}}^{(k+1)} - \tilde{\mathbf{V}}^{(k+1)}. \quad (15c)\end{aligned}$$

The updates of $\tilde{\mathbf{S}}^{(k+1)}$, $\tilde{\mathbf{U}}^{(k+1)}$ and $\tilde{\mathbf{\Lambda}}^{(k+1)}$ at data slice $(k+1)$ can be expressed as:

$$\begin{aligned}\tilde{\mathbf{S}}^{(k+1)} &= \left(\mathbf{N}^{(k+1)} + \rho\left(\tilde{\mathbf{U}}^{(k)} - \tilde{\mathbf{\Lambda}}^{(k)}\right)\right) \\ &\quad \left(\mathbf{M}^{(k+1)} + 2\mu\mathbf{P} + \rho\mathbf{I}\right)^{-1}, \quad (16a)\\ \tilde{\mathbf{U}}^{(k+1)} &= \max\left(\mathbf{0}, \tilde{\mathbf{S}}^{(k+1)} + \tilde{\mathbf{\Lambda}}^{(k)}\right), \quad (16b)\\ \tilde{\mathbf{\Lambda}}^{(k+1)} &= \tilde{\mathbf{\Lambda}}^{(k)} + \tilde{\mathbf{S}}^{(k+1)} - \tilde{\mathbf{U}}^{(k+1)}, \quad (16c)\end{aligned}$$

where $\mathbf{N}^{(k+1)} = \alpha\sum_{l=1}^{k}\tilde{\mathbf{X}}^{(l)}\tilde{\mathbf{A}}^{(l)^T} + (1-\alpha)\tilde{\mathbf{X}}^{(k+1)}\tilde{\mathbf{A}}^{(k+1)^T}$ and $\mathbf{M}^{(k+1)} = \alpha\sum_{l=1}^{k}\tilde{\mathbf{A}}^{(l)}\tilde{\mathbf{A}}^{(l)^T} + (1-\alpha)\tilde{\mathbf{A}}^{(k+1)}\tilde{\mathbf{A}}^{(k+1)^T}$. Following [7], under the assumption $\tilde{\mathbf{S}}^{(k+1)} \approx \tilde{\mathbf{S}}^{(k)}$, we can write $\mathbf{N}^{(k)} \approx \sum_{l=1}^{k}\tilde{\mathbf{X}}^{(l)}\tilde{\mathbf{A}}^{(l)^T}$ and $\mathbf{M}^{(k)} \approx \sum_{l=1}^{k}\tilde{\mathbf{A}}^{(l)}\tilde{\mathbf{A}}^{(l)^T}$. Therefore, $\mathbf{N}^{(k+1)}$ and $\mathbf{M}^{(k+1)}$ read as:

$$\mathbf{N}^{(k+1)} = \alpha\mathbf{N}^{(k)} + (1-\alpha)\tilde{\mathbf{X}}^{(k+1)}\tilde{\mathbf{A}}^{(k+1)^T}, \quad (17)$$

$$\mathbf{M}^{(k+1)} = \alpha\mathbf{M}^{(k)} + (1-\alpha)\tilde{\mathbf{A}}^{(k+1)}\tilde{\mathbf{A}}^{(k+1)T}. \quad (18)$$

Algorithm 1 summarizes the proposed OMDC-ADMM algorithm. It includes two main loops: the outer loop produces estimates of all parameters at each new slice. These estimates are iteratively refined in the inner loop using a fixed number of iterations N_{iter}. For notation simplification, the indices $(k+1)$ in the updates rules are omitted.

B. Convergence

We provide in this section a partial result of convergence for the proposed algorithm: we show that any stationary point generated by a sequence of iterations satisfies the Karush-Kuhn-Tucker (KKT) conditions [6]. To simplify the notation, we gather all the variables to be estimated in $\mathbf{W} = \left(\tilde{\mathbf{A}},\tilde{\mathbf{S}},\tilde{\mathbf{V}},\tilde{\mathbf{U}},\tilde{\mathbf{\Pi}},\tilde{\mathbf{\Lambda}}\right)$. Following [6], [17] and after some basic algebraic manipulations, it can be shown that a point $\mathbf{W}$ is a KKT point for the problem (13) if:

$$\begin{aligned}&(1-\alpha)\tilde{\mathbf{S}}^T\tilde{\mathbf{S}}\tilde{\mathbf{A}} - (1-\alpha)\tilde{\mathbf{S}}^T\tilde{\mathbf{X}} + \tilde{\mathbf{\Pi}} = \mathbf{0}, && (19a)\\ &\tilde{\mathbf{S}}\tilde{\mathbf{M}} - \tilde{\mathbf{N}} + 2\mu\tilde{\mathbf{S}}\mathbf{P} + \tilde{\mathbf{\Lambda}} = \mathbf{0}, && (19b)\\ &\tilde{\mathbf{A}} - \tilde{\mathbf{V}} = \mathbf{0}, && (19c)\\ &\tilde{\mathbf{S}} - \tilde{\mathbf{U}} = \mathbf{0}, && (19d)\\ &\tilde{\mathbf{\Pi}} \le \mathbf{0} \le \tilde{\mathbf{V}}, \tilde{\mathbf{\Pi}} \odot \tilde{\mathbf{V}} = \mathbf{0}, && (19e)\\ &\tilde{\mathbf{\Lambda}} \le \mathbf{0} \le \tilde{\mathbf{U}}, \tilde{\mathbf{\Lambda}} \odot \tilde{\mathbf{U}} = \mathbf{0}, && (19f)\end{aligned}$$

Algorithm 1 OMDC-ADMM

Inputs: $\mathbf{X}$; $\mathbf{P}$; R; α; μ; ρ; N_{iter};
Initialization: $k = 0$; $\mathbf{N} = \text{zeros}(N_\lambda, R)$; $\mathbf{M} = \text{zeros}(R, R)$; $\tilde{\mathbf{S}} = \text{rand}(N_\lambda, R)$; $\tilde{\mathbf{V}} = \text{zeros}(R, N_x)$; $\tilde{\mathbf{U}} = \text{zeros}(N_\lambda, R)$; $\tilde{\mathbf{\Pi}} = \text{zeros}(R, N_x)$; $\tilde{\mathbf{\Lambda}} = \text{zeros}(N_\lambda, R)$; $\mathbf{I} = \text{eye}(R, R)$; $\mathbf{A} = [\,]$; $\mathbf{S} = [\,]$;
Outputs: $\mathbf{A}$; $\mathbf{S}$;
while New sample $(k+1)$ available **do**
  $\tilde{\mathbf{X}} = \tilde{\mathbf{X}}^{(k+1)}$;
  $t = 1$;
  while $t < N_{iter}$ **do**
    $\tilde{\mathbf{A}} = \left((1-\alpha)\tilde{\mathbf{S}}^T\tilde{\mathbf{S}} + \rho\mathbf{I}\right)^{-1}\left((1-\alpha)\tilde{\mathbf{S}}^T\tilde{\mathbf{X}} + \rho\left(\tilde{\mathbf{V}} - \tilde{\mathbf{\Pi}}\right)\right)$;
    $\tilde{\mathbf{V}} = \max\left(\mathbf{0}, \tilde{\mathbf{A}} + \tilde{\mathbf{\Pi}}\right)$;
    $\tilde{\mathbf{\Pi}} \leftarrow \tilde{\mathbf{\Pi}} + \tilde{\mathbf{A}} - \tilde{\mathbf{V}}$;
    $\tilde{\mathbf{N}} = \alpha\mathbf{N} + (1-\alpha)\left(\tilde{\mathbf{X}}\tilde{\mathbf{A}}^T\right)$;
    $\tilde{\mathbf{M}} = \alpha\mathbf{M} + (1-\alpha)\left(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^T\right)$;
    $\tilde{\mathbf{S}} = \left(\tilde{\mathbf{N}} + \rho\left(\tilde{\mathbf{U}} - \tilde{\mathbf{\Lambda}}\right)\right)\left(\tilde{\mathbf{M}} + 2\mu\mathbf{P} + \rho\mathbf{I}\right)^{-1}$;
    $\tilde{\mathbf{U}} = \max\left(\mathbf{0}, \tilde{\mathbf{S}} + \tilde{\mathbf{\Lambda}}\right)$;
    $\tilde{\mathbf{\Lambda}} \leftarrow \tilde{\mathbf{\Lambda}} + \tilde{\mathbf{S}} - \tilde{\mathbf{U}}$;
    $t \leftarrow t + 1$;
  end while
  $\mathbf{N} = \tilde{\mathbf{N}}$;
  $\mathbf{M} = \tilde{\mathbf{M}}$;
  $\mathbf{A} = \left[\mathbf{A}\ \tilde{\mathbf{A}}\right]$;
  $\mathbf{S} = \left[\mathbf{S}\ \tilde{\mathbf{S}}\right]$;
end while

where $\odot$ represents the Hadamard product. Let $\mathbf{W}_t := \left(\tilde{\mathbf{A}}_t, \tilde{\mathbf{S}}_t, \tilde{\mathbf{V}}_t, \tilde{\mathbf{U}}_t, \tilde{\mathbf{\Pi}}_t, \tilde{\mathbf{\Lambda}}_t\right)$ be the estimate of $\mathbf{W}$ at iteration t of **Algorithm 1**. From (15a), the following relation can be written between $\tilde{\mathbf{A}}_t$ et $\tilde{\mathbf{A}}_{t+1}$:

$$\begin{aligned}&\left((1-\alpha)\tilde{\mathbf{S}}_t^T\tilde{\mathbf{S}}_t + \rho\mathbf{I}\right)\left(\tilde{\mathbf{A}}_{t+1} - \tilde{\mathbf{A}}_t\right)\\ &= -\left((1-\alpha)\tilde{\mathbf{S}}_t^T\tilde{\mathbf{S}}_t\tilde{\mathbf{A}}_t - (1-\alpha)\tilde{\mathbf{S}}_t^T\tilde{\mathbf{X}} + \rho\left(\tilde{\mathbf{A}}_t - \tilde{\mathbf{V}}_t\right) + \rho\tilde{\mathbf{\Pi}}_t\right).\end{aligned} \tag{20}$$

Suppose that the algorithm reaches a stationary point *i.e.*, $\mathbf{W}_{t+1} = \mathbf{W}_t = \mathbf{W}^\star$; this implies $\tilde{\mathbf{A}}_{t+1} = \tilde{\mathbf{A}}_t = \mathbf{V}_t = \tilde{\mathbf{A}}^\star$. By replacing it in (20), we obtain $(1-\alpha)\tilde{\mathbf{S}}^{\star T}\tilde{\mathbf{S}}^\star\tilde{\mathbf{A}}^\star - (1-\alpha)\tilde{\mathbf{S}}^{\star T}\tilde{\mathbf{X}} + \rho\tilde{\mathbf{\Pi}}^\star = \mathbf{0}$. Using similar rationale for the other parameters of $\mathbf{W}$, it can be shown that the first four equalities from the KKT conditions (19) are satisfied for every limit point $\mathbf{W}^\star := \left(\tilde{\mathbf{A}}^\star, \tilde{\mathbf{S}}^\star, \tilde{\mathbf{V}}^\star, \tilde{\mathbf{U}}^\star, \tilde{\mathbf{\Pi}}^\star, \tilde{\mathbf{\Lambda}}^\star\right)$. To prove (19e), we can write: $\max\left(\mathbf{0}, \tilde{\mathbf{A}}^\star + \tilde{\mathbf{\Pi}}^\star\right) = \tilde{\mathbf{V}}^\star$. If $\tilde{\mathbf{A}}^\star = \tilde{\mathbf{V}}^\star = \mathbf{0}$, then $\max\left(\mathbf{0}, \tilde{\mathbf{\Pi}}^\star\right) = \mathbf{0}$ which leads to $\tilde{\mathbf{\Pi}}^\star < \mathbf{0}$. If $\tilde{\mathbf{A}}^\star = \tilde{\mathbf{V}}^\star > \mathbf{0}$, then $\tilde{\mathbf{\Pi}}^\star = \mathbf{0}$. The same kind of rationale also applies to (19f). Thus, we have shown that for the problem (13), any stationary point $\mathbf{W}^\star$ given by **Algorithm 1** satisfies the KKT conditions.

The working assumption to establish this result is $\tilde{\mathbf{S}}^{(k+1)} \approx \tilde{\mathbf{S}}^{(k)}$, meaning that the proposed convergence result is valid only when the steady state is reached; it does not provide any insights into the transient behavior. This point will be addressed through numerical simulations in sections V-C and V-D.

C. Computational complexity

In this section we evaluate the computational complexity of the proposed algorithm; this is done by taking into account only the matrix multiplication operations, that dominate the algorithm complexity. By considering K slices of size $(N_x \times N_\lambda)$, N_{iter} iterations and a decomposition rank R, the computational complexity for OMDC-ADMM algorithm is of the order of $2KN_{iter}(RN_xN_\lambda + (N_\lambda + N_x)R^2 + R^3)$. For comparison, we developed a *batch* version of OMDC-ADMM, that we called BMDC-ADMM (for *Batch MDC-ADMM*), which processes the entire hyperspectral data cube at once; this algorithm has complexity of $2N_{iter}(RN_xKN_\lambda + (N_\lambda + N_xK)R^2 + R^3)$. Moreover, the OMVC-NMF algorithm presented in [30], based on multiplicative update rules with minimum volume constraint (8), has a computational complexity of the order of $KN_{iter}(2RN_xN_\lambda + (N_\lambda + N_x)R + (6N_\lambda + 2N_x)R^2)$. Note that the complexity of OMDC-ADMM is comparable to that of BMDC-ADMM and OMVC-NMF. However, as we show in the sequel, OMDC-ADMM requires fewer iterations to converge compared to the two other methods, and thus significantly reduces the computational cost.

V. Experimental results

In order to evaluate the performance of OMDC-ADMM for hyperspectral unmixing, we conducted several experiments on simulated and real hyperspectral images (including the benchmarking image *Jasper Ridge* available at http://lesun.weebly.com/hyperspectral-data-set.html). For these experiments we used Matlab (R2016a) on a 2.7 GHz Macbook Pro with 4-core processor and 16 GB of RAM. These experiments have the following objectives:

1) Showing the efficiency of OMDC-ADMM by examining its convergence speed and comparing it to that of the OMVC-NMF algorithm [30]. The sensitivity of the convergence speed of the proposed method to parameters α and ρ is also studied.
2) Illustrating the rank preserving properties of MDC and MVC.
3) Comparing OMDC-ADMM with its batch counterpart BMDC-ADMM to assess the advantages of on-line processing of hyperspectral images.
4) Studying the tracking capability of OMDC-ADMM.
5) Validating the unmixing performance of our algorithm on a real hyperspectral image with ground truth.

A. Performance criteria

Three performance criteria were used for these experiments: the residual error, Spectral Angle Distance (SAD) and Root Mean Square Error (RMSE).

The residual error was calculated for each slice k as follows:

$$\frac{1}{2}\left\|\tilde{\mathbf{X}} - \hat{\tilde{\mathbf{S}}}\hat{\tilde{\mathbf{A}}}\right\|_F^2, \tag{21}$$

where $\hat{\tilde{\mathbf{S}}}$ and $\hat{\tilde{\mathbf{A}}}$ are the estimated endmembers and abundances, respectively.

SAD measures the similarity between original endmembers $\tilde{\mathbf{S}}$ and estimated endmembers $\hat{\tilde{\mathbf{S}}}$. For a given slice k, it is written as follows:

$$\frac{1}{R}\sum_{r=1}^{R}\cos^{-1}\left(\frac{\tilde{\mathbf{s}}_r^T\hat{\tilde{\mathbf{s}}}_r}{||\tilde{\mathbf{s}}_r||\ ||\hat{\tilde{\mathbf{s}}}_r||}\right), \tag{22}$$

where R is the number of endmembers, $\tilde{\mathbf{S}} = [\tilde{\mathbf{s}}_1, ..., \tilde{\mathbf{s}}_R]$ and $\hat{\tilde{\mathbf{S}}} = \left[\hat{\tilde{\mathbf{s}}}_1, ..., \hat{\tilde{\mathbf{s}}}_R\right]$.

RMSE measures the error between the original abundances $\mathbf{A}$ and the estimated abundances $\hat{\mathbf{A}}$ and is computed as:

$$\frac{1}{R}\sum_{r=1}^{R}\sqrt{\frac{1}{P}\sum_{p=1}^{P}(a_{rp}-\hat{a}_{rp})^2}, \tag{23}$$

where a_{rp} and $\hat{a}_{rp}$ are the ground truth and estimated abundance respectively of the r^{th} endmember at pixel p ($p = 1, ..., P$).

B. Rank preserving properties of MVC and MDC

We performed an experiment to illustrate the rank preserving properties of both volume regularizers, MVC (9) and MDC (11). We used the endmembers shown in Fig. 8. None of the three endmembers has any zero value, which results in a non-unique NMF problem. In other words, without additional constraints the model is non-identifiable. The three abundance maps are matrices of size 36×36, randomly drawn from a continuous uniform distribution on the interval $[0, 1]$. By doing so, we ensure (with high probability) that the sufficiently scattered and full column rank conditions are fulfilled (see subsection III-A). For μ ranging from 0.0001 to 0.003, the quantities $\text{trace}\left(\hat{\tilde{\mathbf{S}}}\mathbf{P}\mathbf{S}\hat{\tilde{\mathbf{S}}}^T\right)$ and $\log\left(\det\left(\hat{\tilde{\mathbf{S}}}^T\hat{\tilde{\mathbf{S}}}+\epsilon\mathbf{I}\right)\right)$ were computed, where $\hat{\tilde{\mathbf{S}}}$ is the endmember matrix estimated by either OMVC-NMF or OMDC-ADMM at the last slice k of the image. For MVC, four different values of ϵ are chosen: $\epsilon = 0, 10^{-4}, 10^{-3}$ and 10^{-2}. The normalized curves (maximum value equal to 1, minimum value equal to 0) are shown in Fig. 6.

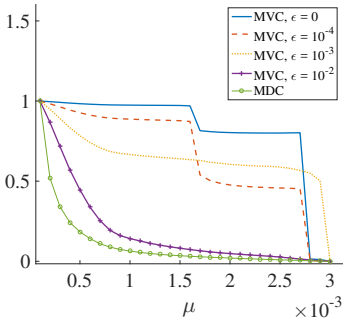
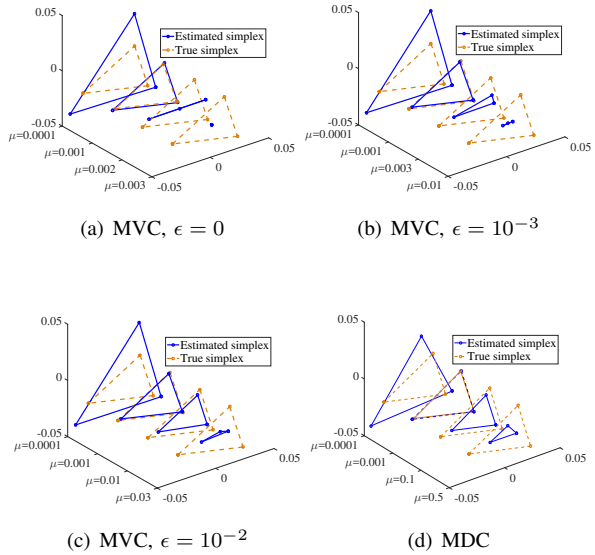


Fig. 6: The values of MVC and MDC as functions of μ

In Fig. 6, several behaviors can be observed. For MVC with $\epsilon = 0$, the plot presents a staircase shape and, as the value of ϵ increases, the steps tend to disappear. Actually, MVC with low values of ϵ is favoring rank deficiency as μ increases. When ϵ increases, rank deficiency is no longer promoted. In Fig. 7, we can compare the true and estimated simplexes for different values of μ, for MVC ($\epsilon = 0$, 10^{-3}, and 10^{-2}) and MDC. For $\epsilon = 0$, as μ increases, the volume of the estimated simplex tends to zero, which means that endmembers become collinear. In the limit case ($\mu \to +\infty$), all endmembers become collinear. On the other hand, as ϵ increases, the rank of the decomposition is preserved for a larger interval of μ. For example, for $\epsilon = 10^{-2}$ and $\mu = 0.03$, although the estimated simplex is included in the true simplex, the rank is preserved which avoids numerical instabilities; this shows the interest of the ϵ parameter in the context of on-line blind unmixing. Similarly, for MDC, the simplex decreases progressively, while preserving the rank of the decomposition.



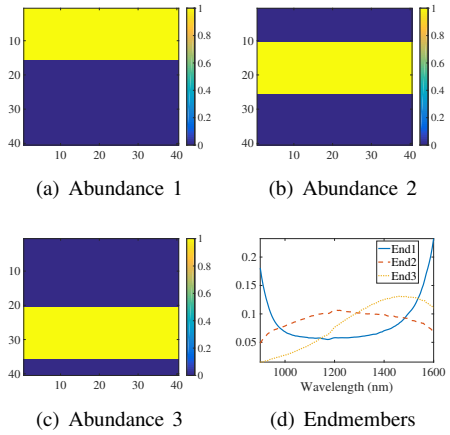
(a) MVC, $\epsilon = 0$ (b) MVC, $\epsilon = 10^{-3}$

(c) MVC, $\epsilon = 10^{-2}$ (d) MDC

Fig. 7: Simulated simplexes as functions of μ for MVC and MDC

C. Convergence rate

The convergence speed of OMDC-ADMM and OMVC-NMF was studied on a synthetic hyperspectral image. A hyperspectral image of size $119 \times 40 \times 40$, composed of $R = 3$ endmembers not varying over time was simulated. Here, 119 corresponds to the number of wavelengths and 40×40 to the (spatial $\times$ time) dimensions. Each new time sample is a 119×40 slice of the hyperspectral image. The abundance maps (considered binary for this experiment) and the corresponding endmembers are shown in Fig. 8. The data were corrupted by a low-level noise. We used for both OMVC-NMF and OMDC-ADMM the same coefficient $\alpha = 0.99$ and regularization parameter $\mu = 0.003$. The parameter ρ for OMDC-ADMM algorithm was set to 0.001 and ϵ for MVC to 0.001.

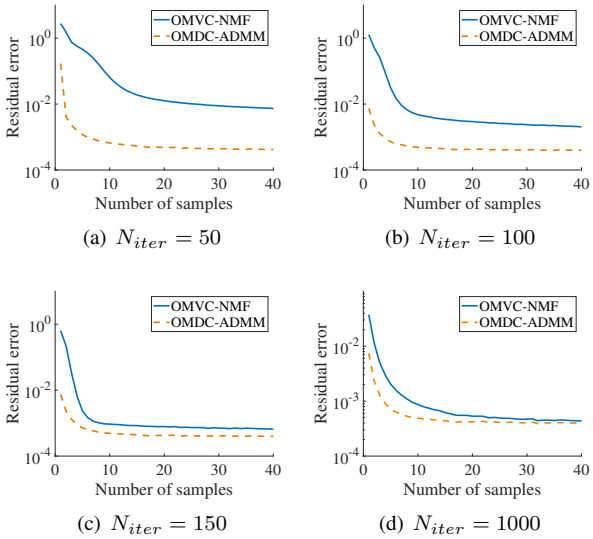


(a) Abundance 1 (b) Abundance 2

(c) Abundance 3 (d) Endmembers

Fig. 8: Simulated abundances and endmembers (End)

To compare the convergence speeds of the two algorithms, the residual error *vs.* time sample (from 1 to 40) was evaluated for different values of N_{iter}. The results are shown in Fig. 9. Note that a logarithmic scale was used for the vertical axis.



(a) $N_{iter} = 50$ (b) $N_{iter} = 100$

(c) $N_{iter} = 150$ (d) $N_{iter} = 1000$

Fig. 9: The residual error as functions of number of samples for OMDC-ADMM and OMVC-NMF

In Fig. 9, we observe that for all the considered values of N_{iter}, OMDC-ADMM has a faster convergence than OMVC-NMF. The latter reaches asymptotically an error close to OMDC-ADMM for approximately $N_{iter} = 1000$ iterations. In order to quantify the quality of estimated endmembers and abundances for each value of N_{iter}, we computed the SAD and RMSE performance criteria. The SAD was calculated using the endmembers estimated at the last slice, which is relevant in the case of stationary sources. The obtained values are given in Tables I and II. One can see that 100 iterations are sufficient for OMDC-ADMM algorithm to converge to a relevant solution; in fact, beyond 100 iterations, the estimation accuracy of OMDC-ADMM no longer improves, and the whole image is processed in about 0.3 s. OMVC-NMF algorithm requires at least 1000 iterations to estimate correctly the endmembers and the abundances, for a processing time of about 1.3 s. This slow convergence rate can be attributed to the use of multiplicative update rules in OMVC-NMF. These results show that there is a real interest in terms of rapidity and accuracy in using OMDC-ADMM for on-line blind unmixing.

	OMVC-NMF	OMDC-ADMM
$N_{iter} = 50$	0.2824	0.0816
$N_{iter} = 100$	0.2339	0.0019
$N_{iter} = 150$	0.1985	0.0019
$N_{iter} = 1000$	0.0028	0.0019

TABLE I: SAD for endmembers

	OMVC-NMF	OMDC-ADMM
$N_{iter} = 50$	0.2246	0.1312
$N_{iter} = 100$	0.2057	0.0029
$N_{iter} = 150$	0.1946	0.0028
$N_{iter} = 1000$	0.0036	0.0028

TABLE II: RMSE for abundances

Another interest of on-line algorithm for spectral unmixing is the processing of large hyperspectral images. In that respect, we compared the performance of OMDC-ADMM algorithm with its batch counterpart BMDC-ADMM. We used the same synthetic hyperspectral image (corrupted by a low-level noise) presented in Fig. 8. Both algorithms used the same $\rho = 0.001$; the minimum dispersion constraint was set to $\mu = 0.003$ for the on-line version and to $\mu = 0.5$ for the batch version. The value of α was set to 0.99. The BMDC-ADMM method was applied to the unfolded version of the hyperspectral image, of size 119×1600. To study the convergence speeds of both algorithms, the SAD and the RMSE were evaluated for different values of N_{iter} ranging from 10 to 2000, and averaged for 20 different random initializations. The results are shown in Fig. 10 and Fig. 11. One can see that our on-line algorithm converges much faster than its batch counterpart for both error measures. Indeed, 100 iterations are enough for OMDC-ADMM to yield accurate estimates, while BMDC-ADMM requires at least 800 iterations. This has strong consequences on the computational cost. Consider the computational complexity of the on-line and batch version established in subsection IV-C. If we consider $R = 3$, $N_x = 40$, $N_\lambda = 119$, $K = 40$, $N_{iter} = 100$ for the on-line version and $N_{iter} \approx 800$ for the batch version, the computation complexity of OMDC-ADMM is about ten times lower than its batch counterpart.

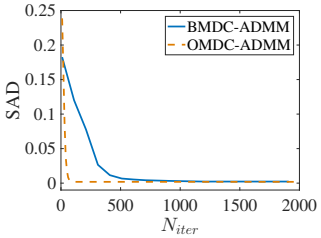


Fig. 10: SAD as functions of N_{iter}

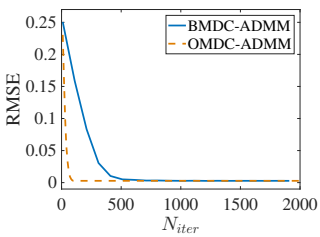


Fig. 11: RMSE as functions of N_{iter}

We also examined the influence of parameter α on the convergence speed of OMDC-ADMM using the same synthetic dataset in the noise-free case. We first varied α from 0.99 to 0.5 while fixing the values of $\rho = 0.001$, $\mu = 0.003$ and $N_{iter} = 1000$. The results are shown in Fig. 12. As α decreases, fewer time samples are required to converge to the correct solution; at the same time, the asymptotic residual error decreases; this can be explained the overfitting of each slice introduced by the low values of α. Next, we varied ρ from 1 to 0.001 while keeping the value of α to 0.9 and μ to 0.003. From Fig. 13, it can be seen that as ρ increases, the OMDC-ADMM convergence rate decreases.

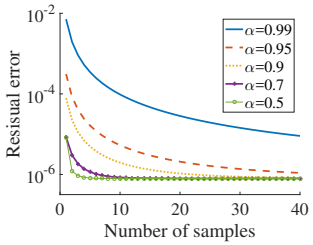


Fig. 12: Residual error as functions of number of samples for different values of α

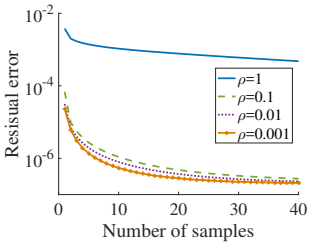
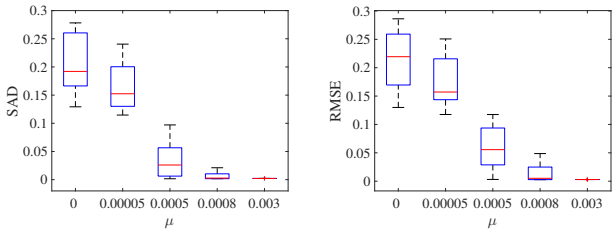


Fig. 13: Residual error as functions of number of samples for different values of ρ

The last aspect assessed by numerical simulation is the sensitivity of the algorithm to initial conditions. In that respect, 20 different random initializations were used for the OMDC-ADMM on the synthetic data corrupted by a low-level noise. The SAD and RMSE were computed for each initialization, using $N_{iter} = 100$ and $\mu = 0, 0.00005, 0.0005, 0.0008$ and 0.003. The results are shown in Fig. 14 as boxplots. When μ is too small, different initializations are likely to produce different estimates of the endmembers. When the value of μ is adequately chosen, the algorithm is not sensitive to initialization.



(a) SAD (b) RMSE

Fig. 14: SAD and RMSE for different initializations as functions of μ

D. Tracking the spectral variability

In hyperspectral imaging, the spectral signature of a component may present intra-class variability. Thus, studying the endmembers evolution between successive samples is sometimes crucial in understanding the underlying physical phenomenon. Integrating a tracking capability (via the α parameter) allows the tracking of dynamic spectral changes. To address this point, a hyperspectral image of size $119 \times 40 \times 100$ with non-stationary endmembers was simulated. The abundance maps are identical to those in Fig. 8, but the endmembers evolve at each slice, according to the following random walk model: $\tilde{\mathbf{S}}^{(k+1)} = \tilde{\mathbf{S}}^{(k)} + \mathbf{W}^{(k)}$, where $\mathbf{W}^{(k)}$ is a low-level random noise drawn from the standard normal distribution. The simulated data were corrupted by noise with an SNR = 26 dB. The parameters μ and ρ were set to 0.01 and 0.001 respectively. In order to assess the ability of our algorithm to follow the evolution of the endmembers, we varied α from 0.9 to 0.4 and, for each slice k, we computed the SAD. The results are shown in Fig. 15. As α decreases, both convergence speed and asymptotic SAD ($k \to +\infty$) increase. Note that for very large values of α an increase of the asymptotic SAD is observed (not shown on this figure). This indicates that there exists an optimal value of α mitigating at best the transient error and tracking error.

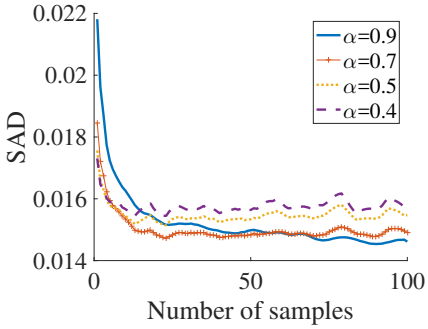


Fig. 15: SAD as functions of number of samples for different values of α

E. Real data application

We also assessed the performance of OMDC-ADMM on a real hyperspectral image with available ground truth. We compared the performance of OMDC-ADMM to its batch version (BMDC-ADMM) and other state-of-the-art (batch) methods: Vertex Component Analysis (VCA) [29], NMF [21], ℓ_1-NMF [16] and $\ell_{1/2}$-NMF [32]. The results of the last four methods come from simulations conducted in [45]. We used the hyperspectral image *Jasper Ridge* [45] for this experiment. This image has a spatial size of 100×100 pixels. Each pixel is recorded at 198 wavelengths ranging from 380 nm to 2500 nm. There are four latent endmembers in this data corresponding to soil, tree, water and road. The OMDC-ADMM algorithm processed the hyperspectral image sequentially along the vertical axis. Note that the sequential processing along the horizontal axis was not possible since rank preservation was not guaranteed from one slice to another. The parameters were set as follows: $R = 4$, $\mu = 0.05$, $\alpha = 0.99$, $\rho = 0.001$ and $N_{iter} = 200$. BMDC-ADMM was applied to the unfolded version of the hyperspectral image, of size 198×10000 with parameters $R = 4$, $\rho = 0.001$, $\mu = 200$ and $N_{iter} = 2000$. For the parameters settings details of the other methods, the reader is referred to [45]. In order to evaluate the quality of

the estimated endmembers and abundances, the SAD and the RMSE for all methods were computed. Each experiment was repeated 50 times for different random initializations and the average results for SAD and RMSE are provided in Table III.

It can be seen from Table III that BMDC-ADMM outperforms the other state-of-the-art methods. Moreover, OMDC-ADMM and BMDC-ADMM yield close estimates for soil, tree, and road. Interestingly, we observed that there is a significant difference between the two methods regarding the SAD of water (0.1113 for OMDC-ADMM *vs.* 0.3141 for BMDC-ADMM). This difference can be explained by the phenomenon of spectral variability, which can modify locally the spectrum of pure materials [11]. The causes of this variability can be diverse, *e.g.*, the changing illumination conditions during the acquisition processing, the intrinsic variability of the components or the atmospheric effects. This aspect is illustrated in Fig. 16 which plots the spectra for all estimated endmembers slice by slice by the on-line algorithm. Thanks to its tracking capability, OMDC-ADMM offers the possibility to study the dynamic content changes over time. We note that the spectral signature of water evolves strongly between slices. In Table III, the SAD for OMDC-ADMM was calculated using the average over the K time samples of the estimated spectra.

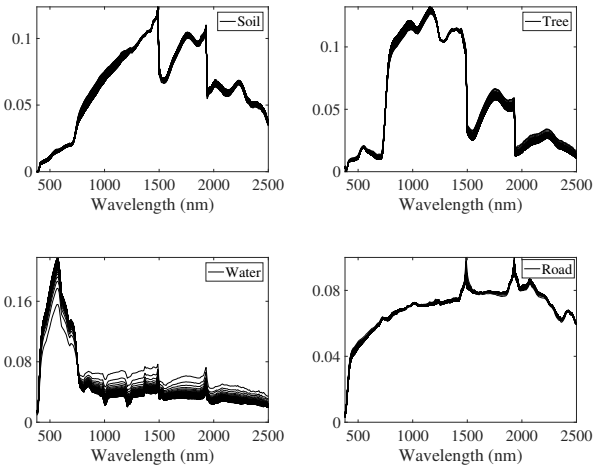
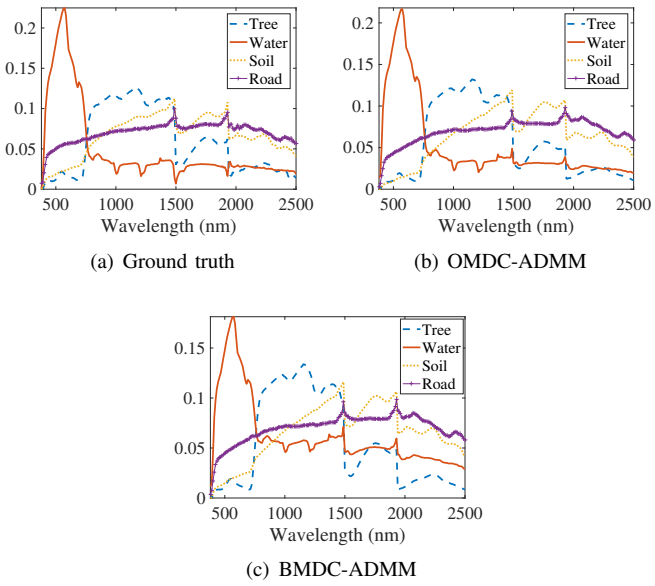


Fig. 16: Estimated endmembers slice by slice by OMDC-ADMM

For the batch algorithm, this spectral variability is not explicitly taken into account, and the estimated endmembers can be interpreted as the average spectra for the entire image. The "best" endmembers and abundance maps estimated by on-line and batch MDC-ADMM, along with the ground truth are represented in Fig. 17 and in Fig. 18. For OMDC-ADMM, the average values of the endmembers are represented. Besides the fact that OMDC-ADMM makes it possible to track the spectral variability, it also requires fewer iterations than the batch version which significantly reduces the processing time (2.77 s *vs.* 4.92 s).



(a) Ground truth (b) OMDC-ADMM

(c) BMDC-ADMM

Fig. 17: True and estimated endmembers by on-line and batch MDC-ADMM

VI. Conclusions

We proposed a new algorithm (OMDC-ADMM), specially designed for the on-line unmixing of pushbroom hyperspectral images. Tests on simulated data have shown that this new algorithm outperforms the state-of-the-art methods based on multiplicative update rules, in terms of convergence speed and estimation accuracy. From a methodological point of view, we showed the interest of using the minimum dispersion constraint compared to the minimum volume one, in particular its capacity to regularize the problem and to stabilize the solution. Finally, experiments on a real hyperspectral image revealed that OMDC-ADMM makes it possible to track the spectral variability of the endmembers over time and significantly reduces the processing time compared to its off-line counterpart, which is a crucial feature for real-time data processing. In future work, we plan to use OMDC-ADMM to perform on-line unmixing of products (pieces of wood) in an industrial application, in order to control and sort them on the fly.

References

[1] MS Andersen Ang and Nicolas Gillis. Volume regularized Non-negative Matrix Factorizations. In *IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–5, 2018.

[2] Mark Berman, Harri Kiiveri, Ryan Lagerstrom, Andreas Ernst, Rob Dunne, and Jonathan F Huntington. ICE: A statistical approach to identifying endmembers in hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 42(10):2085–2095, 2004.

[3] José M Bioucas-Dias. A variable splitting augmented lagrangian approach to linear spectral unmixing. In *IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4, 2009.

[4] José M Bioucas-Dias, Antonio Plaza, Nicolas Dobigeon, Mario Parente, Qian Du, Paul Gader, and Jocelyn Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):354–379, 2012.

	OMDC-ADMM		BMDC-ADMM		VCA		NMF		ℓ_1-NMF		$\ell_{1/2}$-NMF	
Endmembers	SAD	RMSE	SAD	RMSE	SAD	RMSE	SAD	RMSE	SAD	RMSE	SAD	RMSE
Soil	0.0839	0.0897	0.0886	0.0893	0.3584	0.2936	0.1569	0.2557	0.0898	0.2463	0.0506	0.2679
Tree	0.0447	0.0398	0.0905	0.0667	0.2565	0.3268	0.2130	0.1402	0.0680	0.0636	0.0409	0.0707
Water	0.1113	0.0343	0.3141	0.0521	0.2474	0.3151	0.2001	0.1106	0.3815	0.0660	0.1682	0.1031
Road	0.0498	0.0787	0.0473	0.0693	0.5489	0.2829	0.3522	0.2450	0.4118	0.2344	0.3670	0.2737
Average	0.0724	0.0606	0.1351	0.0694	0.3528	0.3046	0.2305	0.1879	0.2378	0.1526	0.1567	0.1789

TABLE III: SAD and RMSE for OMDC-ADMM, BMDC-ADMM, VCA, NMF, ℓ_1-NMF and $\ell_{1/2}$-NMF

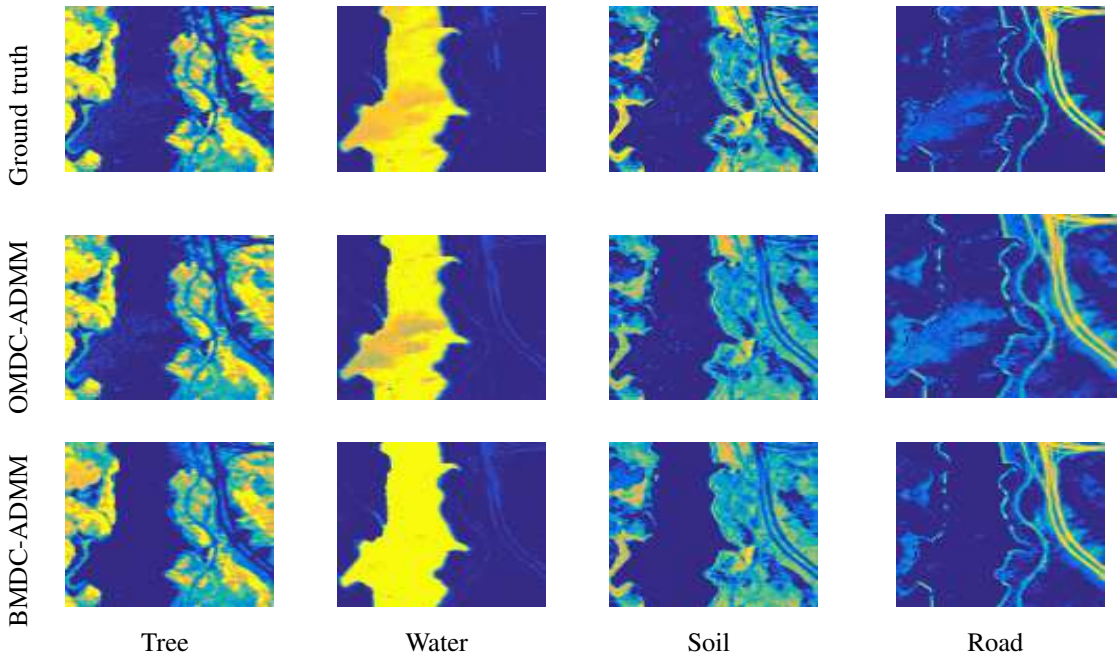


Fig. 18: True and estimated abundance maps by on-line and batch MDC-ADMM

[5] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[6] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[7] Serhat S Bucak and Bilge Gunsel. Incremental subspace learning via Non-negative Matrix Factorization. *Pattern recognition*, 42(5):788–797, 2009.

[8] Tsung-Han Chan, Chong-Yung Chi, Yu-Min Huang, and Wing-Kin Ma. A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing. *IEEE Transactions on Signal Processing*, 57(11):4418–4432, 2009.

[9] Maurice D Craig. Minimum-volume transforms for remotely sensed data. *IEEE Transactions on Geoscience and Remote Sensing*, 32(3):542–552, 1994.

[10] David Donoho and Victoria Stodden. When does Non-negative Matrix Factorization give a correct decomposition into parts? In *Advances in Neural Information Processing Systems*, pages 1141–1148, 2004.

[11] Lucas Drumetz. *Endmember Variability in hyperspectral image unmixing*. PhD thesis, Université Grenoble Alpes, 2016.

[12] Maryam Fazel, Haitham Hindi, and Stephen P Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *IEEE Proceedings of the 2003 American Control Conference*, volume 3, pages 2156–2162, 2003.

[13] Xiao Fu, Kejun Huang, Nicholas D Sidiropoulos, and Wing-Kin Ma. Non-negative Matrix Factorization for signal and data analytics: Identifiability, algorithms, and applications. *arXiv preprint arXiv:1803.01257*, 2018.

[14] Xiao Fu, Kejun Huang, Bo Yang, Wing-Kin Ma, and Nicholas D Sidiropoulos. Robust volume minimization-based matrix factorization for remote sensing and document clustering. *IEEE Transactions on Signal Processing*, 64(23):6254–6268, 2016.

[15] Naiyang Guan, Dacheng Tao, Zhigang Luo, and Bo Yuan. On-line Non-negative Matrix Factorization with robust stochastic approximation. *IEEE Transactions on Neural Networks and Learning Systems*, 23(7):1087–1099, 2012.

[16] Patrik O Hoyer. Non-negative sparse coding. In *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*, pages 557–565, 2002.

[17] Kejun Huang, Nicholas D Sidiropoulos, and Athanasios P Liavas. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing*, 64(19):5052–5065, 2016.

[18] Kejun Huang, Nicholas D Sidiropoulos, and Ananthram Swami. Non-negative Matrix Factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing*, 62(1):211–224, 2014.

[19] Nirmal Keshava and John F Mustard. Spectral unmixing. *IEEE Signal Processing Magazine*, 19(1):44–57, 2002.

[20] Hans Laurberg, Mads Græsbøll Christensen, Mark D Plumbley, Lars Kai Hansen, and Søren Holdt Jensen. Theorems on positive data: On the uniqueness of NMF. *Computational Intelligence and Neuroscience*, 2008.

[21] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by Non-negative Matrix Factorization. *Nature*, 401(6755):788–791, 1999.

[22] Augustin Lefèvre, Francis Bach, and Cédric Févotte. On-line algorithms for Non-negative Matrix Factorization with the Itakura-Saito divergence. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 313–316, 2011.

[23] Valentin Leplat, Andersen MS Ang, and Nicolas Gillis. Minimum-volume rank-deficient Non-negative Matrix Factorizations. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3402–3406, 2019.

[24] Jun Li and José M Bioucas-Dias. Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data. In *IEEE International*

Geoscience and Remote Sensing Symposium, volume 3, pages III–250, 2008.

[25] Chih-Jen Lin. Projected gradient methods for Non-negative Matrix Factorization. *Neural Computation*, 19(10):2756–2779, 2007.

[26] Guolan Lu and Baowei Fei. Medical hyperspectral imaging: a review. *Journal of Biomedical Optics*, 19(1):010901, 2014.

[27] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. On-line learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(Jan):19–60, 2010.

[28] Lidan Miao and Hairong Qi. Endmember extraction from highly mixed data using minimum volume constrained Non-negative Matrix Factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 45(3):765–777, 2007.

[29] José MP Nascimento and José MB Dias. Vertex Component Analysis: A fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):898–910, 2005.

[30] Ludivine Nus, Sebastian Miron, and David Brie. On-line blind unmixing for hyperspectral pushbroom imaging systems. In *IEEE Statistical Signal Processing Workshop*, pages 418–422, 2018.

[31] Bo Peng. The determinant: A means to calculate volume. *Recall*, 21:a22, 2007.

[32] Yuntao Qian, Sen Jia, Jun Zhou, and Antonio Robles-Kelly. Hyperspectral unmixing via $\ell_{1/2}$ sparsity-constrained Non-negative Matrix Factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 49(11):4282–4297, 2011.

[33] Reinhard Schachtner, Gerhard Pöppel, Ana Maria Tomé, and Elmar W Lang. Minimum determinant constraint for Non-negative Matrix Factorization. In *International Conference on Independent Component Analysis and Signal Separation*, pages 106–113. Springer, 2009.

[34] Gilbert Strang. *Linear algebra and its applications*. Thomson, Brooks/Cole, 2006.

[35] Dennis L Sun and Cedric Fevotte. Alternating direction method of multipliers for Non-negative Matrix Factorization with the beta-divergence. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6201–6205, 2014.

[36] Dong Wang and Huchuan Lu. On-line learning parts-based representation via incremental orthogonal projective Non-negative Matrix Factorization. *Signal Processing*, 93(6):1608–1623, 2013.

[37] Fei Wang, Ping Li, and Arnd Christian König. Efficient document clustering via On-line Non-negative Matrix Factorizations. In *Society for Industrial and Applied Mathematics*, volume 11, pages 908–919, 2011.

[38] Michael E Winter. N-FINDR: An algorithm for fast autonomous spectral endmember determination in hyperspectral data. In *Imaging Spectrometry V*, volume 3753, pages 266–276. International Society for Optics and Photonics, 1999.

[39] Yi Wu, Bin Shen, and Haibin Ling. Visual tracking via On-line Non-negative Matrix Factorization. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(3):374–383, 2014.

[40] Yangyang Xu, Wotao Yin, Zaiwen Wen, and Yin Zhang. An alternating direction algorithm for matrix completion with non-negative factors. *Frontiers of Mathematics in China*, 7(2):365–384, 2012.

[41] Yue Yu and Weidong Sun. Minimum distance constrained Non-negative Matrix Factorization for the endmember extraction of hyperspectral images. In *Remote Sensing and GIS Data Processing and Applications; and Innovative Multispectral Technology and Applications*, volume 6790, page 679015. International Society for Optics and Photonics, 2007.

[42] Renbo Zhao and Vincent YF Tan. Online Non-negative Matrix Factorization with outliers. *IEEE Transactions on Signal Processing*, 65(3):555–570, 2016.

[43] Guoxu Zhou, Shengli Xie, Zuyuan Yang, Jun-Mei Yang, and Zhaoshui He. Minimum-volume-constrained Non-negative Matrix Factorization: Enhanced ability of learning parts. *IEEE Transactions on Neural Networks*, 22(10):1626–1637, 2011.

[44] Guoxu Zhou, Zuyuan Yang, Shengli Xie, and Jun-Mei Yang. Online blind source separation using Incremental Non-negative Matrix Factorization with volume constraint. *IEEE Transactions on Neural Networks*, 22(4):550–560, 2011.

[45] Feiyun Zhu. Hyperspectral unmixing: Ground truth labeling, datasets, benchmark performances and survey. *arXiv preprint arXiv:1708.05125*, pages 1–15.