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

Ludivine Nus, Sebastian Miron, David Brie

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

Ludivine NUS, Sebastian MIRON, David BRIE[1]

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 (ADMM) approach, adapted to pushbroom imaging systems. Because of the generally ill-posed nature of the unmixing problem, we impose a minimum endmembers dispersion constraint to regularize the solution; this constraint can be interpreted as a convex relaxation of the minimum volume constraint and therefore, presents interesting optimization properties. The proposed algorithm presents faster convergence rate and lower computational complexity compared to the algorithms based on multiplicative update rules. Experimental results on synthetic and real datasets, and comparison to state-of-the-art algorithms, demonstrate the effectiveness of our method in terms of rapidity and accuracy.

Keywords:
Hyperspectral imaging, Pushbroom acquisition system, On-line unmixing, Alternating Direction Method of Multipliers, Minimum dispersion constraint

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

1. Introduction

Hyperspectral imaging, which combines the power of digital imaging and spectroscopy, is an effective tool in a wide range of applications (see *e.g.*, [1, 2, 3]). 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, hyperspectral 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) unmixing of hyperspectral Near InfraRed (NIR) images acquired by a *pushbroom* imager [4], 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. In particular, the approach proposed in this paper is part of a project that aims at predicting and classifying in real-time, the rendering quality of pieces of wood in an industrial process. This is to be done by *on-line* unmixing method of hyperspectral images of wood surface, which presents several advantages over the off-line methods: *i)* it is well-adapted to real-time data processing for on-line industrial acquisition systems; *ii)* it allows to alleviate computational burden and reduce memory requirements for big hyperspectral data cubes; *iii)* it permits to track the spatial/time variability of the endmembers in a hyperspectral imaging application.

1.1. Hyperspectral image unmixing

Hyperspectral unmixing methods are based on a mixture model describing how the endmembers are combined in the acquired image. Depending on the application and/or on the data, this model can be linear or non-linear [5]. In this article, we focus on the Linear Mixing Model (LMM) [6]; 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 many 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 rows 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. Figure 1 illustrates this geometrical representation for a mixture of $R = 3$ endmembers: all the pixels (red points) belongs to the $(R-1)$-dimensional simplex (blue lines) whose vertices are the endmembers $\mathbf{s}_1, ..., \mathbf{s}_R$ (gray points). When the sum-to-one constraint does not hold, the observations belong to the positive cone defined by the endmembers.

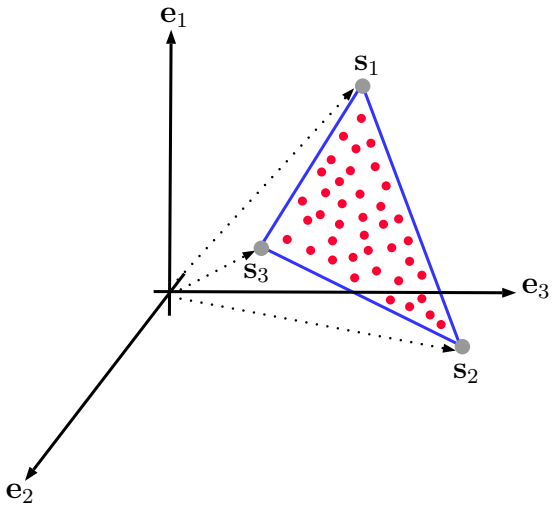


Figure 1: Geometrical representation of the LMM for $R = 3$ endmembers

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.*, [7, 8]. However, the pure pixel assumption is a strong requirement that does not hold in general for highly mixed data, and finding the endmembers is therefore a more challenging task. For this case, another 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 [10, 11]. However the performance and computational efficiency of these methods are often limited because of complicated simplex volume calculations, sensitivity to initialization and lack of rigorous performance analysis. A detailed analysis can be found in [5].

To overcome the up-mentionned problems, Non-negative Matrix Factorization (NMF) [12] 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 [13, 14, 15, 16, 17, 18, 19, 20], which can be considered as state-of-the-art methods in blind hyperspectral unmixing.

1.2. Pushbroom acquisition scheme

In pushbroom imaging systems, hyperspectral data cubes are acquired slice by slice, sequentially in time. Each slice is an image, characterized by a spectral dimension and a spatial dimension (one line of the scene). Figure 2 illustrates the test bench that we consider for the real-time characterization of the wood material. Hyperspectral images are acquired line by line, as the pieces of wood are carried under the imager, via a conveyor.

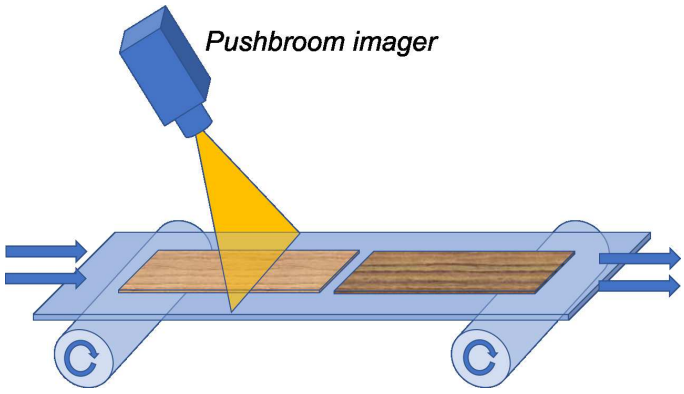


Figure 2: Acquisition of hyperspectral images of wood

The stream of spectral-spatial 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 Figure 3) is a matrix of dimensions $L \times P$, where L denotes the spectral dimension (wavelengths) and P, the across track spatial dimension (one line of the scene).

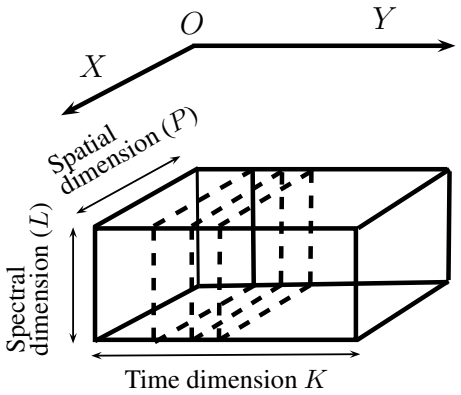


Figure 3: Data structure for pushbroom acquisition

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.

1.3. 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 [21, 22, 23, 24], the endmembers do not vary from one sample to another, while in [25, 26, 27, 28, 29, 30], the endmembers may evolve between successive samples. In particular, the Incremental NMF (INMF) [25] 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 [27, 28, 29]. Recently, the

on-line Minimum Volume Constraint-NMF algorithm (OMVC-NMF) was introduced in [30]. It is a straightforward adaptation of [27] specially designed for pushbroom hyperspectral imaging system. To the best knowledge of the present authors, OMVC-NMF is the only unmixing algorithm in the literature, adapted to on-line processing of hyperspectral images. Thus, it will serve as benchmark for the on-line volume regularized NMF method proposed in this paper. Most of the NMF 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 [31]. These limitations are restrictive in particular for the considered application; indeed, the on-line unmixing algorithms must be fast in order to fulfill the real-time industrial production constraints. An interesting alternative to multiplicative updates is represented by the optimization methods based on Alternating Direction Method of Multipliers (ADMM) in [24]; ADMM proved its superiority over multiplicative updates with respect to both reconstruction accuracy and convergence rate [32, 33, 34].

1.4. 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 state-of-the-art on-line approaches (see *e.g.*, [24]) are:

1. The addition of the Minimum Dispersion Constraint (MDC) [10], 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 to model dynamic content changes.
3. The use of ADMM-based optimization 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 theoretical 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. Comparison to off-line (or *batch*) algorithms as well as to an on-line matrix factorization method, are also provided. We conclude the paper in Section 6.

2. On-line hyperspectral unmixing for pushbroom acquisition scheme

2.1. 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 Figure 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 $\mathbf{A}^{(k+1)} = \left[\mathbf{A}^{(k)}\ \tilde{\mathbf{A}}^{(k+1)}\right]$.

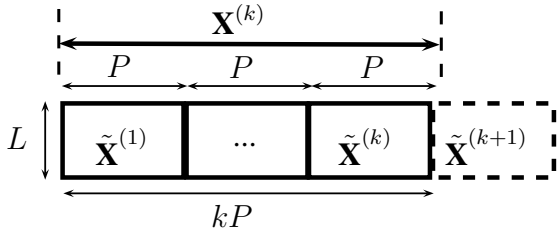


Figure 4: Unfolded pushbroom hyperspectral image

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

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

2.2. 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:

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

Without further assumptions, (3) is just a set of k independent least squares problems 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:

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

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 [25]. 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 $\text{Vol}\left(\tilde{\mathbf{S}}^{(k+1)}\right)$ and integrate it into the cost function as follows:

$$\mathcal{J}_{\text{Vol}}^{(k+1)}\left(\tilde{\mathbf{S}}^{(k+1)}, \tilde{\mathbf{A}}^{(k+1)}\right) = \alpha\mathcal{J}^{(k)} + (1-\alpha)\tilde{\mathcal{J}}^{(k+1)} + \mu\ \text{Vol}\left(\tilde{\mathbf{S}}^{(k+1)}\right), \quad (5)$$

where $\mathcal{J}_{\text{Vol}}$ 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 $\text{Vol}\left(\tilde{\mathbf{S}}^{(k+1)}\right)$ is discussed in the next section. Thus, the approach proposed in this paper aims at solving the following optimization problem:

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

for a particular choice of the volume regularizer term.

3. Volume regularization

The classical measure for the minimum volume constraint is the determinant, *i.e.* $\text{Vol}\left(\tilde{\mathbf{S}}^{(k+1)}\right) = \det\left(\tilde{\mathbf{S}}^{(k+1)}\right)$ [35]. 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 $\text{Vol}\left(\tilde{\mathbf{S}}^{(k+1)}\right)$ is [15, 16]:

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

or its variants presented in [13, 36]. The expression (7) is mathematically justified because it can be interpreted as the square volume of the simplex bounded by the endmembers [37, Theorem 7].

3.1. Identifiability

A lot of work has been done in the last decade to understand the identifiability of the NMF model. The results in [38, 39, 40] 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, the 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. By applying the recent results of [17] to our on-line mixture model, the following sufficient identifiability conditions can be formulated:

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 [17] for the exact definition of sufficiently scattered), then the model (2) is identifiable under the minimum volume constraint.

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

3.2. Volume regularizers

In this section, to simplify the notations, the upper indices indicating the slice number 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 difficulties due to determinant minimization, others volume regularizer surrogates can be considered. An alternative formulation for $\text{Vol}\left(\tilde{\mathbf{S}}\right)$, proposed in [41] 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 [17, 20]:

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

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 $\{\lambda_i^2, i = 1, ..., R\}$ 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:

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

$$\frac{\partial f(\lambda_i)}{\partial\lambda_i}=\frac{2\lambda_i}{\lambda_i^2+\epsilon}.$$

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 $\lambda_i=0$, $\log\left(\det\left(\tilde{\mathbf{S}}^T\tilde{\mathbf{S}}+\epsilon\mathbf{I}\right)\right)$ 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 [42], 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 [10] as:

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

where $\mathbf{P}=\mathbf{I}-\frac{1}{R}\mathbf{1}\mathbf{1}^T$ ($\mathbf{1}$ is an all-ones column vector of size $R\times 1$). 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 Figure 5 the two functions: MVC with varying ϵ and MDC, for increasing values of β; the curves are 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 (*e.g.* 10^{-9}) to avoid a bad conditioning of $\tilde{\mathbf{S}}^T\tilde{\mathbf{S}} + \epsilon\mathbf{I}$. For larger ϵ values (*e.g.* 0.1), 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 3.1 is satisfied.

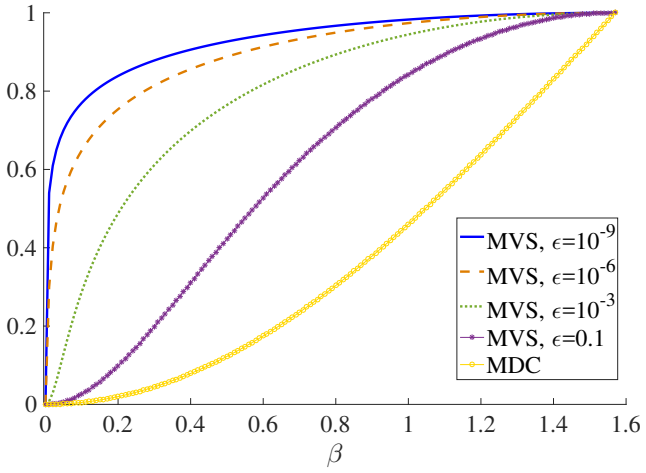


Figure 5: Illustrations of MVC and MDC penalties as functions of the angle β

4. OMDC-ADMM algorithm

In the following, we only consider MDC (rather than MVC) because of its convexity which 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).

4.1. Algorithm derivation

Let us define:

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

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

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

where $\mathbb{I}_{\mathbb{R}_+}$ is the indicator function of $\mathbb{R}_+$ ensuring the non-negativity of the endmembers and abundances. For algorithmic convenience, we use the scaled form of ADMM [43] 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:

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

The parameter $\rho > 0$ controls the convergence rate 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 at data slice $k+1$ can be expressed as:

$$\tilde{\mathbf{A}}^{(k+1)} = \left((1-\alpha)\tilde{\mathbf{S}}^{(k)^T}\tilde{\mathbf{S}}^{(k)} + \rho\mathbf{I}\right)^{-1}\left((1-\alpha)\tilde{\mathbf{S}}^{(k)^T}\tilde{\mathbf{X}}^{(k+1)} + \rho\left(\tilde{\mathbf{V}}^{(k)} - \tilde{\mathbf{\Pi}}^{(k)}\right)\right), \tag{15}$$

$$\tilde{\mathbf{V}}^{(k+1)} = \max\left(\mathbf{0}, \tilde{\mathbf{A}}^{(k+1)} + \tilde{\mathbf{\Pi}}^{(k)}\right), \tag{16}$$

$$\tilde{\mathbf{\Pi}}^{(k+1)} = \tilde{\mathbf{\Pi}}^{(k)} + \tilde{\mathbf{A}}^{(k+1)} - \tilde{\mathbf{V}}^{(k+1)}, \tag{17}$$

$$\tilde{\mathbf{S}}^{(k+1)} = \left(\mathbf{N}^{(k+1)} + \rho\left(\tilde{\mathbf{U}}^{(k)} - \tilde{\mathbf{\Lambda}}^{(k)}\right)\right)\left(\mathbf{M}^{(k+1)} + 2\mu\mathbf{P} + \rho\mathbf{I}\right)^{-1}, \tag{18}$$

$$\tilde{\mathbf{U}}^{(k+1)} = \max\left(\mathbf{0}, \tilde{\mathbf{S}}^{(k+1)} + \tilde{\mathbf{\Lambda}}^{(k)}\right), \tag{19}$$

$$\tilde{\mathbf{\Lambda}}^{(k+1)} = \tilde{\mathbf{\Lambda}}^{(k)} + \tilde{\mathbf{S}}^{(k+1)} - \tilde{\mathbf{U}}^{(k+1)}, \tag{20}$$

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 [25], 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 $\mathbf{M}^{(k)} \approx \sum_{l=1}^{k}\tilde{\mathbf{A}}^{(l)}\tilde{\mathbf{A}}^{(l)^T}$. Thus, $\mathbf{N}^{(k+1)}$ and $\mathbf{M}^{(k+1)}$ can be expressed as:

$$\mathbf{N}^{(k+1)} = \alpha\mathbf{N}^{(k)} + (1-\alpha)\tilde{\mathbf{X}}^{(k+1)}\tilde{\mathbf{A}}^{(k+1)^T}, \tag{21}$$

$$\mathbf{M}^{(k+1)} = \alpha\mathbf{M}^{(k)} + (1-\alpha)\tilde{\mathbf{A}}^{(k+1)}\tilde{\mathbf{A}}^{(k+1)T}. \tag{22}$$

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.

Algorithm 1 OMDC-ADMM

Inputs: $\mathbf{X}$; $\mathbf{P}$; R; α; μ; ρ; N_{iter};
Initialization: $k = 0$; $\mathbf{N} = \text{zeros}(L, R)$; $\mathbf{M} = \text{zeros}(R, R)$; $\tilde{\mathbf{S}} = \text{rand}(L, R)$; $\tilde{\mathbf{V}} = \text{zeros}(R, P)$; $\tilde{\mathbf{U}} = \text{zeros}(L, R)$; $\tilde{\mathbf{\Pi}} = \text{zeros}(R, P)$; $\tilde{\mathbf{\Lambda}} = \text{zeros}(L, R)$; $\mathbf{A} = [\,]$; $\mathbf{S} = [\,]$;
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
  $\mathbf{N} = \tilde{\mathbf{N}}$; $\mathbf{M} = \tilde{\mathbf{M}}$;
  $\mathbf{A} \leftarrow \left[\mathbf{A}\ \tilde{\mathbf{A}}\right]$; $\mathbf{S} \leftarrow \left[\mathbf{S}\ \tilde{\mathbf{S}}\right]$;
end while

4.2. 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 [44]. 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 [44, 34] and after some basic algebraic manipulations, it can be shown that a point $\mathbf{W}$ is a KKT point for the

problem (13) if:

$$(1-\alpha)\tilde{\mathbf{S}}^T\tilde{\mathbf{S}}\tilde{\mathbf{A}} - (1-\alpha)\tilde{\mathbf{S}}^T\tilde{\mathbf{X}} + \tilde{\mathbf{\Pi}} = \mathbf{0}, \tag{23a}$$

$$\tilde{\mathbf{S}}\tilde{\mathbf{M}} - \tilde{\mathbf{N}} + 2\mu\tilde{\mathbf{S}}\mathbf{P} + \tilde{\mathbf{\Lambda}} = \mathbf{0}, \tag{23b}$$

$$\tilde{\mathbf{A}} - \tilde{\mathbf{V}} = \mathbf{0}, \tag{23c}$$

$$\tilde{\mathbf{S}} - \tilde{\mathbf{U}} = \mathbf{0}, \tag{23d}$$

$$\tilde{\mathbf{\Pi}} \leq \mathbf{0} \leq \tilde{\mathbf{V}}, \tilde{\mathbf{\Pi}} \odot \tilde{\mathbf{V}} = \mathbf{0}, \tag{23e}$$

$$\tilde{\mathbf{\Lambda}} \leq \mathbf{0} \leq \tilde{\mathbf{U}}, \tilde{\mathbf{\Lambda}} \odot \tilde{\mathbf{U}} = \mathbf{0}, \tag{23f}$$

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 (15), the following relation can be written between $\tilde{\mathbf{A}}_t$ et $\tilde{\mathbf{A}}_{t+1}$:

$$\begin{aligned}&\left((1-\alpha)\tilde{\mathbf{S}}_t^T\tilde{\mathbf{S}}_t + \rho\mathbf{I}\right)\left(\tilde{\mathbf{A}}_{t+1} - \tilde{\mathbf{A}}_t\right)\\ &= -\left((1-\alpha)\tilde{\mathbf{S}}_t^T\tilde{\mathbf{S}}_t\tilde{\mathbf{A}}_t - (1-\alpha)\tilde{\mathbf{S}}_t^T\tilde{\mathbf{X}} + \rho\left(\tilde{\mathbf{A}}_t - \tilde{\mathbf{V}}_t\right) + \rho\tilde{\mathbf{\Pi}}_t\right).\end{aligned} \tag{24}$$

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 (24), 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 (23) 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 (23e), 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 (23f). 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 5.3 and 5.4.

4.3. 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 $(L \times P)$, N_{iter} iterations and a decomposition rank R, the computational complexity for OMDC-ADMM algorithm is of the order of $2KN_{iter}(RPL + (L + P)R^2 + R^3)$. 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 $2N_{iter}(RPKL + (L + PK)R^2 + R^3)$. Moreover, the OMVC-NMF algorithm presented in [30], based on multiplicative update rules with minimum volume constraint (9), has a computational complexity of the order of $KN_{iter}(2RPL + (L + P)R + (6L + 2P)R^2)$. 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.

5. Experimental results

In order to evaluate the performance of OMDC-ADMM for hyperspectral unmixing, we conducted several experiments on simulated and real hyperspectral images. 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. Illustrating the rank preserving properties of MDC and MVC.
2. 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.
3. Comparing OMDC-ADMM to its batch counterpart BMDC-ADMM to assess the advantages of on-line processing of hyperspectral images.
4. Studying the spectral tracking capability of OMDC-ADMM.

5. Validating the unmixing performance of our algorithm on real hyperspectral images of wood.

5.1. 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:

$$\frac{1}{2}\left\|\tilde{\mathbf{X}} - \hat{\tilde{\mathbf{S}}}\hat{\tilde{\mathbf{A}}}\right\|_F^2, \tag{25}$$

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

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

$$\frac{1}{R}\sum_{r=1}^{R}\cos^{-1}\left(\frac{\tilde{\mathbf{s}}_r^T\hat{\tilde{\mathbf{s}}}_r}{||\tilde{\mathbf{s}}_r||_2\ ||\hat{\tilde{\mathbf{s}}}_r||_2}\right), \tag{26}$$

where R is the number of endmembers, $||.||_2$, the ℓ_2-norm, $\tilde{\mathbf{S}} = [\tilde{\mathbf{s}}_1, ..., \tilde{\mathbf{s}}_R]$ and $\hat{\tilde{\mathbf{S}}} = \left[\hat{\tilde{\mathbf{s}}}_1, ..., \hat{\tilde{\mathbf{s}}}_R\right]$. A low value of SAD indicates that the estimate is close to the ground truth.

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

$$\frac{1}{R}\sum_{r=1}^{R}\sqrt{\frac{1}{KP}\sum_{d=1}^{KP}(a_{rd} - \hat{a}_{rd})^2}, \tag{27}$$

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

Note that expressions (26) and (27) are average values. For some experiments, the SAD and the RMSE values are calculated independently for each source $r = 1, ..., R$.

5.2. Rank preserving properties of MVC and MDC

We performed a numerical experiment to illustrate the rank preserving properties of both volume regularizers, MVC (9) and MDC (11). We used, for the simulations, the endmembers shown in Figure 8(d). 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 3.1). For μ ranging from 0.0001 to 0.003, the quantities trace $\left(\hat{\hat{\mathbf{S}}}\mathbf{P}\hat{\hat{\mathbf{S}}}^T\right)$ and $\log\left(\det\left(\hat{\hat{\mathbf{S}}}^T\hat{\hat{\mathbf{S}}} + \epsilon\mathbf{I}\right)\right)$ were computed, where $\hat{\hat{\mathbf{S}}}$ is the endmember matrix estimated by either OMVC-NMF or OMDC-ADMM at the last slice k of the image. For MVC, we used four different values of ϵ: $10^{-9}, 10^{-4}, 10^{-3}$ and 10^{-2}. The normalized curves (maximum value equal to 1, minimum value equal to 0) are shown in Figure 6.

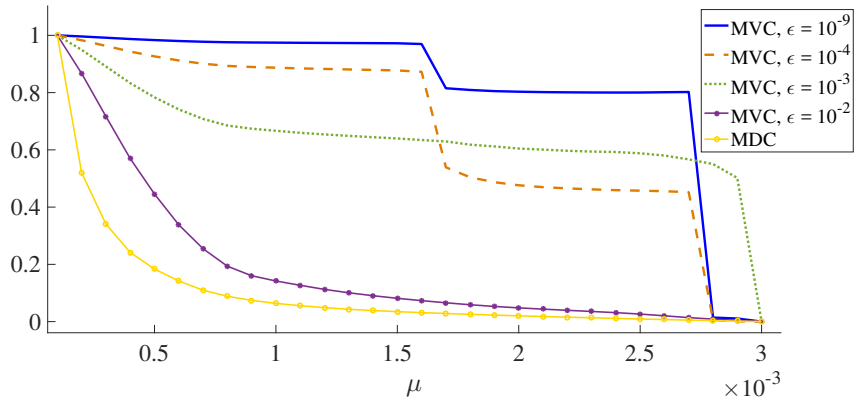
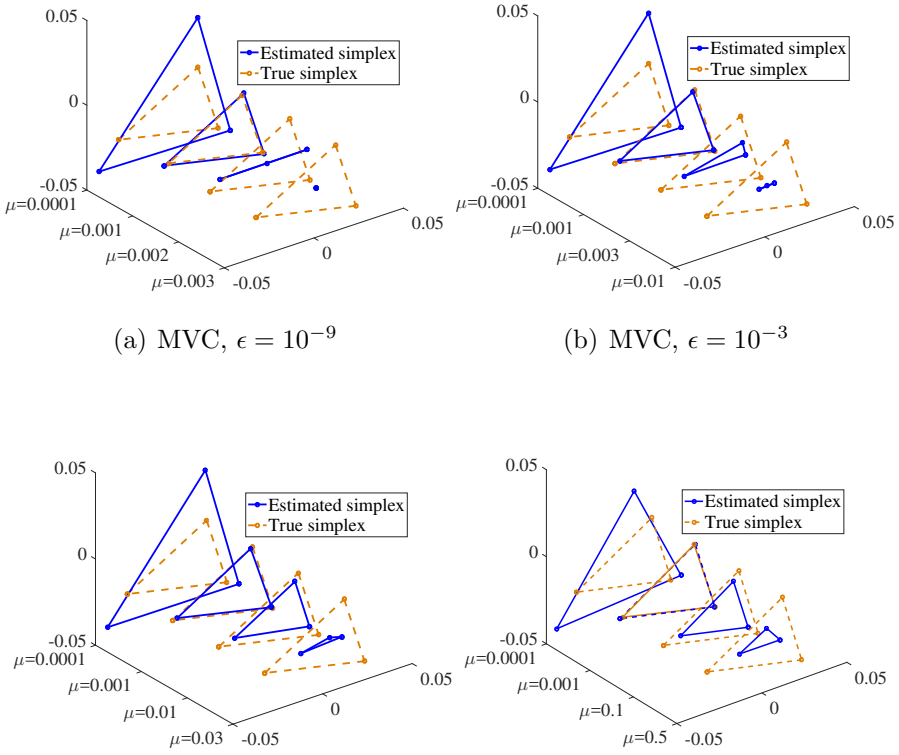


Figure 6: The plots of MVC and MDC cost functions for different values of μ

In Figure 6, several behaviors can be observed. For MVC with $\epsilon = 10^{-9}$, 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; this is reflected on the curves by abrupt staircase shape variations. When ϵ increases, rank deficiency is no longer promoted. In Figure 7, we compared the true and estimated simplexes for different values of μ, for MVC ($\epsilon = 10^{-9}$, 10^{-3}, and 10^{-2}) and MDC. For $\epsilon = 10^{-9}$, as μ

increases, the volume of the estimated simplex tends to zero, meaning that the endmember matrix is rank deficient. In the limit case ($\mu \to +\infty$), all endmembers become collinear and the simplex is transformed into a point. 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 (Figure 7(d)), the simplex decreases progressively, while preserving the rank of the decomposition.



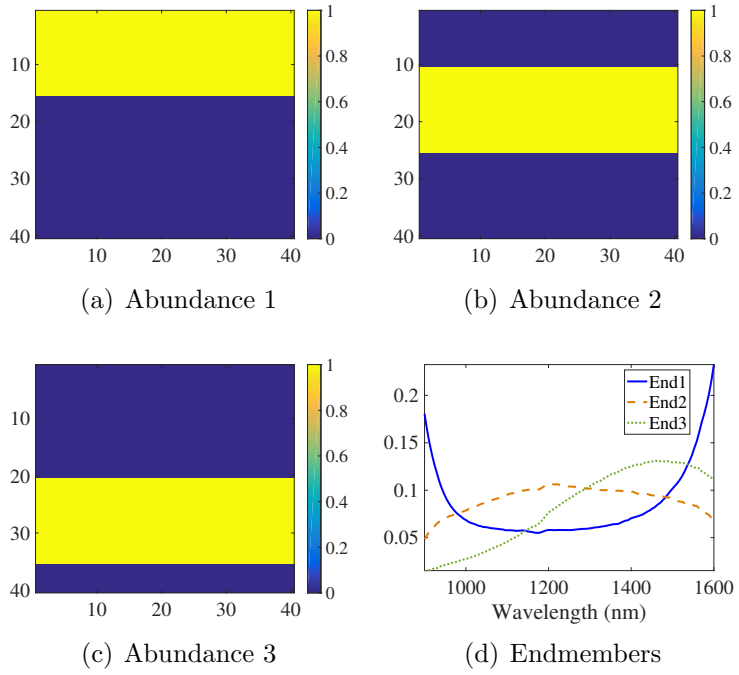
(a) MVC, $\epsilon = 10^{-9}$ (b) MVC, $\epsilon = 10^{-3}$

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

Figure 7: Illustration of simulated and true simplexes as functions of μ for MVC and MDC

5.3. Convergence rate

The convergence speed of OMDC-ADMM and OMVC-NMF was numerically 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 Figure 8. The data were corrupted by a low-level noise.



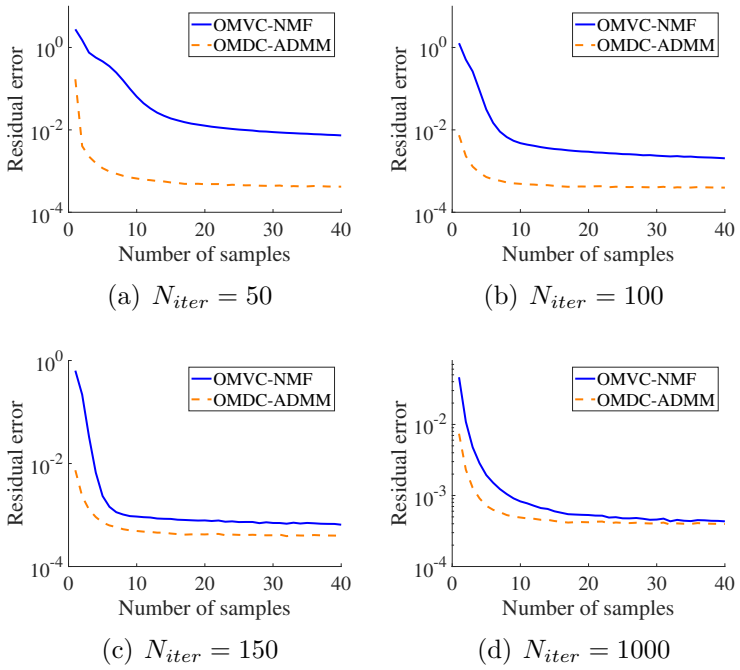
(a) Abundance 1 (b) Abundance 2

(c) Abundance 3 (d) Endmembers

Figure 8: Simulated abundances and endmembers (End)

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. 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 Figure 9. Note that a logarithmic scale was used

for the vertical axis. In Figure 9, we observe that for all the considered values of N_{iter}, OMDC-ADMM has a faster convergence rate than OMVC-NMF. The latter reaches asymptotically an error close to OMDC-ADMM for approximately $N_{iter} = 1000$ iterations.



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

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

Figure 9: The residual error for OMVC-NMF and OMDC-ADMM as functions of number of samples (for the data represented in Figure 8)

In order to quantify the estimation quality of the endmembers and of the abundances for different values of N_{iter}, we computed the SAD and the RMSE performance criteria. The SAD was calculated using the endmembers estimated at the last slice, which is coherent in the case of stationary sources. The obtained values are given in Tables 1 and 2. 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. OMDC-ADMM processes the entire image 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 1: The values of SAD for OMVC-NMF and OMDC-ADMM for different number of iterations (for the unmixing of the data represented in Figure 8)

	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 2: The values of RMSE for OMVC-NMF and OMDC-ADMM for different number of iterations (for the unmixing of the data represented in Figure 8)

Another interest of the sequential algorithm for hyperspectral unmixing is the processing of large hyperspectral datasets. To illustrate this aspect, 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 Figure 8. Both algorithms used the same $\rho = 0.001$; after several trials, the minimum dispersion constraint was set to $\mu = 0.003$ for the on-line version and to $\mu = 0.5$ for the batch version, for optimum performance. 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, while the on-line version processed the image slice by slice. To study the convergence speeds of both algorithms, the SAD was evaluated for different values of N_{iter}, ranging from 10 to 2000. Note that for a given value of N_{iter}, the SAD for OMDC-ADMM was calculated for each slice of the image and then averaged, in order to make a fair comparison between the two algorithms. The results obtained by averaging over 20 trials are illustrated in Figure 10. One can see that our on-line algorithm converges much faster than its batch counterpart. 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 versions established in section 4.3; if $R = 3$, $P = 40$, $L = 119$, $K = 40$, $N_{iter} = 100$ 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.

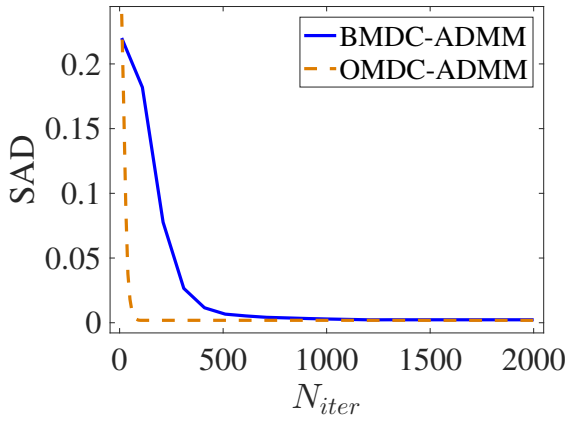


Figure 10: SAD for BMDC-ADMM and OMDC-ADMM as functions of N_{iter} (for the unmixing of the data represented in Figure 8)

We also examined the influence of parameters α and ρ 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 0.8 while fixing the values of $\rho = 0.001$ and $\mu = 0.003$. The values of SAD as functions of N_{iter}, for different values of α, are illustrated in Figure 11. As α increases, fewer iterations are required to converge to the correct solution; this can be explained by the fact that α and μ parameters are strongly linked. Actually, if α changes, the value of μ must also be modified, in order to obtain optimum performance.

Next, we varied ρ from 0.01 to 0.0003 while keeping the value of $\alpha = 0.99$, $\mu = 0.003$ and $N_{iter} = 100$. Figure 12 shows that, when ρ decreases, the asymptotic SAD decreases, while the convergence speed does not seem to be affected. As soon as the value of ρ becomes too low (*e.g.* 0.00003), we notice that the SAD increases, a sign that the algorithm becomes unstable.

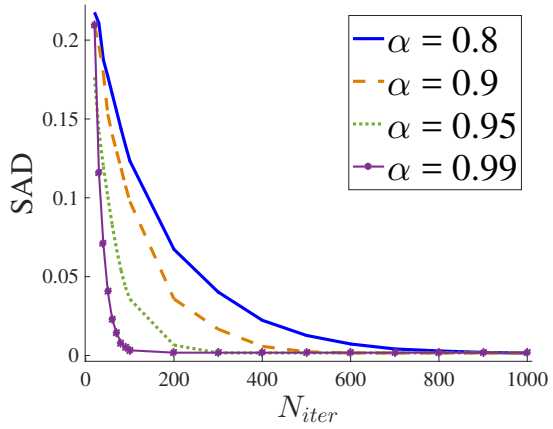


Figure 11: SAD for OMDC-ADMM as functions of N_{iter} for different values of α (for the unmixing of the data represented in Figure 8)

For the proper functioning of OMDC-ADMM, there is a range of suitable values for the parameter ρ; for this experiment, a value of ρ between 0.001 and 0.00005 ensures the stability of the algorithm.

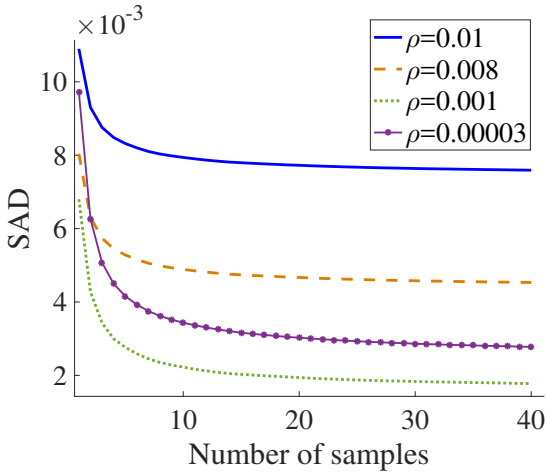


Figure 12: SAD for OMDC-ADMM as functions of number of samples for different values of ρ (for the unmixing of the data represented in Figure 8)

The last aspect studied by numerical simulations is the sensitivity of the algorithm to initial conditions. We used 20 different random initializations for OMDC-ADMM on the synthetic data corrupted by a low-level noise. The SAD and the RMSE were computed for each initialization, using $N_{iter} = 100$ and $\mu = 0, 0.0001, 0.0005, 0.003$ and 0.01. The results are shown in Figure 13 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. Finally, when the value of μ is too high (> 0.003), the SAD and the RMSE increase, indicating that the problem is over-regularized.

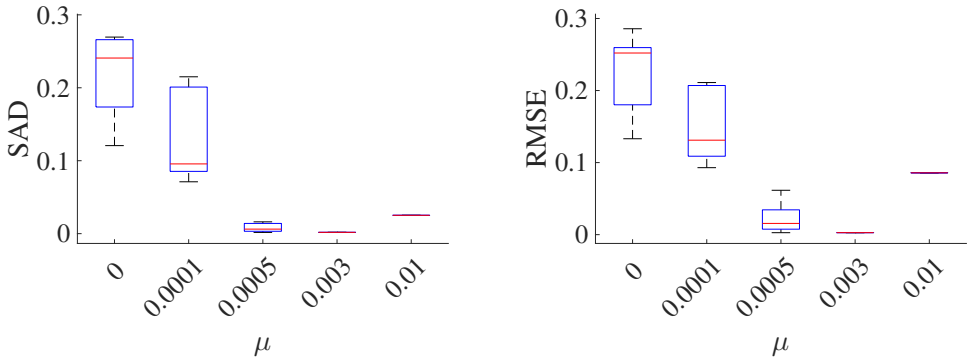


Figure 13: SAD and RMSE for different initializations of OMDC-ADMM as functions of μ (for the unmixing of the data represented in Figure 8)

5.4. Tracking the spectral variability

In hyperspectral imaging, the spectral signature of a component may present intra-class variability [45, 46]. Thus, studying the evolution of the endmembers between successive samples is sometimes crucial in understanding the underlying physical phenomenon. Integrating a tracking capability (via the parameter α) allows the tracking of dynamic spectral changes. To address this point, a hyperspectral image of size $119 \times 40 \times 80$ with non-stationary endmembers was simulated. The abundance maps were identical to those in Figure 8, but the endmembers evolved at each slice, according to the following model: $\tilde{\mathbf{S}}^{(k+1)} = \tilde{\mathbf{S}}^{(k)} + \mathbf{T}^{(k)}$, where $\mathbf{T}^{(k)}$ is a low-level random noise matrix 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 0.99 to 0.7 and, for each slice k, we computed the SAD. The results are shown in Figure 14. As α decreases, both convergence speed and asymptotic SAD ($k \to +\infty$) increase. Note that for very large values of α (*e.g.* 0.99), an increase of the asymptotic SAD is observed. This indicates that there exists an optimal value of α mitigating at best the transient error and tracking error.

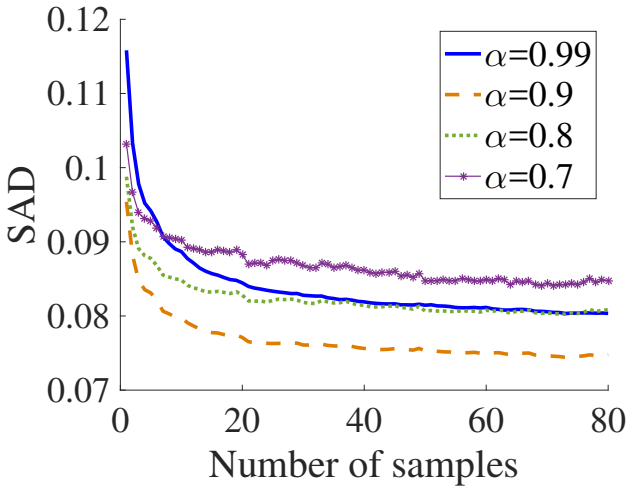


Figure 14: SAD for OMDC-ADMM as functions of number of samples for different values of α, in the case of non-stationary endmembers

5.5. Real data application: on-line analysis of wood surface

In this section, the performance of OMDC-ADMM is tested on two real hyperspectral images of wood, recorded by a pushbroom acquisition system (which is the target application of our method) at CRAN (Centre de Recherche en Automatique de Nancy) laboratory.

Wood is a natural material whose rendering is appreciated in many applications. However, its surface heterogeneity and variability makes it difficult to control the quality of the final product. These technical difficulties generate significant non-quality costs and manual sorting in wood industries. Current industrial technical solutions only detect very pronounced defects, such as knots, using color vision systems. There are other types of defects,

which have high colorimetric variability, and thus, are not systematically detected by the existing systems; this is the case of the sapwood (material located between the bark and the heartwood) and of the dark grain (dark color wood fibers), for example. The sapwood is generally undesirable in many applications due to its porosity, while dark grain-type defects are undesirable, particularly in the parquet industry, because they degrade the aesthetic appearance of the product. Thus, the detection of these defects represents an important economical issue, in many wood industry branches. A relevant technique to overcome the limitations of color vision systems is NIR hyperspectral imaging, which is the target application of this work. For the next experiment, we considered the wood samples shown in Figure 15, measuring 100 cm $\times$ 15 cm.

(a) Sample 1

(b) Sample 2

Figure 15: Selected wood samples (visible range color image)

The sample 1 had a sapwood-type defect in the top left corner and in the bottom right corner. The sample 2 had a dark grain area in the middle, over its entire length. These defects were hardly distinguishable by the colorimetric systems and were manually labeled by experts. The two pieces of wood were scanned by a NIR pushbroom imager working in the spectral range of 900-1600 nm, resulting in hyperspectral images of dimension $224 \times 264 \times 1253$ (for sample 1) and $224 \times 200 \times 1101$ (for sample 2), where 224 represents the number of wavelengths and the other two numbers, the

(spatial $\times$ time) dimensions. The acquired images were processed sequentially by OMDC-ADMM, slice by slice. The parameters of the algorithm were set as follows: $\mu = 3$, $\alpha = 0.99$, $\rho = 0.001$ and $N_{iter} = 15$. The number of endmembers to be extracted was set to $R = 3$, after several preliminary tests.

The three abundance maps generated by OMDC-ADMM for the sample 1 and for the sample 2 are presented in Figure 16 and 17, respectively. On the first abundance map in Figures 16 and 17, we observe that the areas corresponding to the sapwood and to the dark grain are much more contrasted than the rest of the image. These are very promising results, as, for the considered samples, the color of the sapwood/dark grain was very close to the color of heartwood, and therefore, the wood planks were considered as compliant with industrial quality standards by color vision systems. For the considered datasets, the processing time by OMDC-ADMM was of 3 s. The processing time can be largely reduced by low-level implementation of our algorithm directly on the production line to satisfy the real-time industrial constraint of 1 linear m of wood /s.

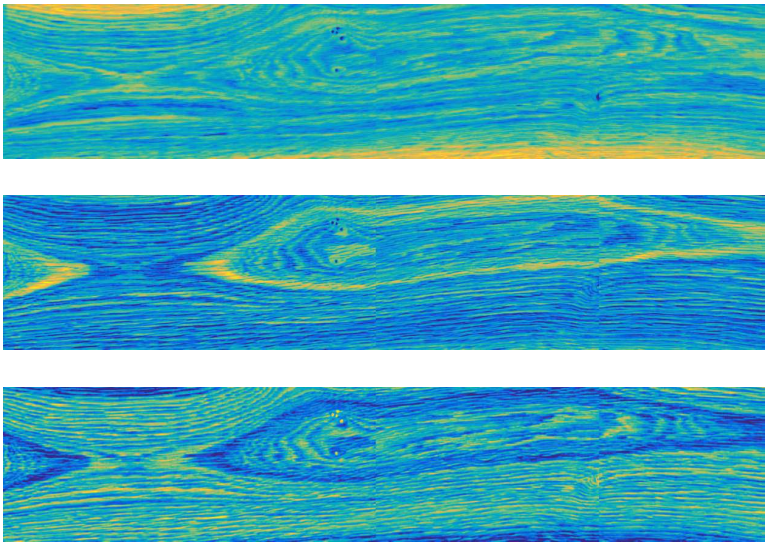


Figure 16: The three abundance maps estimated by OMDC-ADMM (for the wood sample 1 represented in Figure 15(a))

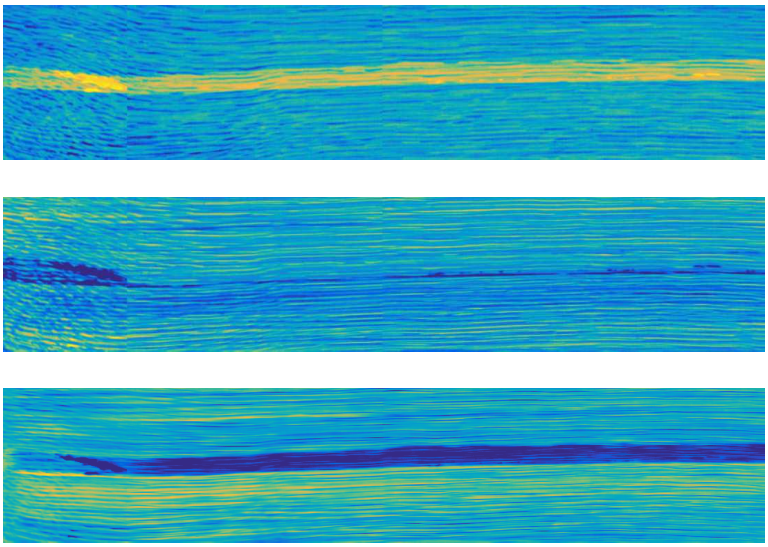


Figure 17: The three abundance maps estimated by OMDC-ADMM (for the wood sample 2 represented in Figure 15(b))

To further assess the performance of the proposed OMDC-ADMM algorithm, we also applied it to a real dataset, called *Jasper Ridge* (`http://lesun.weebly.com/hyperspectral-data-set.html`) with available ground truth. This image is commonly used in hyperspectral unmixing community for benchmarking purposes [47]. The results are given in the AppendixA.

6. 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. Moreover, these experiments 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. From a methodological point of view, we proved the interest of using the

minimum dispersion constraint over the minimum volume one, in particular its capacity to regularize the problem and to stabilize the solution. Finally, OMDC-ADMM proved to be an interesting solution for on-line unmixing of pushbroom hyperspectral images, compliant with the real-time constraints of the industrial wood sorting systems.

AppendixA. Test of OMDC-ADMM on *Jasper Ridge* hyperspectral image

We compared the performance of OMDC-ADMM to its batch version (BMDC-ADMM) and to other state-of-the-art (batch) methods: Vertex Component Analysis (VCA) [8], NMF [12], ℓ_1-NMF [48] and $\ell_{1/2}$-NMF [49]. The results for the last four methods come from simulations conducted in [47]. The *Jasper Ridge* 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 tree, water, soil and road.

In order to respect the pushbroom acquisition scheme that we consider, the image was processed by OMDC-ADMM sequentially, by slice of dimension 198×100, and 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 [47]. 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 the SAD and the RMSE are provided in Table A.3.

It can be seen, from Table A.3, 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 (0.1186 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.

	OMDC-ADMM		BMDC-ADMM		VCA	
Endmembers	SAD	RMSE	SAD	RMSE	SAD	RMSE
Tree	0.0765	0.0579	0.0905	0.0667	0.2565	0.3268
Water	0.1186	0.0306	0.3141	0.0521	0.2474	0.3151
Soil	0.1027	0.0993	0.0886	0.0893	0.3584	0.2936
Road	0.0584	0.0791	0.0473	0.0693	0.5489	0.2829
Average	0.0891	0.0667	0.1351	0.0694	0.3528	0.3046

	NMF		ℓ_1-NMF		$\ell_{1/2}$-NMF	
Endmembers	SAD	RMSE	SAD	RMSE	SAD	RMSE
Tree	0.2130	0.1402	0.0680	0.0636	0.0409	0.0707
Water	0.2001	0.1106	0.3815	0.0660	0.1682	0.1031
Soil	0.1569	0.2557	0.0898	0.2463	0.0506	0.2679
Road	0.3522	0.2450	0.4118	0.2344	0.3670	0.2737
Average	0.2305	0.1879	0.2378	0.1526	0.1567	0.1789

Table A.3: SAD and RMSE for OMDC-ADMM, BMDC-ADMM, VCA, NMF, ℓ_1-NMF and $\ell_{1/2}$-NMF (for the benchmarking image *Jasper Ridge*)

The causes of this variability can be diverse, *e.g.*, the changing illumination conditions during the acquisition, the intrinsic variability of the components or the atmospheric effects. This aspect is illustrated in Figure A.18 which plots the spectra for all estimated endmembers slice by slice for 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 A.3, the SAD for OMDC-ADMM was calculated using the average over the K time samples of the estimated spectra.

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 endmembers and the abundance maps estimated by OMDC-ADMM and BMDC-ADMM, along with the ground truth, are represented in Figure A.19 and A.20. For OMDC-ADMM, the average values of the endmembers are represented. It can be seen in Figure A.20, that some details are missing on the abundance maps estimated

by BMDC-ADMM; this is not the case for the maps generated by OMDC-ADMM.

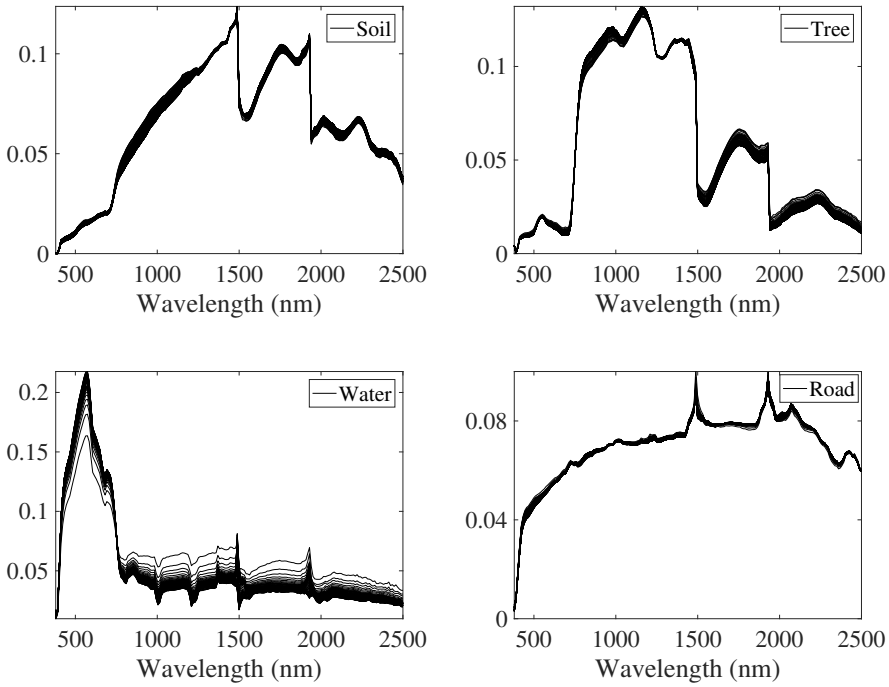
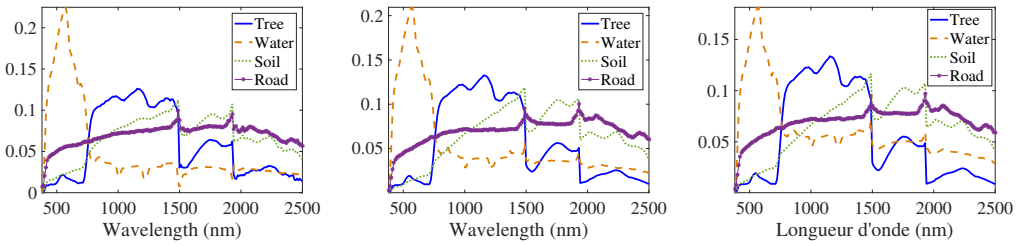


Figure A.18: Estimated endmembers slice by slice for OMDC-ADMM (for the benchmarking image *Jasper Ridge*)



(a) Ground truth (b) OMDC-ADMM (c) BMDC-ADMM

Figure A.19: True and estimated endmembers for OMDC-ADMM and BMDC-ADMM (for the benchmarking image *Jasper Ridge*)

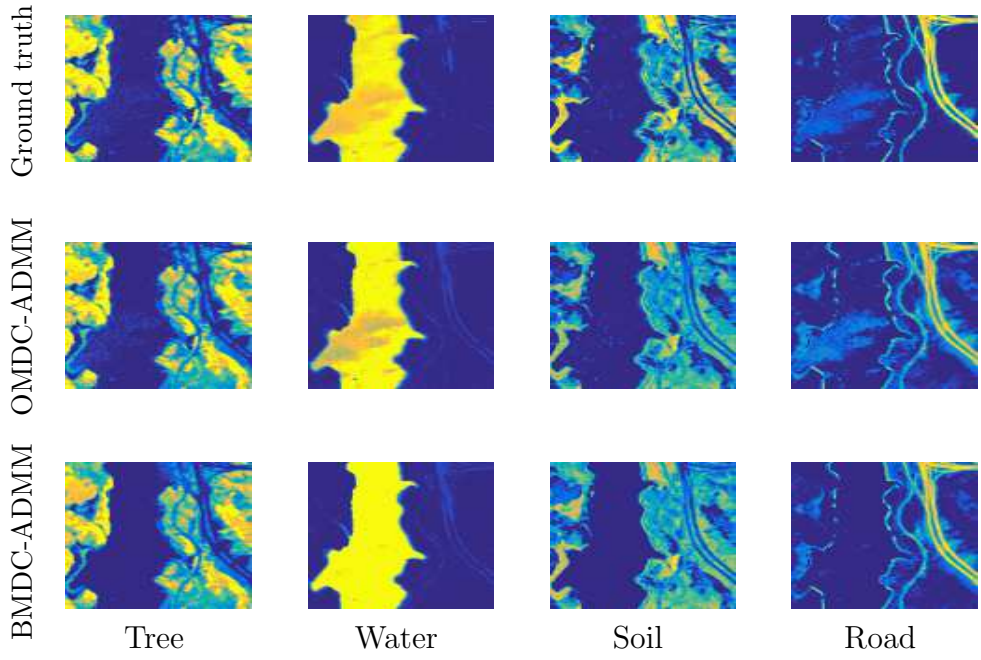


Figure A.20: True and estimated abundance maps for OMDC-ADMM and BMDC-ADMM (for the benchmarking image *Jasper Ridge*)

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).

[1] F. D. Van der Meer, H. M. Van der Werff, F. J. Van Ruitenbeek, C. A. Hecker, W. H. Bakker, M. F. Noomen, M. Van Der Meijde, E. J. M. Carranza, J. B. De Smeth, T. Woldai, Multi-and hyperspectral geologic remote sensing: a review, International Journal of Applied Earth Observation and Geoinformation 14 (2012) 112–128.

[2] J. Qin, M. Kim, K. Chao, D. Chan, S. Delwiche, B.-K. Cho, Line-scan hyperspectral imaging techniques for food safety and quality applications, Applied Sciences 7 (2017) 125.

[3] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, A. Abbas, Modern trends in hyperspectral image analysis: a review, IEEE Access 6 (2018) 14118–14129.

[4] G. Lu, B. Fei, Medical hyperspectral imaging: a review, Journal of Biomedical Optics 19 (2014) 010901.

[5] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, J. Chanussot, Hyperspectral unmixing overview: geometrical, statistical, and sparse regression-based approaches, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 5 (2012) 354–379.

[6] N. Keshava, J. F. Mustard, Spectral unmixing, IEEE Signal Processing Magazine 19 (2002) 44–57.

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

[8] J. M. Nascimento, J. M. Dias, Vertex Component Analysis: a fast algorithm to unmix hyperspectral data, IEEE Transactions on Geoscience and Remote Sensing 43 (2005) 898–910.

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

[10] M. Berman, H. Kiiveri, R. Lagerstrom, A. Ernst, R. Dunne, J. F. Huntington, ICE: a statistical approach to identifying endmembers in hyperspectral images, IEEE Transactions on Geoscience and Remote Sensing 42 (2004) 2085–2095.

[11] J. Li, J. M. Bioucas-Dias, Minimum volume simplex analysis: a fast algorithm to unmix hyperspectral data, in: IEEE International Geoscience and Remote Sensing Symposium (IGARSS), volume 3, 2008, pp. 250–253.

[12] D. D. Lee, H. S. Seung, Learning the parts of objects by Non-negative Matrix Factorization, Nature 401 (1999) 788–791.

[13] L. Miao, H. Qi, Endmember extraction from highly mixed data using minimum volume constrained Non-negative Matrix Factorization, IEEE Transactions on Geoscience and Remote Sensing 45 (2007) 765–777.

[14] Y. Yu, W. 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, 2007, pp. 151–159.

[15] R. Schachtner, G. Pöppel, A. M. Tomé, E. W. Lang, Minimum determinant constraint for Non-negative Matrix Factorization, in: International Conference on Independent Component Analysis and Signal Separation (ICA), Springer, 2009, pp. 106–113.

[16] G. Zhou, S. Xie, Z. Yang, J.-M. Yang, Z. He, Minimum-volume-constrained Non-negative Matrix Factorization: enhanced ability of learning parts, IEEE Transactions on Neural Networks 22 (2011) 1626–1637.

[17] X. Fu, K. Huang, B. Yang, W.-K. Ma, N. D. Sidiropoulos, Robust volume minimization-based matrix factorization for remote sensing and document clustering, IEEE Transactions on Signal Processing 64 (2016) 6254–6268.

[18] M. A. Ang, N. Gillis, Volume regularized Non-negative Matrix Factorizations, in: IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2018, pp. 1–5.

[19] X. Fu, K. Huang, N. D. Sidiropoulos, W.-K. Ma, Non-negative matrix factorization for signal and data analytics: identifiability, algorithms, and applications., IEEE Signal Processing Magazine 36 (2019) 59–80.

[20] V. Leplat, A. M. Ang, N. Gillis, Minimum-volume rank-deficient Non-negative Matrix Factorizations, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 3402–3406.

[21] J. Mairal, F. Bach, J. Ponce, G. Sapiro, On-line learning for matrix factorization and sparse coding, Journal of Machine Learning Research 11 (2010) 19–60.

[22] F. Wang, P. Li, A. C. König, Efficient document clustering via Online Non-negative Matrix Factorizations, in: Society for Industrial and Applied Mathematics (SIAM), volume 11, 2011, pp. 908–919.

[23] N. Guan, D. Tao, Z. Luo, B. Yuan, On-line Non-negative Matrix Factorization with robust stochastic approximation, IEEE Transactions on Neural Networks and Learning Systems 23 (2012) 1087–1099.

[24] R. Zhao, V. Y. Tan, On-line Non-negative Matrix Factorization with outliers, IEEE Transactions on Signal Processing 65 (2016) 555–570.

[25] S. S. Bucak, B. Gunsel, Incremental subspace learning via Non-negative Matrix Factorization, Pattern recognition 42 (2009) 788–797.

[26] A. Lefèvre, F. Bach, C. 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 (WASPAA), 2011, pp. 313–316.

[27] G. Zhou, Z. Yang, S. Xie, J.-M. Yang, On-line blind source separation using Incremental Non-negative Matrix Factorization with volume constraint, IEEE Transactions on Neural Networks 22 (2011) 550–560.

[28] D. Wang, H. Lu, On-line learning parts-based representation via incremental orthogonal projective Non-negative Matrix Factorization, Signal Processing 93 (2013) 1608–1623.

[29] Y. Wu, B. Shen, H. Ling, Visual tracking via On-line Non-negative Matrix Factorization, IEEE Transactions on Circuits and Systems for Video Technology 24 (2014) 374–383.

[30] L. Nus, S. Miron, D. Brie, On-line blind unmixing for hyperspectral pushbroom imaging systems, in: IEEE Statistical Signal Processing Workshop (SSP), 2018, pp. 418–422.

[31] C.-J. Lin, Projected gradient methods for Non-negative Matrix Factorization, Neural Computation 19 (2007) 2756–2779.

[32] Y. Xu, W. Yin, Z. Wen, Y. Zhang, An alternating direction algorithm for matrix completion with non-negative factors, Frontiers of Mathematics in China 7 (2012) 365–384.

[33] D. L. Sun, C. 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 (ICASSP), 2014, pp. 6201–6205.

[34] K. Huang, N. D. Sidiropoulos, A. P. Liavas, A flexible and efficient algorithmic framework for constrained matrix and tensor factorization, IEEE Transactions on Signal Processing 64 (2016) 5052–5065.

[35] G. Strang, Linear algebra and its applications, Thomson, Brooks/Cole, 2006.

[36] T.-H. Chan, C.-Y. Chi, Y.-M. Huang, W.-K. Ma, A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing, IEEE Transactions on Signal Processing 57 (2009) 4418–4432.

[37] B. Peng, The determinant: a means to calculate volume, Recall 21 (2007) a22.

[38] D. Donoho, V. Stodden, When does Non-negative Matrix Factorization give a correct decomposition into parts?, in: Advances in Neural Information Processing Systems, 2004, pp. 1141–1148.

[39] H. Laurberg, M. G. Christensen, M. D. Plumbley, L. K. Hansen, S. H. Jensen, Theorems on positive data: on the uniqueness of NMF, Computational Intelligence and Neuroscience 2008 (2008).

[40] K. Huang, N. D. Sidiropoulos, A. Swami, Non-negative Matrix Factorization revisited: uniqueness and algorithm for symmetric decomposition, IEEE Transactions on Signal Processing 62 (2014) 211–224.

[41] J. 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 (WHISPERS), 2009, pp. 1–4.

[42] M. Fazel, H. Hindi, S. 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 (ACC), volume 3, 2003, pp. 2156–2162.

[43] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends® in Machine learning 3 (2011) 1–122.

[44] S. Boyd, L. Vandenberghe, Convex optimization, Cambridge University Press, 2004.

[45] A. Zare, D. K. Ho, Endmember variability in hyperspectral analysis: addressing spectral variability during spectral unmixing, IEEE Signal Processing Magazine 31 (2013) 95–104.

[46] A. Halimi, N. Dobigeon, J.-Y. Tourneret, Unsupervised unmixing of hyperspectral images accounting for endmember variability, IEEE Transactions on Image Processing 24 (2015) 4904–4917.

[47] F. Zhu, Hyperspectral unmixing: ground truth labeling, datasets, benchmark performances and survey, arXiv preprint arXiv:1708.05125 (2017).

[48] P. O. Hoyer, Non-negative sparse coding, in: IEEE Workshop on Neural Networks for Signal Processing, 2002, pp. 557–565.

[49] Y. Qian, S. Jia, J. Zhou, A. Robles-Kelly, Hyperspectral unmixing via $\ell_{1/2}$ sparsity-constrained Non-negative Matrix Factorization, IEEE Transactions on Geoscience and Remote Sensing 49 (2011) 4282–4297.