IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 3, 2019, accepted September 18, 2019, date of publication September 23, 2019,
date of current version October 4, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2943110*

# Joint-Sparse-Blocks Regression for Total Variation Regularized Hyperspectral Unmixing

**JIE HUANG, TING-ZHU HUANG, XI-LE ZHAO, AND LIANG-JIAN DENG**
School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, China

Corresponding author: Jie Huang (happyjie07mo@163.com)

This work was supported in part by the NSFC under Grant 61772003, Grant 61876203, and Grant 61702083, in part by the Science Strength Promotion Programme of UESTC, and in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2019J093.

**ABSTRACT** Sparse unmixing has attracted much attention in recent years. It aims at estimating the fractional abundances of pure spectral signatures in mixed pixels in hyperspectral images. To exploit spatial-contextual information present in the scene, the total variation (TV) regularization is incorporated into the sparse unmixing formulation, promoting adjacent pixels having similar not only endmembers but also fractional abundances, and thus having similar structural sparsity. It is therefore hoped to impose joint sparsity, instead of classic single sparsity, on these adjacent pixels to further improve the unmixing performance. To this end, we include the joint-sparse-blocks regression into the TV spatial regularization framework and present a new unmixing algorithm, termed *joint-sparse-blocks unmixing via variable splitting augmented Lagrangian and total variation* (JSBUnSAL-TV). In particular, a reweighting strategy is utilized to enhance sparsity along lines within each block. Simulated and real-data experiments show the advantages of the proposed algorithm.

**INDEX TERMS** Hyperspectral images, spectral unmixing, total variation regularization, joint-sparse-blocks regression.

## I. INTRODUCTION

Spectral unmixing is an important and challenging technique for hyperspectral images (HSIs) [1], [2]. It aims at identifying a set of pure spectral signatures, called *endmembers*, and estimating the corresponding fractions, called *abundances*. Linear mixture model (LMM) has been widely adopted for spectral unmixing due to its simplicity and analytically tractable solutions. It assumes that the spectral collected by the imaging spectrometer can be expressed in the form of a linear combination of endmembers, weighted by their corresponding abundances. Along this line, many endmember identification algorithms extract the pure endmembers under the pixel purity assumption that there is at least one pure pixel per endmember in the data [3]–[5]. Hyperspectral signatures, however, in many cases, are highly mixed so that the pixel purity assumption may not be valid, although it has practical advantages such as ease of implementation and flexibility in diverse applications [1]. To address the issue, many other endmember extraction algorithms have been proposed without assuming the presence of pure signatures in the input data [6]–[8]. These algorithms, however, may provide virtual endmembers with no physical meaning. Alternatively, nonnegative matrix factorization (NMF) is a popular technique to find the endmember signatures and associated proportions with more physical meaningful unmixing results, even if there are no pure pixels existing and no available spectral library [9]–[14].

As more spectral libraries become publicly available, a new perspective to tackle the problems of virtual endmembers and pure pixel assumption is to model mixed pixels observations as linear combinations of spectra in a given (potentially very large) spectral library. That said, each mixed pixel in HSIs is potentially composed of only a few endmembers, compared with the large and available spectral library. Sparsity is then incorporated into spectral unmixing, leading to a plethora of abundance estimation algorithms [15]–[22]. Also, in Bayesian framework, suitable sparsity inducing prior distributions are adopted for fractional abundances [23]–[25]. These sparse unmixing algorithms need not extract endmembers from the input data and provide abundance estimations more accurately. Since the sparsity is imposed on each single

The associate editor coordinating the review of this manuscript and approving it for publication was Yi Zhang.

pixel, we therefore call it as *single sparsity* to distinguish other sparsity characters mentioned later.

Based on the overcomplete spectral dictionary, the collaborative (also called "joint" or "row") sparse regression framework assumes that pixels in a region share the same support set of endmembers. There exist several collaborative schemes. All pixels in the data set are assumed to share the same endmembers in [26], [27], whereas the joint sparsity is enforced on pixels in a local window in [28], [29] and is then applied with general segments in [11], [30]. Recently, a joint-sparse-blocks regression model imposes the joint sparsity on local blocks [31], which could contain fewer pixels, compared with a small $3 \times 3$ local window. Assume that some materials in the spectral library are known to exist in the scene, an algorithm, called the sparse unmixing using spectral a priori information (SUnSPI), simultaneously exploits the joint sparsity and the single sparsity [32]. In addition, a centralized collaborative framework couples the collaborative sparse unmixing and abundance estimation error reduction together [33].

Besides the sparsity assumption, spatial information between each pixel and its neighbors is exploited to further improve spectral unmixing performance [9], [34]–[37]. Typically, the total variation (TV) spatial regularization promotes piecewise constant transitions in each abundance map for the same endmember among neighboring pixels [12]–[14], [21], [27], [34], [38]–[41]. Also, using a low-rankness constraint is another stimulating way to exploit the spatial information of HSIs [28], [31], [42]–[44]. The spatial correlation among pixels translates into a linear dependence among their corresponding abundance vectors. This follows a low-rank abundance matrix. The low-rankness constraint has been incorporated with the TV constraint [45], or with the single sparsity [42], or jointly with the row sparsity and the TV spatial regularizer for unmixing problem [46]. Recently, the low-rankness constraint and the joint-sparse-blocks structure have been simultaneously imposed on abundances in [31]. Multiple constraints simultaneously enforced on abundances have greatly improved the unmixing performance.

In this paper, we propose to simultaneously enhance the spatial consistency by using the TV regularizer and enhance the structural sparsity by using the joint-sparse-blocks representation for the hyperspectral unmixing problem. Recall that the TV term promotes piecewise smooth in abundance maps, following that adjacent pixels have both similar mixing endmembers and similar abundance fractions. It leads to similar sparsity pattern in these adjacent pixels. Thus, imposing the local joint sparsity, instead of the classic single sparsity as in [34], on these pixels is expected to better describe the sparsity structure and therefore, to improve unmixing performance. Following this line, we adopt the joint-sparse-blocks regression framework and introduce a new unmixing algorithm called *joint-sparse-blocks unmixing via variable splitting augmented Lagrangian and total variation* (JSBUnSAL-TV). The algorithm is under the classic *alternating direction method of multipliers* (ADMM) framework. In particular, we adopt a two-level reweighting strategy to enhance the sparsity on lines within each block, similarly as in [31]. Simulated and real-data experiments demonstrate the effectiveness of the proposed algorithm.

Finally, we note that both JSBUnSAL-TV and the algorithm in [31] exploit the local joint sparsity property. In addition, the former via TV promotes the piecewise constant transitions in the fractional abundance of the same endmember among neighboring pixels. Instead the latter adopts the low-rank representation, assuming that the correlation among pixels' spectral signatures is reflected as linear dependence among their abundance vectors. Apparently, the TV regularizer is imposed on each abundance map reshaped from one row of the whole abundance matrix, whereas the low-rank constraint is imposed on local abundance matrices corresponding to small sliding square windows. Here we mainly adopt the TV regularizer to impose spatial consistency on each abundance map.

The rest of the paper is organized as follows. Section II briefly reviews sparse unmixing models. In section III we derive our unmixing algorithm called JSBUnSAL-TV. The effectiveness of the proposed algorithm is demonstrated by both simulated experiments in section IV-B and a real-data experiment in section IV-C. Finally, section V gives some concluding remarks.

## II. SPARSE UNMIXING MODEL

Let $\mathbf{Y} \in \mathbb{R}^{L\times n}$ denote the observed matrix of a hyperspectral image with $L$ spectral bands and $n$ pixels. Let $\mathbf{A} \in \mathbb{R}^{L\times m}$ denote the dictionary with $m$ spectral signatures. The LMM can be described as follows:

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{N}, \tag{1}$$

where $\mathbf{X} \in \mathbb{R}^{m\times n}$ is the fractional abundance matrix whose columns correspond with the abundance vectors of $n$ pixels, and $\mathbf{N} \in \mathbb{R}^{L\times n}$ is an independent and identically distributed (i.i.d.) zero-mean Gaussian noise matrix. Due to physical background, the so called *abundance nonnegativity constraint* (ANC) and the *abundance sum-to-one constraint* (ASC), i.e.,

$$\mathbf{X} \geq 0, \quad 1^T\mathbf{X} = 1^T, \tag{2}$$

respectively, are often imposed on the abundance coefficients in $\mathbf{X}$ [47]. We note that $\mathbf{X} \geq 0$ is considered elementwise and 1 is a column vector of 1's. Nevertheless, we relax the ASC to focus on the exploitation of structural characters of $\mathbf{X}$, similarly as in [26], [34], [42]; see more details in [15].

The classic sparse regression model for hyperspectral unmixing problem is as follows:

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \lambda \|\mathbf{X}\|_{1,1}, \quad \text{subject to } \mathbf{X} \geq 0, \tag{3}$$

where $\|\mathbf{X}\|_{1,1} = \sum_{i=1}^{m}\sum_{j=1}^{n}|x_{i,j}|$ is the $\ell_1$ norm of $\mathbf{X}$, $x_{i,j}$ denotes the $(i,j)$th element of $\mathbf{X}$, and $\lambda \geq 0$ is the

regularization parameter. A sparse unmixing by variable splitting and augmented Lagrangian (SUnSAL) algorithm is introduced to solve the above model in [15].

The collaborative sparse regression framework has been presented in [26] to encourage that all pixels in the data set share the same support set. The optimization model is as follows:

$$\min_{\mathbf{X}} \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \lambda \|\mathbf{X}\|_{2,1}, \quad \text{subject to } \mathbf{X} \geq 0, \tag{4}$$

where $\|\mathbf{X}\|_{2,1} = \sum_{i=1}^{m} \left\|\boldsymbol{x}^{[i]}\right\|_2$ is the $\ell_{2,1}$ norm of $\mathbf{X}$, $\boldsymbol{x}^{[i]}$ is the $i$th row of $\mathbf{X}$, and $\lambda \geq 0$ is the regularization parameter. The model is solved by the collaborative SUnSAL (CLSUnSAL) algorithm in [26].

The SUnSPI model assumes that some materials in the spectral library are known to exist in the hyperspectral scene and others may or may not be active. Suppose that $\mathbf{S} = \{1, \cdots, m\}$ is the set of the indices of all the spectral signatures in the spectral library, $\mathbf{P} \subset \mathbf{S}$ is the set of the indices corresponding to the known materials, and $\mathbf{S}/\mathbf{P} = \{i \in \mathbf{S} | i \notin \mathbf{P}\}$. Then the SUnSPI model simultaneously imposes sparsity on $\mathbf{X}$ and joint sparsity on $\mathbf{X}^{\mathbf{S}/\mathbf{P}}$ and leads to an optimization model as follows:

$$\min_{\mathbf{X}\geq 0} \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \lambda_S \|\mathbf{X}\|_{1,1} + \lambda_P \sum_{i\in\mathbf{S}/\mathbf{P}} \left\|\boldsymbol{x}^{[i]}\right\|_2, \tag{5}$$

where $\lambda_S \geq 0$ and $\lambda_P \geq 0$ are regularization parameters.

A joint-sparse-blocks regression model assumes that the $\ell_{2,1}$ prior only promotes sparsity in the activation of the endmembers within each block [31]. For this purpose, the fractional abundance matrix is first partitioned as

$$\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_s], \tag{6}$$

where each column block $\mathbf{X}_j \in \mathbb{R}^{m\times d_j}$, for $j = 1, \ldots, s$, $\sum_{j=1}^{s} d_j = n$, and block number $s$ is a positive integer for $1 \leq s \leq n$. Then each $\mathbf{X}_j$ is assumed joint-sparse. The resulting $\ell_{2,1}$-blocks regularized model becomes:

$$\min_{\mathbf{X}\geq 0} \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \lambda \sum_{j=1}^{s} \left\|\mathbf{X}_j\right\|_{2,1}, \tag{7}$$

where $\lambda \geq 0$ is the regularization parameter. Clearly, the model reduces to the SUnSAL model in (3) if $s = n$ and reduces to the CLSUnSAL model in (4) if $s = 1$.

## III. THE JSBUNSAL-TV ALGORITHM

### A. JOINT-SPARSE-BLOCKS REGRESSION FOR TV REGULARIZED UNMIXING

Besides exploiting sparsity in abundances, simultaneously taking spatial correlation between each pixel and its neighbors into account further improves the abundance estimation performance. The TV spatial regularization is incorporated into classic sparse unmixing scheme and provides promising abundance estimation results [12], [21], [27], [34], [38]–[40], [48]. It promotes spatial homogeneity of the same endmember among neighboring pixels. Therefore, it is hoped to impose a joint sparse structure on these pixels to improve the unmixing performance. Notice that the joint-sparse-blocks structure encourages that pixels in each local block share the same sparse structure, taking both sparsity and spatial information into consideration. Thus, we propose to include the joint-sparse-blocks regression into the TV regularization for abundance estimation. The optimization problem can be written as

$$\min_{\mathbf{X}\geq 0} \frac{1}{2} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 + \lambda \sum_{j=1}^{s} \left\|\mathbf{X}_j\right\|_{2,1} + \lambda_{TV} TV(\mathbf{X}), \tag{8}$$

where

$$TV(\mathbf{X}) \equiv \sum_{\{i,j\}\in\epsilon} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_1, \tag{9}$$

$\boldsymbol{x}_i$ (respectively, $\boldsymbol{x}_j$) denotes the $i$th (respectively, $j$th) column of $\mathbf{X}$, $\epsilon$ denotes the set of horizontal and vertical neighbors in the image, and $\lambda \geq 0$ and $\lambda_{TV} \geq 0$ are regularization parameters.

Let $\mathbf{H}_h : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ and $\mathbf{H}_v : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ be the corresponding discrete gradient operators in the horizontal and vertical directions, respectively. Here we assume periodic boundaries when computing the differences. Define

$$\mathbf{H}\mathbf{X} \equiv \begin{pmatrix} \mathbf{H}_h\mathbf{X} \\ \mathbf{H}_v\mathbf{X} \end{pmatrix}, \tag{10}$$

then clearly $TV(\mathbf{X}) = \|\mathbf{H}\mathbf{X}\|_{1,1}$. In addition, to enhance sparsity along rows in each block in (8), we replace the $\ell_{2,1}$ norm of $\mathbf{X}_j$ by a *weighted* $\ell_{2,1}$ norm defined as

$$\left\|\mathbf{X}_j\right\|_{\boldsymbol{w}_j,2,1} = \sum_{i=1}^{m} w_{i,j} \left\|\mathbf{X}_j^{[i]}\right\|_2, \tag{11}$$

$\mathbf{X}_j^{[i]}$ is the $i$th row of the $j$th block of $\mathbf{X}$, $\boldsymbol{w}_j = [w_{1,j}, \ldots, w_{m,j}]^T \in \mathbb{R}^m$ is a nonnegative weighting vector, for $i = 1, \ldots, m, j = 1, \ldots, s$, and as is usual, $^T$ denotes the transposition. With the above-mentioned two replacements, the model in (8) becomes

$$\min_{\mathbf{X}\geq 0} \frac{1}{2} \|\mathbf{Y}-\mathbf{A}\mathbf{X}\|_F^2 + \lambda \sum_{j=1}^{s} \left\|\mathbf{X}_j\right\|_{\boldsymbol{w}_j,2,1} + \lambda_{TV} \|\mathbf{H}\mathbf{X}\|_{1,1}, \tag{12}$$

Clearly, the weighted $\ell_{2,1}$ norm is always convex for nonnegative $w_{i,j}$ and thus the model in (12) is convex.

### B. JSBUNSAL-TV ALGORITHM

We now solve the proposed model in (12) under the ADMM framework. We note that ADMM has been widely adopted in many areas such as machine learning and image processing, see, e.g., [49]–[53] and references therein. To begin, we partition $\mathbf{X}$ with $s$ blocks as in (6). Introducing three variables

$\mathbf{V}_1$, $\mathbf{V}_2$, and $\mathbf{V}_3$ and partitioning $\mathbf{V}_2 = [\mathbf{V}_{2,1}, \ldots, \mathbf{V}_{2,s}]$ as $\mathbf{X}$, we transform (8) to an equivalent model

$$\begin{aligned}\min_{\mathbf{X},\mathbf{V}_1,\mathbf{V}_2,\mathbf{V}_3,\mathbf{V}_4,\mathbf{V}_5} & \frac{1}{2}\|\mathbf{Y}-\mathbf{V}_1\|_F^2 + \lambda \sum_{j=1}^{s} \|\mathbf{V}_{2,j}\|_{\boldsymbol{w}_j,2,1} \\ & \lambda_{TV}\,\|\mathbf{V}_4\|_{1,1} + \iota_{\mathbb{R}^+}(\mathbf{V}_5), \\ \text{subject to } & \mathbf{A}\mathbf{X} = \mathbf{V}_1,\ \mathbf{X} = \mathbf{V}_2,\ \mathbf{X} = \mathbf{V}_3, \\ & \mathbf{H}\mathbf{V}_3 = \mathbf{V}_4,\ \mathbf{X} = \mathbf{V}_5, \end{aligned} \tag{13}$$

where $\iota_\Omega$ is the indicator function of a set $\Omega$, i.e., $\iota_\Omega(x) = 0$ if $x \in \Omega$ and $\iota_\Omega(x) = +\infty$ otherwise.

To make notations more concisely, we define

$$\mathbf{G} = \begin{pmatrix} \mathbf{A} \\ \mathbf{I} \\ \mathbf{I} \\ 0 \\ \mathbf{I} \end{pmatrix},\ \mathbf{B} = \begin{pmatrix} -\mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & -\mathbf{I} & 0 & 0 & 0 \\ 0 & 0 & -\mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{H} & -\mathbf{I} & 0 \\ 0 & 0 & 0 & 0 & -\mathbf{I} \end{pmatrix},$$
$$\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_1 \\ \boldsymbol{\Lambda}_2 \\ \boldsymbol{\Lambda}_3 \\ \boldsymbol{\Lambda}_4 \\ \boldsymbol{\Lambda}_5 \end{pmatrix},\ \mathbf{V} = \begin{pmatrix} \mathbf{V}_1 \\ \mathbf{V}_2 \\ \mathbf{V}_3 \\ \mathbf{V}_4 \\ \mathbf{V}_5 \end{pmatrix}, \tag{14}$$

and let

$$\begin{aligned} g(\mathbf{X},\mathbf{V}) = & \frac{1}{2}\|\mathbf{Y}-\mathbf{V}_1\|_F^2 + \lambda \sum_{j=1}^{s} \|\mathbf{V}_{2,j}\|_{\boldsymbol{w}_j,2,1} \\ & + \lambda_{TV}\,\|\mathbf{V}_4\|_{1,1} + \iota_{\mathbb{R}^+}(\mathbf{V}_5). \end{aligned} \tag{15}$$

Then we obtain a compact form of (13):

$$\begin{aligned} & \min_{\mathbf{X},\mathbf{V}}\ g(\mathbf{X},\mathbf{V}) \\ & \text{subject to } \mathbf{G}\mathbf{X} + \mathbf{B}\mathbf{V} = 0. \end{aligned} \tag{16}$$

Define

$$\mathcal{L}_\mu(\mathbf{X},\mathbf{V};\boldsymbol{\Lambda}) = g(\mathbf{X},\mathbf{V}) + \frac{\mu}{2}\|\mathbf{G}\mathbf{X}+\mathbf{B}\mathbf{V}-\boldsymbol{\Lambda}\|_F^2, \tag{17}$$

where $\mu > 0$ is the augmented Lagrangian penalty parameter. Then the ADMM framework is derived

$$\begin{cases} \mathbf{X}^{k+1} = arg\,min_{\mathbf{X}}\,\mathcal{L}_\mu(\mathbf{X},\mathbf{V}^k;\boldsymbol{\Lambda}^k), \\ \mathbf{V}^{k+1} = arg\,min_{\mathbf{V}}\,\mathcal{L}_\mu(\mathbf{X}^{k+1},\mathbf{V};\boldsymbol{\Lambda}^k), \\ \boldsymbol{\Lambda}^{k+1} = \boldsymbol{\Lambda}^k - (\mathbf{G}\mathbf{X}^{k+1}+\mathbf{B}\mathbf{V}^{k+1}). \end{cases} \tag{18}$$

We now show how to solve (18). To begin, the $\mathbf{X}$-subproblem, after dropping constant terms, is equivalent to solve

$$\begin{aligned} \mathbf{X}^{k+1} &= \underset{\mathbf{X}}{arg\,min}\ g(\mathbf{X},\mathbf{V}^k) + \frac{\mu}{2}\left\|\mathbf{G}\mathbf{X}+\mathbf{B}\mathbf{V}^k-\boldsymbol{\Lambda}^k\right\|_F^2 \\ &= \underset{\mathbf{X}}{arg\,min}\ \left\|\mathbf{A}\mathbf{X}-\mathbf{V}_1^k-\boldsymbol{\Lambda}_1^k\right\|_F^2 + \left\|\mathbf{X}-\mathbf{V}_2^k-\boldsymbol{\Lambda}_2^k\right\|_F^2 \\ &\quad + \left\|\mathbf{X}-\mathbf{V}_3^k-\boldsymbol{\Lambda}_3^k\right\|_F^2 + \left\|\mathbf{X}-\mathbf{V}_5^k-\boldsymbol{\Lambda}_5^k\right\|_F^2. \end{aligned} \tag{19}$$

It is a least squares problem. A simple calculation gives

$$\begin{aligned} \mathbf{X}^{k+1} = & \left(\mathbf{A}^T\mathbf{A}+3\mathbf{I}\right)^{-1}\left(\mathbf{A}^T\left(\mathbf{V}_1^k+\boldsymbol{\Lambda}_1^k\right)\right. \\ & \left.+\mathbf{V}_2^k+\boldsymbol{\Lambda}_2^k+\mathbf{V}_3^k+\boldsymbol{\Lambda}_3^k+\mathbf{V}_5^k+\boldsymbol{\Lambda}_5^k\right). \end{aligned} \tag{20}$$

We decouple the $\mathbf{V}$-subproblem of (18) to three independent subparts with respect to $\mathbf{V}_1$, $\mathbf{V}_2$, $\mathbf{V}_5$, and a single optimization with respect to $\mathbf{V}_3$ and with respect to $\mathbf{V}_4$. For $\mathbf{V}_1$-subproblem, after dropping constant terms, we obtain

$$\mathbf{V}_1^{k+1} = \underset{\mathbf{V}_1}{arg\,min}\ \frac{1}{2}\|\mathbf{Y}-\mathbf{V}_1\|_F^2 + \frac{\mu}{2}\left\|\mathbf{A}\mathbf{X}^{k+1}-\mathbf{V}_1-\boldsymbol{\Lambda}_1^k\right\|_F^2. \tag{21}$$

Thus,

$$\mathbf{V}_1^{k+1} = \frac{1}{1+\mu}\left(\mathbf{Y}+\mu\left(\mathbf{A}\mathbf{X}^{k+1}-\boldsymbol{\Lambda}_1^k\right)\right). \tag{22}$$

The $\mathbf{V}_2$-subproblem is to solve

$$\mathbf{V}_2^{k+1} = \underset{\mathbf{V}_2}{arg\,min}\ \lambda\sum_{j=1}^{s}\|\mathbf{V}_{2,j}\|_{\boldsymbol{w}_j,2,1} + \frac{\mu}{2}\left\|\mathbf{X}^{k+1}-\mathbf{V}_2-\boldsymbol{\Lambda}_2^k\right\|_F^2. \tag{23}$$

Partition $\mathbf{X}^{k+1} = \left[\mathbf{X}_1^{k+1},\ldots,\mathbf{X}_s^{k+1}\right]$, $\mathbf{V}_2 = [\mathbf{V}_{2,1},\ldots,\mathbf{V}_{2,s}]$, and correspondently, $\boldsymbol{\Lambda}_2^k = [\boldsymbol{\Lambda}_{2,1}^k,\ldots,\boldsymbol{\Lambda}_{2,s}^k]$. Since the object function of (23) is proper, strictly convex, and *separable*, we equivalently decouple it to $s$ sub-problems

$$\min_{\mathbf{V}_{2,j}}\lambda\left\|\mathbf{V}_{2,j}\right\|_{\boldsymbol{w}_j,2,1} + \frac{\mu}{2}\left\|\mathbf{X}_j^{k+1}-\mathbf{V}_{2,j}-\boldsymbol{\Lambda}_{2,j}^k\right\|_F^2, \tag{24}$$

for $j = 1,\ldots,s$. From [54], each sub-problem clearly admits a unique block solution, and we obtain the $i$th row of the $j$th block of $\mathbf{V}_2^{k+1}$, i.e.,

$$\left(\mathbf{V}_{2,j}^{k+1}\right)^{[i]} = \text{vect\_soft}_{\frac{\lambda}{\mu}w_{i,j}}\left(\left(\mathbf{X}_j^{k+1}-\boldsymbol{\Lambda}_{2,j}^k\right)^{[i]}\right), \tag{25}$$

for $i = 1,\ldots,m$, $j = 1,\ldots,s$. Here $\text{vect\_soft}_\alpha(\cdot)$ is a nonlinear operator defined by

$$\text{vect\_soft}_\alpha(\boldsymbol{x}) = \boldsymbol{x}\frac{\max\{\|\boldsymbol{x}\|_2-\alpha,0\}}{\max\{\|\boldsymbol{x}\|_2-\alpha,0\}+\alpha}, \tag{26}$$

for $\forall\,\boldsymbol{x}\in\mathbb{R}^N$ and $\alpha > 0$.

For $\mathbf{V}_3$-subproblem, we equivalently solve the following minimization problem

$$\mathbf{V}_3^{k+1} = \underset{\mathbf{V}_3}{arg\,min}\ \left\|\mathbf{X}^{k+1}-\mathbf{V}_3-\boldsymbol{\Lambda}_3^k\right\|_F^2 + \left\|\mathbf{H}\mathbf{V}_3-\mathbf{V}_4^k-\boldsymbol{\Lambda}_4^k\right\|_F^2, \tag{27}$$

and obtain

$$\mathbf{V}_3^{k+1} = (\mathbf{H}^T\mathbf{H}+\mathbf{I})^{-1}\left(\mathbf{X}^{k+1}-\boldsymbol{\Lambda}_3^k+\mathbf{H}^T\left(\mathbf{V}_4^k+\boldsymbol{\Lambda}_4^k\right)\right). \tag{28}$$

To compute $\mathbf{V}_4^{k+1}$, we solve the optimization problem

$$\mathbf{V}_4^{k+1} = \underset{\mathbf{V}_4}{arg\,min}\ \lambda_{TV}\|\mathbf{V}_4\|_{1,1} + \frac{\mu}{2}\left\|\mathbf{H}\mathbf{V}_3^{k+1}-\mathbf{V}_4-\boldsymbol{\Lambda}_4^k\right\|_F^2, \tag{29}$$

and derive

$$\mathbf{V}_4^{k+1} = soft_{\frac{\lambda_{TV}}{\mu}}\left(\mathbf{H}\mathbf{V}_3^{k+1} - \mathbf{\Lambda}_4^k\right). \tag{30}$$

Here we define $soft_\alpha(\cdot)$ be a nonlinear soft-thresholding operator defined componentwise by

$$\left(soft_\alpha(\boldsymbol{x})\right)_i = x_i \frac{\max\{|x_i| - \alpha, 0\}}{\max\{|x_i| - \alpha, 0\} + \alpha}, \tag{31}$$

for $\forall\, \boldsymbol{x} = [x_1, \ldots, x_N]^T \in \mathbb{R}^N$ and $\alpha > 0$.

For $\mathbf{V}_5$-subproblem, we have

$$\mathbf{V}_5^{k+1} = \underset{\mathbf{V}_5}{arg\,min}\ \iota_{\mathbb{R}^+}(\mathbf{V}_5) + \frac{\mu}{2}\left\|\mathbf{X}^{k+1} - \mathbf{V}_5 - \mathbf{\Lambda}_5^k\right\|_F^2. \tag{32}$$

It is easy to obtain that

$$\mathbf{V}_5^{k+1} = \max\left(\mathbf{X}^{k+1} - \mathbf{\Lambda}_5^k, 0\right). \tag{33}$$

Finally, we update the multipliers

$$\begin{cases} \mathbf{\Lambda}_1^{k+1} = \mathbf{\Lambda}_1^k - (\mathbf{A}\mathbf{X}^{k+1} - \mathbf{V}_1^{k+1}) \\ \mathbf{\Lambda}_2^{k+1} = \mathbf{\Lambda}_2^k - (\mathbf{X}^{k+1} - \mathbf{V}_2^{k+1}) \\ \mathbf{\Lambda}_3^{k+1} = \mathbf{\Lambda}_3^k - (\mathbf{X}^{k+1} - \mathbf{V}_3^{k+1}) \\ \mathbf{\Lambda}_4^{k+1} = \mathbf{\Lambda}_4^k - (\mathbf{H}\mathbf{V}_3^{k+1} - \mathbf{V}_4^{k+1}) \\ \mathbf{\Lambda}_5^{k+1} = \mathbf{\Lambda}_5^k - (\mathbf{X}^{k+1} - \mathbf{V}_5^{k+1}). \end{cases} \tag{34}$$

To make it more clearly, we summarize the proposed joint-sparse-blocks unmixing via variable splitting augmented Lagrangian and total variation (JSBUnSAL-TV) algorithm in the following.

**Algorithm 1** Pseudocode of JSBUnSAL-TV

1. **Input:** $\mathbf{Y}$, $\mathbf{A}$.
2. **Selected parameters:** $\lambda$, $\lambda_{TV}$, $\mu$, maximum iteration, $d_j$, $w_{i,j}$, for $i = 1, \ldots, m, j = 1, \ldots, s$.
3. **Initialization:** $\mathbf{\Lambda}_l^0$, $\mathbf{V}_l^0$, $l = 1, 2, \ldots, 5$, and set $k = 0$.
4. **Repeat:**
5. Compute $\mathbf{X}^{k+1}$ by (20).
6. Compute $\mathbf{V}_1^{k+1}$ by (22).
7. Compute $\mathbf{V}_2^{k+1}$ by (25).
8. Compute $\mathbf{V}_3^{k+1}$ by (28).
9. Compute $\mathbf{V}_4^{k+1}$ by (30).
10. Compute $\mathbf{V}_5^{k+1}$ by (33).
11. Update Lagrange multipliers $\mathbf{\Lambda}_l^{k+1}$ by (34), for $l = 1, \ldots, 5$.
12. **until** some stopping criterion is satisfied.
13. **Output:** $\hat{\mathbf{X}} = \mathbf{X}^{k+1}$.

We note that the proposed JSBUnSAL-TV algorithm is similar to SUnSAL-TV. The only difference of the two algorithms is in Step 7, where the corresponding subproblem of SUnSAL-TV is the classic sparse regression, whereas the one of JSBUnSAL-TV in (23) is the joint-sparse-blocks regression. In addition, each iteration of JSBUnSAL-TV clearly has the same complexity as that of SUnSAL-TV, i.e., $\mathcal{O}(L^2\, n) + \mathcal{O}(Ln \log n)$, see more details in [34].

## IV. EXPERIMENTS

In this section, we demonstrate the effectiveness of JSBUnSAL-TV on both simulated and real data. We will compare JSBUnSAL-TV with six state-of-the-art algorithms: SUnSAL [15], CLSUnSAL [26], SUnSAL-TV[1] [34], ADSpLRU [42][2], DRSU-TV [40], and JSpBLRU [31]. Our tests were done by using MATLAB R2016a on a MacBook Pro laptop with 2.7 GHz Intel Core i7 and 16 GB memory. The floating-point precision is $10^{-16}$.

### A. PARAMETER SETTING

In the following, for the proposed JSBUnSAL-TV algorithm, we first give a partition strategy of $\mathbf{X}$ in (6) and then a reweighting strategy of $w_{i,j}$ in (25), similarly as in [31]. Recall from (6) that partitioning $\mathbf{X}$ is just to determine the values of $s$ and $d_i$, for $i = 1, \ldots, s$. Recall that $n$ denotes the total number of pixels and $s$ is the number of blocks. Assume $d$ is a positive integer for $1 \le d \le n$. Given $n$ and $d$, we now show how to set $s$ and all $d_i$. For simplicity, we assume that most of $\mathbf{X}_j$ contain the same number of pixels. To this end, we let $s = \lfloor n/d \rfloor$ be the largest integer no greater than $n/d$, and define

$$d_1 = d_2 = \cdots = d_{s-1} = d, \quad d_s = n - (s-1)d. \tag{35}$$

Clearly, *once $d$ is given, all $d_i$ can be obtained.* Also, it is easy to check that $n = \sum_{j=1}^s d_j$. As an example, consider $n = 5625$ and $d = 3$, then we have $s = \lfloor n/d \rfloor = 1875$ and $d_1 = d_2 = \cdots = d_{s-1} = d_s = d = 3$. It says that $\mathbf{X}$ is partitioned as $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{1875}]$ and all $\mathbf{X}_j$ contain $d = 3$ pixels. As another example, consider instead $n = 5624$ and $d = 3$, we have $s = \lfloor n/d \rfloor = 1874$, $d_1 = d_2 = \cdots = d_{1873} = d = 3$, and $d_{1874} = 5$. Then $\mathbf{X}$ is partitioned as $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{1874}]$. Each $\mathbf{X}_j$ contains $d = 3$ pixels, for $j = 1, \ldots, 1873$, and $\mathbf{X}_{1874}$ contains the remaining 5 pixels. The authors in [31] have explored the influence of the values of $d$ for the $\ell_{2,1}$-blocks norm regularization and it is hard to obtain an optimal $d$ for all cases. However, $d = 3$ is a good candidate for many experiments. Following the analysis, we fix $d = 3$ in JSBUnSAL-TV for both simulated and real-data experiments.

We now adopt a two-level reweighting strategy proposed in [31] for joint-sparse-blocks regression to enhance the sparsity along the rows in each block. We recall from (25) and (26) that the $i$th row of the $j$th block of $\mathbf{V}_2^{k+1}$, i.e., $(\mathbf{V}_{2,j}^{k+1})^{[i]}$, is obtained by

$$(\mathbf{V}_{2,j}^{k+1})^{[i]} = \boldsymbol{t}_{i,j}^k \frac{\max\{\|\boldsymbol{t}_{i,j}^k\|_2 - \frac{\lambda}{\mu} w_{i,j}, 0\}}{\max\{\|\boldsymbol{t}_{i,j}^k\|_2 - \frac{\lambda}{\mu} w_{i,j}, 0\} + \frac{\lambda}{\mu} w_{i,j}}, \tag{36}$$

where we assume $\boldsymbol{t}_{i,j}^k = (\mathbf{X}_j^{k+1} - \mathbf{\Lambda}_{2,j}^k)^{[i]}$, for $i = 1, \ldots, m$, $j = 1, \ldots, s$. Similarly as in [55], the authors in [31] propose a reweighting strategy of $w_{i,j}$, that is, the weights $w_{i,j}$ at

[1]The MATLAB codes of SUnSAL, CLSUnSAL, and SUnSAL-TV are available at http://www.lx.it.pt/~bioucas/publications.html

[2]The MATLAB code of ADSpLRU is available at http://members.noa.gr/parisg/demo_splr_unmixing.zip

$(k+1)$th iteration for $(\mathbf{V}_{2,j}^{k+1})^{[i]}$ in (36), denoted as $w_{i,j}^{k+1}$, are computed by

$$w_{i,j}^{k+1} = \frac{1}{\|\boldsymbol{t}_{i,j}^{k}\|_2 + \varepsilon}, \tag{37}$$

where $\varepsilon = 10^{-16}$ is a small constant added to avoid singularities. Clearly, the reweighting coefficients treat each row in each block differently and enhance the sparsity along the rows in each block. The effectiveness of the reweighting technique for spectral unmixing problems has been exploited in [31] and will be also shown in the following section IV-B.

For all the algorithms, regularization parameters are tuned to their best performance with respect to signal-to-reconstruction error (SRE) measured in dB and defined by

$$\text{SRE (dB)} = 10\log_{10}\left(\frac{\frac{1}{n}\sum_{i=1}^{n}\|\hat{\boldsymbol{x}}_i\|_2^2}{\frac{1}{n}\sum_{i=1}^{n}\|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|_2^2}\right), \tag{38}$$

where $n$ is the number of pixels, $\hat{\boldsymbol{x}}_i$ and $\boldsymbol{x}_i$ are estimated and exact abundance vectors of the $i$th pixel, respectively. Generally speaking, the higher SRE (dB), the higher quality of the unmixing results. Another metric evaluating the performance of unmixing estimations is the root-mean-square error (RMSE) defined by

$$\text{RMSE} = \sqrt{\frac{1}{mn}\sum_{i=1}^{n}\left\|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\right\|_2^2}, \tag{39}$$

where $m$ is the number of endmembers. Unless otherwise specified, we select optimal regularization parameters in the compared algorithms from the following sequence:

$$\{0,\ 5\cdot 10^{-4}, 0.001,\ 0.005,\ 0.01,\ 0.05,\ 0.1,\ 0.5,\ 1\} \tag{40}$$

to get best SRE (dB) value, similarly as in [32], [34]. Note that each of SUnSAL and CLSUnSAL has only one regularization parameter, but each of other four algorithms has two. Thus, fine tuning the parameters of SUnSAL-TV, ADSpLRU, DRSU-TV, JSpBLRU, and JSBUnSAL-TV costs much more computational time than that of SUnSAL and CLSUnSAL. In particular, we empirically select the optimal rank parameter $\tau$ in ADSpLRU and JSpBLRU from a slightly wider range:

$$\{0,\ 5\cdot 10^{-4}, 0.001,\ 0.005,\ 0.01,\ 0.05,\ 0.1,\ 0.5,\ 1,\ 3,\ 5,\ 10,\ 50,\ 100\}. \tag{41}$$

In addition, we initially set the augmented Lagrangian penalty parameter $\mu$ by choosing an optimal value from:

$$\{0.001,\ 0.01,\ 0.1,\ 1\}. \tag{42}$$

We stop JSBUnSAL-TV if the termination criterion

$$\frac{\left\|\mathbf{X}^{k} - \mathbf{X}^{k-1}\right\|_F}{\left\|\mathbf{X}^{k-1}\right\|_F} \le 10^{-8} \tag{43}$$

is satisfied or when the number of iterations reached 1000. The termination criterions for other algorithms are the same as in [15], [26], [31], [34], [40], [42] with maximum iteration 1000.

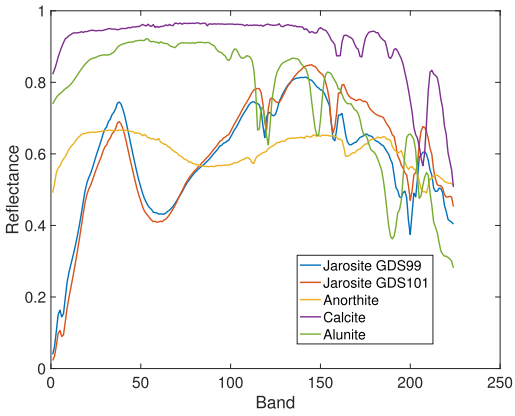


**FIGURE 1.** Spectral characteristic curves of five endmembers in Example 1.

### B. EXPERIMENTS ON SIMULATED DATA

*Example 1:* In this experiment, the simulated data cube contains 75 × 75 pixels with 224 bands per pixel, which has been used in [21], [34], [56]. We use the spectral library $\mathbf{A}_1 \in \mathbb{R}^{224\times 240}$: a randomly selected subset of the U.S. Geological Survey (USGS) spectral library[3], which comprises 498 spectral signatures with reflectance values measured in 224 spectral bands, distributed uniformly ranging from 0.4 to 2.5 $\mu$m. The test data cube is generated according to LMM, with five randomly selected spectral signatures from $\mathbf{A}_1$ as endmembers, shown in Fig. 1, and five corresponding true fractional abundances shown in the first row of Fig. 2. Then, the scene is corrupted by white Gaussian i.i.d. noise, generated by the MATLAB function **randn**, with SNR = 25, 30, 35 and 40 dB, respectively.

*Example 2:* In this example, we use a simulated data cube containing 128 × 128 pixels with 224 spectral bands. The spectral library matrix is $\mathbf{A}_2 \in \mathbb{R}^{224\times 100}$: a randomly selected subset of $\mathbf{A}_1$. To generate the data cube by LMM, we randomly choose five signatures from $\mathbf{A}_2$ and use true fractional abundances shown in the first row of Fig. 3. We note that the fractional abundances are also used in [48]. After the above procedure, the true simulated data cube is contaminated by white Gaussian i.i.d. noise with the same SNR values adopted for Example 1.

Figs. 2 and 3 show the true and estimated abundance maps by different unmixing algorithms for Example 1 with SNR = 35 dB and Example 2 with SNR = 25 dB, respectively. Abundance maps for other SNRs show similar behavior, so we omit here for space considerations. From Fig. 2, we observe that all unmixing algorithms delineate most square regions in each abundance map except SUnSAL. Also, CLSUnSAL estimates the abundances with lower accuracy. Clearly, CLSUnSAL has more smooth background than SUnSAL. As expected, due to the TV regularizer, SUnSAL-TV, DRSU-TV, and JSBUnSAL-TV give abundance maps with better spatial consistency for all endmembers than other algorithms. In particular, JSBUnSAL-TV delineates a few

[3]Available online: http://speclab.cr.usgs.gov/spectral.lib06

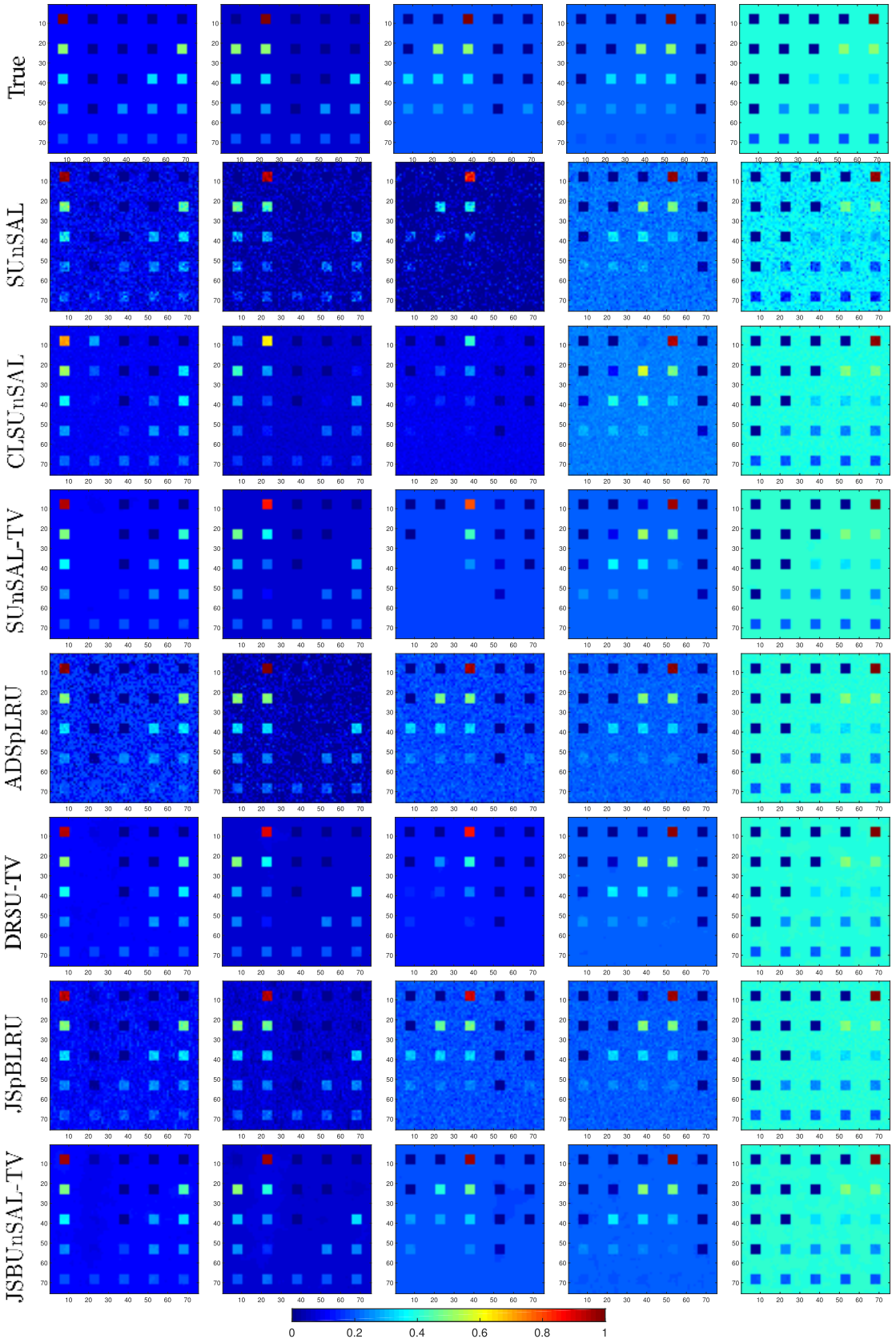


**FIGURE 2.** True and estimated abundance maps for (from left to right) endmembers #1–#5 by different unmixing algorithms for Example 1 with SNR = 35 dB.

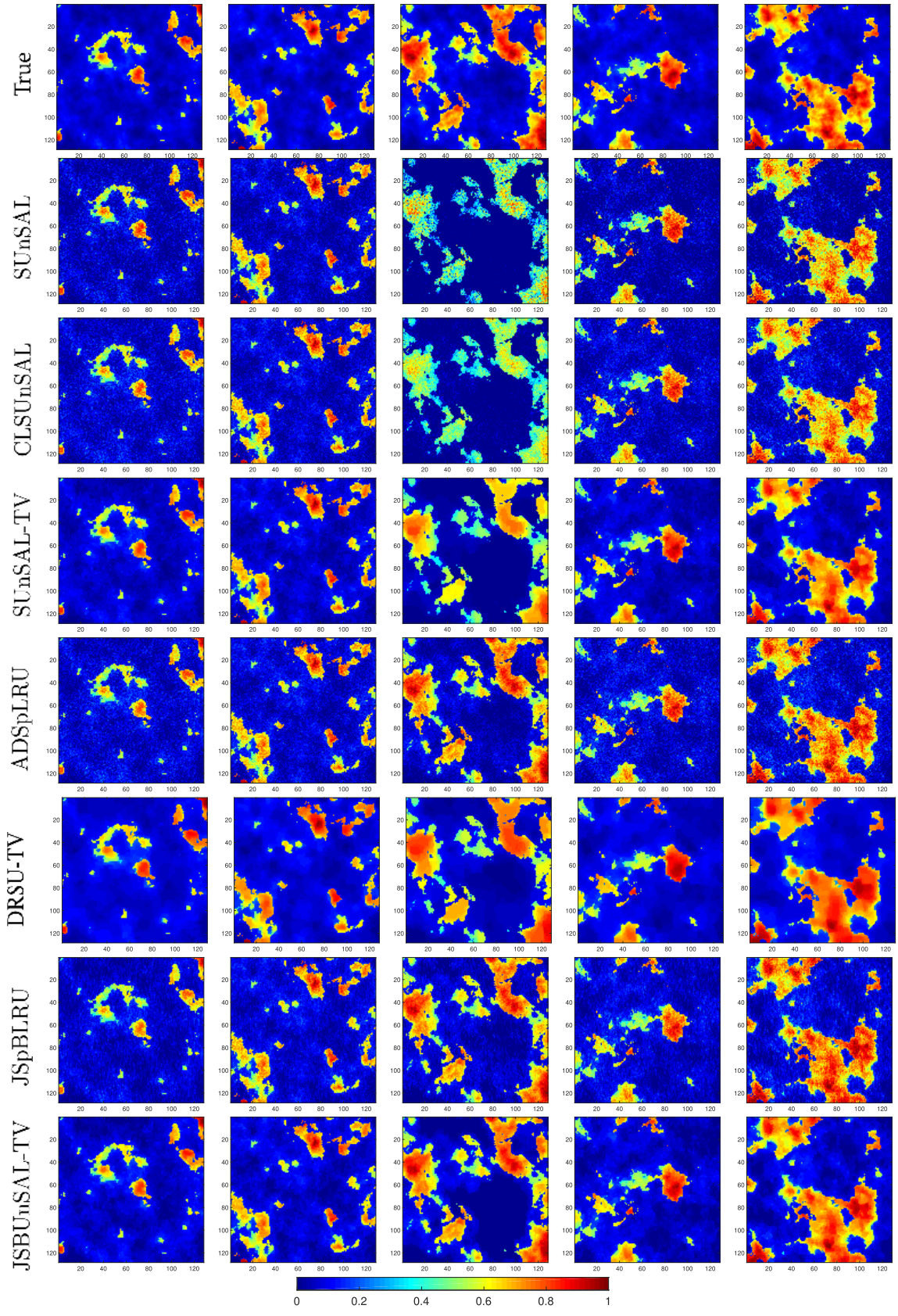


**FIGURE 3.** True and estimated abundance maps for (from left to right) endmembers #1–#5 by different unmixing algorithms for Example 2 with SNR = 25 dB.

**TABLE 1.** SRE (dB), RMSE, and Time (s) by different unmixing algorithms for Examples 1 and 2. Optimal parameters for which the reported values were achieved are indicated in the parentheses.

| Example 1 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| SNR | Criteria | SUnSAL | CLSUnSAL | SUnSAL-TV | ADSpLRU | DRSU-TV | JSpBLRU | JSBUnSAL-TV |
| 25 dB | SRE(dB) | 4.51 | 6.54 | 13.24 | 6.72 | 15.17 | 10.26 | **15.65** |
| | RMSE | 0.0206 | 0.0163 | 0.0075 | 0.0159 | 0.0060 | 0.0106 | **0.0057** |
| | Time(s) | **27.44** ($\lambda = 0.1$) | 30.66 ($\lambda = 5$) | 233.06 ($\lambda = 0.01$, $\lambda_{TV} = 0.05$) | 135.09 ($\lambda = 5 \cdot 10^{-4}$, $\tau = 10$) | 379.26 ($\lambda = 0.05$, $\lambda_{TV} = 0.1$) | 129.53 ($\lambda = 0.005$, $\tau = 50$) | 386.75 ($\lambda = 5 \cdot 10^{-4}$, $\lambda_{TV} = 0.05$) |
| 30 dB | SRE(dB) | 6.13 | 7.51 | 15.56 | 10.82 | 17.91 | 12.67 | **21.05** |
| | RMSE | 0.0170 | 0.0146 | 0.0058 | 0.0099 | 0.0044 | 0.0080 | **0.0031** |
| | Time(s) | **30.06** ($\lambda = 0.05$) | 32.09 ($\lambda = 1$) | 223.23 ($\lambda = 0.005$, $\lambda_{TV} = 0.01$) | 141.56 ($\lambda = 0.001$, $\tau = 50$) | 369.29 ($\lambda = 0.1$, $\lambda_{TV} = 0.05$) | 140.26 ($\lambda = 0.005$, $\tau = 10$) | 406.29 ($\lambda = 5 \cdot 10^{-4}$, $\lambda_{TV} = 0.01$) |
| 35 dB | SRE(dB) | 7.67 | 11.62 | 19.34 | 15.90 | 16.38 | 21.15 | **25.44** |
| | RMSE | 0.0143 | 0.0091 | 0.0037 | 0.0055 | 0.0052 | 0.0030 | **0.0018** |
| | Time(s) | **29.08** ($\lambda = 0.01$) | 31.27 ($\lambda = 1$) | 209.90 ($\lambda = 0.001$, $\lambda_{TV} = 0.01$) | 133.01 ($\lambda = 5 \cdot 10^{-4}$, $\tau = 1$) | 390.91 ($\lambda = 0.01$, $\lambda_{TV} = 0.01$) | 140.14 ($\lambda = 0.001$, $\tau = 1$) | 405.85 ($\lambda = 5 \cdot 10^{-4}$, $\lambda_{TV} = 0.005$) |
| 40 dB | SRE(dB) | 11.20 | 15.85 | 22.50 | 19.02 | 28.01 | 26.98 | **30.96** |
| | RMSE | 0.0095 | 0.0056 | 0.0026 | 0.0039 | 0.0014 | 0.0015 | **0.0010** |
| | Time(s) | **28.98** ($\lambda = 0.01$) | 31.09 ($\lambda = 0.5$) | 363.87 ($\lambda = 0.001$, $\lambda_{TV} = 0.005$) | 134.00 ($\lambda = 5 \cdot 10^{-4}$, $\tau = 1$) | 381.78 ($\lambda = 0.01$, $\lambda_{TV} = 0.005$) | 78.43 ($\lambda = 0.001$, $\tau = 0.5$) | 393.84 ($\lambda = 5 \cdot 10^{-4}$, $\lambda_{TV} = 0.001$) |
| Example 2 | | | | | | | | |
| SNR | Criteria | SUnSAL | CLSUnSAL | SUnSAL-TV | ADSpLRU | DRSU-TV | JSpBLRU | JSBUnSAL-TV |
| 25 dB | SRE(dB) | 9.17 | 10.90 | 15.98 | 16.37 | 18.13 | 18.16 | **20.28** |
| | RMSE | 0.0249 | 0.0204 | 0.0114 | 0.0109 | 0.0089 | 0.0088 | **0.0069** |
| | Time(s) | **34.22** ($\lambda = 0.05$) | 35.80 ($\lambda = 1$) | 359.28 ($\lambda = 0$, $\lambda_{TV} = 0.01$) | 243.41 ($\lambda = 0.001$, $\tau = 50$) | 364.22 ($\lambda = 0.5$, $\lambda_{TV} = 0.05$) | 258.14 ($\lambda = 0.05$, $\tau = 100$) | 425.72 ($\lambda = 0.001$, $\lambda_{TV} = 0.01$) |
| 30 dB | SRE(dB) | 12.77 | 15.10 | 21.10 | 20.72 | 20.40 | 21.59 | **26.90** |
| | RMSE | 0.0164 | 0.0126 | 0.0063 | 0.0066 | 0.0068 | 0.0060 | **0.0032** |
| | Time(s) | **36.67** ($\lambda = 0.01$) | 37.42 ($\lambda = 1$) | 349.10 ($\lambda = 0.005$, $\lambda_{TV} = 0.01$) | 225.15 ($\lambda = 5 \cdot 10^{-4}$, $\tau = 10$) | 360.37 ($\lambda = 0$, $\lambda_{TV} = 0.01$) | 241.93 ($\lambda = 0.01$, $\tau = 50$) | 415.84 ($\lambda = 0.001$, $\lambda_{TV} = 0.01$) |
| 35 dB | SRE(dB) | 16.36 | 19.52 | 24.73 | 25.66 | 26.21 | 26.73 | **29.03** |
| | RMSE | 0.0109 | 0.0076 | 0.0041 | 0.0037 | 0.0035 | 0.0033 | **0.0025** |
| | Time(s) | **33.94** ($\lambda = 0.01$) | 35.41 ($\lambda = 0.5$) | 346.32 ($\lambda = 0.001$, $\lambda_{TV} = 0.005$) | 229.19 ($\lambda = 5 \cdot 10^{-4}$, $\tau = 10$) | 361.25 ($\lambda = 0.05$, $\lambda_{TV} = 0.01$) | 242.07 ($\lambda = 0.001$, $\tau = 10$) | 417.30 ($\lambda = 5 \cdot 10^{-4}$, $\lambda_{TV} = 0.005$) |
| 40 dB | SRE(dB) | 20.36 | 24.15 | 25.42 | 30.43 | 31.34 | 31.62 | **32.08** |
| | RMSE | 0.0069 | 0.0044 | 0.0038 | 0.0022 | 0.0019 | 0.0019 | **0.0018** |
| | Time(s) | **21.47** ($\lambda = 0.005$) | 35.51 ($\lambda = 0.5$) | 335.63 ($\lambda = 0$, $\lambda_{TV} = 0.005$) | 236.10 ($\lambda = 5 \cdot 10^{-4}$, $\tau = 10$) | 361.89 ($\lambda = 0.01$, $\lambda_{TV} = 0.005$) | 244.09 ($\lambda = 5 \cdot 10^{-4}$, $\tau = 5$) | 414.86 ($\lambda = 5 \cdot 10^{-4}$, $\lambda_{TV} = 0.001$) |

**TABLE 2.** Average per-pixel runtime by different unmixing algorithms for the AVIRIS Cuprite subscene.

| | SUnSAL | CLSUnSAL | SUnSAL-TV | ADSpLRU | DRSU-TV | JSpBLRU | JSBUnSAL-TV |
|---|---|---|---|---|---|---|---|
| Time (s) | 0.0050 | 0.0061 | 0.0411 | 0.0587 | 0.0447 | 0.0580 | 0.0459 |

more square regions of endmembers #1–#4 in comparison with SUnSAL-TV and DRSU-TV. Similarly, from Fig. 3, we observe that SUnSAL and CLSUnSAL provide less accurate unmixing results. In addition, the TV based unmixing algorithms: SUnSAL-TV, DRSU-TV, and JSBUnSAL-TV give more smooth background than the low-rank representation based ADSpLRU and JSpBLRU. We also see that SUnSAL-TV provides less accurate abundance for endmember #3 and DRSU-TV makes the regions with high fractional abundance of the considered endmembers over-smooth. The abundance maps by JSBUnSAL-TV are more similar to those in the ground-truth than the ones estimated by other unmixing algorithms.

Table 1 lists the SRE (dB) values, the RMSE values, and the elapsed time in seconds (denoted as Time (s)), along with optimal regularization parameter values, of all compared algorithms for Examples 1 and 2. From Table 1, we see that both SUnSAL and CLSUnSAL are very fast and

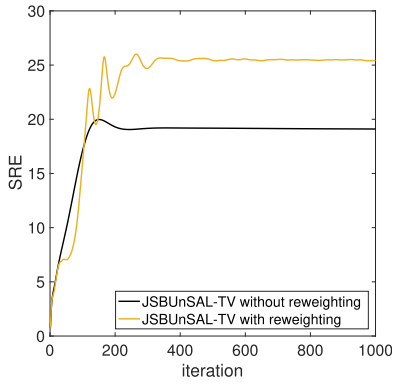


**FIGURE 4.** SRE (dB) versus iteration of JSBUnSAL-TV with/without reweighting for Example 1 with SNR = 35 dB.

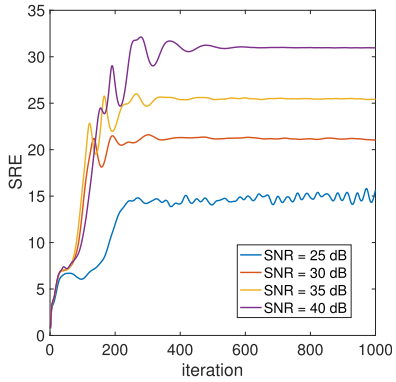


**FIGURE 5.** Convergence histories of JSBUnSAL-TV for Example 1 under different noise levels.

TV based unmixing algorithms: SUnSAL-TV, DRSU-TV, and JSBUnSAL-TV, cost much more computational time than other four algorithms. In addition, JSBUnSAL-TV demands more runtime compared with SUnSAL-TV and DRSU-TV. Clearly, among the seven compared algorithms, JSBUnSAL-TV provides the highest SREs and lowest RMSEs, which is consistent with the visual observation from Figs. 2 and 3. The improvement is particularly clear for Example 1 with SNR = 30, 35, and 40 dB since the changes in SRE are at least greater than 3 dB.

We now demonstrate the effectiveness of the reweighting strategy in JSBUnSAL-TV. To this end, Fig. 4 plots the SRE (dB) values versus the iteration number of JSBUnSAL-TV with and without the reweighting strategy for Example 1 with SNR = 35 dB. We see from this figure that JSBUnSAL-TV with reweighting provides higher SRE values than the one without reweighting as the iteration goes stable. We also see that JSBUnSAL-TV with and without reweighting provide an overall robust convergence behavior. Furthermore, Fig. 5 shows the convergence histories of JSBUnSAL-TV for Example 1 with examined SNRs. It shows that JSBUnSAL-TV goes stable as iteration increases for SNR of 30, 35, and 40 dB, but exhibits slight oscillation for SNR = 25 dB.

Finally, we plot SRE (dB) values by JSBUnSAL-TV as a function of parameters $\lambda$ and $\lambda_{TV}$ for Example 1 in Fig. 6.

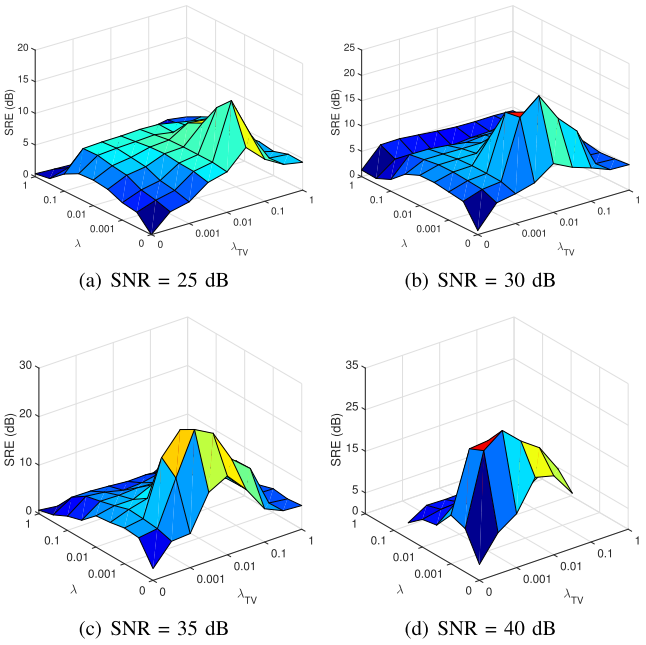


**FIGURE 6.** SRE (dB) as a function of parameters $\lambda$ and $\lambda_{TV}$ in JSBUnSAL-TV for Example 1 under different noise levels. (a) SNR = 25 dB. (a) SNR = 30 dB. (a) SNR = 35 dB. (a) SNR = 40 dB.

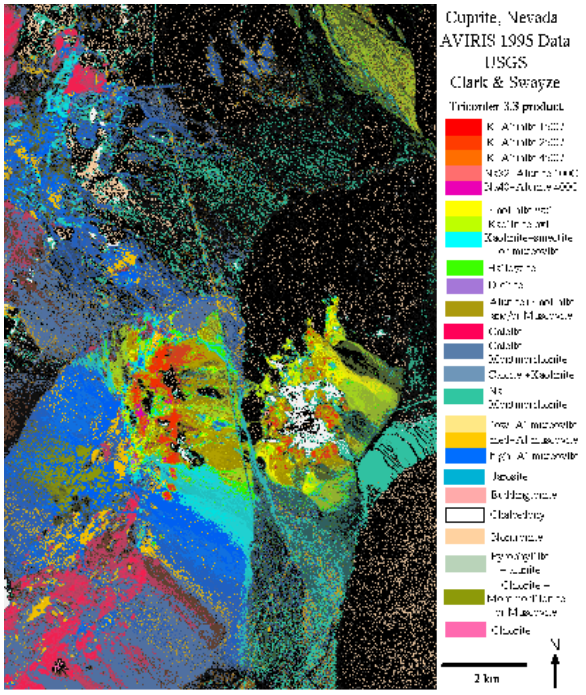


**FIGURE 7.** USGS map showing the location of different minerals in the Cuprite mining district in Nevada.

As shown in this figure, optimal $\lambda_{TV}$ values decrease as SNR gets higher, similarly as the observations for SUnSAL-TV [34]. The optimal $\lambda$ for the $\ell_{2,1}$-blocks norm regularization is relatively stable for all examined SNR values. We also observe that the optimal choice of $\lambda_{TV}$ is always greater than the optimal choice of $\lambda$. Thus, fine-tuning strategies are suggested to be adopted for $\lambda_{TV}$ in JSBUnSAL-TV. Also,

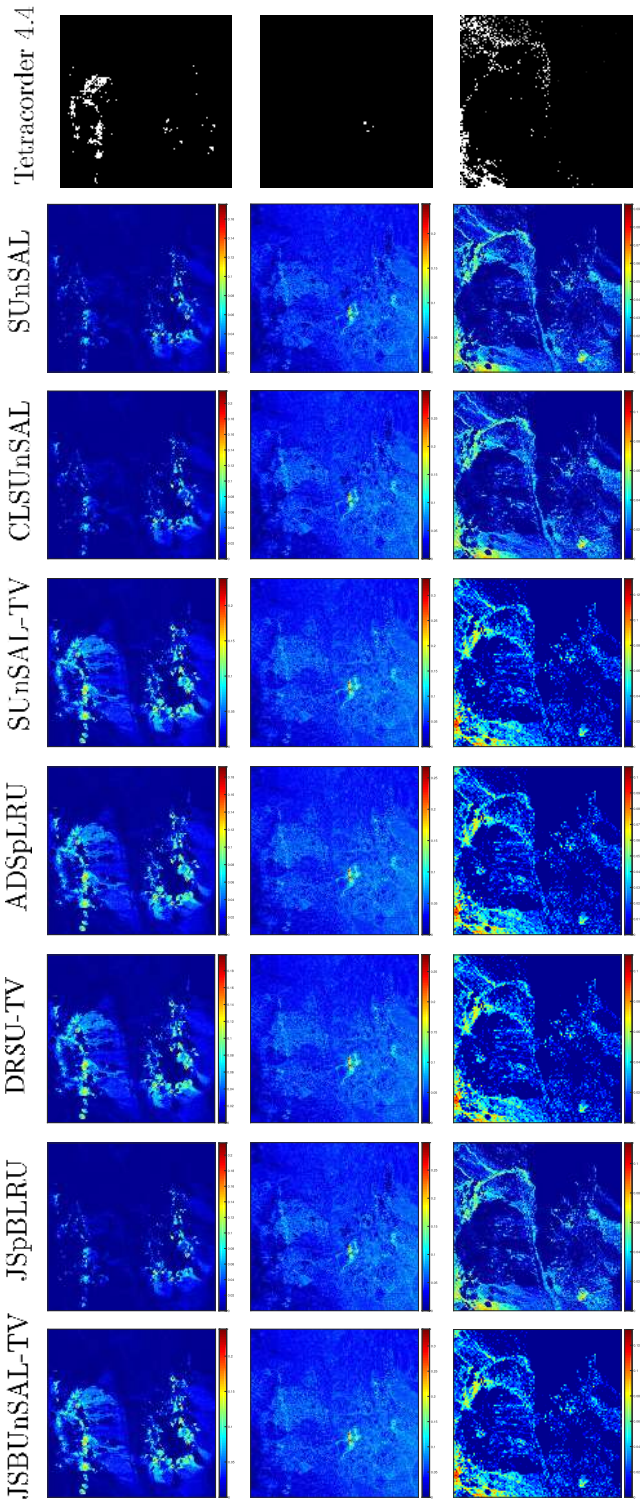


**FIGURE 8.** **Abundance maps estimated for the minerals: (from left to right) *alunite, buddingtonite,* and *muscovite* by applying different unmixing algorithms to the AVIRIS Cuprite scene.**

both optimal $\lambda$ and $\lambda_{TV}$ values are nonzero for all SNR levels, showing the effectiveness of simultaneously imposing the $\ell_{2,1}$-blocks norm and TV regularization terms in JSBUnSAL-TV for hyperspectral unmixing.

### C. EXPERIMENTS ON REAL DATA

In this experiment, we test different unmixing algorithms on the well-known Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Cuprite data set[4]. We use a subscene containing $350 \times 350$ pixels with 188 spectral bands between 0.4 and 2.5 $\mu$m. This data cube has been widely applied to validate the effectiveness of unmixing algorithms in the literature [18], [21], [32], [38], [48]; see [15], [26], [34] for more details. We generate the $188 \times 240$ spectral library matrix from the USGS library which includes all exposed minerals of interest. A mineral map[5] produced in 1995 by USGS is shown in Fig. 7, in which a Tetracorder 3.3 software product [57] was used to map different minerals present in the Cuprite mining district. Note that the publicly available AVIRIS Cuprite data were collected in 1997. Thus, it is hard to give a direct comparison between the 1995 USGS map and the 1997 Cuprite data. We only qualitatively assess fractional abundance maps of different unmixing algorithms by using the mineral map as a reference.

Fig. 8 shows fractional abundance maps estimated by the Tetracorder software product[6] and different unmixing algorithms for three prominent minerals in the subscene: *alunite, buddingtonite,* and *muscovite*. As in [15], [26], [32], the regularization parameter $\lambda$ for SUnSAL and CLSUnSAL is set to 0.001 and 0.01, respectively. We also set the parameters $\lambda = \lambda_{TV} = 0.001$ for SUnSAL-TV, DRSU-TV, and JSBUnSAL-TV and set $\lambda = \tau = 0.001$ for both ADSpLRU and JSpBLRU, similarly as in [31], [34]. From Fig. 8, we can observe that the highest abundances estimated by all seven algorithms are generally consistent with those obtained by the Tetracorder software product. This says that all seven unmixing algorithms are able to assess fractional abundance maps of respective minerals.

Furthermore, we list the runtime per pixel by different unmixing algorithms in Table 2. It says that SUnSAL and CLSUnSAL are very fast and JSBUnSAL-TV uses slightly more computational time than SUnSAL-TV and DRSU-TV, in line with the observation from simulated-data experiments. However, we also observe that both ADSpLRU and JSpBLRU cost more time than other algorithms, which may be due to the expensive step of the singular value decomposition for the abundance matrix with a larger size. In conclusion, the proposed JSBUnSAL-TV algorithm is effective for unmixing real hyperspectral data.

## V. CONCLUSION AND FUTURE WORK

In this paper, we have incorporated the joint-sparsity-blocks regression in TV spatial regularized spectral unmixing framework. The new model promotes adjacent pixels to have similar not only constituent endmembers but also fractional

[4]Available online: http://aviris.jpl.nasa.gov/html/aviris.freedata.html

[5]Available online: http://speclab.cr.usgs.gov/cuprite95.tgif.2.2um_map.gif

[6]Available online: https://speclab.cr.usgs.gov/PAPERS/tetracorder/

abundances, and moreover, to have similar sparsity pattern. We employ the ADMM to solve the proposed model, and obtain an algorithm called JSBUnSAL-TV. In particular, a reweighting strategy is utilized to enhance the sparsity long lines within each block. The experiments on both simulated and real data show that the proposed algorithm efficiently improves the abundance estimation performance. In the future, we will extend the $\ell_{2,1}$-blocks and TV mixed regularization to tensor-based hyperspectral image processing.

## REFERENCES

[1] J. M. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.

[2] W.-K. Ma, J. M. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A. J. Plaza, A. Ambikapathi, and C.-Y. Chi, "A signal processing perspective on hyperspectral unmixing: Insights from remote sensing," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 67–81, Jan. 2014.

[3] J. Boardman, F. A. Kruse, and R. O. Green, "Mapping target signatures via partial unmixing of AVIRIS data," in *Proc. JPL Airborne Earth Sci. Workshop*, vol. 1, 1995, pp. 23–26.

[4] M. E. Winter, "N-FINDR: An algorithm for fast autonomous spectral end-member determination in hyperspectral data," *Proc. SPIE*, vol. 3753, pp. 266–275, Oct. 1999.

[5] J. M. P. Nascimento and J. M. 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.

[6] J. Li and J. Bioucas-Dias, "Minimum volume simplex analysis: A fast algorithm to unmix hyperspectral data," in *Proc. IEEE IGARSS*, vol. 3, Jul. 2008, pp. III-250–III-253.

[7] X. Chen, J. Chen, X. Jia, B. Somers, J. Wu, and P. Coppin, "A quantitative analysis of virtual Endmembers' increased impact on the collinearity effect in spectral Unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 8, pp. 2945–2956, Aug. 2011.

[8] J. M. P. Nascimento and J. M. Bioucas-Dias, "Hyperspectral unmixing based on mixtures of Dirichlet components," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 3, pp. 863–878, Mar. 2012.

[9] X. Liu, W. Xia, B. Wang, and L. Zhang, "An approach based on constrained nonnegative matrix factorization to unmix hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 2, pp. 757–772, Feb. 2011.

[10] W. He, H. Zhang, and L. Zhang, "Sparsity-regularized robust non-negative matrix factorization for hyperspectral unmixing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4267–4279, Sep. 2016.

[11] X. Wang, Y. Zhong, L. Zhang, and Y. Xu, "Spatial group sparsity regularized nonnegative matrix factorization for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6287–6304, Nov. 2017.

[12] W. He, H. Zhang, and L. Zhang, "Total variation regularized reweighted sparse nonnegative matrix factorization for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3909–3921, Jul. 2017.

[13] F. Xiong, Y. Qian, J. Zhou, and Y. Y. Tang, "Hyperspectral unmixing via total variation regularized nonnegative tensor factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2341–2357, Apr. 2019.

[14] X.-R. Feng, H.-C. Li, J. Li, Q. Du, A. Plaza, and W. J. Emery, "Hyperspectral unmixing using sparsity-constrained deep nonnegative matrix factorization with total variation," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 6245–6257, Oct. 2018.

[15] 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.

[16] L. Sun, Z. Wu, L. Xiao, J. Liu, Z. Wei, and F. Dang, "A novel $l_{1/2}$ sparse regression method for hyperspectral unmixing," *Int. J. Remote Sens.*, vol. 34, no. 20, pp. 6983–7001, Oct. 2013.

[17] R. Rajabi and H. Ghassemian, "Spectral unmixing of hyperspectral imagery using multilayer NMF," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 1, pp. 38–42, Jan. 2015.

[18] G. Zhang, Y. Xu, and F. Fang, "Framelet-based sparse unmixing of hyperspectral images," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1516–1529, Apr. 2016.

[19] S. Zhang, J. Li, H.-C. Li, C. Deng, and A. Plaza, "Spectral-spatial weighted sparse regression for hyperspectral image unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3265–3276, Jun. 2018.

[20] S. Wang, T.-Z. Huang, X.-L. Zhao, G. Liu, and Y. Cheng, "Double reweighted sparse regression and graph regularization for hyperspectral unmixing," *Remote Sens.*, vol. 10, no. 7, p. 1046, 2018.

[21] L. Sun, W. Ge, Y. Chen, J. Zhang, and B. Jeon, "Hyperspectral unmixing employing $l_1$-$l_2$ sparsity and total variation regularization," *Int. J. Remote Sens.*, vol. 39, no. 19, pp. 6037–6060, 2018.

[22] J. Zou and J. Lan, "A multiscale hierarchical model for sparse hyperspectral unmixing," *Remote Sens.*, vol. 11, no. 5, p. 500, 2019.

[23] N. Dobigeon, J.-Y. Tourneret, and C.-I. Chang, "Semi-supervised linear spectral unmixing using a hierarchical Bayesian model for hyperspectral imagery," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 2684–2695, Jul. 2008.

[24] K. E. Themelis, A. Rontogiannis, and K. D. Koutroumbas, "A novel hierarchical Bayesian approach for sparse semisupervised hyperspectral unmixing," *IEEE Trans. Signal Process.*, vol. 60, no. 2, pp. 585–599, Feb. 2012.

[25] A. Rontogiannis, K. Themelis, O. Sykioti, and K. D. Koutroumbas, "A fast variational Bayes algorithm for sparse semi-supervised unmixing of OMEGA/Mars express data," in *Proc. 5th WHISPERS*, Jun. 2013, pp. 974–978.

[26] M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Collaborative sparse regression for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 341–354, Jan. 2014.

[27] H. Aggarwal and A. Majumdar, "Hyperspectral unmixing in the presence of mixed noise using joint-sparsity and total variation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4257–4266, Feb. 2016.

[28] Q. Qu, N. M. Nasrabadi, and T. D. Tran, "Abundance estimation for bilinear mixture models via joint sparse and low-rank representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 7, pp. 4404–4423, Jul. 2014.

[29] S. Zhang, J. Li, K. Liu, C. Deng, L. Liu, and A. Plaza, "Hyperspectral unmixing based on local collaborative sparse regression," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 5, pp. 631–635, May 2016.

[30] S. Huang, H. Zhang, and A. Pižurica, "A robust sparse representation model for hyperspectral image classification," *Sensors*, vol. 17, no. 9, p. 2087, Sep. 2017.

[31] J. Huang, T.-Z. Huang, L.-J. Deng, and X.-L. Zhao, "Joint-sparse-blocks and low-rank representation for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2419–2438, Apr. 2019.

[32] W. Tang, Z. Shi, Y. Wu, and C. Zhang, "Sparse unmixing of hyperspectral data using spectral *a Priori* information," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 770–783, Feb. 2015.

[33] R. Wang, H.-C. Li, W. Liao, X. Huang, and W. Philips, "Centralized collaborative sparse unmixing for hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 5, pp. 1949–1962, May 2017.

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

[35] Y. Zhong, R. Feng, and L. Zhang, "Non-local sparse unmixing for hyperspectral remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 1889–1909, Jun. 2014.

[36] Z. Shi, W. Tang, Z. Duren, and Z. Jiang, "Subspace matching pursuit for sparse unmixing of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3256–3274, Jun. 2014.

[37] C. Shi and L. Wang, "Incorporating spatial information in spectral unmixing: A review," *Remote Sens. Environ.*, vol. 149, pp. 70–87, Jun. 2014.

[38] X.-L. Zhao, F. Wang, T.-Z. Huang, M. K. Ng, and R. J. Plemmons, "Deblurring and sparse unmixing for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 7, pp. 4045–4058, Jul. 2013.

[39] J. Sigurdsson, M. O. Ulfarsson, and J. R. Sveinsson, "Blind hyperspectral unmixing using total variation and $\ell_q$ sparse regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 11, pp. 6371–6384, Nov. 2016.

[40] R. Wang, H.-C. Li, A. Pizurica, J. Li, A. Plaza, and W. J. Emery, "Hyperspectral unmixing using double reweighted sparse regression and total variation," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 1146–1150, Jul. 2017.

[41] X. Li, J. Huang, L.-J. Deng, and T.-Z. Huang, "Bilateral filter based total variation regularization for sparse hyperspectral image unmixing," *Inf. Sci.*, vol. 504, pp. 334–353, 2019.

[42] P. V. Giampouras, K. E. Themelis, A. A. Rontogiannis, and K. D. Koutroumbas, "Simultaneously sparse and low-rank abundance matrix estimation for hyperspectral image unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4775–4789, Aug. 2016.

[43] G. Ma, T.-Z. Huang, J. Huang, and C.-C. Zheng, "Local low-rank and sparse representation for hyperspectral image denoising," *IEEE Access*, vol. 7, pp. 79850–79865, 2019.

[44] J.-H. Yang, X.-L. Zhao, T.-H. Ma, Y. Chen, T.-Z. Huang, and M. Ding, "Remote sensing images destriping using unidirectional hybrid total variation and nonconvex low-rank regularization," *J. Comput. Appl. Math.*, vol. 363, pp. 124–144, Jan. 2020.

[45] R. Wang, W. Liao, H. Li, H. Zhang, and A. Pižurica, "Hyperspectral unmixing by reweighted low rank and total variation," in *Proc. 8th Workshop Hyperspectral Image Signal Process. Evol. Remote Sens.*, Los Angeles, CA, USA, Aug. 2016, pp. 1–4.

[46] M. Rizkinia and M. Okuda, "Joint local abundance sparse unmixing for hyperspectral images," *Remote Sens.*, vol. 9, no. 12, p. 1224, 2017.

[47] D. C. Heinz and C.-I. Chang, "Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 3, pp. 529–545, Mar. 2001.

[48] S. Zhang, J. Li, Z. Wu, and A. Plaza, "Spatial discontinuity-weighted sparse unmixing of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5767–5779, Oct. 2018.

[49] C. Wu and X.-C. Tai, "Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models," *SIAM J. Imag. Sci.*, vol. 3, no. 3, pp. 300–339, Jan. 2010.

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

[51] X.-L. Zhao, W. Wang, T. Zeng, T.-Z. Huang, and M. K. Ng, "Total variation structured total least squares method for image restoration," *SIAM J. Sci. Comput.*, vol. 35, no. 6, pp. 1304–1320, 2013.

[52] J. Huang and T.-Z. Huang, "A nonstationary accelerating alternating direction method for frame-based Poissonian image deblurring," *J. Comput. Appl. Math.*, vol. 352, pp. 181–193, May 2019.

[53] L.-J. Deng, M. Feng, and X.-C. Tai, "The fusion of panchromatic and multispectral remote sensing images via tensor-based sparse modeling and hyper-Laplacian prior," *Inf. Fusion*, vol. 52, pp. 76–89, Dec. 2019.

[54] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2479–2493, Jul. 2009.

[55] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted $\ell_1$ minimization," *J. Fourier Anal. Appl.*, vol. 14, nos. 5–6, pp. 877–905, 2008.

[56] J. Yang, Y.-Q. Zhao, J. C.-W. Chan, and S. G. Kong, "Coupled sparse denoising and unmixing with low-rank constraint for hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1818–1833, Mar. 2016.

[57] R. Clark, G. A. Swayze, K. E. Livo, R. F. Kokaly, S. J. Sutley, J. B. Dalton, R. R. McDougal, and C. A. Gent, "Imaging spectroscopy: Earth and planetary remote sensing with the USGS Tetracorder and expert systems," *J. Geophys. Res.*, vol. 108, no. E12, pp. 5131–5135, Dec. 2003.