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

Ludivine Nus, Sebastian Miron, David Brie

An ADMM-based algorithm with minimum dispersion regularization 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 regularization to stabilize the solution; this regularization can be interpreted as a convex relaxation of the minimum volume regularization 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 regularization

[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 endmembers and $\mathbf{A}$ is a matrix containing on its rows the abundances for the recorded samples. We assume that

$$\mathbf{A} \geq \mathbf{0} \text{ and } \mathbf{A}^\top \mathbf{1} = \mathbf{1}.$$

The constraint $\mathbf{A} \geq \mathbf{0}$ denotes the fact that $\mathbf{A}$ is component-wise non-negative, where $\mathbf{0}$ is the matrix of zeros. The constraint $\mathbf{A}^\top \mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ is an all-ones vector, means $\mathbf{A}$ is a column-stochastic matrix, that is, the entries in each column of $\mathbf{A}$ sums to one. Under theses constraints, the LMM admits a geometrical interpretation. Figure 1 illustrates this geometrical representation for a mixture of $R = 3$ endmembers: all the data points (red points) belong to the convex hull defined by the endmembers (gray points) $\mathbf{s}_1, ..., \mathbf{s}_R$ and the origin (denoted conv $\{\mathbf{S}\}$) or, equivalently, to the simplex defined by $\mathbf{s}_1, ..., \mathbf{s}_R$. When the sum-to-one constraint does not hold, the observations belong to the positive convex cone spanned by the endmembers.

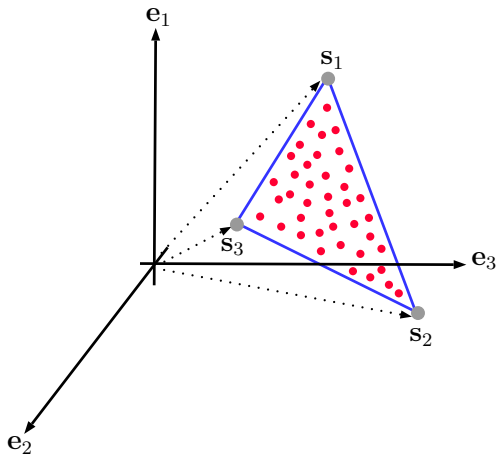


Figure 1: Geometric representation of the LMM for $R = 3$ endmembers

There are many methods in the literature for hyperspectral unmixing; their goal is to recover the factors $\mathbf{S}$ and/or $\mathbf{A}$ from the data matrix $\mathbf{X}$.

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 regularized approaches are *volume minimization (VolMin)*-based NMF such as [13, 14, 15, 16, 17, 18, 19, 20, 21], which can be considered as state-of-the-art methods in blind hyperspectral unmixing. The reason to consider VolMin criterion is motivated by geometric insights: in Figure 1 (in the noiseless case), the columns of $\mathbf{S}$ (the endmembers) are the vertices of a convex hull that contains the data points. In the absence of pure pixels, minimizing the volume of $\text{conv}\,\{\mathbf{S}\}$ allows to recover these endmembers under mild conditions (see Section 3 for more details).

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.

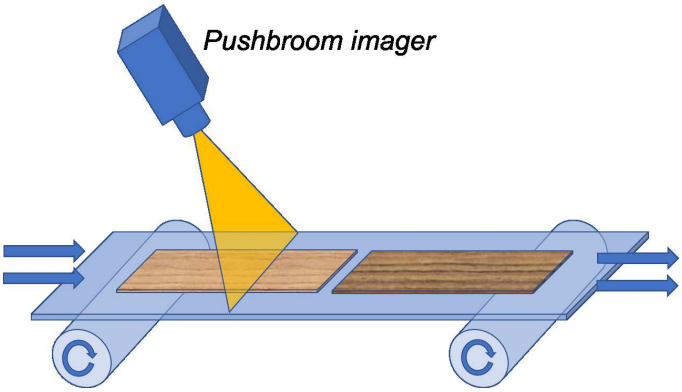


Figure 2: Acquisition of hyperspectral images of wood

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).

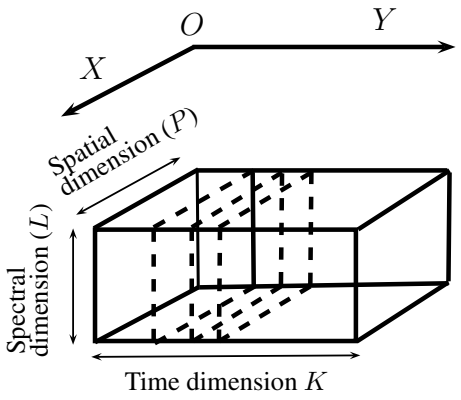


Figure 3: Data structure for pushbroom acquisition

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 [22, 23, 24, 25], the endmembers do not vary from one sample to another, while in [26, 27, 28, 29, 30, 31], the endmembers may evolve between successive samples. In particular, the Incremental NMF (INMF) [26] 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 [28, 29, 30]. Recently, the on-line Minimum Volume Regularization-NMF algorithm (OMVR-NMF) was introduced in [31]. It is a straightforward adaptation of [28] specially designed for pushbroom hyperspectral imaging system. To the best knowledge of the present authors, OMVR-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. However, OMVR-NMF is based on multiplicative update rules which are known to be highly sensitive to initialization and often suffer from slow convergence rate [32]. 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. Therefore, this paper aims to design an efficient algorithm for on-line volume regularized NMF, specially devised for pushbroom acquisition systems, and to compare it with on-line and off-line state-of-the-art methods for blind hyperspectral unmixing.

1.4. Main contributions

The contribution of this manuscript (compared to our previous work [31] and to the state-of-the-art on-line approaches) is twofold: first of all, we propose to replace the minimum volume regularization used in OMVR-NMF by the Minimum Dispersion Regularization (MDR) [10, 16]. MDR can be seen as a surrogate for the minimum volume regularization; it aims at minimizing the sum of the distances between each endmember and the endmembers'

centroid, which indirectly minimizes the volume of conv $\{\mathbf{S}\}$ [33]. Although this regularization could be considered as a rough one, it has some interesting properties, in particular for the considered on-line context:

- Using MDR results in lower computational complexity compared to MVR: only matrix multiplications are required for MDR while MVR needs matrix inversions (required in OMVR-NMF algorithm). This is of upmost importance in the context of fast on-line unmixing.
- Using MDR makes the $\mathbf{S}$ estimation subproblem convex and thus, offers interesting optimization properties by enabling explicit updates of the parameters.
- Unsing MDR results in faster convergence of the algorithm compared to the non-convex regularization approach.
- Using MDR enables preservation of the decomposition rank for a wide range of the regularization hyperparameter, which is essential for the correct performance of the proposed approach.

The second contribution is algorithm design, in which we substitute the on-line multiplicative update rules by using optimization method based on Alternating Direction Method of Multipliers (ADMM) in [25]; ADMM proved its superiority over multiplicative updates with respect to both reconstruction accuracy and convergence rate [34, 35, 36].

Through extensive experimental results reported in AppendixA, we show, first of all, that the use of MDR for on-line volume regularized NMF, allows to significantly increase the convergence speed of the algorithm compared, *e.g.*, to OMVR-NMF, for similar performance in terms of parameter estimates. Moreover, we illustrate the ability of the proposed on-line approach to produce meaningful unmixing results on a real dataset; these results are obtained for a much smaller computational cost compared to the *batch* (or off-line) volume regularized algorithms. To assess its performance, we compare our implementation to two state-of-the-art approaches: RVolMin (Robust Volume Minimization) [18] and VRNMF (Volume-Regularized NMF) [20] algorithms.

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 regularization. Section 4 presents the derivation of the proposed on-line ADMM approach; convergence and computational complexity are also discussed. Finally, in Section 5, the proposed algorithm is applied to on-line analysis of wood surface. Conclusions are drawn 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 $\mathbf{A}^{(k+1)} = \left[\mathbf{A}^{(k)}\ \tilde{\mathbf{A}}^{(k+1)}\right]$.

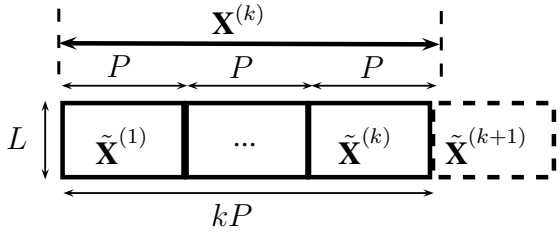


Figure 4: Unfolded pushbroom hyperspectral image

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:

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

In others words, α controls the trade-off between the contribution of the old and the new samples. A version of the cost function (4) is used by the

INMF algorithm proposed in [26]. Nevertheless, without additional regularization, 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 endmember matrix $\tilde{\mathbf{S}}^{(k+1)}$, a geometric regularization which forces the simplex bounded by the endmembers to circumscribe the data as closely as possible. For now, we denote this penalty 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:

$$\begin{aligned} &\underset{\tilde{\mathbf{S}}^{(k+1)}, \tilde{\mathbf{A}}^{(k+1)}}{\text{minimize}}\ \mathcal{J}_{\text{Vol}}^{(k+1)}\left(\tilde{\mathbf{S}}^{(k+1)}, \tilde{\mathbf{A}}^{(k+1)}\right), \\ &\text{subject to } \tilde{\mathbf{S}}^{(k+1)} \geq \mathbf{0}, \tilde{\mathbf{A}}^{(k+1)} \geq \mathbf{0}, \tilde{\mathbf{A}}^{(k+1)^\top}\mathbf{1} = \mathbf{1}, \end{aligned} \quad (6)$$

for a particular choice of the volume regularizer term.

3. Volume regularization

The classical measure for the minimum volume regularization is the determinant, *i.e.* $\text{Vol}\left(\tilde{\mathbf{S}}^{(k+1)}\right) = \det\left(\tilde{\mathbf{S}}^{(k+1)}\right)$ [37]. 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, 17]:

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

or its variants presented in [13, 38]. The reason for considering expression (7) is that $\sqrt{\det\left(\tilde{\mathbf{S}}^{(k+1)^\top}\tilde{\mathbf{S}}^{(k+1)}\right)}/R!$ represents the volume of the convex hull

of the columns of $\tilde{\mathbf{S}}^{(k+1)}$ and the origin. Hence, (7) is, up to a constant factor, the square of that volume.

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 [39, 40, 41] 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 identifiable. This is the case where volume minimization approaches can be successfully used. By applying the recent results of [42, 43, 33] 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 (it means that the columns of $\tilde{\mathbf{A}}$ are spread enough in the non-negative orthant, *i.e.* at least every facet of the non-negative orthant is touched by some columns of $\tilde{\mathbf{A}}$ and some points are close enough to the pure pixel condition; see [33] for the exact definition of sufficiently scattered), then the model (2) is identifiable under the minimum volume regularization.

It is important to note that this condition stipulates that there is no rank variation 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 regularization 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 [44] is:

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

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.* $\det\left(\tilde{\mathbf{S}}^\top\tilde{\mathbf{S}}\right) \to 0$, a modified version was proposed in [18, 21]:

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

where $\epsilon > 0$ is a specified small value and $\mathbf{I}$ is the identity matrix. Let $\{\lambda_i^2, i = 1, ..., R\}$ be the ordered eigenvalues of $\tilde{\mathbf{S}}^\top\tilde{\mathbf{S}}$, that is the λ_i's are the singular values of $\tilde{\mathbf{S}}$. While, it is easy to check that (8) is a concave function of λ_i's, it appears that (9) has a stationary point in $\mathbf{0}$ and is convex in a neighborhood of $\mathbf{0}$, as shown hereafter:

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

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:

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

In other words, in the neighborhood of $\lambda_i = 0$, $\log\left(\det\left(\tilde{\mathbf{S}}^\top\tilde{\mathbf{S}} + \epsilon\mathbf{I}\right)\right)$ behaves as $\text{trace}\left(\tilde{\mathbf{S}}^\top\tilde{\mathbf{S}}\right)$, the extent of the neighborhood being controlled by the value of ϵ. This has an important consequence: while $\log\left(\det\left(\tilde{\mathbf{S}}^\top\tilde{\mathbf{S}}\right)\right)$ will favor rank deficiency as the regularization parameter μ increases [45], $\log\left(\det\left(\tilde{\mathbf{S}}^\top\tilde{\mathbf{S}} + \epsilon\mathbf{I}\right)\right)$ will tend to preserve the full column rank of $\tilde{\mathbf{S}}$ for a

wider range of the regularization parameter.

Finally, a convex surrogate for $\text{Vol}\left(\tilde{\mathbf{S}}\right)$ was proposed in [10, 16] as:

$$\sum_{r=1}^{R} \|\tilde{\mathbf{s}}_r - \bar{\mathbf{s}}\|_2^2\,,\ \bar{\mathbf{s}} = \frac{1}{R}\sum_{r=1}^{R} \tilde{\mathbf{s}}_r,$$

or equivalently in matrix form:

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

where $\mathbf{D} = \mathbf{I} - \frac{1}{R}\mathbf{1}\mathbf{1}^\top$ ($\mathbf{1}$ is an all-ones column vector of size $R \times 1$). As illustrated in Figure 5, geometrically speaking, expression (11) computes the sum of the distances between each endmember and the endmembers' centroid. It can be thus interpreted as a measure of the dispersion of the endmembers around their centroid, and related indirectly to the volume of $\text{conv}\left\{\tilde{\mathbf{S}}\right\}$.

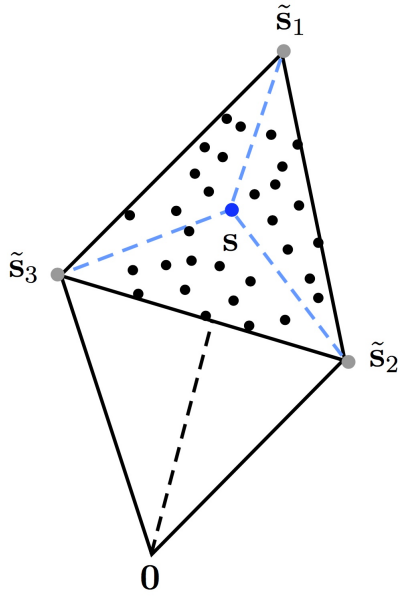


Figure 5: Geometric intuition of the minimum dispersion regularization for $R = 3$ endmembers

In the following, we will denote the regularizer $\log\left(\det\left(\tilde{\mathbf{S}}^\top\tilde{\mathbf{S}}+\epsilon\mathbf{I}\right)\right)$ by MVR (for Minimum Volume Regularization), and $\operatorname{trace}\left(\tilde{\mathbf{S}}\mathbf{D}\tilde{\mathbf{S}}^\top\right)$ by MDR (for Minimum Dispersion Regularization).

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 is 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 6 the two functions: MVR with varying ϵ and MDR, for increasing values of β; the curves are normalized to set their minimum value to 0 and their maximum value to 1. To promote rank-deficient solutions, ϵ should not be chosen too large, say $\epsilon<0.1$. Meanwhile, ϵ should no be chosen too small either (*e.g.* 10^{-9}) to avoid bad conditioning of $\tilde{\mathbf{S}}^\top\tilde{\mathbf{S}}+\epsilon\mathbf{I}$. For large values of ϵ (*e.g.* 0.1), rank deficiency is no longer promoted and both MDR and MVR tend to preserve the full column rank of $\tilde{\mathbf{S}}$ for a wider range of μ. This rank preserving property is essential to ensure that the sufficient identifiability condition of Section 3.1 is satisfied.

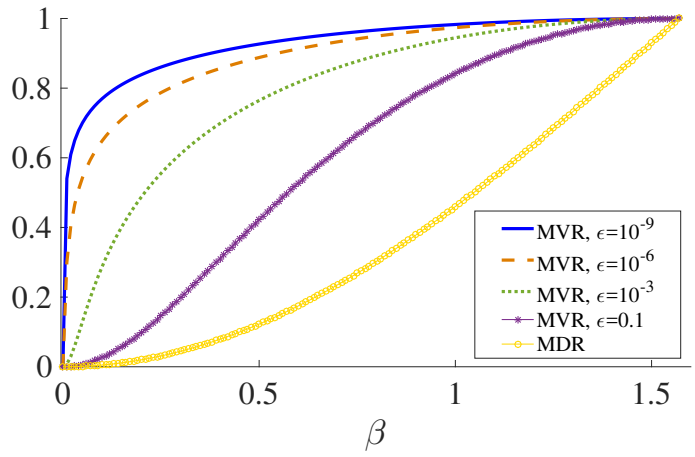


Figure 6: Illustrations of MVR and MDR penalties as functions of the angle β

4. OMDR-ADMM algorithm

In the following, we only consider MDR (rather than MVR) because of its convexity which allows an efficient implementation of on-line ADMM. In addition, according to the numerical experiments from the previous section, MDR tends to preserve full column rank solutions, which allows us to assume that the sufficient identifiability condition for our on-line mixture model is satisfied. The resulting algorithm will be called *On-line MDR-ADMM* (OMDR-ADMM).

4.1. Algorithm derivation

Let us define:

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

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

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

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 [46] 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:

$$\tilde{\mathbf{A}}^{(k+1)} \leftarrow \underset{\tilde{\mathbf{A}}^{(k+1)^\top}\mathbf{1}=\mathbf{1}}{\arg\min}\ (1-\alpha)\tilde{\mathcal{J}}^{(k+1)} + \frac{\rho}{2}\left\|\tilde{\mathbf{A}}^{(k+1)} - \tilde{\mathbf{V}}^{(k+1)} + \tilde{\mathbf{\Pi}}^{(k+1)}\right\|_F^2, \quad (15)$$

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

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

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

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

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

where $\mathbf{N}^{(k+1)} = \alpha\sum_{l=1}^{k}\tilde{\mathbf{X}}^{(l)}\tilde{\mathbf{A}}^{(l)^\top} + (1-\alpha)\tilde{\mathbf{X}}^{(k+1)}\tilde{\mathbf{A}}^{(k+1)^\top}$ and $\mathbf{M}^{(k+1)} = \alpha\sum_{l=1}^{k}\tilde{\mathbf{A}}^{(l)}\tilde{\mathbf{A}}^{(l)^\top} + (1-\alpha)\tilde{\mathbf{A}}^{(k+1)}\tilde{\mathbf{A}}^{(k+1)^\top}$.

Following [26], under the assumption $\tilde{\mathbf{S}}^{(k+1)} \approx \tilde{\mathbf{S}}^{(k)}$, we can write $\mathbf{N}^{(k)} \approx \sum_{l=1}^{k}\tilde{\mathbf{X}}^{(l)}\tilde{\mathbf{A}}^{(l)^\top}$ and $\mathbf{M}^{(k)} \approx \sum_{l=1}^{k}\tilde{\mathbf{A}}^{(l)}\tilde{\mathbf{A}}^{(l)^\top}$. Thus, $\mathbf{N}^{(k+1)}$ and $\mathbf{M}^{(k+1)}$ can be expressed as:

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

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

The update of $\tilde{\mathbf{A}}^{(k+1)}$ in (15) has a closed-form solution as presented in [19], which is computationally cheap[2].

[2]A Matlab implementation of OMDR-ADMM is available at https://github.com/LudivineNus/OMDR-ADMM

Algorithm 1 summarizes the proposed OMDR-ADMM method. For notation simplification, the indices $k+1$ in the updates are omitted.

Algorithm 1 OMDR-ADMM

Inputs: $\mathbf{X}$; $\mathbf{D}$; R; α; μ; ρ; N_1; N_2;
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**
  $\tilde{\mathbf{X}} = \tilde{\mathbf{X}}^{(k+1)}$;
  $t_1 = 1$;
  while $t_1 < N_1$ **do**
    $t_2 = 1$;
    while $t_2 < N_2$ **do**
      $\tilde{\mathbf{A}} \leftarrow \underset{\tilde{\mathbf{A}}^\top \mathbf{1} = \mathbf{1}}{\arg\min} \frac{1}{2}(1-\alpha) \left\| \tilde{\mathbf{X}} - \tilde{\mathbf{S}}\tilde{\mathbf{A}} \right\|_F^2 + \frac{\rho}{2} \left\| \tilde{\mathbf{A}} - \tilde{\mathbf{V}} + \tilde{\mathbf{\Pi}} \right\|_F^2$;
      $\tilde{\mathbf{V}} = \max\left(\mathbf{0}, \tilde{\mathbf{A}} + \tilde{\mathbf{\Pi}}\right)$;
      $\tilde{\mathbf{\Pi}} \leftarrow \tilde{\mathbf{\Pi}} + \tilde{\mathbf{A}} - \tilde{\mathbf{V}}$;
      $t_2 \leftarrow t_2 + 1$;
    end while
    $\tilde{\mathbf{N}} = \alpha\mathbf{N} + (1-\alpha)\left(\tilde{\mathbf{X}}\tilde{\mathbf{A}}^\top\right)$;
    $\tilde{\mathbf{M}} = \alpha\mathbf{M} + (1-\alpha)\left(\tilde{\mathbf{A}}\tilde{\mathbf{A}}^\top\right)$;
    $\tilde{\mathbf{S}} = \left(\tilde{\mathbf{N}} + \rho\left(\tilde{\mathbf{U}} - \tilde{\mathbf{\Lambda}}\right)\right)\left(\tilde{\mathbf{M}} \;\; + 2\mu\mathbf{D} + \rho\mathbf{I}\right)^{-1}$;
    $\tilde{\mathbf{U}} = \max\left(\mathbf{0}, \tilde{\mathbf{S}} + \tilde{\mathbf{\Lambda}}\right)$;
    $\tilde{\mathbf{\Lambda}} \leftarrow \tilde{\mathbf{\Lambda}} + \tilde{\mathbf{S}} - \tilde{\mathbf{U}}$;
    $t_1 \leftarrow t_1 + 1$;
  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

It includes three loops. The inner loop alternately estimates $\tilde{\mathbf{A}}$, $\tilde{\mathbf{V}}$ and

$\tilde{\mathbf{\Pi}}$ using a fixed number of iterations N_2. These estimates are iteratively refined in an outer loop with N_1 iterations, in which the parameters $\tilde{\mathbf{S}}$, $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{\Lambda}}$ are cyclically updated. This procedure is repeated for each new sample $\tilde{\mathbf{X}}^{(k+1)}$.The inner loop of N_2 iterations ensures that the objective function decreases between each iteration of the outer loop. In addition, and this was unexpected, including this additional loop allows to accelerate the convergence speed of the algorithm and consequently to reduce N_1.

It can be shown, using a similar approach to [47, 36], that any stationary point generated by a sequence of iterations of OMDR-ADMM satisfies the Karush-Kuhn-Tucker conditions. This statement is valid only in the case of our working assumption *i.e.* $\tilde{\mathbf{S}}^{(k+1)} \approx \tilde{\mathbf{S}}^{(k)}$, in other words, when the steady state is reached. No theoretical claims can be made regarding the transient behavior. This point is addressed through numerical simulations (see Section 4.3 and AppendixA).

4.2. 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_1 and N_2 iterations and a decomposition rank R, the computational complexity for OMDR-ADMM algorithm is of the order of $KN_1(2RPL + 2LR^2 + 2R^3 + PR^2(1 + N_2))$. For comparison, we developed a *batch* version of OMDR-ADMM, that we called BMDR-ADMM (for Batch MDR-ADMM), which processes the entire hyperspectral data cube at once; this algorithm has complexity of $N_1(2RPKL + 2LR^2 + 2R^3 + PKR^2(1 + N_2))$. Moreover, the OMVR-NMF algorithm presented in [31], based on multiplicative update rules with the minimum volume regularization in (9), has a computational complexity of the order of $KN_1(2RPL + R(L + PN_2) + R^2(6L + 2P))$. For comparison, we have also implemented the on-line algorithm based on multiplicative update rules with MDR, which results in the OMDR-NMF algorithm; the latter has a complexity of $KN_1(2RPL + R(L + PN_2) + R^2(3L + 2P))$. For an identical number of iterations, it is easy to notice that OMDR-NMF has a lower complexity than OMVR-NMF. Note that the complexity of OMDR-ADMM is comparable to that of BMDR-ADMM, OMDR-NMF and OMVR-NMF. However, as we show in the sequel, OMDR-ADMM requires fewer iterations to con-

verge compared to the three other methods, and thus significantly reduces the computational cost.

4.3. Numerical results

In order to evaluate the performance of OMDR-ADMM for hyperspectral unmixing, we conducted several experiments on simulated and real hyperspectral images. These experiments have the following objectives:

1. Illustrating the rank preserving properties of MDR and MVR.
2. Showing the efficiency of OMDR-ADMM by examining its convergence speed and comparing it to that of the OMVR-NMF algorithm [31]. The sensitivity of the convergence speed of the proposed method to parameters α and ρ is also studied.
3. Comparing OMDR-ADMM to its batch counterpart BMDR-ADMM to assess the advantages of on-line processing of hyperspectral images.
4. Studying the spectral tracking capability of OMDR-ADMM.
5. Validating the unmixing performance of our algorithm on a real hyperspectral image called *Jasper Ridge* (`http://lesun.weebly.com/hyperspectral-data-set.html`) with the available endmember and abundance references provided in [48]. This image is commonly used in hyperspectral unmixing community for benchmarking purposes. A comparison to state-of-the-art volume-regularization algorithms [18, 20] is also provided.

   All the results are reported in AppendixA.

5. Application to on-line analysis of wood surface

In this section, the performance of OMDR-ADMM is tested on a real hyperspectral image 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), for example. The sapwood is generally undesirable in many applications due to its porosity, particularly in the parquet industry. Thus, the detection of this defect 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 sample shown in Figure B.25, measuring 120 cm $\times$ 15 cm.

Figure 7: Selected wood sample (visible range color image)

This sample had a sapwood-type defect in the bottom left corner. This defect was hardly distinguishable by the colorimetric systems and was manually labeled by experts. The piece of wood was scanned by a NIR pushbroom imager working in the spectral range of 900-1600 nm, resulting in hyperspectral image of dimension $224 \times 270 \times 1514$, where 224 represents the number of wavelengths and the other two numbers, the (spatial $\times$ time) dimensions. The acquired image was processed sequentially by OMDR-ADMM, slice by slice. The number of endmembers to be extracted was set to $R = 3$, after several preliminary tests. The parameters of the algorithm were set as follows: $\tilde{\mu} = 10^{-5}$, $\alpha = 0.99$, $\rho = 0.01$, $N_1 = 15$ and $N_2 = 10$. The endmember and abundance matrices were randomly initialized from a continuous uniform distribution on the interval $[0, 1]$.

The three abundance maps generated by OMDR-ADMM are presented in Figure 8. On the first abundance map, we observe that the area corresponding to the sapwood is much more contrasted than the rest of the image. Moreover, it is interesting to highlight that all of the knots present on its surface are grouped on the third abundance map. These are very promising results, as, for the selected wood sample, the color of the sapwood 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 dataset, the processing time by OMDR-ADMM was of about 3.22 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.

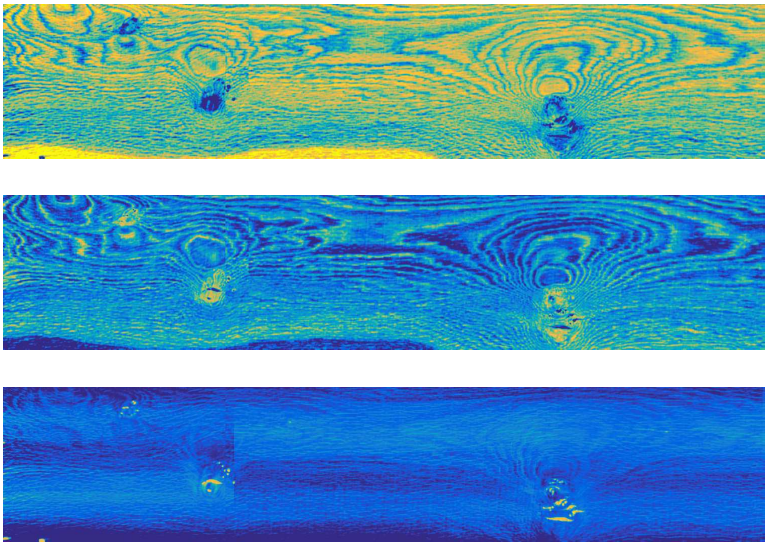


Figure 8: Abundance maps estimated by OMDR-ADMM for the considered wood sample

In AppendixB, we provide, for comparison, the abundance maps and the endmembers generated by the state-of-the-art volume regularization algorithm RVolMin [18], for the wood sample of Figure B.25. For further comparisons we applied the two algorithms (OMDR-ADMM and RVolMin) on a second wood sample presenting another type of defect. The processing results are also given in this appendix.

6. Conclusions

We proposed a new algorithm, OMDR-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.

In addition, we have shown that the algorithms using the minimum dispersion regularization yield results similar to the minimum volume one, but for a much smaller number of iterations, which demonstrates the interest of this type of penalty for fast on-line hyperspectral unmixing. These experiments also revealed that OMDR-ADMM allows to track the spectral variability of the endmembers over time and significantly reduces the processing burden compared to its off-line counterpart, which is a crucial feature for real-time data processing. Finally, OMDR-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-Supplementary material: Numerical results.

AppendixB-Supplementary material: Comparison of OMDR-ADMM to RVolMin on wood hyperspectral images.

AppendixA. Numerical results

In order to evaluate the performance of OMDR-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 MDR and MVR.
2. Showing the efficiency of OMDR-ADMM by examining its convergence speed and comparing it to that of the OMVR-NMF algorithm [31]. The sensitivity of the convergence speed of the proposed method to parameters α and ρ is also studied.
3. Comparing OMDR-ADMM to its batch counterpart BMDR-ADMM to assess the advantages of on-line processing of hyperspectral images.
4. Studying the spectral tracking capability of OMDR-ADMM.
5. Validating the unmixing performance of our algorithm on a real hyperspectral image called *Jasper Ridge* (`http://lesun.weebly.com/hyperspectral-data-set.html`) with the available endmember and abundance references provided in [48]. This image is commonly used in hyperspectral unmixing community for benchmarking purposes. A comparison to state-of-the-art volume-regularization algorithms [18, 20] is also provided.

AppendixA.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:

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

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:

$$\frac{1}{R}\sum_{r=1}^{R}\cos^{-1}\left(\frac{\tilde{\mathbf{s}}_r^{\top}\hat{\tilde{\mathbf{s}}}_r}{||\tilde{\mathbf{s}}_r||_2\ ||\hat{\tilde{\mathbf{s}}}_r||_2}\right), \tag{A.2}$$

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:

$$\frac{1}{R}\sum_{r=1}^{R}\sqrt{\frac{1}{KP}\sum_{d=1}^{KP}(a_{rd}-\hat{a}_{rd})^2}, \tag{A.3}$$

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 (A.2) and (A.3) are average values. For some experiments, the SAD and the RMSE values are calculated independently for each source $r = 1, ..., R$.

AppendixA.2. Rank preserving properties of MVR and MDR

We performed a numerical experiment to illustrate the rank preserving properties of both volume regularizers, MVR (9) and MDR (11). We used, for the simulations, the endmembers shown in Figure A.11(d). None of the three endmembers has any zero value, which results in a non-unique NMF problem. In other words, without additional regularizations the model is non-identifiable. We generated the three abundance maps of size 35×35, following the uniform Dirichlet distribution. By doing so, we ensure that the sufficiently scattered and full column rank conditions are verified with high probability (see Section 3.1). To avoid the pure-pixel case scenario, we discarded the realizations containing values greater than 0.9.

For μ ranging from 0.0001 to 0.001, the quantities trace $\left(\hat{\tilde{\mathbf{S}}}\mathbf{D}\hat{\tilde{\mathbf{S}}}^\top\right)$ and $\log\left(\det\left(\hat{\tilde{\mathbf{S}}}^\top\hat{\tilde{\mathbf{S}}}+\epsilon\mathbf{I}\right)\right)$ were computed, where $\hat{\tilde{\mathbf{S}}}$ is the endmember matrix estimated by either OMVR-NMF or OMDR-ADMM at the last slice k of the image. For MVR, we used four different values of ϵ: $10^{-9}, 10^{-4}, 10^{-3}$ and 10^{-1}. The normalized curves (maximum value equal to 1, minimum value equal to 0) are shown in Figure A.9.

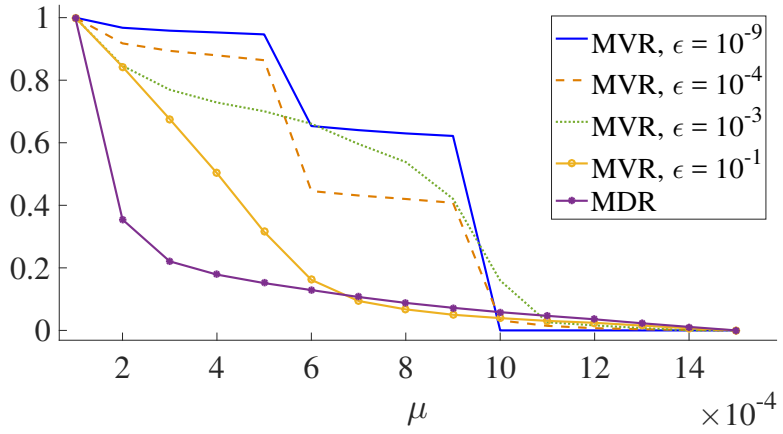
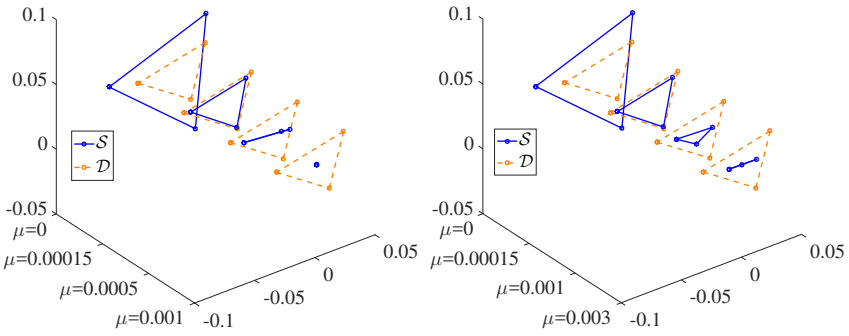


Figure A.9: The plots of MVR and MDR penalties as functions of μ

In Figure A.9, several behaviors can be observed. For MVR with $\epsilon = 10^{-9}$, the plot presents a staircase shape and, as the value of ϵ increases, the steps tend to disappear. Actually, for low values of ϵ, MVR favors rank deficiency as μ increases; this is reflected on the curves by abrupt staircase shape variations. When ϵ increases, the full column rank of $\tilde{\mathbf{S}}$ tends to be better preserved; the rank deficiency is observed only for very large value of μ.

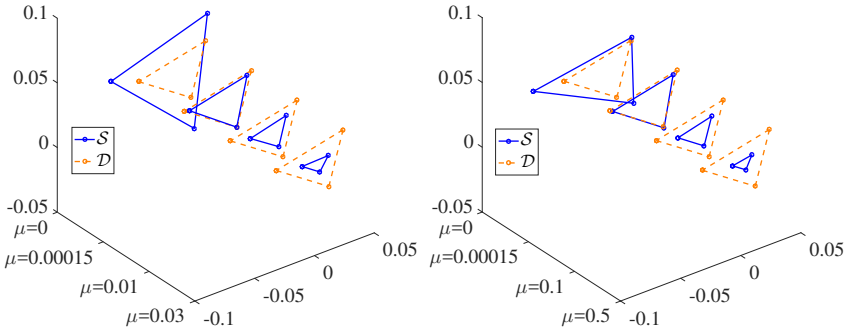
In Figure A.10, we compared the true and estimated simplexes (noted respectively $\mathcal{D}$ and $\mathcal{S}$) for different values of μ, for MVR ($\epsilon = 10^{-9}$, 10^{-3}, and 10^{-1}) and MDR. For $\epsilon = 10^{-9}$, as μ increases, the volume of the estimated simplex tends to zero, meaning 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 $\epsilon = 10^{-1}$ 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 MDR (Figure A.10(d)), the simplex decreases progressively, while preserving the rank of the decomposition.



(a) MVR, $\epsilon = 10^{-9}$

(b) MVR, $\epsilon = 10^{-3}$



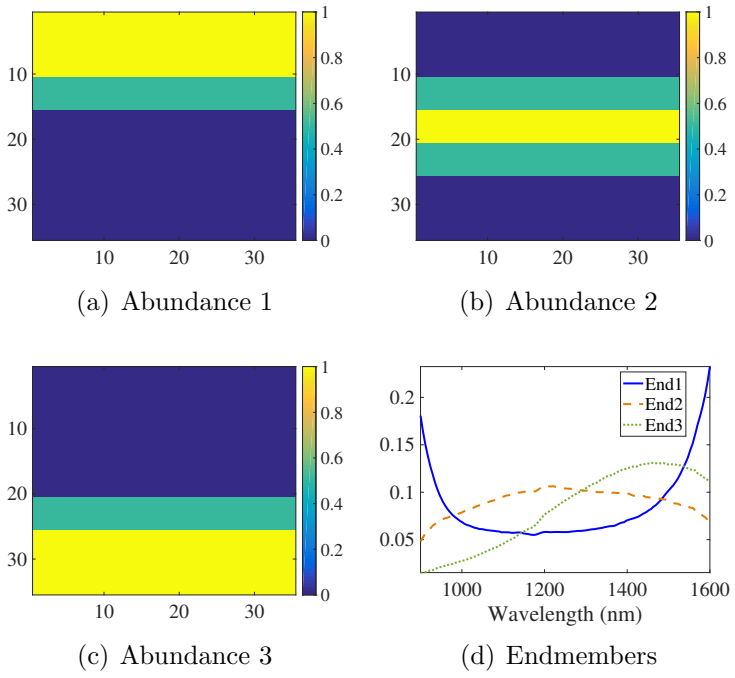
(c) MVR, $\epsilon = 10^{-1}$

(d) MDR

Figure A.10: Illustration of simulated and true simplexes for MVR and MDR as functions of μ

AppendixA.3. Convergence rate

The convergence speed of OMDR-ADMM and OMVR-NMF was numerically studied on a synthetic hyperspectral image. A hyperspectral image of size $119 \times 35 \times 35$, composed of $R = 3$ endmembers not varying over time was simulated. Here, 119 corresponds to the number of wavelengths and 35×35 to the (spatial $\times$ time) dimensions. Each new time sample is a 119×35 slice of the hyperspectral image. The abundance coefficients in each pixel are non-negative and they sum-to-one; the corresponding endmembers are shown in Figure A.11. The data were corrupted by a low-level noise.



(a) Abundance 1 (b) Abundance 2

(c) Abundance 3 (d) Endmembers

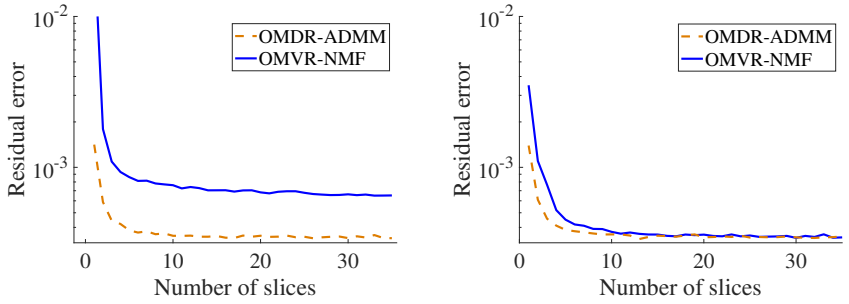
Figure A.11: Simulated abundances and endmembers (End)

We used for OMVR-NMF and OMDR-ADMM the same coefficient $\alpha = 0.99$. The parameter ρ for OMDR-ADMM algorithm was set to 0.01 and ϵ for OMVR-NMF to 1. For all experiments, the endmembers and abundances were randomly initialized using a continuous uniform distribution on the interval $[0, 1]$.

One of the essential parameters for both algorithms is the hyperparameter μ. The latter should usually be chosen small. In fact, a large μ forces

the vertices of the convex hull of the endmembers to be very close to each other, making the endmember matrix ill-conditioned and/or rank-deficient. In order to make the algorithm insensitive to scale variations and given the sequential nature of the processing, we normalize μ in the following way: $\mu = \tilde{\mu} \times \left\| \tilde{\mathbf{X}}^{(1)} \right\|_F^2$, where $\tilde{\mathbf{X}}^{(1)}$ is the first slice of the hyperspectral image. $\tilde{\mu}$ is a tuning parameter that we choose from a fixed search interval in order to minimize the performance criterion (A.2) (see Figure A.18 for more details).

To compare the convergence speeds of the two algorithms, the residual error *vs.* time sample (from 1 to 35) was evaluated for different values of N_1. N_2 was set to 10 and $\tilde{\mu}$ to 10^{-5}. The results are shown in Figure A.12. Note that a logarithmic scale was used for the vertical axis. In Figure A.12, we observe that for all the considered values of N_1, OMDR-ADMM has a faster convergence rate than OMVR-NMF. The latter reaches asymptotically an error close to OMDR-ADMM for approximately $N_1 = 400$ iterations.



(a) $N_1 = 100$ (b) $N_1 = 400$

Figure A.12: The residual error for OMVR-NMF and OMDR-ADMM as functions of number of slices (for the data represented in Figure A.11)

In order to quantify the estimation quality of the endmembers and of the abundances for different values of N_1, we computed the SAD and the RMSE performance criteria. For a given value of N_1, the SAD was calculated at each slice k of the hyperspectral image and then averaged over the entire dataset. For this simulation, we compared OMDR-ADMM and OMVR-NMF to OMDR-NMF algorithm. The results obtained by averaging over 20 trials are illustrated in Figure A.13. First of all, we note that the use of MDR, compared to MVR, makes it possible to reach good quality estimates with fewer iterations. Moreover, OMDR-ADMM has the fastest convergence

rate; the slow convergence rate of OMDR-NMF and OMVR-NMF can be attributed to the use of multiplicative update rules.

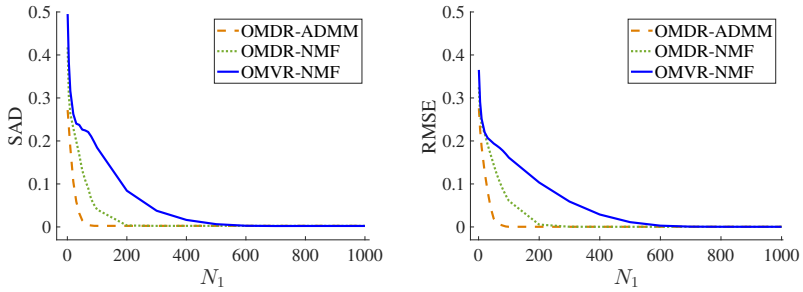


Figure A.13: SAD and RMSE convergence speeds for OMDR-ADMM, OMDR-NMF and OMVR-NMF as functions of N_1 (for the data represented in Figure A.11)

Table A.1 gives the values of N_1 for which the algorithms reach approximately the same asymptotic SAD and RMSE, as well as the associated processing time. In terms of accuracy, MDR performs slightly worse in terms of SAD and RMSE than MVR, because it is a coarse approximation of the simplex volume. However, OMDR-ADMM requires the lowest number of iterations which makes it the algorithm with the lowest processing time. These results show that there is a real interest in terms of rapidity and accuracy in using OMDR-ADMM for fast on-line blind unmixing.

	SAD	RMSE	N_1	Time (s)
OMDR-ADMM	0.0025	0.0004	100	0.52
OMDR-NMF	0.0025	0.0004	300	0.63
OMVR-NMF	0.0023	0.0003	700	1.32

Table A.1: Asymptotic SAD and RMSE values for OMDR-ADMM, OMDR-NMF and OMVR-NMF, related to the processing time and the number of iterations (for the data represented in Figure A.11)

To study the influence of N_2, the SAD and the processing time were evaluated for OMDR-ADMM, for different values of the couple (N_1, N_2). The results are illustrated in Figure A.14 as a bar graph. We note that the couples which minimize the SAD are (100, 10) and (300, 1). However, they differ in their processing times which are 0.52 s and 0.82 s respectively. We see that there is a real interest in using the N_2 loop of iterations, both to ensure

that the abundance matrice tends to sum-to-one but also to accelerate the convergence speed of the algorithm.

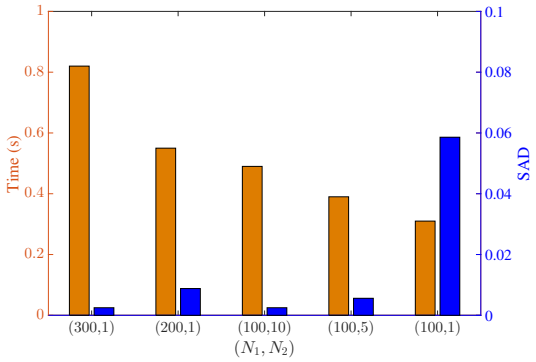


Figure A.14: SAD and processing time for OMDR-ADMM as functions of different values of the couple (N_1, N_2) (for the unmixing of the data represented in Figure A.11)

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 OMDR-ADMM algorithm with its batch counterpart BMDR-ADMM. We used the same synthetic hyperspectral image (corrupted by a low-level noise) presented in Figure A.11. Both algorithms used the same $\rho = 0.01$ and $N_2 = 10$; for the minimum dispersion regularization, $\tilde{\mu}$ was set to 10^{-4} for the batch version, for optimum performance. The BMDR-ADMM method was applied to the unfolded version of the hyperspectral image, of size 119×1225, 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_1, ranging from 1 to 3000. The results obtained by averaging over 20 trials are illustrated in Figure A.15. One can see that our on-line algorithm converges much faster than its batch counterpart. Indeed, 100 iterations are enough for OMDR-ADMM to yield accurate estimates, while BMDR-ADMM requires at least 2000 iterations to achieve the same SAD value. This has strong consequences on the computational cost. Consider the computational complexity of the on-line and batch versions established in Section 4.2; if $R = 3$, $P = 35$, $L = 119$, $K = 35$, $N_1 = 100$ for the on-line version, $N_1 \approx 2000$ for the batch version and $N_2 = 10$, the computation complexity of OMDR-ADMM is about ten times lower than its batch counterpart. For the considered dataset, the

processing time by BMDR-ADMM was of about 1.50 s.

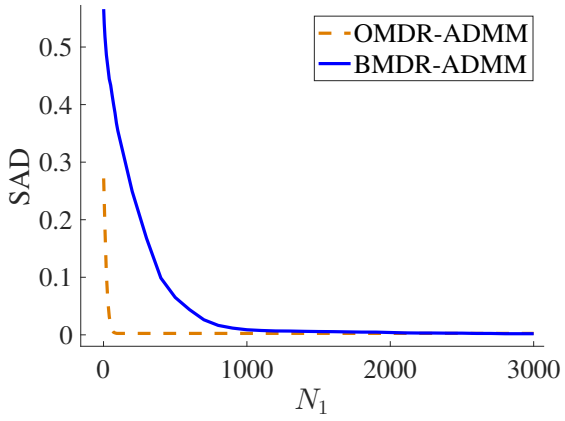


Figure A.15: SAD for BMDR-ADMM and OMDR-ADMM as functions of N_1 (for the unmixing of the data represented in Figure A.11)

We also examined the influence of parameters α and ρ on the convergence speed of OMDR-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 $\rho = 0.01$, $\tilde{\mu} = 10^{-5}$ and $N_2 = 10$. The values of SAD as functions of N_1, for different values of α, are illustrated in Figure A.16. 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 while keeping the value of $\alpha = 0.99$, $\tilde{\mu} = 10^{-5}$, $N_1 = 100$ and $N_2 = 10$. Figure A.17 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.* 5×10^{-6}), we notice that the SAD increases, a sign that the algorithm becomes unstable. For the proper functioning of OMDR-ADMM, there is a range of suitable values for the parameter ρ; for this experiment, a value of ρ between 0.01 and 10^{-5} ensures the stability of the algorithm.

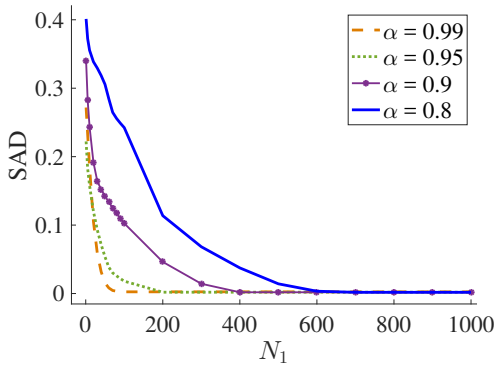


Figure A.16: SAD for OMDR-ADMM as functions of N_1 for different values of α (for the unmixing of the data represented in Figure A.11)

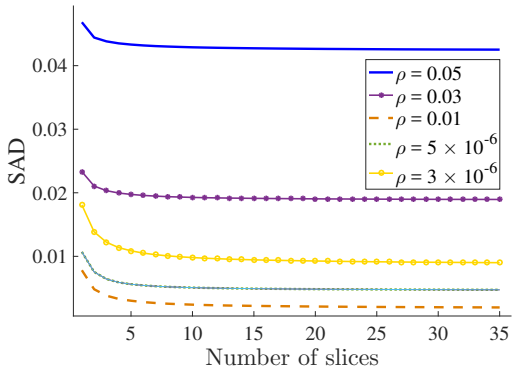


Figure A.17: SAD for OMDR-ADMM as functions of number of slices for different values of ρ (for the unmixing of the data represented in Figure A.11)

The last aspect studied by numerical simulations is the sensitivity of OMDR-ADMM to initial conditions. For the hyperspectral image (corrupted by a low-level noise) presented in Figure A.11, we used 20 different random initializations for the endmember and abundance matrices. The SAD and the RMSE were computed for each initialization, using $N_1 = 100$, $N_2 = 10$ and $\tilde{\mu} = 0$, 10^{-6}, 10^{-5}, 10^{-4} and 10^{-3}. The results are shown in Figure A.18 as box plots. When $\tilde{\mu}$ is too small, different initializations are likely to produce different estimates of the endmembers and of the abundances. When the value of $\tilde{\mu}$ is adequately chosen, the algorithm is not sensitive to initialization. Finally, when the value of $\tilde{\mu}$ is too high ($> 10^{-4}$), the SAD

and the RMSE increase, indicating that the problem is over-regularized.

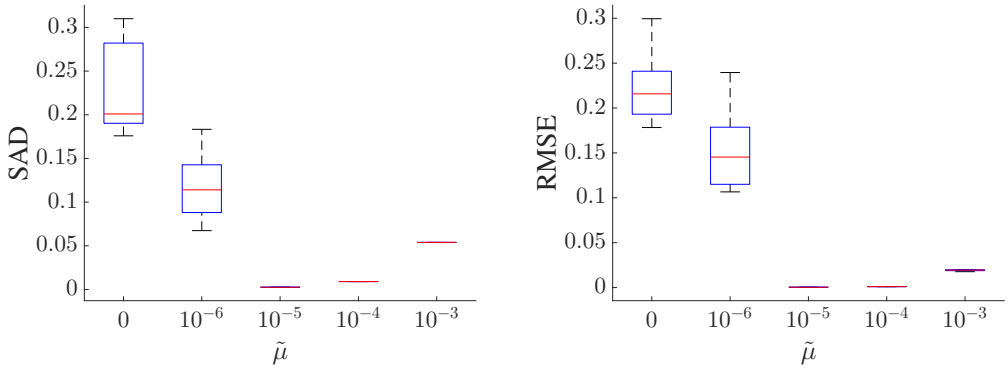


Figure A.18: SAD and RMSE for OMDR-ADMM as functions of $\tilde{\mu}$ for different initializations (for the unmixing of the data represented in Figure A.11)

AppendixA.4. Tracking the spectral variability

In hyperspectral imaging, the spectral signature of a component may present intra-class variability [49, 50]. 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 $119 \times 35 \times 100$ with non-stationary endmembers was simulated. The abundance maps were identical to those in Figure A.11, 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 initialized from the standard normal distribution. The simulated data were corrupted by noise with an SNR = 26 dB. The parameters $\tilde{\mu}$ and ρ were set to 10^{-4} and 0.01 respectively. In order to assess the ability of our algorithm to follow the evolution of the endmembers, we varied α from 0.99 to 0.5 and, for each slice k, we computed the SAD. The results are shown in Figure A.19. 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.

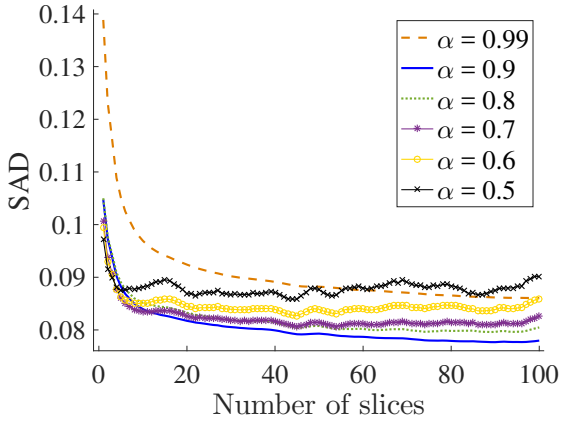


Figure A.19: SAD for OMDR-ADMM as functions of number of slices for different values of α (in the case of non-stationary endmembers)

AppendixA.5. Test of OMDR-ADMM on Jasper Ridge hyperspectral image

In this part, we used a real hyperspectral image (with the endmember and abundance references provided in [48]) to compare the performance of OMDR-ADMM to its batch version (BMDR-ADMM) and to other state-of-the-art volume-regularization algorithms: RVolMin [18] and VRNMF [20]. Note that there are three versions of the VRNMF algorithm: the first uses the minimum volume regularization based on the determinant, the second uses the logarithm of the determinant and the last uses the nuclear norm.

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 OMDR-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 the rank preservation was not guaranteed from one slice to another. The batch methods were applied to the unfolded version of the hyperspectral image, of size 198×10000.

For OMDR-ADMM, the parameters were set as follows: $R = 4$, $\tilde{\mu} = 10^{-5}$, $\alpha = 0.99$, $\rho = 0.01$, $N_1 = 100$ and $N_2 = 10$. For the batch algorithms, N_1 was

set to 500 and we have tuned the regularization parameter of each method in the same way as for the on-line setup. The endmember and abundance matrices was randomly generated and we kept the same initialization for all methods. In order to evaluate the quality of the estimated endmembers and abundances, the SAD and the RMSE for all methods were computed. The results are provided in Table A.2. First of all, it can be seen, from Table A.2, that $\text{VRNMF}_{\text{logdet}}$, BMDR-ADMM and RVolMin yield lower SAD and RMSE that $\text{VRNMF}_{\text{det}}$, although the regularization terms used in these algorithms are not an exact measure of simplex volume (such as the determinant regularizer). $\text{VRNMF}_{\text{logdet}}$ outperforms the state-of-the-art batch methods as it yields the lowest SAD and RMSE values. We note that BMDR-ADMM presents similar performances to $\text{VRNMF}_{\text{logdet}}$. However, the processing time is considerably reduced compared to the latter (5.28 s *vs.* 11.45 s). RVolMin generates slightly worse estimates than BMDR-ADMM, for twice the processing time, while $\text{VRNMF}_{\text{nuclear}}$ has the worst performance.

	OMDR-ADMM		BMDR-ADMM		RVolMin	
Endmembers	SAD	RMSE	SAD	RMSE	SAD	RMSE
Tree	0.0757	0.1179	0.1630	0.1448	0.1379	0.1520
Water	0.0611	0.1800	0.1931	0.1565	0.2622	0.1659
Soil	0.0570	0.1221	0.0650	0.1010	0.0521	0.1234
Road	0.0514	0.0843	0.0664	0.0831	0.0837	0.0770
Average	0.0613	0.1261	0.1219	0.1213	0.1340	0.1296
Time (s)	2.38		5.28		10.64	

	$\text{VRNMF}_{\text{det}}$		$\text{VRNMF}_{\text{logdet}}$		$\text{VRNMF}_{\text{nuclear}}$	
Endmembers	SAD	RMSE	SAD	RMSE	SAD	RMSE
Tree	0.1682	0.2181	0.1227	0.1259	0.1524	0.1452
Water	0.2800	0.1270	0.2368	0.0862	0.2848	0.2241
Soil	0.1384	0.1333	0.0735	0.1554	0.1981	0.2174
Road	0.0451	0.1102	0.0482	0.0671	0.0744	0.1372
Average	0.1579	0.1471	0.1203	0.1086	0.1774	0.1810
Time (s)	13.37		11.45		8.20	

Table A.2: SAD and RMSE for all methods (for the real image *Jasper Ridge*)

Interestingly, we observe that the algorithm that best estimates the end-

members is OMDR-ADMM, while exhibiting the lowest processing time (2.38 s). This difference, compared to the batch methods, can be explained by the fact that, unlike the off-line approaches, it can account for the spectral variability, that modifies locally the spectrum of pure materials. 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.20 which plots the spectra for all estimated endmembers slice by slice for the on-line algorithm. Thanks to its tracking capability, OMDR-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.2, the SAD, for OMDR-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.

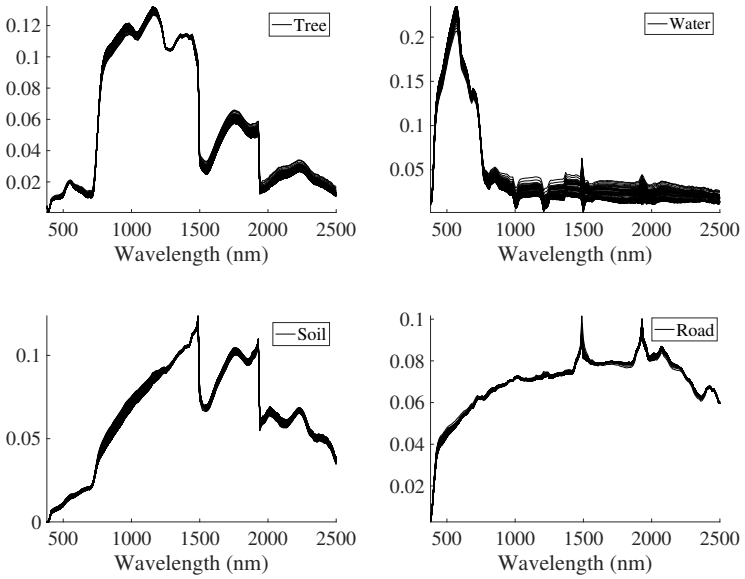


Figure A.20: Estimated endmembers slice by slice for OMDR-ADMM (for the real image *Jasper Ridge*)

The endmembers and the abundance maps estimated by all methods, along with the references provided in [48], are represented in Figure A.21 and A.22. For OMDR-ADMM, the average values of the endmembers are plotted.

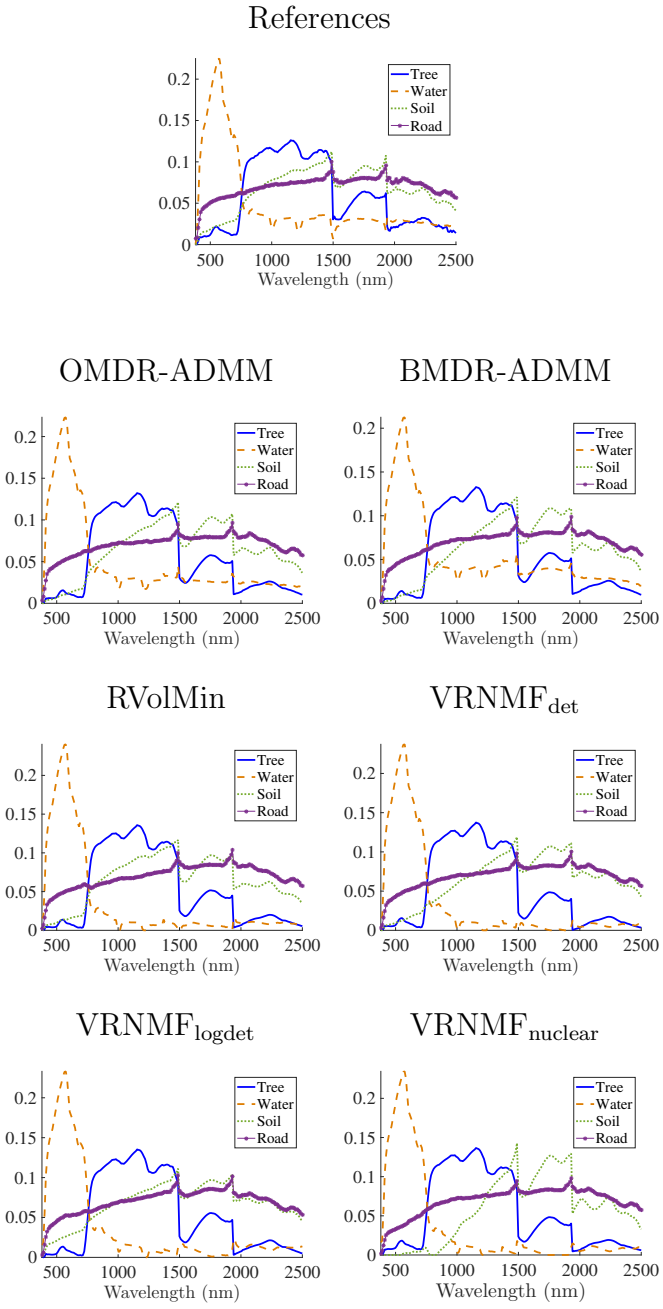


Figure A.21: References and estimated endmembers by all methods (for the real image *Jasper Ridge*)

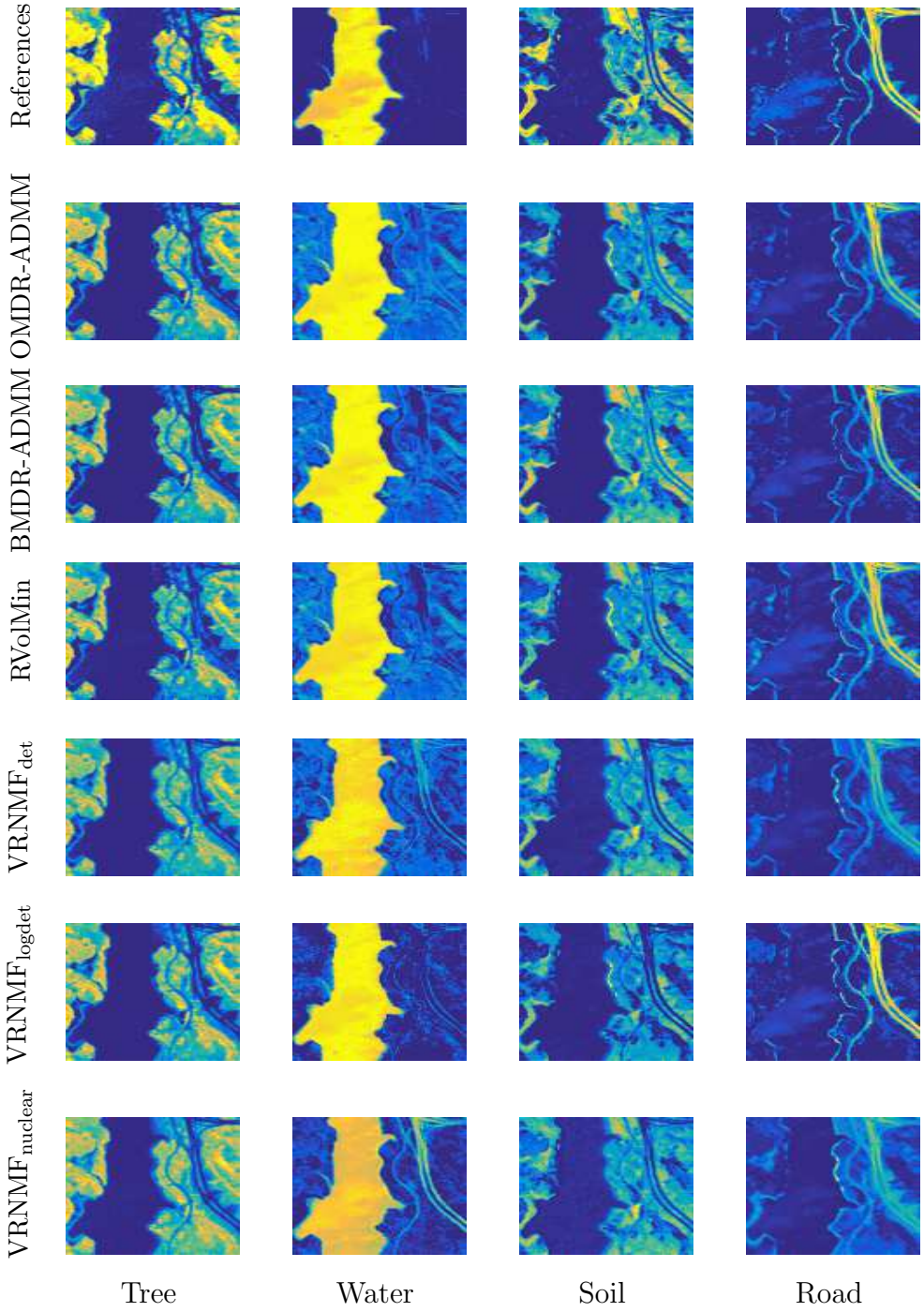


Figure A.22: References and estimated abundance maps by all methods (for the real image *Jasper Ridge*)

AppendixB. Application to on-line analysis of wood surface

We compared the results generated by OMDR-ADMM to the state-of-the-art volume-regularization algorithm RVolMin [18]. For the latter, we used the same initialization as for our algorithm, the same approach to select the value of the regularization parameter and the number of iterations was set to 500. The abundance maps obtained for the wood sample considered in the paper are presented in Figure B.23.

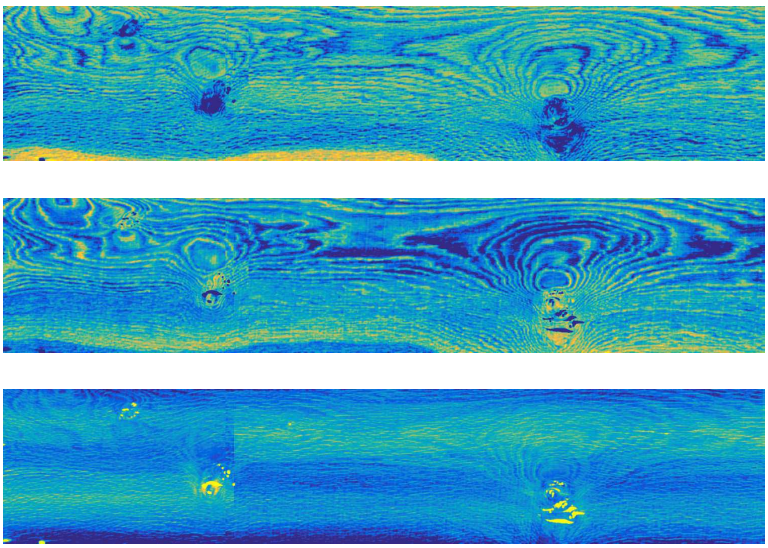
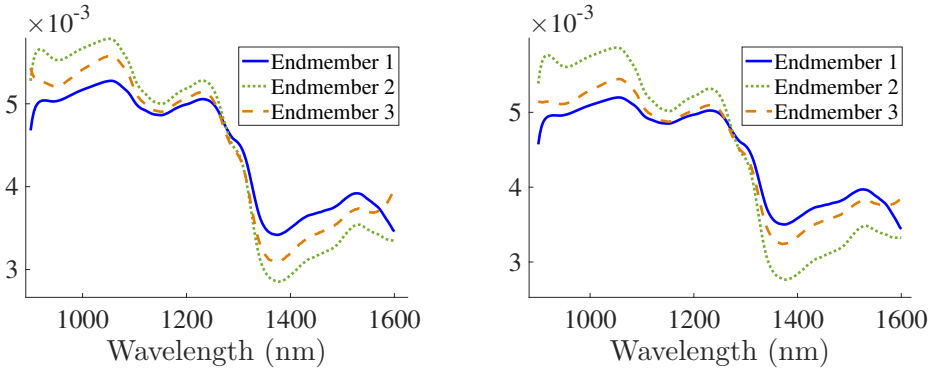


Figure B.23: Abundance maps estimated by RVolMin

Th estimated endmembers for OMDR-ADMM and RVolMin are represented in Figure B.24.



(a) OMDR-ADMM (b) RVolMin

Figure B.24: Estimated endmembers

With a few differences, the abundance maps and the endmembers generated by RVolMin are very similar to those generated by OMDR-ADMM, which validates the effectiveness of the proposed approach. However, the major difference between the two methods is the processing time; RVolMin processed the entire hyperspectral image in 2 min 15 s, compared to 3.22 s for OMDR-ADMM. This shows that there is a real interest in performing on-line blind hyperspectral unmixing.

We also propose to compare the two methods on the wood sample shown in Figure B.25, measuring 120 cm × 15 cm.

Figure B.25: Selected wood sample (visible range color image)

This sample had a dark grain area in the middle, over its entire length. This defect is undesirable, particularly in the parquet industry, because it degrades the aesthetic appearance of the product. Thus, the detection of this defect represents an important economical issue, in many wood industry branches. The piece of wood was scanned by a NIR pushbroom imager working in the spectral range of 900-1600 nm, resulting in hyperspectral image

of dimension $224 \times 200 \times 1100$, where 224 represents the number of wavelengths and the other two numbers, the (spatial $\times$ time) dimensions. The acquired image was processed sequentially by OMDR-ADMM, slice by slice. The number of endmembers to be extracted was set to $R = 3$, after several preliminary tests. The parameters of the algorithm were set as follows: $\tilde{\mu} = 10^{-5}$, $\alpha = 0.99$, $\rho = 0.01$, $N_1 = 15$ and $N_2 = 10$. The endmember and abundance matrices were randomly initialized from a continuous uniform distribution on the interval $[0, 1]$. For RVolMin, we used the same initialization as for our algorithm, the same approach to select the value of the regularization parameter and the number of iterations was set to 500.

The abundance maps obtained by OMDR-ADMM and RVolMin are presented in Figure B.26 and B.27 respectively.

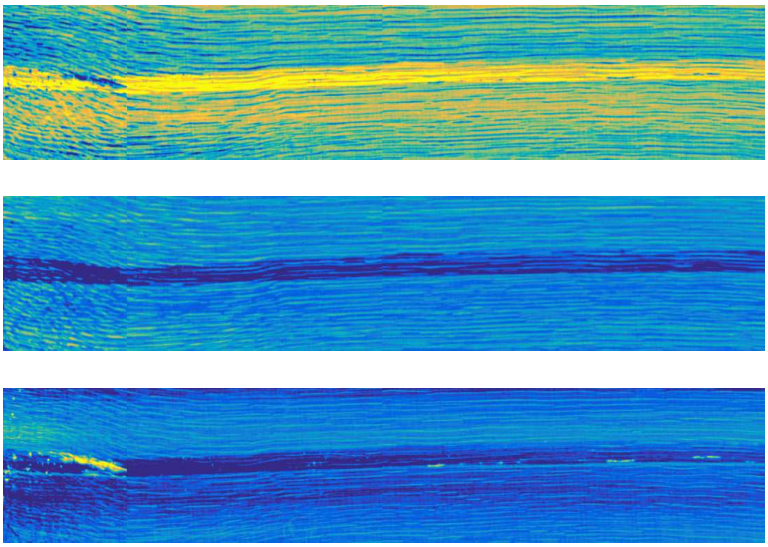


Figure B.26: Abundance maps estimated by OMDR-ADMM

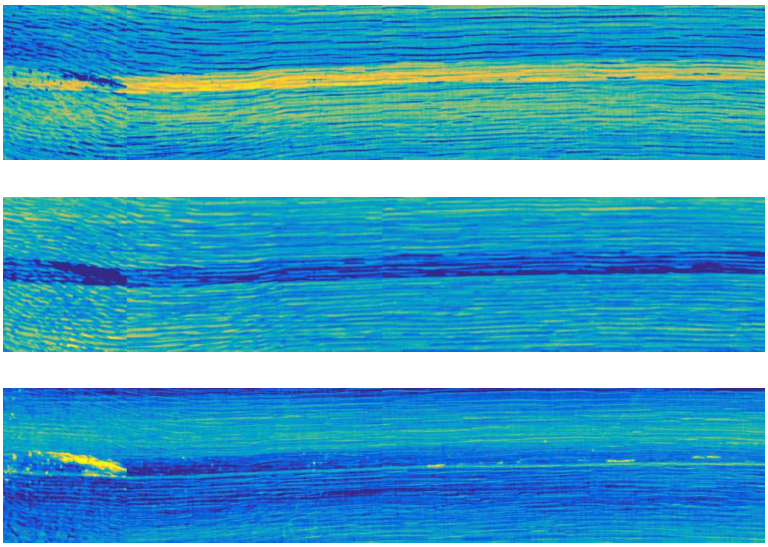


Figure B.27: Abundance maps estimated by RVolmin

In both cases, in the first abundance map, we observe that the area corresponding to the dark grain are much more contrasted than the rest of the image. These are very promising results, as, for the considered sample, the color of the 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. In general, the results generated by the two methods are essentially the same and validate the reliability of our approach. In addition, for the considered dataset, OMDR-ADMM has a very low processing time compared to RVolMin (2.24 s *vs.* 1 min 35 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] A. Huck, M. Guillaume, J. Blanc-Talon, Minimum dispersion constrained nonnegative matrix factorization to unmix hyperspectral data, IEEE Transactions on Geoscience and Remote Sensing 48 (2010) 2590–2602.

[17] 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.

[18] 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.

[19] X. Fu, W.-K. Ma, K. Huang, N. D. Sidiropoulos, Robust volume minimization-based matrix factorization via alternating optimization, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 2534–2538.

[20] A. M. S. Ang, N. Gillis, Algorithms and comparisons of nonnegative matrix factorizations with volume regularization for hyperspectral unmixing, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2019).

[21] 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.

[22] 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.

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

[24] 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.

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

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

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

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

[33] 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.

[34] 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.

[35] 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.

[36] 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.

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

[38] 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.

[39] 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.

[40] 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).

[41] 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.

[42] C.-H. Lin, W.-K. Ma, W.-C. Li, C.-Y. Chi, A. Ambikapathi, Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case, IEEE Transactions on Geoscience and Remote Sensing 53 (2015) 5530–5546.

[43] X. Fu, W.-K. Ma, K. Huang, N. D. Sidiropoulos, Blind separation of quasi-stationary sources: Exploiting convex geometry in covariance domain, IEEE Transactions on Signal Processing 63 (2015) 2306–2320.

[44] 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.

[45] 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.

[46] 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.

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

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

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

[50] 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.