Minimum Volume Simplex Analysis: A Fast Algorithm for Linear Hyperspectral Unmixing

Jun Li, *Member, IEEE*, Alexander Agathos, Daniela Zaharie, José M. Bioucas-Dias, *Member, IEEE*,
Antonio Plaza, *Fellow, IEEE*, and Xia Li

***Abstract*—Linear spectral unmixing aims at estimating the number of pure spectral substances, also called *endmembers*, their spectral signatures, and their abundance fractions in remotely sensed hyperspectral images. This paper describes a method for unsupervised hyperspectral unmixing called minimum volume simplex analysis (MVSA) and introduces a new computationally efficient implementation. MVSA approaches hyperspectral unmixing by fitting a minimum volume simplex to the hyperspectral data, constraining the abundance fractions to belong to the probability simplex. The resulting optimization problem, which is computationally complex, is solved in this paper by implementing a sequence of quadratically constrained subproblems using the interior point method, which is particularly effective from the computational viewpoint. The proposed implementation (available online: www.lx.it.pt/%7ejun/DemoMVSA.zip) is shown to exhibit state-of-the-art performance not only in terms of unmixing accuracy, particularly in nonpure pixel scenarios, but also in terms of computational performance. Our experiments have been conducted using both synthetic and real data sets. An important assumption of MVSA is that pure pixels may not be present in the hyperspectral data, thus addressing a common situation in real scenarios which are often dominated by highly mixed pixels. In our experiments, we observe that MVSA yields competitive performance when compared with other available algorithms that work under the nonpure pixel regime. Our results also demonstrate that MVSA is well suited to problems involving a high number of endmembers (i.e., complex scenes) and also for problems involving a high number of pixels (i.e., large scenes).**

***Index Terms*—Endmember identification, hyperspectral imaging, interior point method, minimum volume simplex analysis (MVSA), spectral unmixing.**

I. INTRODUCTION

HYPERSPECTRAL unmixing is a source separation problem which focuses on the decomposition of the pixel spectra into a set of constituent spectra, also termed *endmembers*, and their corresponding fractional abundances present in the pixel [1]. Compared with the canonical source separation scenario, the sources in hyperspectral unmixing are statistically dependent, and the observed mixtures are either linear or nonlinear in nature [2]. The linear mixing model holds when the mixing scale is macroscopic [3]–[6]. In this case, we assume that the acquired spectral vectors are a linear combination of the endmember signatures present in the scene, weighted by their respective fractional abundances. In turn, nonlinear mixing holds when the light suffers multiple scattering involving different materials [7]–[10]. This model assumes that incident solar radiation is scattered by the scene through multiple bounces involving several endmembers [7]. In practice, nonlinear mixtures happen very often in real scenarios, although linear models can approximate these complex mixtures with a good degree of confidence [11]. These characteristics, together with the high dimensionality of hyperspectral vectors and the large number of pixels present in real scenes, place the unmixing of hyperspectral mixtures beyond the reach of most source separation algorithms, thus fostering active research in the field (see [1] for a recent overview of advances in this area).

Linear unmixing techniques can be classified into statistical and geometrical based. The former category addresses spectral unmixing as an inference problem, often formulated under the Bayesian framework, whereas the latter category exploits the fact that the spectral vectors (under the linear mixing model) lie in a simplex whose vertices correspond to the endmembers. Here, we focus on the geometrical approach to spectral unmixing (additional details about the statistical approach can be found in [1] and references therein). It should be noted that the overview does not intend to be exhaustive, but to introduce some of the most relevant methods that will be compared with our proposed approach. For instance, important recent techniques such as sparse unmixing [12] or support vector machine-based unmixing [13] are not described in detail.

The geometrical approach exploits the fact that, under the linear mixing model, hyperspectral vectors belong to a simplex set whose vertices correspond to the endmembers. Therefore, finding the endmembers is equivalent to identifying the vertices of the aforementioned simplex. The main research lines presented in recent years under this framework belong to two different groups. Pure pixel-based algorithms assume that the scene contains at least one pure pixel per endmember [1]. More recently, several algorithms dropped this assumption by assuming that no pure pixels may be present in real hyperspectral scenes [14]. In the following, we outline these two approaches.

Manuscript received August 9, 2014; revised January 30, 2015; accepted March 23, 2015. This work was supported by the Portuguese Science and Technology Foundation, Projects UID/EEA/50008/2013 and PTDC/EEI-PRO/1470/2012.

J. Li and X. Li are with the Guangdong Provincial Key Laboratory of Urbanization and Geo-simulation, School of Geography and Planning, Sun Yat-sen University, Guangzhou 510275, China.

A. Agathos and D. Zaharie are with the Computer Science Department, West University of Timisoara, 300223 Timisoara, Romania.

J. M. Bioucas-Dias is with the Instituto de Telecomunicações, Instituto Superior Técnico, Universidade de Lisboa, 1649-004 Lisboa, Portugal.

A. Plaza is with the Hyperspectral Computing Laboratory Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, 10071 Cáceres, Spain.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2015.2417162

A. Spectral Unmixing With the Pure Pixel Assumption

If there exists at least one pure pixel per endmember (i.e., a pixel containing just one material), then unmixing amounts to finding the spectral vectors in the data set corresponding to the vertices of the data simplex. Some popular algorithms implemented using this assumption are the vertex component analysis (VCA) [15] and N-FINDR [16], among many others (see [1], [11], and [17] for more extensive overviews). Among representative algorithms in this category, maximum volume simplex-based techniques like VCA or N-FINDR are based on the fact that, in p spectral dimensions, the p-dimensional volume defined by a simplex formed by the purest pixels is larger than any other volume defined by any other combination of pixels. The VCA algorithm iteratively projects data onto a direction orthogonal to the subspace spanned by the endmembers already determined. The new endmember signature corresponds to the extreme of the projection. The algorithm iterates until all endmembers are exhausted [15]. The N-FINDR algorithm finds the set of pixels defining the largest volume by inflating a simplex inside the data. This strategy is opposite to that explored by minimum volume algorithms (addressed in the next section), which instead minimize the volume of the simplex that encloses all pixel observations [18], [19].

B. Spectral Unmixing Without the Pure Pixel Assumption

If the pure pixel assumption is not fulfilled (this is a more realistic scenario since hyperspectral data are often dominated by highly mixed pixels [1], [2]), the unmixing process is a rather challenging task. This is because the endmembers, or at least some of them, are not included in the data set. In his seminal work Craig [18] presented this idea which has also been explored by other authors providing different algorithms for minimizing the same objective function such as the minimum-volume enclosing simplex (MVES) [19]. The robust MVES [20] and the simplex identification via split augmented Lagrangian (SISAL) algorithm [21] are variants of these. Other techniques minimize a regularized least squares fit of the data, including the iterative constrained endmembers (ICEs) [22], the sparsity-promoting ICE [23], and the minimum volume constrained nonnegative matrix factorization (MVC-NMF) [24]. MVC-NMF uses constrained NMF to decompose mixed pixels in multispectral and hyperspectral remote sensing images. Specifically, MVC-NMF adopts a volume-based constraint together with NMF for the decomposition of mixed pixels. In this regard, the main difference between ICE and MVC-NMF is the measure of the simplex that they use as a regularizer [25]. Craig[18] and MVES find a simplex by minimizing the simplex volume subject to the constraint that all the dimensionally-reduced pixels are enclosed by the simplex. The MVES algorithm is based on a cyclic minimization procedure, in which a sequence of linear programs is solved. SISAL implements a robust version of the minimum volume concept which allows violations of the abundance nonnegativity constraint.

At this point, it is important to emphasize that the main difference between the seminal algorithm introduced by Craig [18] and other strategies like MVES or SISAL lies in the solution of the optimization problem. It has been found that the solutions provided by greedy solvers are strongly dependent on the initialization [1]. This handicap was circumvented by MVES and SISAL by reformulating the optimization problem with respect to the inverse of the matrix of estimated endmembers [25]. In this paper, we present a new computationally efficient implementation of the minimum volume simplex analysis (MVSA) algorithm introduced in [26] that uses sequential quadratic programming (SQP) to solve the optimization problem. The optimization process adopted by the algorithm will be shown to be faster than the following: 1) the original solver introduced in [26]; 2) MVES which solves exactly the same optimization; and 3) MVC-NMF.

C. Proposed Approach

In this paper, we specifically focus on the minimum volume-based approach for hyperspectral unmixing and further develop a computationally efficient version of the MVSA method [26]. The MVSA algorithm fits a minimum volume simplex to the hyperspectral data by constraining the abundance fractions to belong to the probability simplex. The resulting optimization problem, which is computationally very complex, is solved in this paper by implementing a sequence of quadratically constrained subproblems using the interior point method [27], thus providing a completely new perspective on the MVSA method based on an efficient implementation that allows, for the first time in the literature, a detailed comparison of its performance with that of other standard methods based on minimum volume concepts and the nonpure pixel assumption. The main contributions and differences of the proposed work with regard to [26] can be summarized as follows.

1) An interior point algorithm is used to solve the optimization problem, whereas in [26], an SQP approach was used. The proposed optimization greatly reduces the computational complexity of the algorithm and allows for its practical utilization with moderately large and complex hyperspectral data sets.
2) An implementation of the new algorithm is available in the form of an online demonstration.[1] This optimized demonstration includes the Matlab source code of the algorithm, together with different analysis examples including difficult cases with a large number of samples, large number of endmembers, etc. The availability of the source code will allow interested readers to reproduce our results and to conduct their own experiments.

The remainder of this paper is organized as follows. Section II describes the fundamentals of the MVSA algorithm. Section III describes our proposed implementation, with particular emphasis on the optimizations conducted. Section IV presents a detailed experimental evaluation of the algorithm using synthetic data sets. In Section V, we use a subset of the popular Airborne Visible/Infrared Imaging Spectrometer

---

[1] Available from http://www.lx.it.pt/%7ejun/DemoMVSA.zip.

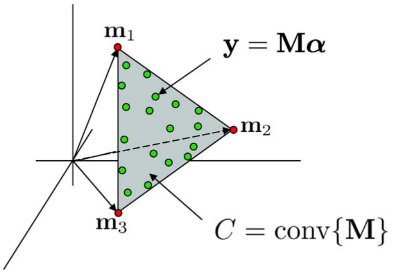


Fig. 1. Illustration of the simplex set $\mathcal{C}$ for $p = 3$. $\mathcal{C}$ is the convex hull of the columns of $\mathbf{M}$. Green circles represent spectral vectors. Red circles represent vertices of the simplex and correspond to the endmembers.

(AVIRIS) Cuprite data for evaluation. Section VI summarizes this paper and hints at plausible future research lines.

II. MVSA

Let $\mathbf{Y} \equiv [\mathbf{y}_1, \ldots, \mathbf{y}_N] \in \mathbb{R}^{L\times N}$ denote a matrix collecting N measured spectral vectors of size L. We assume that these vectors are well approximated by the linear mixing model [1]; that is, for $i \in \{1, 2, \ldots, N\}$, we have

$$\begin{aligned} &\mathbf{y}_i = \mathbf{M}\boldsymbol{\alpha}_i + \mathbf{n}_i \\ &\text{s.t.:} \quad \boldsymbol{\alpha}_i \geq 0, \;\; \mathbf{1}_p^T\boldsymbol{\alpha}_i = 1 \end{aligned} \tag{1}$$

where $\mathbf{M} \equiv [\mathbf{m}_1, \ldots, \mathbf{m}_p] \in \mathbb{R}^{L\times p}$ is the mixing matrix ($\mathbf{m}_j$ denotes the jth endmember signature and p is the number of endmembers), $\boldsymbol{\alpha}_i = [\alpha_{i1}, \alpha_{i2}, \ldots, \alpha_{ip}]^T$ is the abundance vector, $\mathbf{1}_p = [1, 1, \ldots, 1]^T$ is a column vector of size p (the notation $[\cdot]^T$ stands for the vector or matrix transpose), and $\mathbf{n}_i$ accounts for additive noise. The constraints $\boldsymbol{\alpha}_i \geq 0$ (the notation $\mathbf{A} \geq 0$ is to be understood componentwise) and $\mathbf{1}_p^T\boldsymbol{\alpha}_i = 1$ stem from a physical interpretation of the abundance vector according to which the components of $\boldsymbol{\alpha}_i$ represent fractions occupied by the corresponding endmembers; therefore, they are nonnegative, and their sum is equal to one.

By collecting the abundance vectors in the matrix $\mathbf{A} \equiv [\boldsymbol{\alpha}_1, \ldots, \ldots, \boldsymbol{\alpha}_N] \in \mathbb{R}^{p\times N}$ and the noise vectors in the matrix $\mathbf{N} \equiv [\mathbf{n}_1, \ldots, \mathbf{n}_N] \in \mathbb{R}^{L\times N}$, we may write the observation equations (1), for $i \in \{1, 2, \ldots, N\}$, in the compact matrix form

$$\begin{aligned} &\mathbf{Y} = \mathbf{M}\mathbf{A} + \mathbf{N} \\ &\text{s.t.:} \quad \mathbf{A} \geq 0, \;\; \mathbf{1}_p^T\mathbf{A} = \mathbf{1}_N^T. \end{aligned} \tag{2}$$

The set $\mathcal{C} \equiv \{\mathbf{x} = \mathbf{M}\boldsymbol{\alpha} \in \mathbb{R}^L : \boldsymbol{\alpha} \geq 0, \mathbf{1}_p^T\boldsymbol{\alpha} = 1\}$, assuming that $\mathbf{M}$ is full rank, is a $(p-1)-$simplex, meaning that $\mathcal{C}$ has p vertices corresponding to the columns of $\mathbf{M}$. MVSA aims at finding the vertices of the simplex $\mathcal{C}$, and therefore the matrix $\mathbf{M}$, by fitting a simplex of minimum volume to the observed data $\mathbf{Y}$. This concept is schematized in Fig. 1 for $p = 3$, where $\mathcal{C}$ denotes the convex hull of the columns of $\mathbf{M}$, the green circles represent spectral vectors, and the red circles represent vertices of the simplex, which correspond to the endmembers. If there exist enough samples in the facets of the simplex, then the minimum volume simplex containing the spectral vectors corresponds to the true one, as illustrated in Fig. 1. Hence, the identification of the minimum volume simplex is, in the absence of noise, equivalent to the identification of $\mathbf{M}$ (see [1] for further details about the minimum volume approach to hyperspectral unmixing). In addition to the mixing matrix $\mathbf{M}$, MVSA also estimates the abundance matrix $\mathbf{A}$.

A. MVSA Preprocessing

As discussed before, the vectors $\mathbf{M}\boldsymbol{\alpha}_i$ belong to the simplex set $\mathcal{C}$. However, this is not the case of the measured vectors $\mathbf{y}_i = \mathbf{M}\boldsymbol{\alpha}_i + \mathbf{n}_i$, owing to the presence of the observation noise $\mathbf{n}_i$. Another degradation mechanism that displaces the measured vectors further away from the original simplex set is the spectral variability due to, namely, variations in the illumination and surface topography. Spectral variability is often characterized by pixel-dependent scaling factors affecting the abundance vectors. That is, instead of $\boldsymbol{\alpha}_i$, we have $\boldsymbol{\gamma_i}\boldsymbol{\alpha}_i$, with $\gamma_i > 0$, for $i \in \{1, 2, \ldots, N\}$, and therefore, the sum-to-one constraint does not hold true.

The observation noise and the spectral variability are two degradation mechanisms which have a negative impact on the inference of the simplex of minimum volume. In order to mitigate these negative effects, we introduce two processing steps. In the first step, the signal subspace is identified using the hyperspectral subspace identification by minimum error (HySime) [28] algorithm. In the second step, the scale factors $\gamma_i > 0$ are removed by means of a projection on a suitable affine set. Below, we summarize these steps (for more details, see [1]).

1) Signal Subspace Identification: The objective of signal subspace identification is the estimation of the subspace span$(\mathbf{M})$, i.e., the subspace spanned by the columns of $\mathbf{M}$. Under the observation model (2), the identification of span$(\mathbf{M})$ may be obtained via the eigendecomposition of the sample correlation matrix $\mathbf{Y}\mathbf{Y}^T/N$. However, if the noise is band dependent, the inference of span$(\mathbf{M})$ is more complex. We use HySime [28], which assumes band-dependent noise, to estimate the noise covariance matrix and the signal subspace. HySime outputs the estimated subspace in the form of an orthonormal matrix $\mathbf{U} \equiv [\mathbf{u}_1, \ldots, \mathbf{u}_p] \in \mathbb{R}^{L\times p}$ whose columns span the same subspace as $\mathbf{M}$. Notice that in this paper the term orthogonal matrix is to be understood in the sense of matrices with orthonormal columns. Based on $\mathbf{U}$, we compute the coordinates of $\mathbf{y}_i$, for $i \in \{1, 2, \ldots, N\}$, as

$$\mathbf{Y} \leftarrow \mathbf{U}^T\mathbf{Y} = \mathbf{U}^T\mathbf{M}\mathbf{A} + \mathbf{U}^T\mathbf{N}. \tag{3}$$

For simplicity, here, we use $\mathbf{Y}$ to denote both the observation and its projection onto the identified subspace. Consequentially $\mathbf{y}_i$ is used to denote the observed and projected vectors.

According to (3), the projected vectors still follow a linear mixing model with mixing matrix $\mathbf{U}^T\mathbf{M} \in \mathbb{R}^{p\times p}$ and noise vectors $\mathbf{U}^T\mathbf{N}$. The new model has, however, two significant advantages with respect to the original one: 1) computational, because $p \ll L$ in most applications, and 2) improved SNR, because $\|\mathbf{U}^T\mathbf{N}\|_F \ll \|\mathbf{N}\|_F$, where $\|\mathbf{X}\|_F^2 \equiv \mathrm{trace}(\mathbf{X}\mathbf{X}^T)$ denotes the Frobenius norm of the matrix $\mathbf{X}$.

Given that $\text{span}(\mathbf{U}) = \text{span}(\mathbf{M})$, apart from estimation errors which are very small if N is large, then we have $\mathbf{M} = \mathbf{U}\mathbf{U}^T\mathbf{M}$. The implication of this property is that we may estimate the matrix $\mathbf{U}^T\mathbf{M} \in \mathbb{R}^{p\times p}$, for example, $\widehat{\mathbf{M}}_U$, and then obtain the estimate of the original mixing matrix as $\widehat{\mathbf{M}} = \mathbf{U}\widehat{\mathbf{M}}_U$.

2) Affine Projection: The objective of this step is to remove the effect of pixel-dependent scale factors and, thus, to recover the sum-to-one constraint. This goal may be achieved by projecting each spectral vector onto the hyperplane that best represents the measured data set in the least squares sense. Here, we follow closely the work in [19].

Let

$$\overline{\mathbf{y}} \equiv \frac{1}{N}\sum_{i=1}^{N} \mathbf{y}_i, \quad \text{and} \quad \overline{\mathbf{Y}} = \overline{\mathbf{y}}\mathbf{1}_N^T.$$

The hyperplane that best represents the measured data set in the least squares sense is given by $\mathcal{H}_y \equiv \{\mathbf{y} \in \mathbb{R}^p : \mathbf{y} = \overline{\mathbf{y}} + \mathbf{E}_{p-1}^T\boldsymbol{\beta}\, \boldsymbol{\beta} \in \mathbb{R}^{p-1}\}$, where $\mathbf{E}_{p-1}$ holds in its columns the $p-1$ eigenvalues of the sample covariance matrix $(\mathbf{Y}-\overline{\mathbf{Y}})(\mathbf{Y}-\overline{\mathbf{Y}})^T/N$ corresponding to the $p-1$ largest eigenvalues of the same matrix [19]. The orthogonal projection of the measured vectors onto $\mathcal{H}_y$ is given by

$$\mathbf{y}_i \leftarrow \overline{\mathbf{y}} + \mathbf{E}_{p-1}^T(\mathbf{y}_i - \overline{\mathbf{y}}), \quad i \in \{1, 2, \ldots, N\}.$$

Given that, after the projection, the vectors $\mathbf{y}_i \in \mathbb{R}^p$ belong to a $(p-1)$ affine set, then the sum-to-one constraint is recovered; that is, any $\mathbf{y} \in \mathcal{H}_y$ may be written as an affine combination of p linearly independent vectors lying in $\mathcal{H}_y$.

At this point, we would like to call attention to the fact that the affine projection may introduce angle displacements between the original measured vectors and the corresponding projected ones. These displacements increase with the spread of the scaling factors [22]. This shortcoming may be avoided by using the projective projection instead of the affine projection (see [1] for a detailed discussion on this issue).

B. MVSA Inference Criterion

Our goal is to perform hyperspectral linear unmixing under the linear mixing model assumption. Following the rationale introduced in [21] and [26], we formulate the problem by seeking the smallest $(p-1)$—simplex $\mathcal{C}$ that contains the data samples $\mathbf{Y}$. Assuming that the noise term $\mathbf{U}^T N$ shown in (3)—obtained after the data projection step—is negligible, the hyperspectral unmixing problem may be formulated as

$$\begin{aligned} \widehat{\mathbf{M}} &= \arg\min_{\mathbf{M}} |\det(\mathbf{M})|, \\ \text{s.t.:} &\quad \mathbf{QY} \geq 0,\ \ \mathbf{1}_p^T\mathbf{QY} = \mathbf{1}_N^T \end{aligned} \qquad (4)$$

where $\mathbf{Q} \equiv \mathbf{M}^{-1}$ [see (2)]; since $|\det(\mathbf{M})|$ is the volume defined by the origin and the columns of $\mathbf{M}$, the interpretation of optimization (4) is clear: We seek a mixing matrix $\mathbf{M} = \mathbf{Q}^{-1}$ defining the smaller simplex that contains the observed data in the simplex $\mathcal{C}$, which is a facet of the simplex defined by the columns of $\mathbf{M}$ and the origin.

Since $\det(\mathbf{Q}) = 1/\det(\mathbf{M})$, we can replace the problem (4) with the following:

$$\begin{aligned} \widehat{\mathbf{Q}} &= \arg\max_{\mathbf{Q}}\ \log|\det(\mathbf{Q})|, \\ \text{s.t.:} &\quad \mathbf{QY} \geq 0,\ \ \mathbf{1}_p^T\mathbf{QY} = \mathbf{1}_N^T. \end{aligned} \qquad (5)$$

As already mentioned, in this paper, we are assuming that the noise after the projection step is negligible. We call attention, however, to a robust to noise and outlier version of MVSA introduced in [26] and further developed in [21]. This robustness is the result of replacing the hard constraint $\mathbf{QY} \geq 0$ with a soft constraint $-\mathbf{1}^T\text{hinge}(-\mathbf{QY})\mathbf{1}$, where $\text{hinge}(\mathbf{x})$ is an elementwise operator that, for each component, yields the negative part of $\mathbf{x}$. However, in this paper, we only address the unmixing problem under the hard constraint $\mathbf{QY} \geq 0$ because our objective is mainly focused on solving (5) in a computationally very efficient way. In the following section, we describe our proposed implementation of MVSA which aims at obtaining "good" (but suboptimal) solutions to the optimization problem (5).

III. PROPOSED IMPLEMENTATION

A. Constraint Reduction

Hyperspectral data sets are often very large (in the sense of the number of pixels that they comprise), and thus, the optimization problem described in the previous section is complex from a computational point of view. In order to lighten the computational load of the MVSA algorithm, we adopt the strategy followed in [21] and [26] to reduce the number of constraints, which exploits the following fact:

$$\underbrace{\left\{\mathbf{Q} \in \mathbb{R}^{p\times p}\ :\ \mathbf{1}_p^T\mathbf{QY} = \mathbf{1}_N^T\right\}}_{\mathcal{A}} = \underbrace{\left\{\mathbf{Q} \in \mathbb{R}^{p\times p}\ :\ \mathbf{1}_p^T\mathbf{QYY}^T = \mathbf{1}_N^T\mathbf{Y}^T\right\}}_{\mathcal{B}}. \qquad (6)$$

To prove that $\mathcal{A} = \mathcal{B}$, we show that $\mathcal{A} \subset \mathcal{B}$ and that $\mathcal{B} \subset \mathcal{A}$. The former relation is trivial. We prove the latter by reduction to absurdity. Suppose that we are given a matrix $\mathbf{Q}_a \in \mathcal{A}$ and a matrix $\mathbf{Q}_b \in \mathcal{B} - \mathcal{A}$. It follows that $\mathbf{1}_p^T(\mathbf{Q}_a - \mathbf{Q}_b)\mathbf{Y} \neq \mathbf{0}$, or equivalently, $\mathbf{Y}^T\boldsymbol{\xi} \neq \mathbf{0}$, where $\boldsymbol{\xi}^T \equiv \mathbf{1}_p^T(\mathbf{Q}_a - \mathbf{Q}_b)$, and because $\mathbf{Q}_a \in \mathcal{B}$, $\mathbf{YY}^T\boldsymbol{\xi} = \mathbf{0}$. That is, $\mathbf{Y}^T\boldsymbol{\xi}$ belongs to the null space of $\mathbf{Y}$. This is, however, impossible because the intersection between the null space of $\mathbf{Y}$ and the range of $\mathbf{Y}^T$ is just the zero vector.

Now, assuming that $\mathbf{Y}$ is full rank, then $\mathbf{YY}^T$ is invertible, and we may then write

$$\mathbf{1}_p^T\mathbf{QY} = \mathbf{1}_N^T \Leftrightarrow \mathbf{1}_p^T\mathbf{Q} = \mathbf{q}_p \qquad (7)$$

where $\mathbf{q}_p \equiv \mathbf{1}_N^T \mathbf{Y}^T (\mathbf{Y}\mathbf{Y^T})^{-1}$ can be obtained beforehand. Then, the problem in (5) is simplified to the following form:

$$\begin{aligned} \widehat{\mathbf{Q}} &= \arg\max_{\mathbf{Q}} \ \log|\det(\mathbf{Q})| \\ \text{s.t.:} \quad & \mathbf{Q}\mathbf{Y} \geq 0, \ \ \mathbf{1}_p^T \mathbf{Q} = \mathbf{q}_p. \end{aligned} \tag{8}$$

Notice that, by applying the constraint reduction, we can greatly reduce the number of active constraints in the equality constraint (from pN to p).

MVSA is initialized with the set of endmembers $\mathbf{M} \equiv [\mathbf{m}_1, \ldots, \mathbf{m}_p]$ generated by the VCA [15] algorithm. We selected VCA because it is one of the fastest among the state-of-the-art pure pixel-based methods. In order to ensure that most vectors belong to the convex set generated by the columns of $\mathbf{M}$, we expand the initial simplex to increase the number of pixels that are inside the convex hull of the identified endmembers, which leads to very few active nonnegativity constraints, i.e., $\mathbf{Q}\mathbf{y}_i > 0$ for most pixels. For instance, if there are n samples outside of the current simplex, with $n \ll N$, we reduce the number of active constraints in the inequality constraint from pN to pn. This reduces computational complexity because, during the computation, inactive constraints are temporarily ignored, although we continue to track them.

B. MM Optimization. Sequence of Convex Subproblems

The optimization problem (8) becomes convex only when $\mathbf{Q}$ is restricted to the cone of symmetric positive definite matrices. This is not the case in our application where $\mathbf{Q}$ is neither symmetric nor positive definite yielding a nonconvex and thus quite challenging optimization problem. Herein, we adopt the "minorize-maximization" (MM) framework [29] to find the local optima for (8). The MM scheme is an iterative procedure that, at each iteration, builds a minorizer of the objective function and maximizes it. When the minorizer function is optimized, the original objective function is driven downhill as needed. For the MM procedure to make sense, the sequence of minorizers should be much easier to optimize than the original problem.

Let $\mathbf{x} \equiv \text{vec}(\mathbf{Q})$ denote the operator that stacks the columns of $\mathbf{Q}$ in the column vector $\mathbf{x}$, $f(\mathbf{x}) \equiv \log|\det(\mathbf{Q})|$, and $\phi(\mathbf{x}; \mathbf{x}^{(t)})$ denote a minorizer for f at $\mathbf{x}^{(t)}$; that is, $f(\mathbf{x}^{(t)}) = \phi(\mathbf{x}^{(t)}; \mathbf{x}^{(t)})$, and $f(\mathbf{x}) \geq \phi(\mathbf{x}; \mathbf{x}^{(t)})$ for all $\mathbf{x}$. Given that $\text{vec}\,(\mathbf{A}\mathbf{B}) = (\mathbf{B}^T \otimes \mathbf{I})\,\text{vec}\,(\mathbf{A}) = (\mathbf{I} \otimes \mathbf{A})\,\text{vec}(\mathbf{B})$, where $\otimes$ denotes the Kronecker operator and $\mathbf{I}$ is the identity matrix with suitable dimension, then our MM iterative procedure is given by

$$\begin{aligned} \mathbf{x}^{(t+1)} = \ & \arg\max_{\mathbf{x}} \ \phi\left(\mathbf{x}; \mathbf{x}^{(t)}\right) \\ & \text{s.t.: } \mathbf{A}_I \mathbf{x} \geq \mathbf{b}_I, \quad \mathbf{A}_E \mathbf{x} = \mathbf{b}_E \end{aligned} \tag{9}$$

where

$$\begin{aligned} \mathbf{A}_I &\equiv (\mathbf{Y}^T \otimes \mathbf{I}) \in \mathbb{R}^{pN \times p^2} \\ \mathbf{A}_E &\equiv \left(\mathbf{I} \otimes \mathbf{1}_p^T\right) \in \mathbb{R}^{p \times p^2} \\ \mathbf{b}_I &\equiv \mathbf{0} \in \mathbb{R}^{pN} \\ \mathbf{b}_E &\equiv \mathbf{q}_p \in \mathbb{R}^{p}. \end{aligned} \tag{10}$$

Let $\mathbf{g}(\mathbf{x}) \equiv \text{vec}(\mathbf{Q}^{-T})$ and $\mathbf{H}(\mathbf{x}) \equiv -\mathbf{K}_n[\mathbf{Q}^{-T} \otimes \mathbf{Q}^{-1}]$, where $\mathbf{K}_n$ is the commutation matrix (i.e., $\mathbf{K}_n \text{vec}(\mathbf{A}) = \text{vec}(\mathbf{A}^T)$), denote respectively the gradient and the Hessian of f. As a minorizer for f, we use the quadratic function

$$\begin{aligned} \phi\left(\mathbf{x}; \mathbf{x}^{(t)}\right) \equiv & f\left(\mathbf{x}^{(t)}\right) + \mathbf{g}^{(t)^T}\left(\mathbf{x} - \mathbf{x}^{(t)}\right) \\ & + \frac{1}{2}\left(\mathbf{x} - \mathbf{x}^{(t)}\right)^T \mathbf{G}^{(t)}\left(\mathbf{x} - \mathbf{x}^{(t)}\right) \qquad (11) \\ = & f\left(\mathbf{x}^{(t)}\right) + \mathbf{c}^{(t)^T}\mathbf{x} + \frac{1}{2}\mathbf{x}^T \mathbf{G}^{(t)} \mathbf{x} \qquad (12) \end{aligned}$$

where $\mathbf{G} \equiv \min\{\lambda_{\min}(\mathbf{H}), -\upsilon)\}\mathbf{I}$, with $\lambda_{\min}(\mathbf{H})$ standing for the minimum eigenvalue of $\mathbf{H}$ and $\upsilon > 0$ being a small positive number, and $\mathbf{c}^{(t)} \equiv \mathbf{g}^{(t)} - \mathbf{G}^{(t)}\mathbf{x}^{(t)}$.

We conclude therefore that the core step in the MVSA algorithm is the computation, in each iteration, of the solution of a quadratic problem with linear inequality and equality constraints with the following structure:

$$\begin{aligned} & \max \mathbf{c}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \mathbf{G}\mathbf{x} \\ & \text{s.t.: } \mathbf{A}_I \mathbf{x} \geq \mathbf{b}_I, \ \ \mathbf{A}_E \mathbf{x} = \mathbf{b}_E \end{aligned} \tag{13}$$

where $\mathbf{A}_I \mathbf{x} \geq \mathbf{b}_I$ and $\mathbf{A_E}\mathbf{x} = \mathbf{b}_E$ are defined in (10). Since $\mathbf{G}$ is negative definite, the quadratic problem (13) is strictly convex, and its difficulty is equal to finding a solution to a linear optimization problem [27]. Thus, we have transformed the nonconvex optimization problem into the solution of a sequence of convex quadratic problems.

The pseudocode for the MVSA algorithm is shown in Algorithm 1. As mentioned in Section III-A, the initialization $\mathbf{x}_0$ is provided by an expansion of the VCA estimate to increase the number of pixels that are in the convex hull of the identified endmembers. The gradient and the Hessian of f are computed in line 4. In line 5, $\lambda_{\min}(\mathbf{H})$ represents the minimum eigenvalue of $\mathbf{H}$, which is a real number because $\mathbf{H}$ is symmetric. Since $\min\{\lambda_{\min}(\mathbf{H}), -\upsilon)\} < 0$, matrix $\mathbf{G}$ is negative definite.

Algorithm 1 MVSA pseudocode

```
1: INPUT: $\mathbf{A}_I, \mathbf{A}_E, \mathbf{b}_I, \mathbf{b}_E, \mathbf{x}_0$ (initialization)
2: Convergence $\leftarrow$ false
3: repeat
4:    $\mathbf{g} \leftarrow \nabla f(\mathbf{x}_0), \mathbf{H} \leftarrow \nabla^2 f(\mathbf{x}_0)$
5:    $\mathbf{G} \leftarrow \min\{\lambda_{\min}(\mathbf{H}), -\upsilon)\}\mathbf{I}$
6:    $\mathbf{c} \leftarrow \mathbf{g} - \mathbf{G}\mathbf{x}$
7:    $\mathbf{x} \leftarrow$ solution of the quadratic optimization (13)
8:    if $f(\mathbf{x}_0) > f(\mathbf{x})$ then
9:       do line search until $f(\mathbf{x}_0) \leq f(\mathbf{x})$
10:   end if
11:   if $|f(\mathbf{x}_0) - f(\mathbf{x})|/|f(\mathbf{x})| <$ threshold then
12:      Convergence $\leftarrow$ true
13:   end if
14:   $\mathbf{x}_0 \leftarrow \mathbf{x}$
15: until Convergence
```

To avoid the computation of $\lambda_{\min}(\mathbf{H})$ at each iteration, an alternative to the definition of $\mathbf{G}$ shown in line 6, which we found heuristically to work very well in practice, is

$$\mathbf{G} = -\upsilon\mathbf{I} + \operatorname{diag}(\mathbf{g}^2) \tag{14}$$

where $\operatorname{diag}(\mathbf{g}^2)$ stands for a diagonal matrix with diagonal elements given by the square of the elements of $\mathbf{g}$.

Because it can only be guaranteed that $\mathbf{G} - \mathbf{H}$ is negative definite in a neighborhood of $\mathbf{x}^{(t)}$, then it cannot be guaranteed that $\phi(\mathbf{x}; \mathbf{x}^{(t)})$ is, in fact, a minorizer of f. In order to ensure that $f(\mathbf{x}_0) \geq f(\mathbf{x})$ after solving the optimization (13), i.e., to have a monotonic behavior, we implement in line 9 of MVSA a line search between $\mathbf{x}$ and $\mathbf{x}_0$ if $f(\mathbf{x}_0) > f(\mathbf{x})$.

C. Fast Interior Point Method to Solve the Quadratic Problem (13)

At this point, it is also important to notice that $\mathbf{A}_I$ is of size of $pN \times p^2$, which brings difficulties for solving problem (13) in terms of computational time and especially in terms of RAM. For instance, for a problem with $p = 20$ endmembers and $N = 512 \times 512$, it would be prohibitive to manipulate $\mathbf{A}_I$. This roadblock has been a major limitation of MVSA in the past. In this paper, we address this problem by using the interior point method to solve the quadratic problem (13).

The Karush–Kuhn–Tucker (KKT) conditions for the quadratic problem (13) are

$$\begin{aligned}
\mathbf{G}\mathbf{x} - \mathbf{A}_I^T\boldsymbol{\lambda} + \mathbf{A}_E^T\boldsymbol{\mu} + \mathbf{c} &= 0\\
\mathbf{A}_I\mathbf{x} - \mathbf{b}_I &\geq 0\\
\mathbf{A}_E\mathbf{x} - \mathbf{b}_E &= 0\\
(\mathbf{A}_I\mathbf{x} - \mathbf{b}_I)_i\lambda_i &= 0,\ i = 1 \ldots n_I \equiv Np\\
\boldsymbol{\lambda} &\geq 0
\end{aligned} \tag{15}$$

where $\boldsymbol{\lambda} \equiv [\lambda_1, \ldots, \lambda_{n_I}]^T$, $\boldsymbol{\mu} \in \mathbb{R}^p$ are the Lagrangian multipliers for the inequality and equality constraints, respectively, n_I is the number of inequality constraints, and the notation $(\mathbf{X})_i$ stands for the ith row of matrix $\mathbf{X}$.

By introducing a slack vector $\mathbf{s} \equiv [s_1, \ldots, s_{n_I}]^T$, the nonlinear system (15) is transformed into a nonlinear system of equations that can be solved by the interior point method so that the problem becomes

$$\begin{aligned}
\mathbf{G}\mathbf{x} - \mathbf{A}_I^T\boldsymbol{\lambda} + \mathbf{A}_E^T\boldsymbol{\mu} + \mathbf{c} &= 0\\
\mathbf{A}_I\mathbf{x} - \mathbf{s} - \mathbf{b}_I &= 0\\
\mathbf{A}_E\mathbf{x} - \mathbf{b}_E &= 0\\
s_i\lambda_i &= 0,\ i = 1 \ldots n_I\\
\boldsymbol{\lambda},\ \mathbf{s} &\geq 0.
\end{aligned} \tag{16}$$

A predictor corrector interior point algorithm [27] is used to solve (16). The predictor corrector algorithm solves two times the calculation of the Newton step of the system of equations (16): one time to get the *affine* Newton step and the other to correct the affine step getting the final Newton step. Notice that the Newton steps in both cases should be constrained such that the vectors $\boldsymbol{\lambda}$ and $\mathbf{s}$ are nonnegative and strictly positive, respectively. The two systems that need to be solved in order to get the affine and final Newton steps are respectively

$$\begin{bmatrix} \mathbf{G} & \mathbf{A}_E^T & \mathbf{0} & -\mathbf{A}_I^T\\ \mathbf{A}_I & \mathbf{0} & -\mathbf{I} & \mathbf{0}\\ \mathbf{A}_E & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}\mathbf{x}^{\text{aff}}\\ \boldsymbol{\Delta\mu}^{\text{aff}}\\ \boldsymbol{\Delta}\mathbf{s}^{\text{aff}}\\ \boldsymbol{\Delta\lambda}^{\text{aff}} \end{bmatrix} = \begin{bmatrix} -\mathbf{r_d}\\ -\mathbf{r_I}\\ -\mathbf{r_E}\\ -\boldsymbol{\Lambda}\mathbf{S}\mathbf{e} \end{bmatrix} \tag{17}$$

where

$$\begin{aligned}
\mathbf{r_d} &\equiv \mathbf{G}\mathbf{x} - \mathbf{A}_I^T\boldsymbol{\lambda} + \mathbf{A}_E^T\boldsymbol{\mu} + \mathbf{c}\\
\mathbf{r_I} &\equiv \mathbf{A}_I\mathbf{x} - \mathbf{s} - \mathbf{b}_I\\
\mathbf{r_E} &\equiv \mathbf{A}_E\mathbf{x} - \mathbf{b}_E\\
\boldsymbol{\Lambda} &\equiv \operatorname{diag}(\lambda_1, \ldots, \lambda_{n_I})\\
\mathbf{S} &\equiv \operatorname{diag}(s_1, \ldots, s_{n_I})\\
\mathbf{e} &\equiv [1, \ldots, 1]^T
\end{aligned} \tag{18}$$

$$\begin{bmatrix} \mathbf{G} & \mathbf{A}_E^T & \mathbf{0} & -\mathbf{A}_I^T\\ \mathbf{A}_I & \mathbf{0} & -\mathbf{I} & \mathbf{0}\\ \mathbf{A}_E & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Lambda} & \mathbf{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Delta}\mathbf{x}\\ \boldsymbol{\Delta\mu}\\ \boldsymbol{\Delta}\mathbf{s}\\ \boldsymbol{\Delta\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{r_d}\\ -\mathbf{r_I}\\ -\mathbf{r_E}\\ -\boldsymbol{\Lambda}\mathbf{S}\mathbf{e} - \boldsymbol{\Delta\Lambda}^{\text{aff}}\boldsymbol{\Delta}\mathbf{S}^{\text{aff}}\mathbf{e} + \sigma\rho\mathbf{e} \end{bmatrix} \tag{19}$$

where

$$\begin{aligned}
\boldsymbol{\Delta\Lambda}^{\text{aff}} &\equiv \operatorname{diag}\left(\Delta\lambda_1^{\text{aff}}, \ldots, \Delta\lambda_{n_I}^{\text{aff}}\right)\\
\boldsymbol{\Delta}\mathbf{S}^{\text{aff}} &\equiv \operatorname{diag}\left(\Delta s_1^{\text{aff}}, \ldots, \Delta s_{n_I}^{\text{aff}}\right)\\
\rho &\equiv \frac{\mathbf{s}^T\boldsymbol{\lambda}}{n_I}
\end{aligned} \tag{20}$$

and $\sigma \in (0, 1]$. The predictor corrector interior point algorithm for the solution of the quadratic problem is shown in Algorithm 2.

Algorithm 2 The predictor corrector interior point algorithm

```
Initialize $(\mathbf{x}_0, \boldsymbol{\mu}_0, \mathbf{s}_0, \boldsymbol{\lambda}_0)$ with $\mathbf{s}_0, \boldsymbol{\lambda}_0 > \mathbf{0}$
$k \leftarrow 0$
while $\sigma, \rho \geq 10^{-8}$ do
    $(\mathbf{x}, \boldsymbol{\mu}, \mathbf{s}, \boldsymbol{\lambda}) \leftarrow (\mathbf{x}_k, \boldsymbol{\mu}_k, \mathbf{s}_k, \boldsymbol{\lambda}_k)$
    solve (17) and get $(\boldsymbol{\Delta}\mathbf{x}^{\text{aff}}, \boldsymbol{\Delta\mu}^{\text{aff}}, \boldsymbol{\Delta}\mathbf{s}^{\text{aff}}, \boldsymbol{\Delta\lambda}^{\text{aff}})$
    $\rho \leftarrow (\mathbf{s}^T\boldsymbol{\lambda}/n_I)$
    $\hat{\alpha}_{\text{aff}} \leftarrow \max\{\alpha \in (0, 1] | (\mathbf{s}, \boldsymbol{\lambda}) + \alpha(\boldsymbol{\Delta}\mathbf{s}^{\text{aff}}, \boldsymbol{\Delta\lambda}^{\text{aff}}) \geq 0\}$
    $\rho_{\text{aff}} \leftarrow (\mathbf{s} + \hat{\alpha}_{\text{aff}}\boldsymbol{\Delta}\mathbf{s}^{\text{aff}})^T(\boldsymbol{\lambda} + \hat{\alpha}_{\text{aff}}\boldsymbol{\Delta\lambda}^{\text{aff}})/n_I$
    $\sigma \leftarrow (\rho_{\text{aff}}/\rho)^3$
    solve (19) and get $(\boldsymbol{\Delta}\mathbf{x}, \boldsymbol{\Delta\mu}, \boldsymbol{\Delta}\mathbf{s}, \boldsymbol{\Delta\lambda})$
    $\tau_k \leftarrow 1 - (1/(k + 1))$
    $\hat{\alpha} \leftarrow \max\{\alpha \in (0, 1] | (\mathbf{s}, \boldsymbol{\lambda}) + \alpha(\boldsymbol{\Delta}\mathbf{s}, \boldsymbol{\Delta\lambda}) \geq$
        $(1 - \tau_k)(\mathbf{s}, \boldsymbol{\lambda})\}$
    $(\mathbf{x}_{k+1}, \boldsymbol{\mu}_{k+1}, \mathbf{s}_{k+1}, \boldsymbol{\lambda}_{k+1}) \leftarrow (\mathbf{x}_k, \boldsymbol{\mu}_k, \mathbf{s}_k, \boldsymbol{\lambda}_k) +$
        $\hat{\alpha}(\boldsymbol{\Delta}\mathbf{x}, \boldsymbol{\Delta\mu}, \boldsymbol{\Delta}\mathbf{s}, \boldsymbol{\Delta\lambda})$
    $k \leftarrow k + 1$
end while
return $\mathbf{x}_k$
```

It should be noted that, since the interior point method converges to the solution of the KKT conditions and the quadratic problem is convex, the solution will be unique and the optimal solution of the quadratic problem. Therefore, Algorithm 2 converges to the optimal solution. The number of iterations needed for convergence is dependent on the allowed error. In all our experiments, we have observed that a maximum number of iterations set empirically to 150 yielded an error that is negligible from a practical point of view. In this respect, it should be mentioned that the convergence of Algorithm 2 is quadratic when the error is small. This behavior is certainly related to the observed negligible error after 150 iterations. In view of these observations, we have set the maximum number of iterations to 150 in all experiments.

D. Normal Equations and Implementation

From the interior point method description in Algorithm 2, it can be seen that the main computational tasks are those described in lines 5 and 10 of Algorithm 2, i.e., those related to the computation of the Newton step. In the case of MVSA, the number of unknowns is p^2, the number of inequality constraints is $n_I = Np$, and the number of equality constraints is $n_E = p$, where N is the number of pixels in the hyperspectral data. As a result, the Jacobian matrix of the systems (17) and (19) is of size $(2Np + p^2 + p) \times (2Np + p^2 + p)$. This means that, for an image with 250 × 190 pixels and $p = 20$ endmembers, the size of the matrices is already prohibitively large for the systems to be solved directly, both computationally and in terms of RAM consumption. However, by exploiting the Jacobian structure system, the problem can be solved progressively by deriving the "normal equations" [27] as follows:

$$\begin{aligned}
&\left(\mathbf{G} + \mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\mathbf{A}_I\right)\boldsymbol{\Delta}\mathbf{x} + \mathbf{A}_E^T\boldsymbol{\Delta}\boldsymbol{\mu}\\
&\qquad = -\mathbf{r}_d + \mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}(-\mathbf{r}_I - \boldsymbol{\Lambda}^{-1}\mathbf{r}_{\Lambda S})\\
&\mathbf{A}_E\boldsymbol{\Delta}\mathbf{x} = -\mathbf{r}_E\\
&\quad\boldsymbol{\Delta}\mathbf{s} = \mathbf{A}_I\boldsymbol{\Delta}\mathbf{x} + \mathbf{r}_I\\
&\quad\boldsymbol{\Delta}\boldsymbol{\lambda} = -\mathbf{S}^{-1}\boldsymbol{\Lambda}(\boldsymbol{\Lambda}^{-1}\mathbf{r}_{\Lambda S} + \boldsymbol{\Delta}\mathbf{s})
\end{aligned} \tag{21}$$

where $\mathbf{r}_{\Lambda S}$ is the last right term of both systems (17) and (19).

It can be seen in (21) that, in order to solve both systems (17) and (19), the first two equations can be solved by forming a $(p^2 + p) \times (p^2 + p)$ linear system, thus obtaining $\boldsymbol{\Delta}\mathbf{x}$ and $\boldsymbol{\Delta}\boldsymbol{\mu}$. Then, $\boldsymbol{\Delta}\mathbf{s}$ is obtained from the third equation using $\boldsymbol{\Delta}\mathbf{x}$, and finally, $\boldsymbol{\Delta}\boldsymbol{\lambda}$ is obtained from the last equation using $\boldsymbol{\Delta}\mathbf{s}$.

The multiplication of the diagonal matrices $\mathbf{S}^{-1}$, $\boldsymbol{\Lambda}$ (and their inverses) with vectors can be done by the membership multiplication of the vector in the diagonal of the matrices with the respective vector. As a result, there is no need to store explicitly the diagonal matrices, but just the vectors in the diagonal. The multiplication of the very large and sparse matrix $\mathbf{A}_I$ with a vector $\mathbf{v}$ in our case can be computed as a matrix by matrix multiplication, i.e., $\mathbf{A}_I\mathbf{v} = \mathbf{V}\mathbf{Y}$ and $\mathbf{A}_I^T\mathbf{v} = \mathbf{V}\mathbf{Y}^T$, where $\mathbf{Y}$ is the sample matrix and $\mathbf{V}$ is the matrix formed by $\mathbf{v}$ with a column-major order. These multiplications can be done efficiently by dense matrix by matrix multiplication. The same concept applies to the multiplication of $\mathbf{A}_I^T\mathbf{v}$. It can be also observed that the symmetric matrix of the linear system, formed by the first two equations of (21), is the same for the solution of $\boldsymbol{\Delta}\mathbf{x}$ and $\boldsymbol{\Delta}\boldsymbol{\mu}$ in both systems (17) and (19); therefore, the inverse is computed just once. Since p, in practice, is small, for example, less than 25, this matrix is of low dimension and can be computed rapidly.

Up until now, the only remaining challenge is how to efficiently compute the term $\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\mathbf{A}_I$. Since we want to avoid using the matrix $\mathbf{A}_I$ due to its large size, a methodology will be presented for exploiting the structure of $\mathbf{A}_I$. We will examine first the computation of $\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}$. Our goal is to create a compact dense form of this computation, i.e., without zero elements and without storing $\mathbf{A}_I^T$ and the diagonal matrix $\mathbf{S}^{-1}\boldsymbol{\Lambda}$. Let $\mathbf{s}_{inv}\boldsymbol{\lambda}$ be the matrix formed by the diagonal of $\mathbf{S}^{-1}\boldsymbol{\Lambda}$. We can express the multiplication using zero initial index as follows:

$$\begin{aligned}
&\left(\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\right)_{\text{compact}}\\
&= \begin{bmatrix}
\frac{\lambda_0}{s_0Y_{00}} & \frac{\lambda_p}{s_pY_{01}} & \cdots & \frac{\lambda_{(N-1)p}}{s_{(N-1)p}Y_{0N-1}}\\
\vdots & \vdots & \vdots & \vdots\\
\frac{\lambda_{p-1}}{s_{p-1}Y_{00}} & \frac{\lambda_p}{s_pY_{01}} & \cdots & \frac{\lambda_{Np-1}}{s_{Np-1}Y_{0N-1}}\\
\vdots & \vdots & \vdots & \vdots\\
\frac{\lambda_0}{s_0Y_{(p-1)0}} & \frac{\lambda_p}{s_pY_{(p-1)1}} & \cdots & \frac{\lambda_{(N-1)p}}{s_{(N-1)p}Y_{(p-1)N-1}}\\
\vdots & \vdots & \vdots & \vdots\\
\frac{\lambda_{p-1}}{s_{p-1}Y_{(p-1)0}} & \frac{\lambda_p}{s_pY_{(p-1)1}} & \cdots & \frac{\lambda_{Np-1}}{s_{Np-1}Y_{(p-1)N-1}}
\end{bmatrix}.
\end{aligned} \tag{22}$$

The multiplication $\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\mathbf{A}_I$ can be described also in compact form by the dense matrix by matrix multiplication $(\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda})_{\text{compact}}\mathbf{Y}^T$. The final product is a $p^2 \times p$ matrix which can be easily factored into the desired $\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\mathbf{A}_I$ $p^2 \times p^2$ matrix, using Algorithm 3.

Algorithm 3 Formulation of the matrix $\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\mathbf{A}_I$

```
$\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\mathbf{A}_I \leftarrow \mathbf{0}$
for i = 0 to p do
    for k = 0 to p do
        for j = 0 to p do
            $\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\mathbf{A}_I[k + j * p + (k + i * p) * p^2] \leftarrow$
            $(\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\mathbf{A}_I)_{\text{comp}}[j + (k + i * p) * p]$
        end for
    end for
end for
```

From the preceding discussion, we have seen that there is no need to store the large matrix $\mathbf{A}_I$. The largest matrix used is the $p^2 \times N$ dimensional $(\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda})_{\text{compact}}$ which is p times smaller than $\mathbf{A}_I$. This makes the problem feasible for values of p of the order of 20.

From the discussion in this section, we conclude that the main computational tasks in our presented method are the

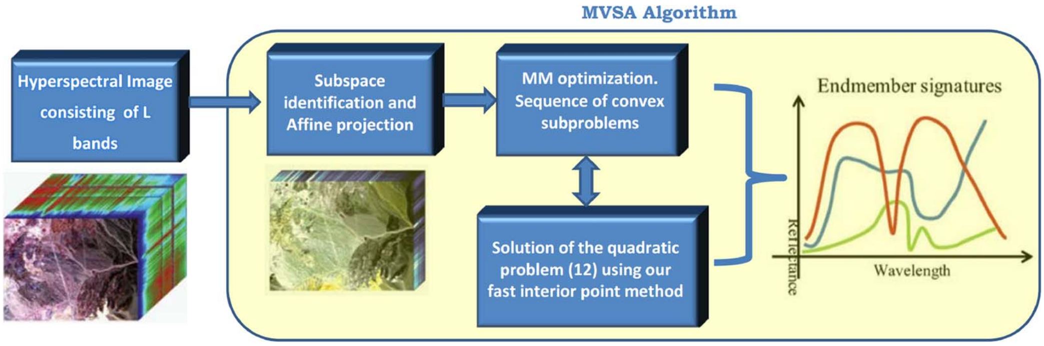


Fig. 2. Flowchart of the proposed MVSA algorithm, from the input (hyperspectral image) to the output (endmember signatures).

calculation of $(\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda})_{\text{compact}}\mathbf{Y}^T$ and the matrix multiplications of the type $\mathbf{A}_I\mathbf{v}$ and $\mathbf{A}_I^T\mathbf{v}$. Using a naive matrix by matrix multiplication, this leads to a computational complexity of the order $O(p^3N + p^2N)$.

In Algorithm 4, we present the predictor corrector algorithm with all the described optimizations. We will explain the algorithm in relation to Algorithm 2. Line 5 of Algorithm 2 is replaced by lines 5–20 of Algorithm 4. In these lines, the solution of (17) is obtained using the normal equations (21). It can also be seen in these lines that we use the optimizations that we described, like calculating the compact representation of (22) in line 9 and applying Algorithm 3 in line 11. Also, in line 7, we use a notation also used in Matlab, specifically the (:) notation, which means to vectorize the matrix in column-major order. Also, when we use the capital letter of a vector, like $\mathbf{X}$, we mean that the vector $\mathbf{x}$ is made a matrix in column-major order. Continuing line 7 of Algorithm 2 is replaced by lines 22–24 of Algorithm 4. Line 10 of Algorithm 2 is replaced by lines 27–35 of Algorithm 4. In these lines, the linear system (19) is solved using the normal equations (21). Note that, here, we reuse the inverse matrix obtained from line 13. Finally, line 12 of Algorithm 2 is replaced by lines 37–39 of Algorithm 4. Having presented Algorithm 4, line 9 of the basic MVSA Algorithm 1 should call Algorithm 4 for obtaining the solution of the quadratic optimization problem. A flowchart describing the full process of the algorithm from the input (hyperspectral image) to the output (endmembers) is included in Fig. 2.

Algorithm 4 Pseudocode of an optimized predictor corrector interior point algorithm

```
1: INPUT: $\mathbf{Y}, \mathbf{c}, \mathbf{G}, \mathbf{A}_E, \mathbf{b}_I, \mathbf{b}_E$
2: $(\mathbf{x}, \boldsymbol{\mu}, \mathbf{s}, \boldsymbol{\lambda}) \leftarrow (\mathbf{x_0}, \mathbf{e}, \mathbf{e}, \mathbf{e}), \mathbf{e} = [1, \ldots, 1]^T$
3: $\rho \leftarrow 1, \sigma \leftarrow 1, k \leftarrow 1$
4: while $(\sigma > 10^{-8}$ or $\rho > 10^{-8})$ do
5:     $\mathbf{s}^{-1}\boldsymbol{\lambda} \leftarrow \mathbf{s}^{-1}. * \boldsymbol{\lambda}$
6:     $\mathbf{r_d} \leftarrow \mathbf{G} * \mathbf{x} + \mathbf{c} - (\mathbf{L} * \mathbf{Y}^T)(:) + \mathbf{A}_E^T * \boldsymbol{\mu}$
7:     $\mathbf{r_I} \leftarrow (\mathbf{X} * \mathbf{Y})(:) - \mathbf{s} - \mathbf{b}_I$
8:     $\mathbf{r_E} \leftarrow \mathbf{A_E} * \mathbf{x} - \mathbf{b}_E$
9:     $(\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda})_{\text{compact}} \leftarrow \text{CalculateCompact}(\mathbf{Y}, \mathbf{s}^{-1}\boldsymbol{\lambda}^T)$
       //Calculate Equation (22)
10:    $(\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\mathbf{A}_I)_{\text{compact}} \leftarrow (\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda})_{\text{compact}} * \mathbf{Y}^T$
11:    $\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\mathbf{A}_I \leftarrow \text{ConstructMatrix}((\mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\mathbf{A}_I)_{\text{compact}})$
       //Apply Algorithm 3
12:    $\mathbf{K} \leftarrow \mathbf{G} + \mathbf{A}_I^T\mathbf{S}^{-1}\boldsymbol{\Lambda}\mathbf{A}_I$
13:    $\mathbf{Inv} \leftarrow \begin{bmatrix} \mathbf{K} & \mathbf{A}_E^T \\ \mathbf{A_E} & \mathbf{0} \end{bmatrix}^{-1}$
14:    $\mathbf{rh} \leftarrow \mathbf{s}^{-1}\boldsymbol{\lambda}. * (\mathbf{r_I} + \mathbf{s})$
15:    $\mathbf{rh} \leftarrow -\mathbf{r_d} - (\mathbf{RH} * \mathbf{Y}^T)(:)$
16:    $\mathbf{rh} \leftarrow \begin{bmatrix} \mathbf{rh} \\ -\mathbf{r_E} \end{bmatrix}$
17:    $\boldsymbol{\Delta}\mathbf{x}\boldsymbol{\Delta}\mathbf{m} \leftarrow \mathbf{Inv} * \mathbf{rh}$
18:    $\boldsymbol{\Delta}\mathbf{x}^{\text{aff}} \leftarrow \boldsymbol{\Delta}\mathbf{x}\boldsymbol{\Delta}\mathbf{m}(1 : p^2)$
19:    $\boldsymbol{\Delta}\mathbf{s}^{\text{aff}} \leftarrow (\boldsymbol{\Delta}\mathbf{X}^{\text{aff}} * \mathbf{Y})(:) + \mathbf{r_I}$
20:    $\boldsymbol{\Delta\lambda}^{\text{aff}} \leftarrow -\mathbf{s}^{-1}\boldsymbol{\lambda}. * (\mathbf{s} + \boldsymbol{\Delta}\mathbf{s}^{\text{aff}})$
21:    $\rho \leftarrow \mathbf{s}^T\boldsymbol{\lambda}/n_I$
22:    $\alpha_{\boldsymbol{\Delta}\mathbf{s}} \leftarrow \min_{\boldsymbol{\Delta}\mathbf{s}^{\text{aff}}<0} -\mathbf{s}./\boldsymbol{\Delta}\mathbf{s}^{\text{aff}}$
23:    $\alpha_{\boldsymbol{\Delta\lambda}} \leftarrow \min_{\boldsymbol{\Delta\lambda}^{\text{aff}}<0} -\boldsymbol{\lambda}./\boldsymbol{\Delta\lambda}^{\text{aff}}$
24:    $\alpha_{\text{aff}} \leftarrow \min(\alpha_{\boldsymbol{\Delta}\mathbf{s}}, \alpha_{\boldsymbol{\Delta\lambda}}, 1)$
25:    $\rho_{\text{aff}} \leftarrow (\mathbf{s} + \alpha_{\text{aff}}\boldsymbol{\Delta}\mathbf{s}^{\text{aff}})^T(\boldsymbol{\lambda} + \alpha_{\text{aff}}\boldsymbol{\Delta\lambda}^{\text{aff}})/n_I$
26:    $\sigma \leftarrow (\rho_{\text{aff}}/\rho)^3$
27:    $\mathbf{s}_{\text{corrected}} \leftarrow \mathbf{s} + \boldsymbol{\lambda}^{-1}. * \boldsymbol{\Delta\lambda}^{\text{aff}}. * \boldsymbol{\Delta}\mathbf{s}^{\text{aff}} - \sigma * \rho * \boldsymbol{\lambda}^{-1}$
28:    $\mathbf{rh} \leftarrow \mathbf{s}^{-1}\boldsymbol{\lambda}. * (\mathbf{r_I} + \mathbf{s}_{\text{corrected}})$
29:    $\mathbf{rh} \leftarrow -\mathbf{r_d} - (\mathbf{RH} * \mathbf{Y}^T)(:)$
30:    $\mathbf{rh} \leftarrow \begin{bmatrix} \mathbf{rh} \\ -\mathbf{r_E} \end{bmatrix}$
31:    $\boldsymbol{\Delta}\mathbf{x}\boldsymbol{\Delta}\mathbf{m} \leftarrow \mathbf{Inv} * \mathbf{rh}$
32:    $\boldsymbol{\Delta}\mathbf{x} \leftarrow \boldsymbol{\Delta}\mathbf{x}\boldsymbol{\Delta}\mathbf{m}(1 : p^2)$
33:    $\boldsymbol{\Delta\mu} \leftarrow \boldsymbol{\Delta}\mathbf{x}\boldsymbol{\Delta}\mathbf{m}(p^2 + 1 : \text{end})$
34:    $\boldsymbol{\Delta}\mathbf{s} \leftarrow (\boldsymbol{\Delta}\mathbf{X} * \mathbf{Y})(:) + \mathbf{r_I}$
35:    $\boldsymbol{\Delta\lambda} \leftarrow -\mathbf{s}^{-1}\boldsymbol{\lambda}. * (\mathbf{s}_{\text{corrected}} + \boldsymbol{\Delta}\mathbf{s})$
36:    $\tau \leftarrow 1 - (1/(k + 1))$
37:    $\alpha_{\text{primal}} \leftarrow \min_{\boldsymbol{\Delta}\mathbf{s}<0} -\tau * \mathbf{s}./\boldsymbol{\Delta}\mathbf{s}$
38:    $\alpha_{\text{dual}} \leftarrow \min_{\boldsymbol{\Delta\lambda}<0} -\tau * \boldsymbol{\lambda}./\boldsymbol{\Delta\lambda}$
39:    $\alpha \leftarrow \min(\alpha_{\text{primal}}, \alpha_{\text{dual}}, 1)$
40:    $(\mathbf{x}, \boldsymbol{\mu}, \mathbf{s}, \boldsymbol{\lambda}) \leftarrow (\mathbf{x}, \boldsymbol{\mu}, \mathbf{s}, \boldsymbol{\lambda}) + \alpha(\boldsymbol{\Delta}\mathbf{x}, \boldsymbol{\Delta\mu}, \boldsymbol{\Delta}\mathbf{s}, \boldsymbol{\Delta\lambda})$
41:    $k \leftarrow k + 1$
42: end while
```

IV. SIMULATED EXPERIMENTS

In this section, we compare the proposed implementation of MVSA with some state-of-the-art endmember extraction algorithms such as VCA [15] MVES [19], and MVC-NMF

TABLE I

COMPARISON OF ENDMEMBER EXTRACTION ALGORITHMS ON A SYNTHETIC IMAGE WITH SIZE OF $N = 100 \times 100$ PIXELS AND $p = 5$ ENDMEMBERS CONTAINING PURE MINERAL SIGNATURES FROM THE USGS LIBRARY UNDER DIFFERENT NOISE LEVELS. THE COMPUTATIONAL TIMES (SECONDS) ARE ALSO INCLUDED. ALL THE RESULTS ARE OBTAINED BY AVERAGING 30 INDEPENDENT RUNS

	VCA				MVES				MVC-NMF				MVSA			
dB	$\parallel \epsilon \parallel_F$	$r\epsilon$	SAD	time	$\parallel \epsilon \parallel_F$	$r\epsilon$	SAD	time	$\parallel \epsilon \parallel_F$	$r\epsilon$	SAD	time	$\parallel \epsilon \parallel_F$	$r\epsilon$	SAD	time
90	0.033	0.003	1.652	0.134	0.004	0.0	0.209	8.071	0.040	0.002	1.856	80.246	0.0004	0.0	0.026	1.125
70	0.038	0.004	1.687	0.135	0.004	0.0	0.258	7.752	0.038	0.002	1.597	82.608	0.0004	0.0	0.025	1.224
50	0.043	0.003	1.792	0.131	0.009	0.0	0.408	6.327	0.041	0.001	1.508	79.745	0.003	0.0	0.163	1.114
30	0.064	0.005	1.828	0.128	0.036	0.0	1.750	3.644	0.040	0.002	1.599	78.747	0.036	0.0	1.543	1.147

[24]. VCA is a pure pixel-based algorithm, while MVES and MVC-NMF are nonpure pixel-based algorithms.

Concerning the parameters involved in the algorithms we use the following settings. MVSA depends on three parameters: the parameter ν ensuring that the matrix $\mathbf{G}$ shown in (14) is negative definite, the maximum number of iterations needed for MVSA to converge (outer loop), and the maximum number of iterations that the interior point method needs to converge (inner loop). In all the experiments conducted in this section, these parameters were set to 10^{-6}, 4, and 150. The MVES algorithm, as implemented by the authors of [19], depends on the number of full cycles over the rows of the matrix of the unknowns needed to converge. The maximum number of iterations has been set by the authors to $10 * p$, where p is the number of endmembers. In the following comparisons, we kept this parameterization. Furthermore, in this experiment, we use a fast implementation of the MVES algorithm distributed by the authors of MVES,[2] in which they propose a solution to reduce the dimensionality of the problem by discarding samples from the interior of the convex hull when the number of endmembers are below ten. Here, we assume that the authors do not use the convex hull for higher dimensions because the problem of calculating it is computationally expensive and also due to the fact that in very high dimensions the data accumulate close to the convex hull even if a uniform distribution is assumed. Finally, MVC-NMF, along with its parameters, has been optimized for execution in accordance with the guidelines provided in [24]. Specifically the value of the MVC-NMF regularization parameter used in our experiments is $\tau = 0.01$.

To evaluate the performance of the different algorithms, the estimated abundance fractions $\widehat{\mathbf{A}}$ and the estimated mixing matrix $\widehat{\mathbf{M}}$ are compared with the true ones ($\mathbf{A}$ and $\mathbf{M}$, respectively). We recall that MVSA is an unmixing algorithm that estimates simultaneously the mixing matrix $\widehat{\mathbf{M}} = \widehat{\mathbf{Q}}^{-1}$ and the abundances $\widehat{\mathbf{A}} = \widehat{\mathbf{Q}}\mathbf{Y}$.

In all experiments, the number of endmembers, p, was estimated using the HySime method in [28], which has been shown to be effective for this task and also for dimensionality reduction purposes [1]. We use several metrics to evaluate the proposed approach. The first one is the mean square error (MSE), denoted as $\parallel \epsilon \parallel_F = \parallel \widehat{\mathbf{M}} - \mathbf{M} \parallel_F$ where $\parallel \cdot \parallel_F$ stands for the Frobenius norm. Another metric considered in our experiments is the reconstruction error, computed as $r\epsilon = \parallel \widehat{\mathbf{Y}} - \mathbf{Y} \parallel_F = \parallel \widehat{\mathbf{M}}\widehat{\mathbf{A}} - \mathbf{Y} \parallel_F$. The third metric used in this work is the spectral angle distance (SAD) (in degrees) expressed as $\mathrm{SAD} = \cos^{-1}(\mathbf{m}_i^T \widehat{\mathbf{m}}_i / \parallel \mathbf{m}_i \parallel \parallel \widehat{\mathbf{m}}_i \parallel)$ (degrees) [2]. Although SAD may not be completely accurate for matching libraries to endmembers, particularly if the endmembers are themselves mixtures or if the atmospheric correction process conducted on the image is not perfect, we have decided to use SAD as it is a standard metric for spectral signature comparison. Our implementation of MVSA has been carried out in Matlab and compared with the Matlab implementations of the other algorithms tested as provided by their authors.

A. Pure Pixel-Based Experiments

This experiment aims at evaluating MVSA for scenarios with pure pixels. In this experiment, the synthetic image, with a size of $N = 100 \times 100$ pixels and $p = 5$ endmembers, is constructed according to the linear model given by (1) using the procedure described in [15] with a maximum purity of 1. That is, for each endmember, there is at least one pure pixel in the simulated image. The spectral signatures were randomly selected from the USGS library [30] (convolved and downsampled to AVIRIS wavelengths). It should be noted that the USGS signatures considered in experiments are randomly sampled from a subset of the USGS library formed by retaining 62 signatures so that the minimum angle between any couple of signatures was larger than 10°. Zero-mean white Gaussian noise, defined as $\mathrm{SNR} = 10\log_{10}(\mathbb{E}\|\mathbf{Y}\|_F^2 / \mathbb{E}\|\mathbf{N}\|_F^2)$ (dB), has been added to the synthetic scene. In our experiments, the proposed MVSA algorithm only considers the hard constraint $\mathbf{QY} \geq 0$. As shown in [26], under the hard constraint, MVSA performs very good under relatively low noise levels. Finally, it should be noted that the results are obtained by averaging 30 independent Monte Carlo runs and all simulated experiments have been performed in a desktop personal computer with the latest Intel Core I5 CPU and 4 GB of RAM.

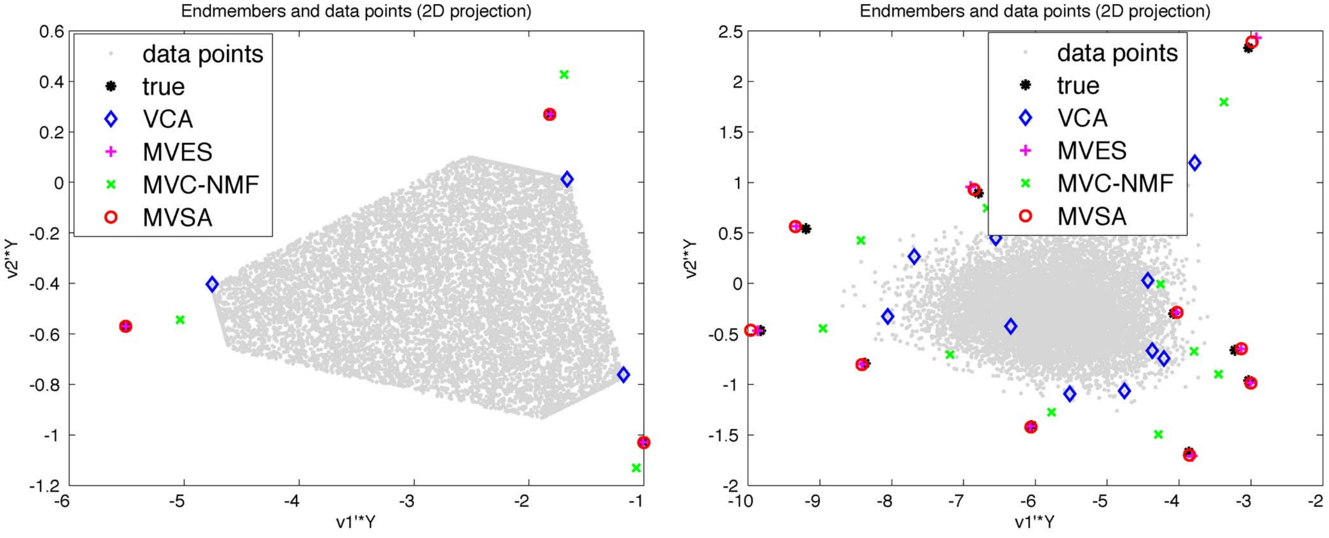
Table I shows the results obtained by the aforementioned methods for the considered scene with different noise levels. It can be observed that all algorithms provide comparable results, which reveals that nonpure pixel-based algorithms such as MVSA, MVES, and MVC-NMF can tackle well problems with pure pixels. Furthermore, it can be observed that our algorithm provides slightly better results than the other tested methods

---

[2] Go to http://mx.nthu.edu.tw/~tsunghan/download/MVES_code.zip.

TABLE II
COMPARISON OF ENDMEMBER EXTRACTION ALGORITHMS ON A SYNTHETIC IMAGE WITH SIZE OF $N = 100 \times 100$ PIXELS AND $p = 5$ ENDMEMBERS CONTAINING NONPURE MINERAL SIGNATURES (MAXIMUM PURITY OF 0.8) FROM THE USGS LIBRARY UNDER DIFFERENT NOISE LEVELS. THE COMPUTATIONAL TIMES (SECONDS) ARE ALSO INCLUDED. ALL THE RESULTS ARE OBTAINED BY AVERAGING 30 INDEPENDENT RUNS

	VCA				MVES				MVC-NMF				MVSA			
dB	$\| \epsilon \|_F$	$r\epsilon$	SAD	time	$\| \epsilon \|_F$	$r\epsilon$	SAD	time	$\| \epsilon \|_F$	$r\epsilon$	SAD	time	$\| \epsilon \|_F$	$r\epsilon$	SAD	time
90	0.146	0.024	5.909	0.130	0.004	0.0	0.171	7.712	0.050	0.002	1.993	81.432	0.0004	0.0	0.023	1.120
70	0.157	0.028	6.307	0.129	0.004	0.0	0.189	9.369	0.060	0.002	1.849	79.467	0.0005	0.0	0.026	1.110
50	0.150	0.029	6.274	0.133	0.009	0.0	0.496	6.838	0.049	0.002	1.848	80.188	0.003	0.0	0.151	1.078
30	0.160	0.033	6.551	0.128	0.039	0.0	1.598	4.463	0.053	0.002	1.896	77.981	0.030	0.0	1.421	1.111



(a) $p = 3$ endmembers

(b) $p = 10$ endmembers

Fig. 3. Unmixing results for a simulation with nonpure pixels using different numbers of endmembers: (a) $p = 3$ and (b) $p = 10$ for VCA MVSA, MVC-NMF, and MVES algorithms, respectively, where $\mathbf{Y}$ denotes the spectral vectors $\mathbf{v}1 \equiv [1, 0, \ldots, 0]^T$ and $\mathbf{v}2 \equiv [0, 1, 0 \ldots, 0]^T$.

in terms of MSE reconstruction error and SAD. Concerning the computational time, MVSA is notably faster than MVES and MVC-NMF. A final aspect that should be underlined is that all algorithms obtained very good reconstruction error, particularly MVSA and MVES. Both algorithms obtained reconstruction error close to zero. This is expected due to the nonnegative constraint that both algorithms use which forces all pixels into the simplex and leads to very low reconstruction error.

B. No Pure Pixel-Based Experiments

In this section, we evaluate MVSA by assuming that no pure pixels exist in the considered image. The same experimental setting (based on the procedure described in [15]) was constructed as in the previous experiments, with a size of $N = 100 \times 100$ pixels and $p = 5$ endmembers. In order to make sure that there are no pure pixels in the simulated image, abundance fractions with *purities* [22] (i.e., maximum abundance fractions) greater than 0.8 are discarded in the simulation so that only mixed pixels exist. Table II shows the obtained results from the same aforementioned methods for the considered scene with different noise levels. As expected, the algorithms without the pure pixel assumption such as MVSA, MVES, and MVC-NMF largely outperform the pure pixel-based VCA algorithm. Another important observation is that, as it was the case in the previous experiment, the two minimum volume-based algorithms (MVES and MVSA) obtained very

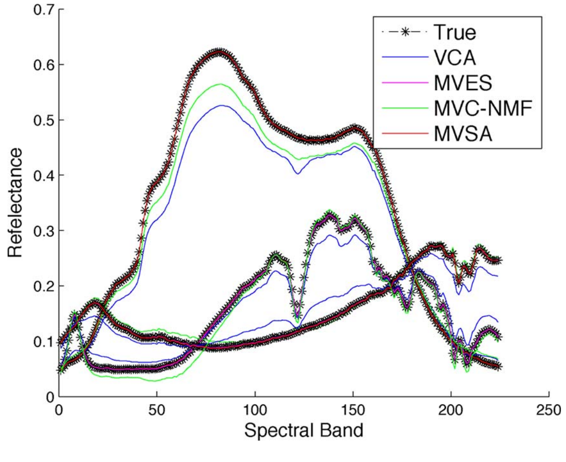


Fig. 4. Endmember signatures estimated by MVSA, VCA, MVES, and MVC-NMF in a simulation with SNR = 50 dB noise and nonpure pixels, which corresponds to the experiment reported in Fig. 3(a).

TABLE III
PROCESSING TIME (SECONDS) OBTAINED FOR THE PROPOSED MVSA ALGORITHM ON SYNTHETIC IMAGES WITH $N = 50 \times 50$, $N = 100 \times 100$, $N = 150 \times 150$, SNR = 70, dB AND MAXIMUM PURITY OF 0.8 FOR DIFFERENT NUMBERS OF ENDMEMBERS

	MVSA					MVES				
p	4	8	12	16	20	4	8	12	16	20
$N = 50 \times 50$	0.3	0.5	0.8	1.4	3.23	1.9	29	237	527	866
$N = 100 \times 100$	0.7	1.6	2.8	5.6	8.9	1	141	757	1759	3453
$N = 150 \times 150$	1.3	4.4	9	14.4	23.3	4.3	158	1778	4197	7959

low reconstruction errors, i.e., close to zero. This is due to the fact that, under the nonnegativity constraint, the minimum volume-based algorithms enclose all observations into the simplex. It can also be observed that, among the nonpure pixel-based algorithms, MVSA obtains the best results with respect to SAD reconstruction error and MSE for all considered noise levels. Concerning the computational time, MVSA is much faster than MVES and MVC-NMF.

For illustrative purposes Fig. 3 compares the four methods graphically, using a simulation with nonpure pixels, with (a) $p = 3$ and (b) $p = 10$ endmembers, $N = 100 \times 100$ spectral vectors, maximum purity of 0.8, and noise level of SNR = 50 dB. Finally, Fig. 4 shows the obtained spectral signatures after conducting the experiment reported in Fig. 3(a). These two figures reveal the quality of MVSA estimates with regard to those obtained by other algorithms.

C. Evaluation of the Efficiency

An important aspect in this experiment is to analyze the efficiency of the proposed MVSA algorithm from a computational viewpoint. In order to explore this issue, we now discuss the computational performance of Matlab implementations of MVSA (by us) and MVES (by the authors of [19]) using USGS library endmembers. All our experiments were conducted using the latest Intel Core I7 CPU and 32 GB of RAM. Notice that, here, we only report the results obtained by MVES and MVSA as both algorithms solve similar optimization problems but using a completely different strategy. Table III reports the processing time for problems with $N = 50 \times 50$, $N = 100 \times 100$, and $N = 150 \times 150$ pixels using different numbers of endmembers. In Table III, the number of endmembers p goes up to 20. This is a very difficult problem, and (as we mentioned in the previous experiment) it is difficult to have $p = 20$ endmembers in one given pixel or a local area. As a result, the main purpose of using $p = 20$ is to show the computational efficiency of our algorithm for problems with large scale. At this point, we also emphasize that, in our experiments, we have not considered purities lower than 0.8 since the probability of having an abundance larger than a given value of p vanishes as p increases. It can be seen that, in Table III, MVSA is very efficient for moderately large and complex problems, which would be impractical for other methods like MVES. For instance, it only took 23.3 s for $p = 20$ and $N = 150 \times 150$, which is prohibitive for the previous MVSA implementation [26] developed in Matlab, from the viewpoints of either RAM memory requirements and computational time. This problem is also extremely time consuming for the Matlab version of MVES distributed by the authors of the algorithm, as shown in Table III.

V. REAL DATA EXPERIMENTS

The scene used in our real data experiments is the well-known AVIRIS Cuprite data set, available online in reflectance units.[3] This scene has been widely used to validate the performance of endmember extraction algorithms. The portion used in experiments corresponds to a 250 × 191 pixel subset of the f970619t01p02r02 online data set in reflectance units.[4] The scene comprises 224 spectral bands between 0.4 and 2.5 μm, with a nominal spectral resolution of 10 nm. Prior to the analysis, bands 1–6, 105–115, 150–170, and 222–224 were removed due to water absorption and low SNR in those bands, leaving a total of 183 spectral bands, for which according to the HySime [28] algorithm we obtain an estimate of $p = 14$ endmembers. Here, we use HySime to estimate the number of endmembers as we believe that the result of HySime is quite accurate judging from the computed low noise. The Cuprite site is well understood mineralogically and has several exposed minerals of interest, all included in the USGS library considered in experiments, denoted splib06[5] and released in September 2007. In our experiments, we use spectra obtained from this library (convolved and downsampled to AVIRIS wavelengths) in order to substantiate the quality of the endmembers derived by MVSA and compare them with those produced by other algorithms. For illustrative purposes, Fig. 5(a) shows a mineral map produced in 1995 by USGS, in which the Tricorder 3.3 software product was used to map different minerals present in the Cuprite mining district.[6] The 250 × 190 pixel subscene used in our experiments is shown in Fig. 5. It should be noted that all experiments with this subscene have been performed in a desktop personal computer with an Intel Core I5 CPU and 4 GB of RAM. Concerning the parameters involved in the considered algorithms we consequentially follow the settings in the simulated experiments. Regarding the affine projection we have used the projective projection instead of the affine one (see [1]) as the former works slightly better in this example.

A fundamental assumption in the minimum volume unmixing algorithms is that, in a given data set, the spectral samples

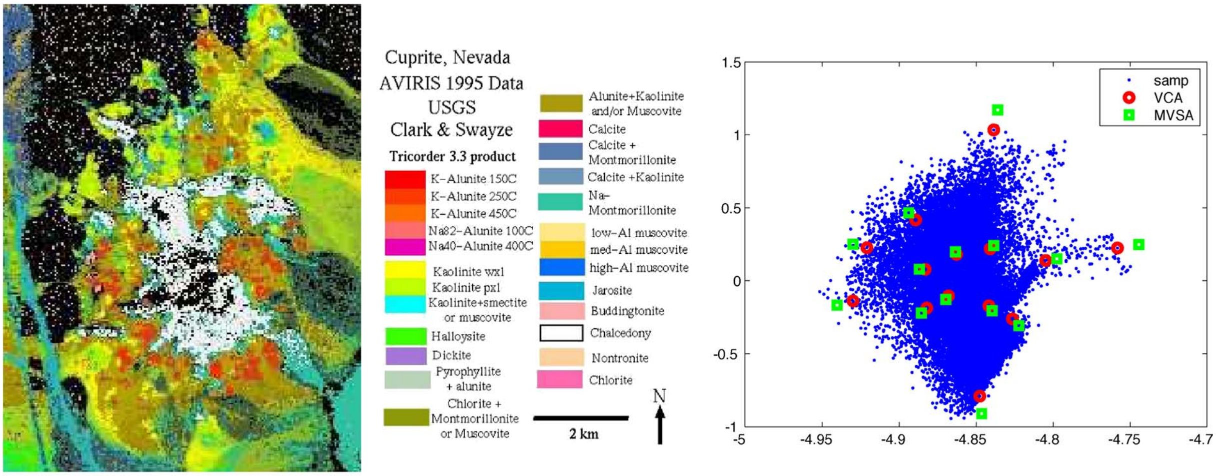
---

[3] http://aviris.jpl.nasa.gov/html/aviris.freedata.html

[4] Subscene is available from http://www.lx.it.pt/%7ebioucas/code.htm

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

[6] http://speclab.cr.usgs.gov/cuprite95.tgif.2.2um_map.gif



(a) (b)

Fig. 5. (a) 250 × 190 pixel subscene used in our experiments, showing the location of different minerals in the Cuprite mining district in Nevada. The map is available online. (b) Projection of the data on the first two PCA components.

belong to a simplex and that there are at least $p-1$ samples on, or in the neighborhood of, each simplex facet [1]. It happens that the spatial distribution of Cuprite spectral vectors does not comply with that assumption, which renders the associated unmixing an ill-posed problem. The fact is that, given an estimated simplex, the large majority of the spectral vectors are outside of that simplex; it is a clear symptom of that ill-posedness. For example, the simplexes estimated by VCA [15], N-FINDR [16], and the successive volume maximization [31] leave respectively 47733, 47732, and 47734 samples outside the simplex in a maximum of 47750 samples, which is the size of the data set. A similar pattern is observed using minimum volume-based algorithms.

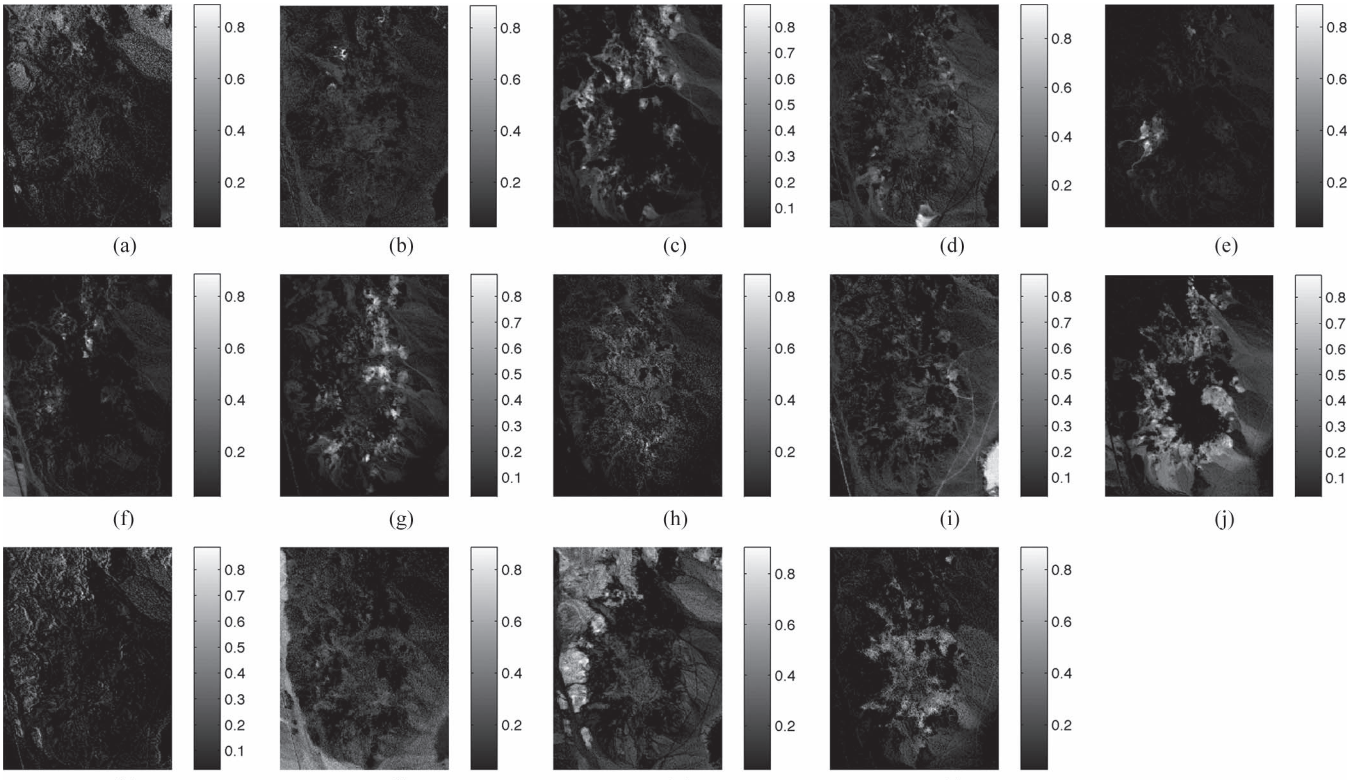
We conclude therefore that the spatial distribution of Cuprite spectral vectors is far away from that envisaged in the minimum volume unmixing framework, thus precluding those methods to perform optimally. In order to regularize the facets of the simplex, we conceived a very simple procedure that nevertheless produces useful results. We start by running VCA t times and retain the simplex of maximum volume. In the case of VCA, this makes sense given the random directions that this algorithm uses to find the extremes of the simplex. For $t = 30$, this procedure takes just 2 s in a standard personal computer. Next, we project the data set in an inflated simplex obtained by allowing the abundances to take negative values. That is, we solve a modified fully constrained least square (MFLCS) problem with the constraints $\alpha_i \geq -\varepsilon$, where $\varepsilon > 0$ and $\mathbf{1}_p^T \boldsymbol{\alpha}_i = 1$, for $i = 1, \ldots, N$. The MFLCS is solved by a minor modification of the SUNSAL algorithm available in [32]. We apply then MVSA to the regularized data $\mathbf{Y}_{\text{reg}} = \widehat{\mathbf{A}}_{\text{vca}} \widehat{\mathbf{X}}_t$, where $\widehat{\mathbf{A}}_{\text{vca}}$ is the mixing matrix estimated by VCA and $\widehat{\mathbf{X}}_t$ is the result of the MFLCS just described. The complete procedure is available in our online demo: http://www.lx.it.pt/%7ejun/DemoMVSA.zip.

Fig. 5(b) shows a scatterplot of the original data jointly with the VCA and the MVSA endmembers for $\epsilon = 0.01$. It is clear that the simplex defined by MVSA is an enlarged version of the one defined by VCA. Table IV shows the processing time for MVSA, MVES, and MVC-NMF, respectively. It is remarkable that the advantages of MVSA in terms of efficiency are significant as, for the considered data set, MVSA took less than 3 min to perform the computation while MVES spent around 7 h and MVC-NMF took around 50 min in the considered environment. Such computational efficiency makes MVSA more applicable in real scenarios.

TABLE IV
PROCESSING TIME (SECONDS) TAKEN BY MVSA, MVES, AND MVC-NMF, RESPECTIVELY, WHEN PROCESSING THE CONSIDERED AVIRIS CUPRITE DATA SET

Method	MVSA	MVES	MVC-NMF
Time (Seconds)	149.93	24909	2896

For illustrative purposes Fig. 6 shows the abundance maps obtained by the MVSA algorithm, where the minerals are identified by visual interpretation of the estimated abundances with regard to the ground-truth map in Fig. 5. In addition, Fig. 7 shows the spectral signatures of the estimated endmembers. This figure reveals a good match between the real and estimated ones. The individual abundance maps estimated by MVES, MVC-NMF, and VCA are not presented here due to space considerations. Furthermore, we refer to [19], in which the same real data were analyzed by MVES, and to [24] where a portion of the current data set was processed by MVC-NMF. Overall, it has been observed that the algorithms produce some abundance maps that are similar to each other. Although the results provided by HySime are reasonably judged from the computed low noise, it is possible that the number of endmembers is overestimated which affects the performance of MVSA and MVES. On the other hand, the abundance maps estimated by MVC-NMF were found to be more distinct from each other. This indicates that the algorithms are sensitive to the estimation of the number of endmembers, which, in this paper, is performed by an external algorithm. In order to have a fair comparison of algorithms, we decided to report results with

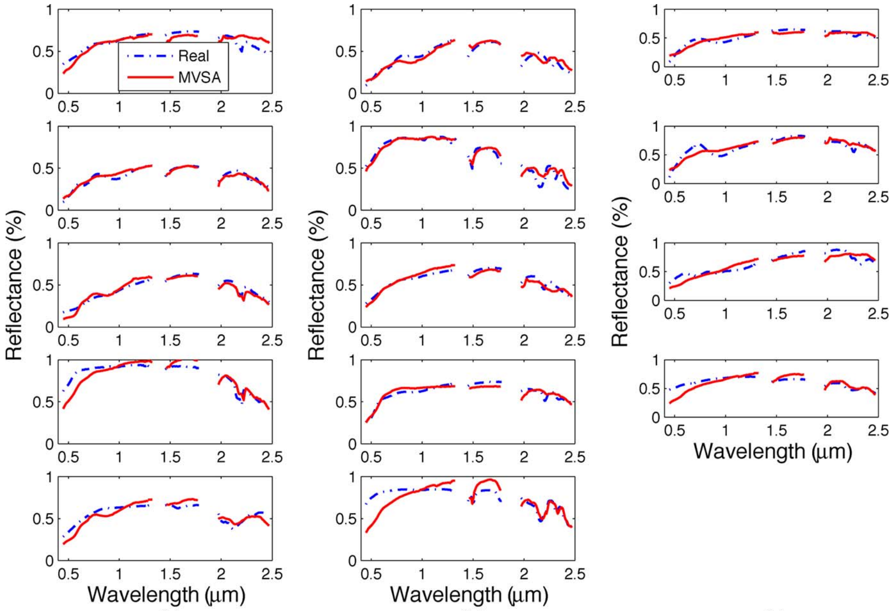


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n)

Fig. 6. Abundance fraction maps estimated by the proposed MVSA algorithm: (a) Chlorapatite WS423. (b) Nontronite NG-1.a. (c) Kaolin/Smect KLF508 85%K. (d) Kaolinite KGa-2 (pxyl). (e) Buddingtonite GDS85 D-206. (f) Nontronite SWa-1.a. (g) Alunite GDS84 Na03. (h) Montmorillonite+Illi CM42. (i) Montmorillonite+Illi CM37. (j) Alunite AL706 Na__. (k) Jarosite WS368 Pb. (l) Jarosite JR2501 K. (m) Chlorite SMR-13.e < 30um. (n) Chalcedony CU91-6A.



(a) (b) (c)

Fig. 7. Endmember signatures in the USGS library and the endmember estimates obtained by our MVSA algorithm. The corresponding signatures are the following (from top to bottom): (a) Chlorapatite WS423, Nontronite NG-1.a, Kaolin/Smect KLF508 85%K, Kaolinite KGa-2 (pxyl), and Buddingtonite GDS85 D-206; (b) Nontronite SWa-1.a, Alunite GDS84 Na03, Montmorillonite+Illi CM42, Montmorillonite+Illi CM37, and Alunite AL706 Na__; and (c) Jarosite WS368 Pb, Jarosite JR2501 K, Chlorite SMR-13.e < 30um, and Chalcedony CU91-6A.

$p = 14$ (the HySime estimate) for all the compared methods in this experiment. Overall, the experimental results reported in this section reveal that the proposed MVSA can produce similar results to those provided by other state-of-the-art algorithms like MVES or MVC-NMF, but in a more computationally efficient fashion.

VI. CONCLUSION AND FUTURE LINES

In this paper, we have described an MVSA algorithm for unsupervised hyperspectral unmixing and its efficient implementation using the interior point method. This algorithm is a representative method of a class of algorithms for endmember extraction that does not need the presence of pure pixels in the hyperspectral data. Despite the interest and proven effectiveness of the method in toy examples and experiments with small data sets, the algorithm had rarely been used in real applications due to its computational complexity, resulting from the fact that the involved optimization problem was very difficult to handle. In this regard, one of the main contributions of this paper is the presentation of a series of strategies in order to lighten the computational load of MVSA, making it appealing for real hyperspectral imaging applications. Another contribution has been the detailed comparison of MVSA to other algorithms (with and without the pure pixel assumption) using both simulated and real data sets. Our experiments demonstrate that, with the presented modifications, MVSA is competitive with other state-of-the-art solutions in terms of endmember identification and spectral unmixing accuracy and also in terms of computational complexity, thus allowing the application of the algorithm to problems characterized by a high number of endmembers (i.e., complex scenes) and also by a high number of pixels (i.e., large scenes). In future work, we will include a soft constraint in the proposed MVSA algorithm in order to make it more robust to noise and outliers. Furthermore, the proposed algorithm can also be adapted to extract endmember bundles using the framework described in [33] to address issues of endmember variability.

ACKNOWLEDGMENT

The authors would like to thank the developers of the minimum-volume enclosing simplex, minimum volume constrained nonnegative matrix factorization, and vertex component analysis algorithms for sharing their Matlab implementations. The authors also thank the associate editor and the four anonymous reviewers for their outstanding comments and suggestions which greatly improved the technical quality and presentation of this manuscript.

REFERENCES

[1] J. Bioucas-Dias *et al.*, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.

[2] N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 44–57, Jan. 2002.

[3] J. B. Adams, M. O. Smith, and P. E. Johnson, "Spectral mixture modeling: A new analysis of rock and soil types at the Viking Lander 1 site," *J. Geophys. Res.*, vol. 91, no. B8, pp. 8098–8112, Jul. 1986.

[4] J. J. Settle and N. A. Drake, "Linear mixing and the estimation of ground cover proportions," *Int. J. Remote Sens.*, vol. 14, no. 6, pp. 1159–1177, Apr. 1993.

[5] M. Petrou and P. G. Foschi, "Confidence in linear spectral unmixing of single pixels," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 1, pp. 624–626, Jan. 1999.

[6] J. Settle, "On the effect of variable endmember spectra in the linear mixture model," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 2, pp. 389–396, Feb. 2006.

[7] C. Borel and S. Gerstl, "Nonlinear spectral mixing models for vegetative and soil surfaces," *Remote Sens. Environ.*, vol. 47, no. 3, pp. 403–416, Mar. 1994.

[8] W. Fan, B. Hu, J. Miller, and M. Li, "Comparative study between a new nonlinear model and common linear model for analysing laboratory simulated-forest hyperspectral data," *Int. J. Remote Sens.*, vol. 30, no. 11, pp. 2951–2962, Jun. 2009.

[9] K. J. Guilfoyle, M. L. Althouse, and C.-I. Chang, "A quantitative and comparative analysis of linear and nonlinear spectral mixture models using radial basis function neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 10, pp. 2314–2318, Oct. 2001.

[10] Y. Altmann, A. Halimi, N. Dobigeon, and J.-Y. Tourneret, "Supervised nonlinear spectral unmixing using a post-nonlinear mixing model for hyperspectral imagery," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 3017–3025, Jun. 2012.

[11] W.-K. Ma *et al.*, "A signal processing perspective on hyperspectral unmixing: Insights from remote sensing," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 67–81, Jan. 2014.

[12] M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2014–2039, Jun. 2011.

[13] L. Wang and X. Jia, "Integration of soft and hard classifications using extended support vector machines," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 3, pp. 543–547, Jul. 2009.

[14] M. Parente and A. Plaza, "Survey of geometric and statistical unmixing algorithms for hyperspectral images," in *Proc. 2nd IEEE WHISPERS*, 2010, pp. 1–4.

[15] J. Nascimento and J. Bioucas-Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 4, pp. 898–910, Apr. 2005.

[16] M. E. Winter, "N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data," in *Proc. SPIE Imaging Spectrom. V*, 2003, vol. 3753, pp. 266–277.

[17] A. Plaza, P. Martinez, R. Perez, and J. Plaza, "A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 3, pp. 650–663, Mar. 2004.

[18] M. Craig, "Minimum-volume transforms for remotely sensed data," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 3, pp. 542–552, May 1994.

[19] T.-H. Chan, C.-Y. Chi, Y.-M. Huang, and W.-K. Ma, "A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmixing," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4418–4432, Nov. 2009.

[20] A. Ambikapathi, T.-H. Chan, W.-K. Ma, and C.-Y. Chi, "Chance-constrained robust minimum-volume enclosing simplex algorithm for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4194–4209, Nov. 2011.

[21] J. Bioucas-Dias, "A variable splitting augmented Lagragian approach to linear spectral unmixing," in *Proc. 1nd IEEE Workshop Hyperspectr. Image Signal Process., Evol. Remote Sens.*, 2009, pp. 1–4.

[22] M. Berman *et al.*, "ICE: A statistical approach to identifying endmembers in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 10, pp. 2085–2095, Oct. 2004.

[23] A. Zare and P. Gader, "Sparsity promoting iterated constrained endmember detection in hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 3, pp. 446–450, Jul. 2007.

[24] L. Miao and H. Qi, "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 3, pp. 765–777, Mar. 2007.

[25] J. M. Bioucas-Dias *et al.*, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.

[26] J. Li and J. Bioucas-Dias, "Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data," *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2008, vol. 3, pp. 250–253.

[27] J. Nocedal and S. J. Wright, *Numerical Optimization*. Berlin, Germany: Springer-Verlag, 2006.

[28] J. Bioucas-Dias and J. Nascimento, "Hyperspectral subspace identification," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 8, pp. 2435–2445, Aug. 2008.

[29] K. Lange, D. Hunter, and I. Yang, "Optimization transfer using surrogate objective functions," *J. Comput. Graph. Stat.*, vol. 9, no. 1, pp. 1–59, Mar. 2000.

[30] R. N. Clark *et al.*, "USGS digital spectral library splib06a," in *Survey, Digital Data Series 231*, vol. 1, Denver, CO, USA: U.S. Geological Survey, 2007.

[31] T.-H. Chan, W.-K. Ma, A. Ambikapathi, and C.-Y. Chi, "A simplex volume maximization framework for hyperspectral endmember extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4177–4193, Nov. 2011.

[32] J. Bioucas-Dias and M. Figueiredo, "Alternating direction algorithms for constrained sparse regression: Application to hyperspectral unmixing," in *Proc. 2nd IEEE WHISPERS*, 2010, pp. 1–4.

[33] B. Somers, M. Zortea, A. Plaza, and G. P. Asner, "Automated extraction of image-based endmember bundles for improved spectral unmixing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 396–408, Apr. 2012.

Jun Li (M'13) received the B.S. degree in geographic information systems from Hunan Normal University, Changsha, China, in 2004, the M.E. degree in remote sensing from Peking University, Beijing, China, in 2007, and the Ph.D. degree in electrical engineering from the Instituto de Telecomunicações, Instituto Superior Técnico (IST), Universidade Técnica de Lisboa, Lisbon, Portugal, in 2011.

From 2007 to 2011, she was a Marie Curie Research Fellow with the Departamento de Engenharia Electrotécnica e de Computadores and the Instituto de Telecomunicações, IST, Universidade Técnica de Lisboa, in the framework of the European Doctorate for Signal Processing (SIGNAL). She has also been actively involved in the Hyperspectral Imaging Network, a Marie Curie Research Training Network involving 15 partners in 12 countries and intended to foster research, training, and cooperation on hyperspectral imaging at the European level. Since 2011, she has been a Postdoctoral Researcher with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, Cáceres, Spain. Her research interests include hyperspectral image classification and segmentation, spectral unmixing, signal processing, and remote sensing.

Dr. Li is an Associate Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. She has been a reviewer of several journals, including the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, *Pattern Recognition, Optical Engineering, Journal of Applied Remote Sensing*, and *Inverse Problems and Imaging*.

Alexander Agathos received the Ph.D. degree in computer science from the University of the Aegean, Greece.

He is a Researcher at the Computer Science Department of the West University of Timisoara, Timisoara, Romania. He is currently working on the EU-funded project HOST under the coordination of Prof. Dana Petcu. His research interests involve computer graphics, image processing, and parallel and distributed programming with emphasis on developing algorithms for accelerators like GPUs and Xeon Phi. He has worked as a PostDoc in Tel Aviv University. He has published several papers on a plethora of subjects involving computer graphics, image processing, and 3-D computer vision.

Daniela Zaharie is a Professor of computer science at the West University of Timisoara, Timisoara, Romania. Her main research topics are data mining, machine learning, evolutionary computing, computational biology, and image processing. She was recently involved in national and international projects related to nature-inspired metaheuristics, medical data mining, remote sensing, and high performance computing. Since 2005, she has been a coeditor of the proceedings of the International Symposium of Symbolic and Numeric Algorithms for Scientific Computing.

José M. Bioucas-Dias (S'87–M'95) received the EE, M.Sc., Ph.D., and "Agregado" degrees in electrical and computer engineering from Instituto Superior Técnico (IST), the engineering school of the Technical University of Lisbon (TULisbon), Lisbon, Portugal, in 1985, 1991, 1995, and 2007, respectively.

Since 1995, he has been with the Department of Electrical and Computer Engineering, IST, where he was an Assistant Professor from 1995 to 2007 and has been an Associate Professor since 2007. Since 1993, he is also a Senior Researcher with the Pattern and Image Analysis group of the Instituto de Telecomunicações, which is a private nonprofit research institution. His research interests include inverse problems, signal and image processing, pattern recognition, optimization, and remote sensing.

Dr. Bioucas-Dias was an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS (1997–2000), and he is an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. He was a Guest Editor of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING for the *Special Issue on Spectral Unmixing of Remotely Sensed Data* and of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING for the *Issue on Hyperspectral Image and Signal Processing*, and he is a Guest Editor of the IEEE SIGNAL PROCESSING MAGAZINE for the *Special Issue on Signal and Image Processing in Hyperspectral Remote Sensing*. He was the General Cochair 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.

Antonio Plaza (M'05–SM'07–F'15) was born in Caceres, Spain, in 1975. He is an Associate Professor (with accreditation for Full Professor) with the Department of Technology of Computers and Communications, University of Extremadura, Cáceres, Spain, where he is the Head of the Hyperspectral Computing Laboratory (HyperComp). His main research interests comprise hyperspectral data processing and parallel computing of remote sensing data. He has been the advisor of 12 Ph.D. dissertations and more than 30 Ms.C. dissertations. He was the Coordinator of the Hyperspectral Imaging Network, a European project with a total funding of 2.8 million euro. He has authored more than 400 publications, including 140 journal papers (90 in IEEE journals), 20 book chapters, and over 240 peer-reviewed conference proceeding papers (94 in IEEE conferences). He has edited a book on *High-Performance Computing in Remote Sensing* (CRC Press/Taylor and Francis) and guest edited eight special issues on hyperspectral remote sensing for different journals.

Dr. Plaza is a fellow of IEEE "for contributions to hyperspectral data processing and parallel computing of Earth observation data." He is a recipient of the recognition of Best Reviewers of the IEEE Geoscience and Remote Sensing Letters (in 2009) and a recipient of the recognition of Best Reviewers of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING (in 2010), a journal for which he served as Associate Editor in 2007–2012. He is also an Associate Editor for IEEE ACCESS and was a member of the Editorial Board of the IEEE Geoscience and Remote Sensing Newsletter (2011–2012) and the IEEE GEOSCIENCE AND REMOTE SENSING MAGAZINE (2013). He was also a member of the steering committee of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING (JSTARS). He is a recipient of the 2013 Best Paper Award of the JSTARS journal and a recipient of the most highly cited paper (2005–2010) in the Journal of Parallel and Distributed Computing. He received best paper awards at the IEEE International Conference on Space Technology and the IEEE Symposium on Signal Processing and Information Technology. He is a recipient of the Best Ph.D. Dissertation award at the University of Extremadura, a recognition also received by five of his Ph.D. students. He served as the Director of Education Activities for the IEEE Geoscience and Remote Sensing Society (GRSS) in 2011–2012 and is currently serving as the President of the Spanish Chapter of IEEE GRSS (since November 2012). He has served as a proposal evaluator for the European Commission, the National Science Foundation, the European Space Agency, the Belgium Science Policy, the Israel Science Foundation, and the Spanish Ministry of Science and Innovation. He has reviewed more than 500 manuscripts for over 50 different journals. He is currently serving as the Editor-in-Chief of the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING journal.

Xia Li is a Professor and the Director of the Centre for Remote Sensing and Geographical Information Sciences, School of Geography and Planning, Sun Yat-sen University. He is also a Guest Professor in the Department of Geography at University of Cincinnati. He got the ChangJiang scholarship and the award of the Distinguished Youth Fund of NSFC in 2005. He is currently on the editorial boards of international journals, including the *International Journal of Geographical Information Science, Computers, Environment and Urban Systems* and *GeoJournal*. He is the associated editor of a Chinese journal, *Tropical Geography*. He has about 200 articles, of which many appeared in international journals. His major research interests include the development of urban cellular automata and agent-based models for simulating urban growth and land use changes. Some of his research works focus on the development of methodologies for calibrating these simulation models. Recently, he has carried out research on using ant intelligence for spatial optimization. His other research works include the use of radar remote sensing for urban applications. His papers are widely published in top international GIS and remote sensing journals, such as the *Remote Sensing of Environment, International Journal of Remote Sensing, Photogrammetric Engineering & Remote Sensing, International Journal of Geographical Information Science, Environment and Planning A, and Environment and Planning B*.