Sparse Distributed Multitemporal Hyperspectral Unmixing

Jakob Sigurdsson, Magnus O. Ulfarsson, Johannes R. Sveinsson, and José M. Bioucas-Dias

***Abstract*—Blind hyperspectral unmixing jointly estimates spectral signatures and abundances in hyperspectral images. Hyperspectral unmixing is a powerful tool for analyzing hyperspectral data. However, the usual huge size of hyperspectral images may raise difficulties for classical unmixing algorithms, namely due to limitations of the hardware used. Therefore, some researchers have considered distributed algorithms. In this paper, we develop a distributed hyperspectral unmixing algorithm that uses the alternating direction method of multipliers (ADMM), and ℓ_1 sparse regularization. The hyperspectral unmixing problem is split into a number of smaller subproblems which are individually solved and then the solutions are combined. A key feature of the proposed algorithm is that each subproblem does not need to have access to the whole hyperspectral image. The algorithm may also be applied to multitemporal hyperspectral images with due adaptations accounting for variability that often appears in multitemporal images. The effectiveness of the proposed algorithm is evaluated using both simulated data and real hyperspectral images.**

***Index Terms*—Hyperspectral unmixing, feature extraction, blind signal separation, linear unmixing, alternating direction method of multipliers, distributed algorithms, multitemporal unmixing**

I. Introduction

INTEREST in hyperspectral images (HSIs) has increased significantly over the past decade, mainly due to high spectral resolution, which enables precise material identification using spectroscopic analysis. HSIs were introduced decades ago in mining and geology. Since then, their use has spread to diverse fields such as manuscript research, medical imaging, and remote sensing [1]–[3].

However, the spatial resolution of HSIs in remote sensing applications is often of the order of meters or tens of meters, such that more than one material may be present within one pixel. Pixels containing more than one material are called *mixed pixels*, in contrast to *pure pixels*, which only contain one material. Each pixel in a HSI is thus composed of a mixture of the spectral signatures of materials within the spatial boundaries of the pixel.

The term *endmember* is used to describe one specific material in a HSI, and an *abundance map* specifies the percentage of one specific material for all the pixels. Hyperspectral unmixing is the process of estimating the number of endmembers, their spectral reflectance, termed endmember signatures, and their corresponding abundance maps [2], [3].

This work was supported by the Postdoctoral Research Fund at the University of Iceland, the Research Fund of the University of Iceland and Portuguese Fundação para a Ciência e Tecnologia (FCT) under grants UID/EEA/5008/2013 and ERANETMED/0001/2014.

The linear mixture model (LMM) has extensively been used to model HSIs. If the mixing is assumed to be macroscopic and the image is flat, the LMM is a good approximation for the light scattering phenomenon [2]. However, a remote sensed HSI is not flat and the illumination conditions may vary between sections of the image. For these reasons, the spectral signatures measured by the sensor may vary both in amplitude and form. Also, HSIs of the same region acquired at different times may have endmembers that differ since the acquisition conditions may not be identical for all acquisitions, i.e., the endmembers from these different HSIs may have temporal variability.

HSIs can be very large and with ever-improving optical, computing, and processing equipment, they will only continue to grow in size. This increase in size means that the memory requirements of the hardware used to unmix these big hyperspectral data will also increase. Distributed methods can be used to cope with these ever increasing requirements. The alternative direction method of multipliers (ADMM) [4], [5] is a framework that can be used to develop distributed algorithms. Using ADMM, the optimization problem is solved iteratively; in each iteration, a number of independent subproblems are solved independently and then the different solutions are merged. In this paper, we exploit the structure of the ADMM iterative solver to design a distributed hyperspectral unmixing algorithm, where each independent problem is solved by a processor. Then, the solutions obtained by the processors are merged using moderate communication resources.

A. Regularizers and Constraints In Hyperspectral Unmixing

Hyperspectral unmixing is a blind source separation problem that often uses regularizers to incorporate additional information into the unmixing. From an inverse problem perspective, regularizers help to cope with the ill-conditioned nature of hyperspectral unmixing. Sparsity promoting regularizers have been widely used in hyperspectral unmixing [2], [6]–[9]. A paradigmatic example is the use of sparsity inducing regularizers to promote sparse abundance maps, as it is unlikely for every material in a HSI to be present in every pixel in the image.

Constraining both the endmember signatures and abundances to be nonnegative is common practice in hyperspectral unmixing, since the former are reflectances and the latter are fractions. Constraining the sum of the abundances for each pixel to be one is widely used in hyperspectral unmixing. This constraint is called the abundance sum constraint (ASC). The ASC also stabilize the unmixing solution. However HSIs are noisy, and the ASC does not take into account variations in reflectance from the same material. For these reasons, the ASC has received some criticism [10], and it is not obvious whether to relax the ASC or consider it a part of the modeling error [2]. Another constraint that also stabilizes the solution is the endmember norm constraint (ENC). The ENC puts an unit norm constraint on the endmembers, forcing the endmembers to have the same energy. The ENC has been widely used in hyperspectral unmixing [9], [11]–[13].

B. Related Work

A representative class of blind, non-pure pixel based, hyperspectral unmixing methods are variants of nonnegative matrix factorization (NMF). NMF became widely used in many fields of research following the publication of the Lee-Seung algorithm [14] for NMF. Many NMF-type unmixing methods apply a sparsity regularizer on the abundances, typically the ℓ_1 norm [15]–[17]. In [18], the ℓ_0 norm is used to promote sparse abundance maps and it is compared to the ℓ_1 norm. In [9], the ℓ_q $(0 \leq q \leq 1)$ regularizers are thoroughly evaluated for hyperspectral unmixing. The minimum volume regularizer on the mixing matrix [2], [19] is also a common regularizer in hyperspectral unmixing.

Distributed and parallel hyperspectral unmixing has received some attention recently. A distributed parallel algorithm was presented in [20], where the HSI is split either by bands or spectra. This approach still requires access to the whole dataset. Two parallel endmembers algorithms are discussed in [21], namely automatic morphological endmember extraction [22], and orthogonal subspace projection [23]. In [12], ADMM was used to develop a distributed version of NMF, and in [24], a partial version of the algorithm proposed here was presented.

In [25], a multitemporal unmixing algorithm is proposed, where a dynamical model based on the linear mixing processes is used at each time instant. The estimated endmember spectra are constrained to be similar to a set of endmember spectra that is assumed to be known or estimated in some manner. The abundances are similarly constrained to vary smoothly temporally.

Endmember spectral variability has been addressed by various researchers. In [26], the endmember spectra are assumed to be realizations of multivariate distributions, and in [27], they are represented as spectral signature bundles. In [28], a review of different methods that address spectral variability in HSIs was presented.

The topic of unmixing multitemporal images has also recently received attention. In [29]–[33], multitemporal unmixing was used for change detection, and in [34], an online unmixing of HSIs was introduced, accounting for spectral variability among the different images.

C. Paper Contribution

In this paper, a novel cost function is minimized by using ADMM, to solve two different distributed sparse hyperspectral unmixing problems. The first problem is distributed hyperspectral unmixing and the second problem is distributed multitemporal unmixing. A regularizer is used to promote sparse abundance maps. To handle multitemporal unmixing, the algorithm proposed in [24] is extended to account for spectral variability between the different subproblems by adopting the approach used in [34]. The algorithm can therefore be used to unmix multitemporal HSIs affected by temporal variability.

The proposed algorithm differs from the one in [34], where the hyperspectral unmixing is formulated as a two-stage stochastic program, since it is based on the ADMM and uses dyadic cyclic descent optimization [35]–[38] to solve the ADMM subproblems. The method does not assume that any endmembers are known *apriori*, or estimated independently, as was done in [25].

When unmixing a HSI, the image is first spatially split into a number of smaller HSIs, which are then independently unmixed. The obtained individual solutions are then merged together. Each individual subproblem is solved using dyadic cyclic descent and the solutions are merged together using ADMM. In this case, we do not assume that there is spectral variability between the different subimages. Each individual subproblem does not need to have access to the full hyperspectral image. The algorithm can thus be applied to very large datasets where traditional methods may fail because of hardware limitations.

When using the algorithm to unmix multitemporal HSIs, each temporal HSI is treated as one subproblem. The spectral variability between the HSIs is properly addressed. Each temporal HSI is unmixed independently and the solutions are merged in a similar way to that of distributed unmixing.

D. Notation

The following notations are used in this paper:

$\boldsymbol{s}$ vectors are denoted by lower case bold letters.
$\boldsymbol{S}$ matrices are denoted by upper case bold letters.
$\boldsymbol{s}_p^T$ pth row of matrix $\boldsymbol{S}$.
$\boldsymbol{s}_{(j)}$ jth column vector of $\boldsymbol{S}$.
$\boldsymbol{S}_{\text{-}j}$ $\boldsymbol{S}$ with its jth column removed.
$\boldsymbol{s}^k$ an estimate of $\boldsymbol{s}$ at iteration k.
$\text{tr}(\boldsymbol{S})$ the trace of $\boldsymbol{S}$.
$\|\boldsymbol{s}\|_1$ the ℓ_1 norm of $\boldsymbol{s}$, which is the absolute sum of the vector.
$\|\boldsymbol{S}\|_{1,1}$ the mixed ℓ_1 norm of $\boldsymbol{S}$, i.e. $\sum_i ||\mathbf{s}_{(j)}||_1$.
$\|\boldsymbol{S}\|_F$ the Frobenius norm of $\boldsymbol{S}$.
$\boldsymbol{\Gamma}^{M\times r}$ the set of elementwise nonnegative $M \times r$ matrices where each column has the unit norm.
$\mathbb{R}_+^{P\times r}$ the set of elementwise nonnegative $P \times r$ matrices.
$\max(\boldsymbol{s}, \boldsymbol{0})$ the elementwise maximum operator.
$I(\boldsymbol{s})$ The elementwise identity operator.

E. Paper Structure

The paper is organized as follows. In Section II, the problem formulation and the proposed algorithm are described. In Section III, the algorithm is evaluated using simulated data. In Section IV, the algorithm is applied to real HSIs. In Section V, conclusions are drawn. In the Appendix, the details of the estimation methods are described.

II. Problem Formulation

A. The Minimization Problem

In this paper, we will solve the following minimization problem,

$$\min_{\substack{\boldsymbol{S}\in\mathbb{R}_{+}^{P\times r}\\ \boldsymbol{A}\in\boldsymbol{\Gamma}^{M\times r}}} \frac{1}{2}\|\boldsymbol{Y}-\boldsymbol{S}\boldsymbol{A}^{T}\|_F^2 + h\|\boldsymbol{S}\|_{1,1}, \tag{1}$$

where P, M, and r are positive integers representing, respectively, the number of pixels, the number of spectral bands, and the number of endmembers. $\boldsymbol{Y}$ is a $P\times M$ matrix, where each row represents the spectral vector observed at a pixel of the HSI, $\boldsymbol{A}$ is an $M\times r$ matrix where each column holds endmember spectra, and $\boldsymbol{S}$ is a $P\times r$ abundance matrix where each column is a vectorized abundance map. Here, $r \ll P$, since it is assumed that the number of endmembers in the image is much smaller than the number of pixels.

In this paper, we have traded the usual hardness associated with ℓ_0 regularization with the joint use of ℓ_1 and normalization of the columns of the mixing matrix, which is, from the computational point of view, more manageable than ℓ_0 yielding, nevertheless, very good results.

B. Geometrical representation

Promoting sparse abundances has a similar flavor as the minimum volume regularizer [2]. However, it is simpler to use, since dealing with the ℓ_1 regularizer is easier than dealing with the determinant of $\boldsymbol{A}^T\boldsymbol{A}$, which is typically used as a volume regularizer. To illustrate this, we use the Urban data set described in Subsection IV-A. We unmix this data set using different values of the sparsity parameter h, using Algorithm 2 (given in the Appendix), to solve the optimization (1). This is a blind unmixing problem using an ℓ_1 sparsity inducing regularizer on the abundances, while constraining the endmembers to have unit norm.

Fig. 1 shows, for different values of h, projection of the spectral vectors on the plane defined by bands #60 and #140. The endmembers, at the same bands, are also shown as red triangles. The lines between the origin and the endmembers are also shown in green.

From the four figures, we may conclude that when the sparsity parameter is increased, the endmembers shown as red triangles, are being pulled towards each other, implying that the volume defined by the endmembers decreases. This behavior is further highlighted in Fig 2, where the volume ($\det\sqrt{\boldsymbol{A}^T\boldsymbol{A}}$) is shown as a function of the sparsity parameter.

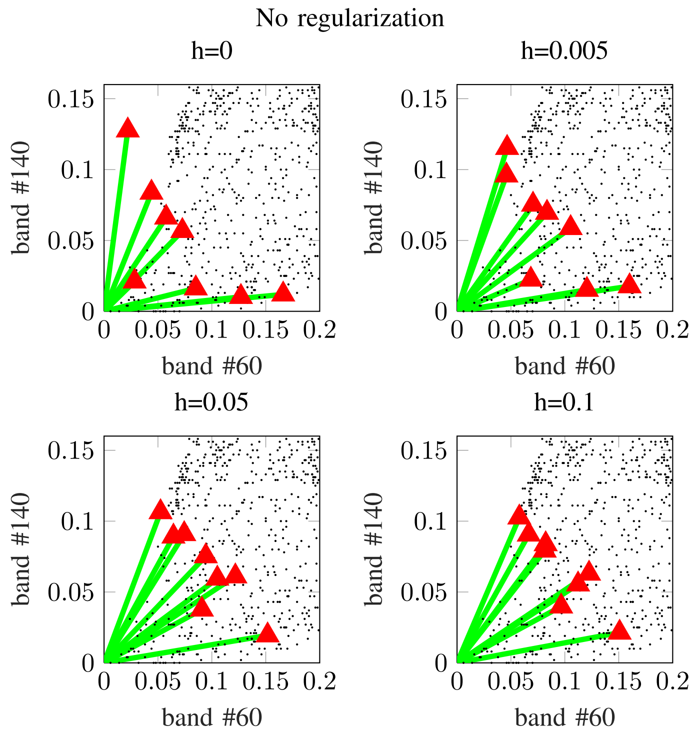


Fig. 1. A geometrical representation of the effects the ℓ_1 regularizer, showing how the endmembers change when the sparsity parameter h, changes.

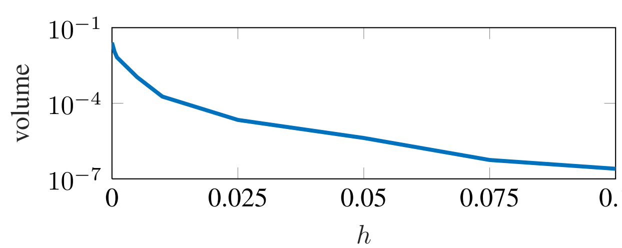


Fig. 2. The volume that the endmembers enclose as a function of the sparsity parameter h.

C. Distributed Unmixing

In order to develop a distributed algorithm, the aim is to decompose (1) into smaller problems. To accomplish this, problem (1) is reformulated into N constrained subproblems as

$$\begin{aligned}\min_{\substack{\boldsymbol{S}_i\in\mathbb{R}_{+}^{P_i\times r}\\ \boldsymbol{A}_i,\boldsymbol{Z}\in\boldsymbol{\Gamma}^{M\times r}}} \quad & \sum_{i=1}^{N}\Big(\frac{1}{2}\|\boldsymbol{Y}_i-\boldsymbol{S}_i\boldsymbol{A}_i^T\|_F^2+h\|\boldsymbol{S}_i\|_{1,1}\Big)\\ \text{s.t. } & \boldsymbol{A}_i-\boldsymbol{Z}=\boldsymbol{0},\end{aligned} \tag{2}$$

where the subscript i corresponds to subproblem number i, $\boldsymbol{Y}=[\boldsymbol{Y}_1^T,\boldsymbol{Y}_2^T,...,\boldsymbol{Y}_N^T]^T$, $\boldsymbol{S}=[\boldsymbol{S}_1^T,\boldsymbol{S}_2^T,...,\boldsymbol{S}_N^T]^T$, $\boldsymbol{A}_i$ is the endmember matrix corresponding to subproblem i, $\sum P_i = P$, and $\boldsymbol{Z}$ is the consensus matrix representing the endmember matrix of the global problem. Problem (2) is nonconvex, since $\boldsymbol{\Gamma}^{M\times r}$ is a nonconvex set and the data fidelity term is nonconvex.

D. Distributed Multitemporal Unmixing

For a multitemporal distributed unmixing algorithm, the original problem is decomposed into N subproblems, each one corresponding to the unmixing of a complete HSI acquired at a given time instant. To account for variability in the endmembers, linked to the different acquisition times and seasonal variations, we introduce N perturbation matrices, $\boldsymbol{M}_i$, into (2) as

$$\min_{\substack{\boldsymbol{S}_i \in \mathbb{R}_+^{P_i \times r} \\ \boldsymbol{A}_i, \boldsymbol{Z} \in \boldsymbol{\Gamma}^{M \times r}}} \sum_{i=1}^{N} \Big(\frac{1}{2} \|\boldsymbol{Y}_i - \boldsymbol{S}_i(\boldsymbol{A}_i + \boldsymbol{M}_i)^T\|_F^2 + h\|\boldsymbol{S}_i\|_{1,1} + \frac{\psi}{2}\|\boldsymbol{M}_i\|_F^2 \Big) \quad (3)$$
$$\text{s.t. } (\boldsymbol{A}_i + \boldsymbol{M}_i) \in \mathbb{R}_+^{M \times r}, \ \boldsymbol{A}_i - \boldsymbol{Z} = \boldsymbol{0}.$$

Each matrix $\boldsymbol{M}_i$ holds the variable part of endmembers. Therefore matrix $\boldsymbol{M}_i$ accounts for the spectral variability of the endmembers associated with the ith problem, while $\boldsymbol{A}_i$ accounts for the non-variable consensus part.

E. Algorithm

To solve the multitemporal distributed unmixing problem in (3), an algorithm based on ADMM [5], [39] is developed. The ADMM subproblems are solved using dyadic cyclic descent (DCD) [35], [36]. We will focus on solving (3), since (2) is a special case of (3) with $\boldsymbol{M}_i = \boldsymbol{0}$. The algorithm used to solve (3) is based on using the augmented Lagrangian (for one subproblem), i.e.,

$$\begin{aligned} L_i(\boldsymbol{Y}_i, \boldsymbol{S}_i, \boldsymbol{A}_i, \boldsymbol{M}_i, \boldsymbol{\Lambda}_i, \boldsymbol{Z}) = \\ \frac{1}{2}\|\boldsymbol{Y}_i - \boldsymbol{S}_i(\boldsymbol{A}_i + \boldsymbol{M}_i)^T\|_F^2 \\ + h\|\boldsymbol{S}_i\|_{1,1} + \frac{\psi}{2}\|\boldsymbol{M}_i\|_F^2 \quad (4) \\ + \operatorname{tr}\big(\boldsymbol{\Lambda}_i^T(\boldsymbol{A}_i - \boldsymbol{Z})\big) + \frac{\rho(k)}{2}\|\boldsymbol{A}_i - \boldsymbol{Z}\|_F^2, \\ i = 1,..., N, \end{aligned}$$

where $\boldsymbol{\Lambda}_i$ is a matrix of size $M \times r$ holding the Lagrange multipliers, and $\rho(k)$ is the augmented Lagrangian weight. The complete Lagrangian is

$$L(\boldsymbol{Y}, \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{M}, \boldsymbol{\Lambda}, \boldsymbol{Z}) = \sum_{i=1}^{N} L_i(\boldsymbol{Y}_i, \boldsymbol{S}_i, \boldsymbol{A}_i, \boldsymbol{M}_i, \boldsymbol{\Lambda}_i, \boldsymbol{Z}). \quad (5)$$

An ADMM-type algorithm is used to solve (3), where (5) is iteratively minimized w.r.t. each optimization variable. The pseudo code for algorithm, termed distributed multitemporal unmixing (DMU), is shown in Algorithm 1. In this algorithm, we use Algorithm 2, shown in the Appendix, to solve the individual subproblems.

Note that each subproblem in Algorithm 1 does not need to have access to the whole image matrix, but only to the corresponding subimage. All the subproblems can thus be solved in parallel. The optimization problem that is solved here is nonconvex and, therefore, there is no guarantee of convergence. We have however systematically observed that the algorithm does perform well, and that the variables of interest $(\boldsymbol{A}_i, \boldsymbol{S}_i, \boldsymbol{M}_i)$ do converge using both simulated and real data.

The stop criteria for Algorithm 2 is

$$\frac{\|\boldsymbol{A}_i^{(k)} + \boldsymbol{M}_i^k - (\boldsymbol{A}_i^{k-1} + \boldsymbol{M}_i^{k-1})\|_F}{\|\boldsymbol{A}_i^k + \boldsymbol{M}_i^k\|_F} < 10^{-7}, \quad (7)$$

and

$$\frac{\|\boldsymbol{S}_i^k - \boldsymbol{S}_i^{k-1}\|_F}{\|\boldsymbol{S}_i^k\|_F} < 10^{-7}. \quad (8)$$

Algorithm 1: The distributed multitemporal unmixing algorithm (DMU).

Input: $\boldsymbol{Y}$, r, N, h
Initialization:
Split $\boldsymbol{Y}$ into N matrices, $\boldsymbol{Y}_1, \boldsymbol{Y}_2, ..., \boldsymbol{Y}_N$.
Initialize $\rho(0)$, $\boldsymbol{A}_i^0$, $\boldsymbol{S}_i^0$, $\boldsymbol{M}_i^0 = 0$, $i = 1, ..., N$
Set $\boldsymbol{Z}^{(0)} = \boldsymbol{0}, \boldsymbol{\Lambda}^{(0)} = \boldsymbol{0}$
for $k = 0...$ **do**
for $i = 1...N$ **do**
Using Algorithm 2 (see Appendix), estimate[1]
$$(\boldsymbol{A}_i^{k+1}, \boldsymbol{S}_i^{k+1}, \boldsymbol{M}_i^{k+1}) = \arg\min L_i(\boldsymbol{Y}_i, \boldsymbol{S}_i, \boldsymbol{A}_i, \boldsymbol{M}_i, \boldsymbol{\Lambda}_i^k, \boldsymbol{Z}^k, h)$$
$$\text{s.t. } \boldsymbol{S}_i \in \mathbb{R}_+^{P_i \times r}, \quad \boldsymbol{A}_i \in \boldsymbol{\Gamma}^{M \times r}, \quad (\boldsymbol{A}_i + \boldsymbol{M}_i) \in \mathbb{R}_+^{M \times r}$$
Estimate $\boldsymbol{Z}^{k+1}$ using
$$\tilde{\boldsymbol{Z}} = \max\left(\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{A}_i^{k+1} + \frac{1}{\rho(k)}\boldsymbol{\Lambda}_i^k), 0\right),$$
$$\boldsymbol{z}_{(j)}^{k+1} = \frac{\tilde{\boldsymbol{z}}_{(j)}}{\|\tilde{\boldsymbol{z}}_{(j)}\|}, \ j = 1,..., r. \quad (6)$$
Estimate $\boldsymbol{\Lambda}_i^{k+1}$ using
$$\boldsymbol{\Lambda}_i^{k+1} = \boldsymbol{\Lambda}_i^k + \rho(k)(\boldsymbol{A}_i^{k+1} - \boldsymbol{Z}^{k+1}).$$
Update $\rho(k)$ using (14)
$$\rho(k) = 10 k^{\frac{8}{30}} + 0.02MP\sigma^2$$
Output: $\widehat{\boldsymbol{Z}}$, $\widehat{\boldsymbol{M}}_i$, $\widehat{\boldsymbol{S}}_i$, $i = 1, ..., N$

Similarly, Algorithm 1 is terminated if the endmembers estimated in all the subproblems have converged to the consensus endmembers, i.e., if

$$\frac{\|\boldsymbol{Z}^k - \boldsymbol{A}_i^k\|_F}{\|\boldsymbol{Z}^k\|_F} < 10^{-6}, \ i = 1, ..., N, \quad (9)$$

or if the maximum number of iteration is achieved.

F. Tuning Parameter Selection

There are five tuning parameters that need to be estimated: the number of endmembers r, the sparsity parameter h, the variability parameter ψ, the number of subproblems N, and the augmented Lagrangian weight $\rho(k)$. To estimate the sparsity parameter h, in the cost function, the extended Bayesian information criteria (EBIC) [40] is minimized. EBIC is a criterion for model selection, where the model with the lowest EBIC is preferred. The EBIC tries to find the best model, here, the best fit of for the data, while minimizing the number of free parameters. The number of free parameters increases when the sparsity of $\boldsymbol{S}$ decreases. The EBIC is therefore used to find a

[1]The variables from the previous iteration used as initial values in the minimization.

balance, where the data fidelity term is low while the sparsity of sparsity $\boldsymbol{S}$ preferred. The EBIC objective function is [41]

$$\text{EBIC} = M\log(\widehat{\sigma}^2) + \frac{1}{P}\frac{\|\boldsymbol{Y}-\widehat{\boldsymbol{S}}\widehat{\boldsymbol{A}}^T\|}{\widehat{\sigma}^2} + \frac{\big(\log(P)+4\alpha\log(M)\big)d}{P}, \tag{10}$$

where $\alpha\in[0,1]$, and d is the number of free parameters in the model, i.e.,

$$d = \|\widehat{\boldsymbol{S}}\|_0 + Mr - r^2, \tag{11}$$

where $\|\widehat{\boldsymbol{S}}\|_0$ is the number of nonzero values in $\widehat{\boldsymbol{S}}$. Unless stated otherwise, we use $\alpha = 0.5$, and

$$\widehat{\sigma}^2 = \frac{1}{PM}\|\boldsymbol{Y}-\widehat{\boldsymbol{S}}\widehat{\boldsymbol{A}}^T\|_F^2. \tag{12}$$

The simultaneous estimation of all the parameters is computationally intensive, since we would need to estimate $\boldsymbol{A}$, $\boldsymbol{S}$ and $\boldsymbol{M}$ for all combinations of the parameters. Therefore, we will first estimate the number of endmembers, r, using $h=0$; in this case, the number of free parameters is

$$d = PM + Mr - r^2. \tag{13}$$

The augmented Lagrangian weight $\rho(k)$ is set to a low value and it is incremented in each iteration forcing all endmember matrices $\boldsymbol{A}_i$ to converge to $\boldsymbol{Z}$ as the algorithm iterations evolve. A maximum of 30 iterations is allowed, and $\rho(k)$ is set according to

$$\rho(k) = 10^{k\frac{8}{30}} + 0.02MP\sigma^2, \tag{14}$$

where k is the iteration number, and σ^2 is variance of the HSI. The median absolute deviation [42] is used to calculate σ^2 for each band and the average value is used.

Using (14), $\rho(k)$ is initially very low and in each iteration it is incremented, and in the last iteration $\rho(30) \approx 10^8$. Using this scheme, the different endmember matrices, $\boldsymbol{A}_i$, will converge to $\boldsymbol{Z}$. Our experiments show that using (14) gives good results, and the variables of interest do converge.

Setting the number of subproblems, N, is done manually, but when unmixing temporal images, the number of subproblems is the same as the number of temporal HSIs.

The variation parameter ψ is set according to

$$\psi = 10^3\sigma^2, \tag{15}$$

where σ^2 is estimated using [42]. For this value of ψ, the the temporal perturbation matrices $\boldsymbol{M}_i$ will be constrained, but still be able to adapt and capture the temporal variations of the endmembers.

The procedure for estimating the parameters is:

1) Choose the number of subproblems, N.
2) Estimate the dimensionality, r, with $h=0$ using (10), (12) and (13). This is done by unmixing the data using different values of r and then calculating the EBIC using the solutions obtained. The value of r that yields the lowest EBIC is then chosen.
3) Estimate the sparsity parameter, h, for a given r, using (10), (11) and (12). In a similar manner as when estimating r, different solutions are obtained using different values of h. The value of h that yields the lowest EBIC is chosen.

To lighten the overall computational complexity in our simulations and experiments, we use one subproblem to estimate the parameters. We have observed in our simulations that using only one subproblem gives an accurate estimate of the parameters and yields good results. The estimated set of parameters is then used for all the subimages.

G. The Subimages

There are no constraints on how the HSI is split up, but different splittings may affect the speed of the algorithm. Here, we show two methods to split the HSI into subimages, the first method splits the HSI vertically into spatially continuous N strips. The second method will randomly place each pixel in the image into one of the N subimages. By randomly selecting pixels into subproblems, we increase the probability that all the endmembers will be present in all the subimages. This will decrease the convergence time of the algorithm and improve the performance. Our proposed algorithm uses the random splitting. An illustration of these two methods is shown in Fig. 3. These two methods to split the data into subimages will be denoted *spatial* and *random* splitting, respectively.

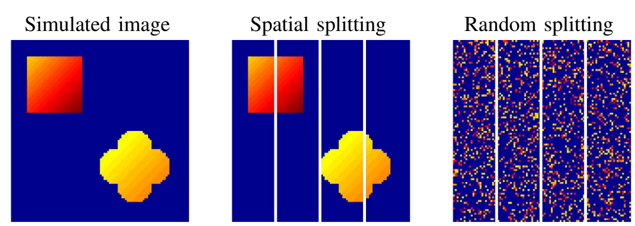


Fig. 3. Left: A simulated image. Center: spatial splitting into 4 subimages. Right: random splitting.

III. Simulations

To evaluate the algorithm, we will use simulated data. The endmembers are mineral signatures from a pruned USGS spectral library[2], containing 100 distinct spectral signatures where the minimum spectral angle distance between any two signatures is 0.16 rad (9^o). The first and last spectral bands are removed so the number of bands used is $M=222$.

In these simulations, five endmembers from the pruned library are used. The abundances are generated following a Dirichlet distribution, with unit parameters (i.e., uniform on the simplex) [43]. Abundance variability is also added by allowing the sum of each abundance vector to be between 0.7 and 1.3. The sparsity of the abundance maps is approximately 35%, which means that 35% of all abundance values are set to zero. This is implemented by randomly setting entries in the abundance map to zero. Gaussian i.i.d. noise is added to the simulated data, resulting in a signal to noise ration of 35dB. The maximum purity of any pixel in the image is 85%.

The spatial dimensions of the simulated image is (200×80) pixels. The image is split into $N=4$ equal subimages, where

[2]http://speclab.cr.usgs.gov/spectral.lib06

the each subimage is (200 × 20) pixels in size. We will use both spatial and random splitting, respectively. The simulated data will also be unmixed using $N = 1$ subimage, i.e., unmix the whole image in the classical way without splitting it.

In these simulations, the sparsity parameter is estimated using the EBIC, the number of endmembers is manually set to the correct value, $r = 5$. We manually set r to the correct value so we are able to evaluate the algorithm.

However, to show that EBIC does yield good results, we show in Fig. 4 the calculated EBIC using the simulated data generated as described in Subsection III-A. The number of endmembers was varied in the interval $r = 3, ..., 10$. EBIC correctly estimated the rank for this simulated data. In Fig. 5, EBIC and SAD plots obtained when estimating λ, when $r = 5$ are given.

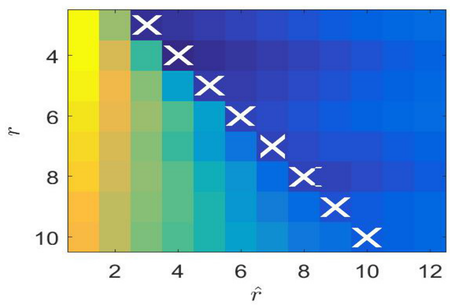


Fig. 4. A surface plot of EBIC, calculated for simulation data with $r = 3, ..., 10$. Each row shows EBIC calculated for $\hat{r} = 1, ..., 12$. The white 'x' shows the minima of EBIC, which is the rank estimated by the EBIC.

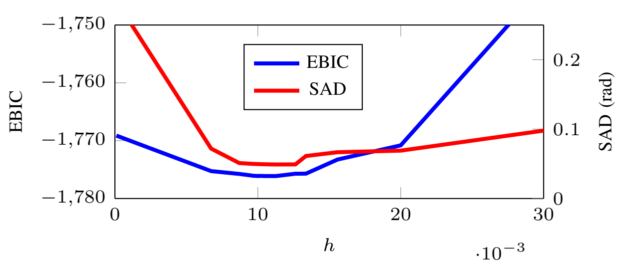


Fig. 5. The EBIC and SAD (rad), obtained when estimating h using simulated data.

Figures 4 and 5 displays accurate estimates of r. Also, the EBIC gives a very good estimate for h, which is very close to minimizing the SAD between the estimated endmembers and the true endmembers.

We will consider three test simulations. In the first two simulations, the data will not suffer from spectral variability and $\psi = 0$ and $\boldsymbol{M}_i = 0$. In the third simulation, the endmembers used will have spectral variability.

Simulation 1 All of the endmembers are found in all of the subimages.

Simulation 2 Two of the endmembers are only present in one subimage. So, these two endmembers are not present in the other subimages. The other three endmembers are present in all four subimages.

Simulation 3 All of the endmembers are found in all of the subimages and spectral variability is added to the different subimages.

For each simulation, 10 simulations are performed, new endmembers are randomly selected from the library, and new abundance maps are generated. The following metrics are used to evaluate the algorithm;

$$\text{nMSE}_{\boldsymbol{AS}} = \frac{\|\boldsymbol{S}\boldsymbol{A}^T - \widehat{\boldsymbol{S}}\widehat{\boldsymbol{A}}^T\|_F^2}{\|\boldsymbol{S}\boldsymbol{A}^T\|_F^2}, \tag{16}$$

$$\text{nMSE}_{\boldsymbol{S}} = \frac{\|\boldsymbol{S} - \widehat{\boldsymbol{S}}\|_F^2}{\|\boldsymbol{S}\|_F^2}, \tag{17}$$

$$\text{sad}(\widehat{\boldsymbol{a}}_{(j)}, \boldsymbol{a}_{(j)}) = \arccos\left(\frac{\widehat{\boldsymbol{a}}_{(j)}^T \boldsymbol{a}_{(j)}}{\|\widehat{\boldsymbol{a}}_{(j)}\|\|\boldsymbol{a}_{(j)}\|}\right), \tag{18}$$

$$\text{SAD}(\widehat{\boldsymbol{A}}, \boldsymbol{A}) = \frac{1}{r}\sum_{j=1}^{r} \text{sad}(\widehat{\boldsymbol{a}}_{(j)}, \boldsymbol{a}_{(j)}). \tag{19}$$

$\text{nMSE}_{\boldsymbol{AS}}$ measures the reconstruction error, and $\text{nMSE}_{\boldsymbol{S}}$ measures the reconstruction error of the abundance maps. The columns in $\widehat{\boldsymbol{A}}$ are scaled to have the same ℓ_2 norm as the columns of $\boldsymbol{A}$. The columns of $\widehat{\boldsymbol{S}}$ are also scaled so that the product, $\widehat{\boldsymbol{S}}\widehat{\boldsymbol{A}}^T$, is not changed, before calculating the $\text{nMSE}_{\boldsymbol{S}}$. The spectral angle distance (SAD) is given in (18) and calculates the angle between the two endmembers being considered and a low value means that the two endmembers being compared are similar and a high value means that the two endmembers are dissimilar. The SAD given in (19) calculates the average SAD for all the endmembers in a HSI.

The algorithms used to unmix the simulated data are:

- The proposed distributed algorithm (DMU).
- The non-distributed version of DMU, with $N = 1$ (DMU_1).
- Vertex Component analysis, an endmember estimation algorithm (VCA) [44].
- A statistical approach to identifying endmembers in hyperspectral images (ICE) [45].
- Minimum volume constrained nonnegative matrix factorization (MVC) [46].

VCA is a geometrical endmember estimation algorithm which assumes that there are pure pixels in the image. We include VCA in our comparisons despite the fact that there are not pure pixels in our simulations as it is a well known and used method.

A. Simulation 1

Here, all of the endmembers are found in every subimage. Table I shows the average SAD, along with one standard deviation between the estimated endmembers and the original endmembers. The HSI is split into subimages using the spatial method.

In Table I, we can see that the distributed unmixing and the classical unmixing results are very similar. The metrics calculated are virtually identical and show that both the endmembers and abundances are very similar to the original data. The sparsity (% of zero values) of the abundance matrix is also shown for the proposed method and for MVC, along with

TABLE I
THE METRICS CALCULATED USING THE RESULTS OF SIMULATION 1, WHERE ALL OF THE ENDMEMBERS ARE FOUND IN ALL OF THE SUBIMAGES

Endm. #	average SAD(rad) ± 1 std DMU	DMU_1	VCA	ICE	MVC
1	0.005 ± 0.003	0.005 ± 0.003	0.051 ± 0.029	0.023 ± 0.022	0.018 ± 0.008
2	0.018 ± 0.020	0.018 ± 0.020	0.072 ± 0.056	0.058 ± 0.057	0.021 ± 0.008
3	0.009 ± 0.010	0.009 ± 0.010	0.062 ± 0.023	0.037 ± 0.041	0.033 ± 0.031
4	0.021 ± 0.036	0.021 ± 0.036	0.073 ± 0.059	0.048 ± 0.074	0.025 ± 0.029
5	0.032 ± 0.053	0.032 ± 0.053	0.117 ± 0.099	0.137 ± 0.210	0.032 ± 0.034
average	0.017	0.017	0.075	0.061	0.026
$\text{nMSE}_{\boldsymbol{AS}}$(dB)	-49.93 ± 0.27	-49.93 ± 0.27	N/A	N/A	-51.567 ± 0.06
$\text{nMSE}_{\boldsymbol{S}}$(dB)	-28.42 ± 3.58	-28.41 ± 3.59	N/A	N/A	-20.987 ± 3.74
Sparsity of $\boldsymbol{S}$	18%	18%	N/A	N/A	3%
Average Computation time	196s	71s	0.16s	63s	168s

TABLE II
THE METRICS CALCULATED USING THE RESULTS OF SIMULATION 2, WHERE TWO OF THE ENDMEMBERS ARE ONLY PRESENT IN ONE SUBIMAGE EACH

Endm. #	average SAD(rad) ± 1 std DMU(a)	DMU(b)	DMU_1	VCA	ICE	MVC
1	0.162 ± 0.102	0.047 ± 0.043	0.061 ± 0.052	0.090 ± 0.092	0.090 ± 0.099	0.075 ± 0.072
2	0.108 ± 0.122	0.023 ± 0.022	0.027 ± 0.020	0.059 ± 0.042	0.056 ± 0.067	0.049 ± 0.034
3	0.052 ± 0.034	0.028 ± 0.038	0.030 ± 0.025	0.077 ± 0.078	0.025 ± 0.016	0.090 ± 0.067
4	0.112 ± 0.125	0.056 ± 0.091	0.046 ± 0.041	0.084 ± 0.079	0.095 ± 0.132	0.056 ± 0.052
5	0.143 ± 0.124	0.036 ± 0.030	0.087 ± 0.102	0.153 ± 0.132	0.093 ± 0.108	0.136 ± 0.142
average	0.115	0.038	0.050	0.092	0.072	0.081
$\text{nMSE}_{\boldsymbol{AS}}$(dB)	-41.61 ± 4.54	-49.23 ± 0.83	-47.87 ± 2.39	N/A	N/A	-51.562 ± 0.10
$\text{nMSE}_{\boldsymbol{S}}$(dB)	-3.14 ± 9.09	-11.25 ± 13.48	-7.70 ± 14.88	N/A	N/A	-7.805 ± 9.42

the average computational time needed. The proposed method is more computationally intensive than the other methods. Note that the parallel calculations for DMU are simulated as described in subsection III-D. In Fig. 6, the SAD is shown as a function of the signal to noise ratio (SNR) of the simulation data. The proposed method does very well compared to the other when the SNR is larger than 26dB, but ICE achieves a lower SAD when the noise is less than 26dB.

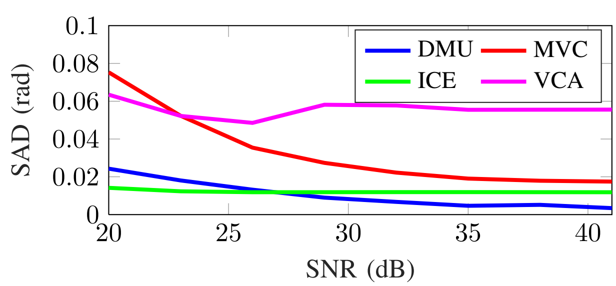


Fig. 6. The SAD, calculated when the signal to noise ration (SNR) of the simulation data varies from 20dB to 40dB.

B. Simulation 2

In this simulation, we will investigate how the distributed algorithm performs when some endmembers are not present in all subimages. The simulation is split into two parts. In the first simulation (denoted by a), endmembers 1 and 5 are only present in subimages 1 and 4, respectively. The presence of endmembers in each of the subimages is shown in Table III. The HSI is split into subimages using the spatial method.

In the second part (denoted by b), the same data is used as in the first part, but here the random splitting method was used.

TABLE III
THE PRESENCE OF ENDMEMBERS IN THE SUBIMAGES. IN SIMULATION (A), ENDMEMBERS 1 AND 5 ARE ONLY PRESENT IN SUBIMAGES 1 AND 4, RESPECTIVELY

Simu-lation	Subimage #	Endmember # 1	2	3	4	5
(a)	1	x	x	x		
	2		x	x	x	
	3		x	x	x	
	4			x	x	x

splitting, the probability of every endmember being present in every subimage is greatly increased.

In Table II, we can see that the results of the distributed unmixing are not as good as the results obtained using the non-distributed approach when the spatial splitting is used, and the results are in fact worse than the methods we compare to. The SAD is high and the variance is also quite high. This is also reflected in the higher values of the reconstruction metrics, $\text{nMSE}_{\boldsymbol{AS}}$ and $\text{nMSE}_{\boldsymbol{S}}$. The proposed method (with spatial splitting) has trouble coping with this data because the number of endmembers in the subimages is not the same as the number of estimated endmembers. However, when the random splitting is used, the distributed approach works as well as the non-distributed approach.

C. Simulation 3

Here, the endmembers will be present in all subimages, but the endmembers in each image will have spectral variability. The variability is added to each endmembers so that the endmembers in each subproblem will not be identical. In Fig. 7, the variability vectors, along with two endmembers in one of the simulations are shown. The endmembers for one subimage are composed of a consensus endmember from the library, and variability is introduced by adding one of the variability vectors shown in Fig. 7 to the endmembers.

In this simulation, VCA, ICE and MVC are used to unmix each subimage separately and the average metrics are calculated and compared to the proposed algorithm, and given in Table IV. In addition to evaluation of the perturbed endmembers ($\boldsymbol{Z}+\boldsymbol{M}_i$) for each subproblem, we will also examine the consensus endmembers ($\boldsymbol{Z}$).

1) DMU($\boldsymbol{Z}$): Here we are only evaluating the non-variable part of the endmembers without perturbations.
2) DMU($\boldsymbol{Z}+\boldsymbol{M}_i$): $\boldsymbol{M}_i$ is the perturbation matrix. Therefore, $\boldsymbol{Z}+\boldsymbol{M}_i$ are the endmembers found for subproblem i.

In Table IV, we can see that the algorithm is able to estimate the perturbed endmembers very well. The common endmembers are similar to the real endmembers, but with the addition of the perturbation matrix ($\boldsymbol{M}_i$), the estimation is greatly improved. Thus, the majority of the endmember morphology is captured by $\boldsymbol{Z}$ and the variations are captured by $\boldsymbol{M}_i$. This is shown in Fig. 8

D. Computations

In this section, we will examine the running time of DMU, which is programmed in Matlab. The data needed for each subproblem is saved in a Matlab (.mat) file. Each individual subproblem reads all the data it needs from the mat file and saves the result in mat-file. The combination part of the algorithm reads all the individual results files and combines the results. The parallel calculations are simulated, meaning that the computational time for the individual subproblems are assumed to be equal to the longest computational time of individual subproblems.

We unmix simulated data using $N \in \{1,...,128\}$. The setting is the same as in Simulation 1, but the number of pixels is $P = 90000$. The sparsity parameter is estimated to be $h = 0.0089$, and this value is used in all the simulations.

As can be seen in Fig. 9, if the subproblems are solved in parallel, the distributed algorithm can be faster than the non distributed one ($N = 1$), given that the number of subproblems is not very low, i.e., when $N > 30$. If the number of subproblems is small, the computation times may increase because of the extra computations needed to solve the distributed problem. This is apparent in Fig. 9 when $2 \leq N \leq 30$.

When a hyperspectral image is split into N subimages, the data each subproblem needs to have access to is the subimage, $\boldsymbol{Y}_i$ of size $P \times M$, and its corresponding endmember and abundance data, i.e., $\boldsymbol{A}_i$ of size $M \times r$ and $\boldsymbol{S}_i$ of size $P \times r$. The overhead data that is needed by each subproblem are three matrices: $\boldsymbol{\Lambda}_i$, $\boldsymbol{Z}$, and $\boldsymbol{M}_i$, each of size $M \times$r. Given that for a typical hyperspectral image, $P_i \gg M$ and $M \gg r$, and $P_i = P/N$, the memory requirements for each subproblem is reduced by a factor of N, compared to the memory needed to process the whole image.

IV. Real Hyperspectral Images

A. The HYDICE Urban Image

The first real HSI used here is a HYDICE image[3] of an urban landscape. The number of spectral bands in this data set is 210 and covers the 400-2500nm spectral range.

This image is 307×307 pixels and the whole image is used. Spectral bands numbered [74-77 86-90 102-111 136-153 202-210] are manually identified as water absorption or noisy bands, and are removed, resulting in 164 usable bands. Matrix $\boldsymbol{Y}$ is thus of dimensions $307^2 \times 164$. An RGB image, generated using the hyperspectral data, is shown in Fig. 11. The RGB image is created by using specific spectral bands from the data set, to represent the red, green, and blue channels of the RGB image. The HSI image is split into $N = 4$ matrices using random splitting. Using EBIC, the number of estimated number are estimated $r = 8$, and the sparsity parameter is $h = 0.002$ (resulting in 13% sparsity in the abundance maps).

The Urban image will also be unmixed without splitting the image (using the proposed algorithm), and with MVC, respectively. Three out of eight abundance maps estimated by the algorithms are shown in Fig. 11. The results obtained by the DMU and the DMU$_1$ algorithm are very similar. The abundance maps shown are easily associated with material seen in the RGB image. The three maps represent trees, grassy areas, and rooftops, respectively.

B. The AVIRIS Cuprite Image

The second real world hyperspectral image we will consider is the Aviris Cuprite image[4] which has 224 spectral bands covering 410-2450nm. The image is 350×350 pixels in size. We discard spectral bands [1 2 105-115 150-170 222-224] and use the remaining 187. An RGB generated image of the scene in shown in Fig. 10.

We unmix the image using four values for N, respectively, i.e., $N = \{1, 8, 16, 32\}$. Using EBIC, the sparsity parameter was estimated to be $h = 0.0022$ (resulting in approximately 10% sparsity), and the number of endmembers was $r = 11$.

A ground truth for this image is not available, but the image has been used extensively by researchers and there are many well documented minerals exposed in the landscape. These minerals are known to be in the USGS spectral library. We compare the endmembers obtained, to the spectral signatures in this library. The spectral signatures in the library that have the lowest SAD from the estimated endmenbers are found and given in Table V. The SAD values in Table V can not be viewed as quality metrics, since a unmixing ground truth is not available for this data set. Here, the SAD only shows how similar the different solutions are. Both the endmembers and the abundances estimated using different values for N were very similar. Many of the endmembers that had the lowest SAD from the estimated endmembers are known to be in the area. The reconstruction error,

$$\mathrm{ERR} = \frac{\|\boldsymbol{Y} - \widehat{\boldsymbol{S}}\widehat{\boldsymbol{A}}^T\|_F^2}{\|\boldsymbol{Y}\|_F^2}, \tag{20}$$

[3]http://www.agc.army.mil/hypercube/

[4]http://aviris.jpl.nasa.gov/data/free_data.html

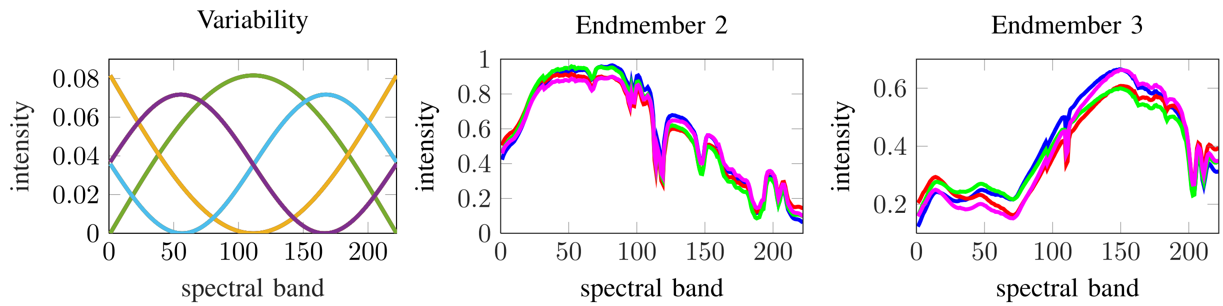


Fig. 7. Left: The variability added to the subimage endmembers. Center and right: The 4 different variations used.

TABLE IV
THE METRICS CALCULATED USING THE RESULTS OF SIMULATION 3, WHERE VARIABILITY IS ADDED TO THE ENDMEMBERS

	average SAD(rad) ± 1 std				
Endm. #	DMU($\boldsymbol{Z}+\boldsymbol{M}_i$)	DMU($\boldsymbol{Z}$)	VCA	ICE	MVC
1	0.006 ± 0.003	0.046 ± 0.030	0.060 ± 0.054	0.031 ± 0.038	0.033 ± 0.030
2	0.007 ± 0.005	0.034 ± 0.013	0.041 ± 0.034	0.053 ± 0.056	0.016 ± 0.008
3	0.009 ± 0.012	0.041 ± 0.032	0.045 ± 0.028	0.043 ± 0.056	0.030 ± 0.021
4	0.030 ± 0.062	0.074 ± 0.105	0.089 ± 0.096	0.090 ± 0.121	0.048 ± 0.072
5	0.007 ± 0.005	0.044 ± 0.027	0.046 ± 0.036	0.033 ± 0.041	0.031 ± 0.027
average	0.012	0.048	0.056	0.050	0.032
nMSE$_{\boldsymbol{AS}}$(dB)	-51.52 ± 0.34	N/A	N/A	N/A	-37.745 ± 4.10
nMSE$_{\boldsymbol{S}}$(dB)	-20.36 ± 6.56	N/A	N/A	N/A	-15.844 ± 3.39

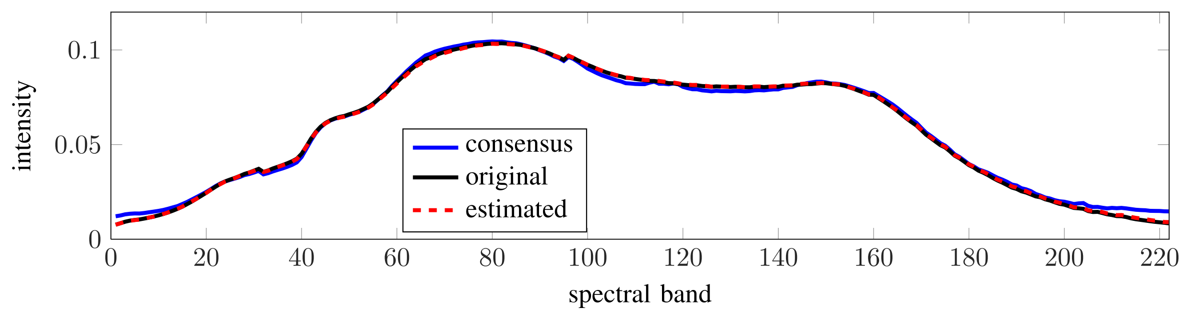


Fig. 8. One of the estimated endmembers in Simulation 3. The consensus endmember ($\boldsymbol{z}_{(j)}$) is similar to the original, but the estimated endmember ($\boldsymbol{z}_{(j)} + \boldsymbol{m}_{i(j)}$) is much closer to the original.

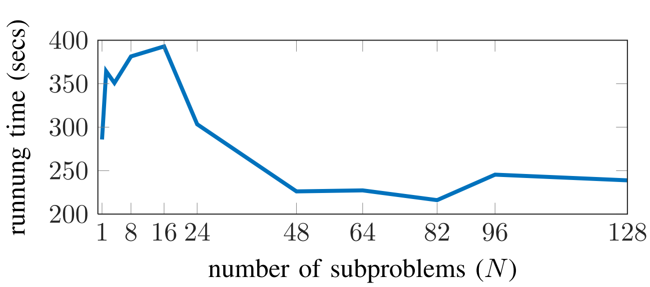


Fig. 9. The computational time as a function of the number of subproblems.

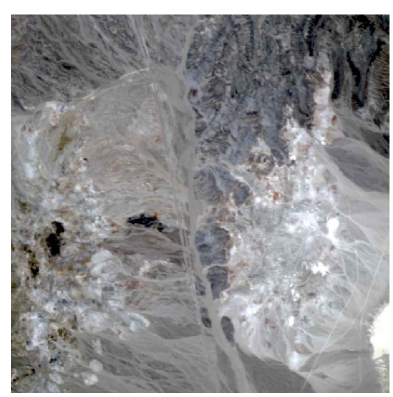


Fig. 10. An RGB generated image of the Cuprite data set.

is also calculated and given in Table V.

In Fig. 12, three out of eleven endmembers and abundance maps are shown. All the DMU solutions and the ERR, for different values of N are very similar.

C. The AVIRIS Lake Tahoe Images

In this subsection multiple AVIRIS HSIs from the Lake Tahoe region[5] are unmixed. We will focus on a small 200×150

[5] Available at http://aviris.jpl.nasa.gov/alt_locator/

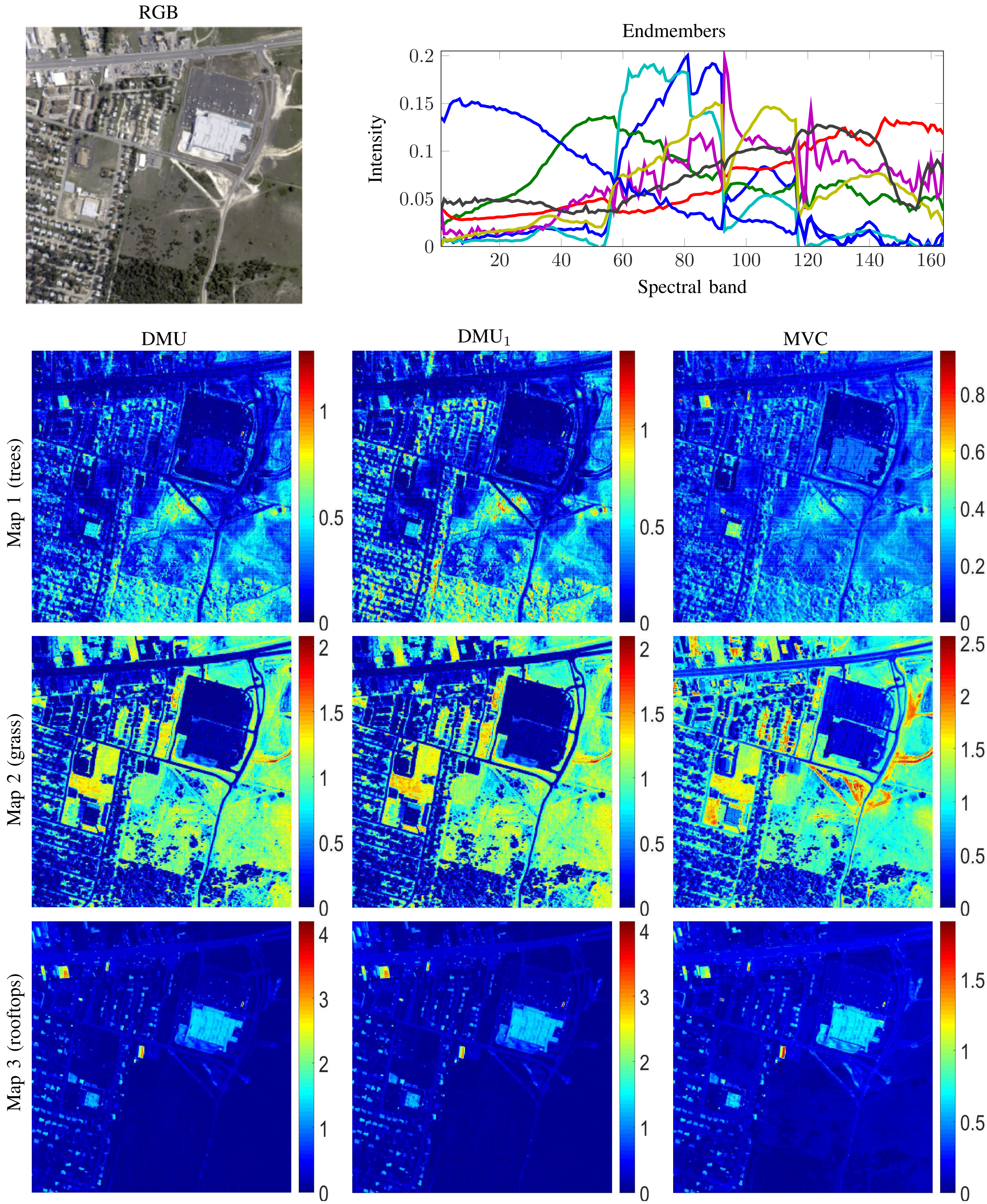


Fig. 11. In the top row, an RGB image of the Urban dataset is show along with the endmembers estimated by the DMU algorithm. In the other rows, three out of eight abundance maps are shown, estimated by the DMU, DMU$_1$ and MVC algorithms, respectively.

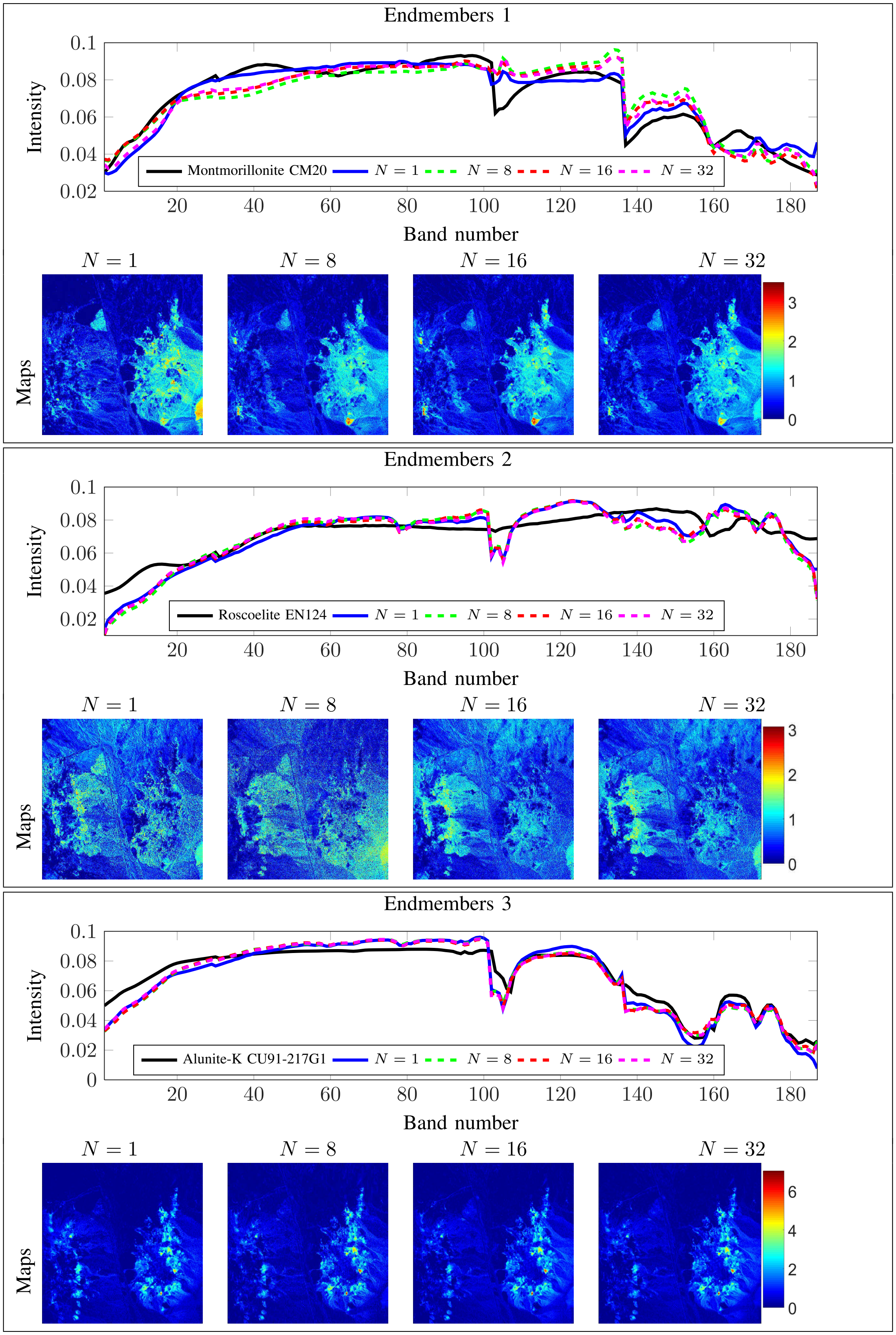


Fig. 12. Three of eleven endmembers and corresponding abundance maps estimated by DMU for $N = \{1, 8, 16, 32\}$.

pixel region surrounding a small lake named Mud Lake[6]. These HSIs are referred to as *Mud Lake*. The HSIs have 244

[6]Latitude: 38.84197^o, longitude: -119.7356347^o

TABLE V
THE SPECTRAL ANGLE DISTANCE BETWEEN LIBRARY SIGNATURES AND THE ENDMEMBERS OBTAINED.

Endmember # / Mineral	SAD			
	$N=1$	$N=8$	$N=16$	$N=32$
1. Montmorillonite CM20	0.078	0.120	0.097	0.098
2. Roscoelite EN124	0.110	0.133	0.122	0.124
3. Alunite-K	0.093	0.093	0.092	0.090
4. Hematitic_Alt._Tuff	0.107	0.120	0.117	0.116
5. Fluorapatite WS416	0.060	0.058	0.052	0.057
6. Illite IL105	0.113	0.096	0.107	0.103
7. Jarosite WS368	0.090	0.111	0.107	0.101
8. Hematite_Thin_Film	0.073	0.082	0.108	0.127
9. Chlorite_Serpentine	0.105	0.080	0.081	0.087
10. Sagebrush YNP-SS-2	0.084	0.120	0.115	0.112
11. Alun_Na+Kaol+Hemat	0.148	0.149	0.148	0.139
Average SAD	0.0966	0.1056	0.1041	0.1050
ERR	6.3e-05	6.3e-05	5.9e-05	5.9e-05

spectral bands. We remove water absorption bands, leaving 179 spectral bands.

Five images are used in this evaluation. These images and their acquisition dates are shown in Fig. 13. Eleven pixels from image (a) and 14 pixels from image (e) are removed from the data. These are outlier pixels, some of them having overflow errors.

The results obtained by the proposed algorithm are compared to the results obtained when unmixing the images with MVC. To acquire consistent results using MVC, we initialize the algorithm, using the endmembers obtained using VCA [44] on the first image (April 10, 2014). MVC is thus initialized identically for all the HSIs. The proposed algorithm is initialized with random values.

The number of subproblems is equal to the number of images, i.e., $N = 5$. Using a majority vote, the number of endmembers used is $r = 11$. The sparsity parameters are estimated to be h_i={5.1e-4, 1.6e-3, 1e-3, 1e-3, 5e-4}, resulting in an average sparsity of 17% for the abundance maps. A ground truth for this image is not available so we will evaluate the algorithm by examining the consistency of the endmembers and we will also calculate the average reconstruction error,

$$\text{avERR} = \frac{1}{N}\sum_{i=1}^{N}\frac{\|\boldsymbol{Y}_i - \widehat{\boldsymbol{S}}_i\widehat{\boldsymbol{A}}_i^T\|_F^2}{\|\boldsymbol{Y}_i\|_F^2}, \tag{21}$$

for the proposed algorithm, and also for MVC.

In Fig. 13, two abundance maps estimated using DMU and MVC are shown. The abundance maps have similarities with the RGB image, one map is associated with the circular agricultural region in the upper left corner of the image while the other is associated with the water region in the lower part of the image. We will refrain from drawing more conclusion from the abundance maps since an accurate ground truth is not available.

In Fig. 14, the endmembers corresponding to the abundance maps in Fig. 13 are shown. The endmembers estimated by DMU are more consistent and have less variations than the endmembers estimated by MVC. The average reconstruction errors, calculated using (21) and shown in Table VI, and are also very similar, albeit the MVC reconstruction error is slightly lower.

TABLE VI
THE AVERAGE RECONSTRUCTION ERRORS CALCULATED FOR THE MUD LAKE DATASET.

	DMU	MVC
avERR	4.30e-05	4.13e-05

V. Conclusions

In this paper, a sparse distributed hyperspectral unmixing algorithm is developed using ADMM and an ℓ_1 regularizer. The model parameters are estimated using the extended Bayesian information criteria. The hyperspectral image is split into N subproblems, and each subproblem is independently solved. The unmixing results from these subproblems is then merged into a global solution using ADMM. As each subproblem does not need to have access to the whole dataset, the algorithm can be applied on very large datasets. The algorithm is able to account for spectral variability between the different subproblems and is thus well suited to unmix multitemporal hyperspectral data. The algorithm is extensively evaluated using simulation data and compared to other well established unmixing algorithms. Using this simulated data, the algorithm performs very well and is able to achieve comparable results to non-distributed algorithms. An evaluation is also performed using three real hyperspectral images, one of which is a multitemporal image, acquired on five different dates. The algorithm performs well on all the real hyperspectral images, and the results of the multitemporal unmixing gives more consistent endmembers than independently unmixing each temporal HSI.

Appendix
Estimation Methods

In this appendix, we detail the estimation methods used to solve the minimization problem in Algorithm 1. A cyclic descent method is used which iteratively estimates the variables interest. In the following subsections, the algorithm is derived.

A. $\boldsymbol{S}$-step

Estimating $\boldsymbol{S}_i$ is done by minimizing

$$\boldsymbol{S}_i^{k+1} = \underset{\boldsymbol{S}_i \in \mathbb{R}_+^{P_i\times r}}{\arg\min}\,\frac{1}{2}\|\boldsymbol{Y}_i - \boldsymbol{S}_i(\boldsymbol{A}_i^k + \boldsymbol{M}_i^k)^T\|_F^2 + h\|\boldsymbol{S}_i\|_{1,1} \tag{22}$$

The task of estimating one column of $\boldsymbol{S}_i$ is done by minimizing

$$\boldsymbol{s}_{i(j)}^{k+1} = \underset{\boldsymbol{s}_{i(j)} \in \mathbb{R}_+^{P_i\times 1}}{\arg\min}\,\frac{1}{2}\|\boldsymbol{R}_{ij} - \boldsymbol{s}_{i(j)}(\boldsymbol{a}_{i(j)}^k + \boldsymbol{m}_{i(j)}^k)^T\|_F^2 + \|\boldsymbol{s}_{i(j)}\|_1 \tag{23}$$

where $\boldsymbol{R}_{ij} = \boldsymbol{Y}_i - \boldsymbol{S}_{i\text{-}j}(\boldsymbol{A}_{i\text{-}j}^k + \boldsymbol{M}_{i\text{-}j}^k)^T$. The update rule for $\boldsymbol{s}_{i(j)}^{k+1}$ is nonnegative soft thresholding [47], i.e.,

$$\boldsymbol{s}_{i(j)}^{k+1} = \alpha\max(0, \boldsymbol{R}_{ij}(\boldsymbol{a}_{i(j)}^k + \boldsymbol{m}_{i(j)}^k) - h), \tag{24}$$

where $\alpha = 1/\|\boldsymbol{a}_{i(j)}^k + \boldsymbol{m}_{i(j)}^k\|^2$. When the problem is multitemporal unmixing, and $\boldsymbol{M}_i$ needs to be estimated, we add a debiasing step which attempts to remove the bias caused by

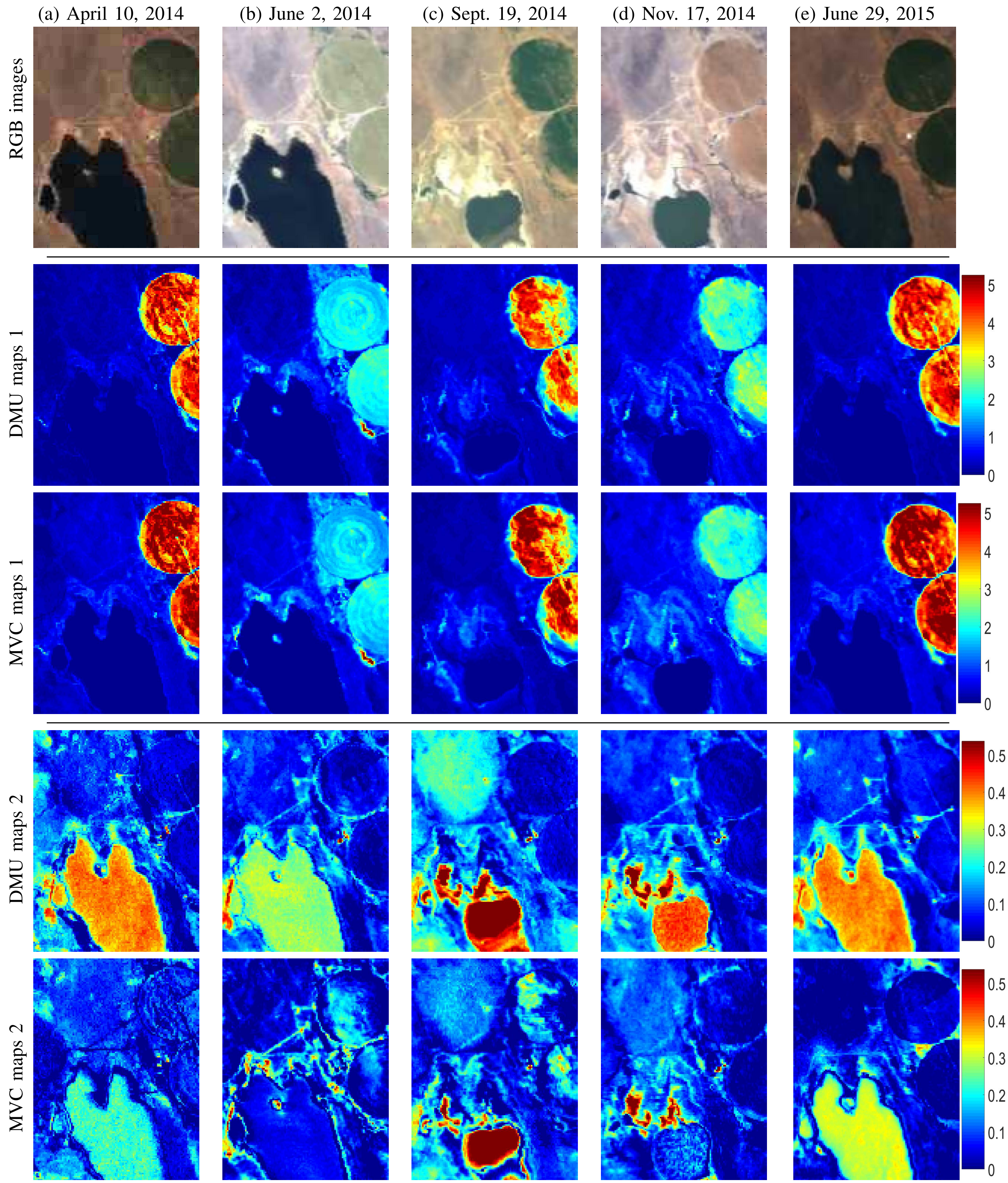


Fig. 13. In top top row are are the generated RGB images of the five Mud Lake HSIs used in the evaluation. In the second and fourth rows are two out of 12 abundance maps estimated by the DMU algorithm and in rows three and five are the corresponding abundance maps estimated by MVC.

the soft thresholding done in (24). The debiasing step involves adding back αh to the non-zero values of $\boldsymbol{s}_{i(j)}^{k+1}$, i.e.,

$$\tilde{\boldsymbol{s}}_{i(j)}^{k+1} = (\boldsymbol{s}_{i(j)}^{k+1} + \alpha h) * I(\boldsymbol{s}_{i(j)}^{k+1}) \tag{25}$$

where $*$ denotes elementwise multiplication and $I(\cdot)$ is the identity operator. We stress that this debiasing step is only applied in the multitemporal case, when $\boldsymbol{M}_i$ needs to be estimated, and it is otherwise not applied.

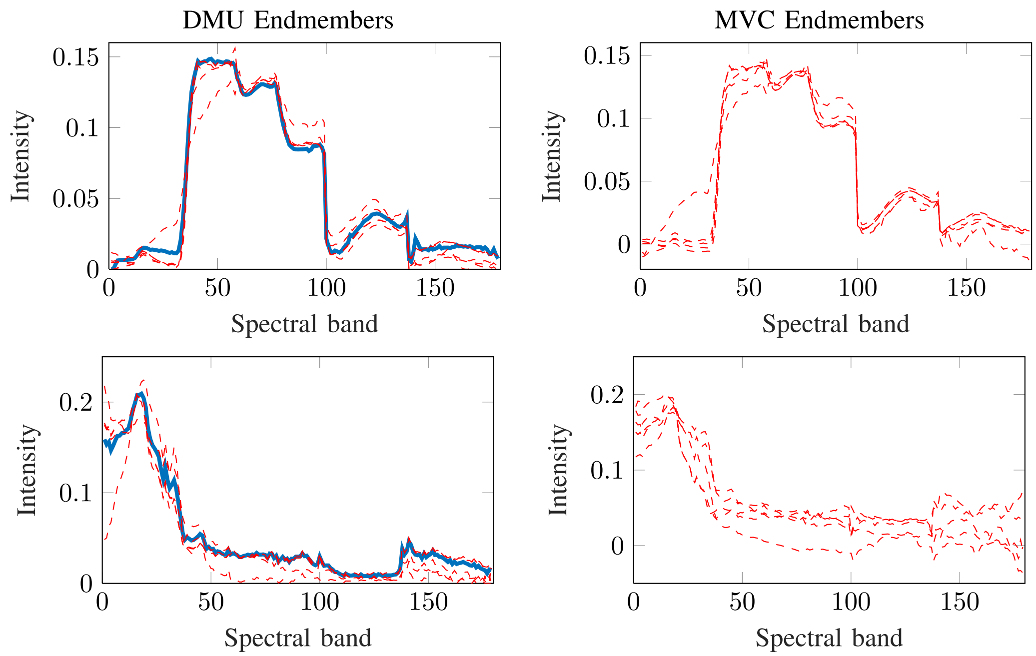


Fig. 14. Two endmembers estimated by DMU and MVC for all five Mud Lake HSIs. In the DMU plots, the blue line shows the consensus endmembers and the red line show each of the estimated endmembers.

B. $\boldsymbol{A}$-step

The minimization task when estimating $\boldsymbol{A}_i$ is

$$\begin{aligned}\boldsymbol{A}_i^{k+1} =& \arg\min_{\boldsymbol{A}\in\boldsymbol{\Gamma}^{M\times r}} \frac{1}{2}\|\boldsymbol{Y}_i - \boldsymbol{S}_i^{k+1}(\boldsymbol{A}_i+\boldsymbol{M}_i^k)^T\|_F^2 \\ &+ \mathrm{tr}\big((\boldsymbol{\Lambda}_i^k)^T(\boldsymbol{A}_i - \boldsymbol{Z}^k)\big) + \frac{\rho(k)}{2}\|\boldsymbol{A}_i - \boldsymbol{Z}^k\|_F^2\end{aligned} \tag{26}$$

Estimating one column in $\boldsymbol{A}_i$ is done by optimizing

$$\boldsymbol{a}_{i(j)}^{k+1} = \arg\min_{\boldsymbol{a}_{(j)}\in\boldsymbol{\Gamma}^{M\times 1}} f_a(\boldsymbol{a}_{i(j)})$$

where

$$f_a(\boldsymbol{a}_{i(j)}) = \frac{1}{2}\|\boldsymbol{R}_{ij} - \boldsymbol{s}_{(j)}^{k+1}(\boldsymbol{a}_{i(j)} + \boldsymbol{m}_{i(j)}^k)^T\|_F^2 + (\boldsymbol{\lambda}_{i(j)}^k)^T(\boldsymbol{a}_{i(j)} - \boldsymbol{z}_{(j)}^k) + \frac{\rho(k)}{2}(\boldsymbol{a}_{i(j)} - \boldsymbol{z}_{(j)}^k)(\boldsymbol{a}_{i(j)} - \boldsymbol{z}_{(j)}^k)^T,$$

and $\boldsymbol{R}_{ij} = \boldsymbol{Y}_i - \boldsymbol{S}_{i\text{-}j}^{k+1}(\boldsymbol{A}_{i\text{-}j} + \boldsymbol{M}_{i\text{-}j}^k)^T$. To solve this minimization problem, we use Lagrangian multiplier theory. The Lagrange function is chosen as

$$L_a(\boldsymbol{a},\gamma) = f_a(\boldsymbol{a}) + \gamma(\boldsymbol{a}^T\boldsymbol{a} - 1)$$

where γ is the Lagrange multiplier. Finding the differential of $L_a(\boldsymbol{a},\gamma)$ w.r.t. $\boldsymbol{a}$, and setting it to zero yields (omitting sub- and superscripts)

$$\frac{dL_a}{d\boldsymbol{a}} = \boldsymbol{0} \Rightarrow \boldsymbol{a} = \frac{(\boldsymbol{R}^T - \boldsymbol{m}\boldsymbol{s}^T)\boldsymbol{s} - \boldsymbol{\lambda} + \rho(k)\boldsymbol{z}}{\boldsymbol{s}^T\boldsymbol{s} + \rho(k) + 2\gamma}. \tag{27}$$

By defining

$$\tilde{\boldsymbol{a}} = \max\Big(\big(\boldsymbol{R}_{ij}^T - \boldsymbol{m}_{i(j)}^k(\boldsymbol{s}_{i(j)}^{k+1})^T\big)\boldsymbol{s}_{i(j)}^{k+1} - \boldsymbol{\lambda}_{i(j)}^k + \rho(k)\boldsymbol{z}_{(j)}^k, \boldsymbol{0}\Big),$$

and choosing γ such that $\boldsymbol{a} =\in \boldsymbol{\Gamma}^{M\times 1}$, yields the update rule for $\boldsymbol{a}_{i(j)}^{k+1}$,

$$\boldsymbol{a}_{i(j)}^{k+1} = \frac{\tilde{\boldsymbol{a}}}{\|\tilde{\boldsymbol{a}}\|}. \tag{28}$$

C. $\boldsymbol{M}$-step

The minimization task when estimating $\boldsymbol{M}_i$ is

$$\begin{aligned}\boldsymbol{M}_i^{k+1} = & \arg\min_{(\boldsymbol{A}_i+\boldsymbol{M}_i)\in\mathbb{R}_+^{M\times r}} \frac{1}{2}\|\boldsymbol{Y}_i - \boldsymbol{S}_i^{k+1}(\boldsymbol{A}_i^{k+1} + \boldsymbol{M}_i)^T\|_F^2 \\ &+ \frac{\psi}{2}\|\boldsymbol{M}_i\|_F^2\end{aligned} \tag{29}$$

Using $\boldsymbol{R}_{ij} = \boldsymbol{Y}_i - \boldsymbol{S}_{i\text{-}j}^{k+1}(\boldsymbol{A}_{i\text{-}j}^{k+1} + \boldsymbol{M}_{i\text{-}j})^T$, the task of estimating one column in $\boldsymbol{M}_i$ becomes

$$\begin{aligned}&\boldsymbol{m}_{i(j)} = \arg\min f_m(\boldsymbol{m}_{i(j)}) \\ &\text{s.t. } (\boldsymbol{a}_{i(j)} + \boldsymbol{m}_{i(j)}) \in \mathbb{R}_+^{M\times 1}\end{aligned} \tag{30}$$

where

$$f_m(\boldsymbol{m}_{i(j)}) = \frac{1}{2}\|\boldsymbol{R}_{ij} - \boldsymbol{s}_{i(j)}^{k+1}(\boldsymbol{a}_{i(j)}^{k+1} + \boldsymbol{m}_{i(j)})^T\|_F^2 + \frac{\psi}{2}\|\boldsymbol{m}_{i(j)}\|_F^2.$$

Finding the differential of $f_m(\boldsymbol{m}_{i(j)})$ w.r.t. $\boldsymbol{m}$, and setting it to zero yields

$$\frac{df_m}{d\boldsymbol{m}} = \boldsymbol{0} \Rightarrow \boldsymbol{m}_{i(j)}^{k+1} = \frac{(\boldsymbol{R}_{ij}^T - \boldsymbol{a}_{i(j)}^{k+1}(\boldsymbol{s}_{i(j)}^{k+1})^T)\boldsymbol{s}_{i(j)}^{k+1}}{(\boldsymbol{s}_{i(j)}^{k+1})^T\boldsymbol{s}_{i(j)}^{k+1} + \psi}. \tag{31}$$

To account for the nonnegativity of $(\boldsymbol{a}_{i(j)}^{k+1} + \boldsymbol{m}_{i(j)}^{k+1})$, we threshold $\boldsymbol{m}_{i(j)}^{k+1}$ such that $(\boldsymbol{m}_{i(j)}^{k+1} + \boldsymbol{a}_{i(j)}^{k+1}) \geq 0$.

D. $\boldsymbol{Z}$-step

Estimating $\boldsymbol{Z}$ is done by minimizing

$$\begin{aligned}\boldsymbol{Z}^{k+1} = \operatorname*{argmin}_{\boldsymbol{Z}\in\boldsymbol{\Gamma}^{M\times r}} \sum_{i=1}^{N} \Big\{&\mathrm{tr}(\boldsymbol{\Lambda}_i^k)^T(\boldsymbol{A}_i^{k+1} - \boldsymbol{Z}) \\ &+ \frac{\rho(k)}{2}\|\boldsymbol{A}_i^{k+1} - \boldsymbol{Z}\|_F^2\Big\},\end{aligned} \tag{32}$$

and the update rule for $\boldsymbol{Z}$ is

$$\tilde{\boldsymbol{Z}} = \max\left(\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{A}_i^{k+1} + \frac{1}{\rho(k)}\boldsymbol{\Lambda}_i^k), 0\right),$$
$$\boldsymbol{z}_{(j)}^{k+1} = \frac{\tilde{\boldsymbol{z}}_{(j)}}{\|\tilde{\boldsymbol{z}}_{(j)}\|},\ j = 1,...,r. \tag{33}$$

E. DCD algorithm

A cyclic descent algorithm is used to estimate $\boldsymbol{A}_i$, $\boldsymbol{S}_i$ and $\boldsymbol{M}_i$ in the DMU algorithm. The algorithm is iterative and estimates one column in one of the matrices while holding all other columns fixed.

Algorithm 2: The DCD algorithm used to estimate $\boldsymbol{A}_i$, $\boldsymbol{S}_i$ and $\boldsymbol{M}_i$.

Input: $\boldsymbol{Y}_i$, $\boldsymbol{A}_i$, $\boldsymbol{S}_i$, $\boldsymbol{M}_i$, ρ, h
for $k = 0...$ **do**
  for $j = 1...r$ **do**
    Estimate column j in $\boldsymbol{S}_i$ using (24)
    Estimate column j in $\boldsymbol{A}_i$ using (28)
    Estimate column j in $\boldsymbol{M}_i$ using (31)
Output: $\widehat{\boldsymbol{S}}_i$, $\widehat{\boldsymbol{A}}_i$, $\widehat{\mathrm{M}}_i$

References

[1] K. Rapantzikos and C. Balas, "Hyperspectral imaging: potential in non-destructive analysis of palimpsests," *IEEE International Conference on Image Processing*, vol. 2, pp. II–618–21, Sept 2005.

[2] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354 –379, April 2012.

[3] N. Keshava and J. Mustard, "Spectral unmixing," *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 44 –57, Jan. 2002.

[4] M. Afonso, J. Bioucas-Dias, and M. Figueiredo., "An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 681–695, March 2011.

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[6] S. Alliney and S. Ruzinsky, "An algorithm for the minimization of mixed l1 and l2 norms with application to bayesian estimation," *IEEE Transactions on Signal Processing*, vol. 42, no. 3, pp. 618–627, Mar 1994.

[7] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society (Series B)*, vol. 58, pp. 267–288, 1996.

[8] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Total variation spatial regularization for sparse hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 11, pp. 4484 –4502, Nov. 2012.

[9] J. Sigurdsson, M. Ulfarsson, and J. R. Sveinsson, "Hyperspectral unmixing with l_q regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 6793–6806, Nov 2014.

[10] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 6, pp. 2014 –2039, June 2011.

[11] J. Sigurdsson, M. Ulfarsson, and J. R. Sveinsson, "Endmember constrained semi-supervised hyperspectral unmixing," in *Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, Jul 2014.

[12] M. Ulfarsson, V. Solo, J. Sigurdsson, and J. R. Sveinsson, "Distributed dyadic cyclic descent for non-negative matrix factorization," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[13] J. Sigurdsson, M. Ulfarsson, and J. R. Sveinsson, "Blind hyperspectral unmixing using total variation and ℓ_q sparse regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 11, pp. 6371–6384, Nov 2016.

[14] D. Lee and S. Seung, "Algorithms for Non-negative Matrix Factorization," *NIPS*, pp. 556–562, 2000.

[15] A. Zymnis, S. Kim, J. Skaf, M. Parente, and S. Boyd, "Hyperspectral image unmixing via alternating projected subgradients," in *Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers, 2007. ACSSC 2007*, 2007, pp. 1164–1168.

[16] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 6, pp. 2014–2039, 2011.

[17] J. Sigurdsson, M. Ulfarsson, J. R. Sveinsson, and J. A. Benediktsson, "A smooth hyperspectral unmixing method using cyclic descent," in *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2012, pp. 3082–3086.

[18] J. Sigurdsson, M. Ulfarsson, and J. R. Sveinsson, "Smooth and sparse hyperspectral unmixing using an l_0 penalty," in *Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, June 2013.

[19] W.-K. Ma, J. Bioucas-Dias, J. Chanussot, and P. Gader, "Signal and image processing in hyperspectral remote sensing [from the guest editors]," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 22–23, January 2014.

[20] S. Robila and D. Ricart, "Distributed algorithms for unmixing hyperspectral data using nonnegative matrix factorization with sparsity constraints," *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 2156–2159, July 2013.

[21] F. Liu, F. Seinstra, and A. Plaza, "Parallel hyperspectral image processing on distributed multicluster systems," *Journal of Applied Remote Sensing*, vol. 5, no. 1, pp. 051501–051501–14, 2011.

[22] A. Plaza, P. Martinez, R. Perez, and J. Plaza, "Spatial/spectral endmember extraction by multidimensional morphological operations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 9, pp. 2025–2041, Sep 2002.

[23] H. Ren and C.-I Chang, "Automatic spectral target recognition in hyperspectral imagery," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1232–1249, Oct 2003.

[24] J. Sigurdsson, M. Ulfarsson, J. R. Sveinsson, and J. Bioucas-Dias, "Sparse distributed hyperspectral unmixing," *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2016.

[25] S. Henrot, J. Chanussot, and C. Jutten, "Dynamical spectral unmixing of multitemporal hyperspectral images," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3219–3232, July 2016.

[26] O. Eches, N. Dobigeon, C. Mailhes, and J. Tourneret, "Bayesian estimation of linear mixtures using the normal compositional model. application to hyperspectral imagery," *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1403–1413, June 2010.

[27] B. Somers, M. Zortea, A. Plaza, and G. Asner, "Automated extraction of image-based endmember bundles for improved spectral unmixing," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 396–408, April 2012.

[28] A. Zare and K. Ho, "Endmember variability in hyperspectral analysis: Addressing spectral variability during spectral unmixing," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 95–104, Jan 2014.

[29] S. Liu, L. Bruzzone, F. Bovolo, and P. Du, "Unsupervised multitemporal spectral unmixing for detecting multiple changes in hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 5, pp. 2733–2748, May 2016.

[30] A. Ertrk, M.-D. Iordache, and A. Plaza, "Sparse unmixing-based change detection for multitemporal hyperspectral images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 2, pp. 708–719, Feb 2016.

[31] A. Ertrk, M. D. Iordache, and A. Plaza, "Sparse unmixing with dictionary pruning for hyperspectral change detection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 1, pp. 321–330, Jan 2017.

[32] H. L. Yang and M. M. Crawford, "Spectral and spatial proximity-based manifold alignment for multitemporal hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 1, pp. 51–64, Jan 2016.

[33] S. Liu, L. Bruzzone, F. Bovolo, and P. Du, "Hierarchical unsupervised change detection in multitemporal hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 1, pp. 244–260, Jan 2015.

[34] P. Thouvenin, N. Dobigeon, and J. Tourneret, "Online unmixing of multitemporal hyperspectral images accounting for spectral variability," *IEEE Transactions on Image Processing*, vol. PP, no. 99, pp. 1–1, 2016.

[35] A. Cichocki, A. Phan, and C. Caiafa, "Flexible hals algorithms for sparse non-negative matrix/tensor factorization," in *Proceedings of 2008 IEEE International Workshop on Machine Learning for Signal Processing*, 2008, pp. 73–78.

[36] M. Ulfarsson and V. Solo, "Sparse component analysis via dyadic cyclic descent," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4234–4238, May 2014.

[37] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Learning overcomplete dictionaries based on atom-by-atom updating," *Signal Processing, IEEE Transactions on*, vol. 62, no. 4, pp. 883–891, Feb 2014.

[38] R. Bro and N. Sidiropoulos, "Least squares algorithms under unimodality and non-negativity constraints," *Journal Of Chemometrics*, , no. 12, pp. 223–247, 1998.

[39] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "A fast algorithm for the constrained formulation of compressive image reconstruction and other linear inverse problems," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, March 2010, pp. 4034–4037.

[40] J. Chen and Z. Chen, "Extended Bayesian information criteria for model selection with large model spaces," *Biometrika*, vol. 95, no. 3, pp. 759–771, 2008.

[41] M. Ulfarsson, V. Solo, and G. Marjanovic, "Sparse and low rank decomposition using l0 penalty," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3312–3316, April 2015.

[42] E. Feigelson and G. Babu, *Statistical Challenges in Astronomy*, Springer, 2003.

[43] B.A. Frigyik, A. Kapila, and M.R. Gupta, "Introduction to the Dirichlet Distribution and Related Processes," Tech. Rep., University of Washington Seattle, 2010.

[44] J. Nascimento and J. Bioucas-Dias, "Vertex component analysis: a fast algorithm to unmix hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898 – 910, April 2005.

[45] M. Berman, H. Kiiveri, R. Lagerstrom, A. Ernst, R. Dunne, and J.F. Huntington, "Ice: A statistical approach to identifying endmembers in hyperspectral images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 10, pp. 2085–2095, Oct 2004.

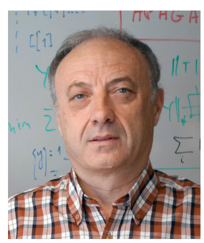
[46] L. Miao and H. Qi, "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 3, pp. 765–777, March 2007.

[47] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 267–288, 1994.

Jakob Sigurdsson (S'10-M'15) received the B.S. and M.S. degree in 2011 and the Ph.D. degree in 2015, from the University of Iceland where hi is currently a postdoctoral researcher. His research interests include statistical signal- and image processing and remote sensing.

Magnus Orn Ulfarsson (S'00-M'05) received the B.S. and the M.S. degrees from the University of Iceland in 2002, and the Ph.D. degree from the University of Michigan in 2007. He joined the University of Iceland in 2007, where he is currently a Professor. He has been affiliated with deCODE Genetics, Reykjavik, Iceland, since 2013. His research interests include statistical signal processing, genomics, medical imaging, and remote sensing.

Johannes R. Sveinsson (SM'02) received the B.S. degree from the University of Iceland, Reykjavk, and the M.S. and Ph.D. degrees from Queen's University, Kingston, ON, Canada, all in electrical engineering. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Iceland. He was with the Laboratory of Information Technology and Signal Processing from 1981 to 1982 and the Engineering Research Institute and the Department of Electrical and Computer Engineering as a Senior Member of research staff and a Lecturer, respectively, from 1991 to 1998. He was a Visiting Research Student with the Imperial College of Science and Technology, London, U.K., from 1985 to 1986. At Queen's University, he held teaching and research assistantships. His current research interests are in systems and signal theory. Dr. Sveinsson received the Queen's Graduate Awards from Queen's University. He is a co-recipient of the 2013 IEEE GRSS Highest Impact Paper Award.



José M. Bioucas-Dias José Bioucas-Dias (S'87–M'95–SM'15–F'17) received the EE, MSc, PhD, and Habilitation degrees in electrical and computer engineering from Instituto Superior Técnico (IST), Universidade Técnica de Lisboa (now Universidade de Lisboa), Portugal, in 1985, 1991, 1995, and 2007, respectively. Since 1995, he has been with the Department of Electrical and Computer Engineering, IST, where he is an Associate Professor and teaches inverse problems in imaging and electric communications. He is also a Senior Researcher with the Pattern and Image Analysis group of the Instituto de Telecomunicações, which is a private non-profit research institution.

His research interests include inverse problems, signal and image processing, pattern recognition, optimization, and remote sensing. Dr. Bioucas-Dias has authored or co-authored more than 300 scientific publications including more than 90 journal papers (78 of which published in IEEE journals) and 210 peer-reviewed international conference papers and book chapters. He is included in Thomson Reuters' Highly Cited Researchers 2015 list.

Dr. Bioucas-Dias was an Associate Editor for the IEEE Transactions on Circuits and Systems (1997-2000) and IEEE Transactions on Image Processing (2010-2014) and he is a Senior Area Editor for the IEEE Transactions on Image Processing and an Associate Editor for the IEEE Transactions on Geoscience and Remote Sensing. He was a Guest Editor for 6 special issues of IEEE journals (IEEE Transactions on Geoscience and Remote Sensing, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2), IEEE Signal Processing, IEEE Journal of Selected Topics in Signal Processing, IEEE Geoscience and Remote Sensing Magazine). He was the General Co-Chair of the 3rd IEEE GRSS Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote sensing (WHISPERS'2011) and has been a member of program/technical committees of several international conferences.