Bilateral Joint-Sparse Regression for Hyperspectral Unmixing

Jie Huang, Wu-Chao Di, Jin-Ju Wang, Jie Lin and Ting-Zhu Huang

Abstract—Sparse hyperspectral unmixing has been a hot topic in recent years. Joint sparsity assumes that each pixel in a small neighborhood of hyperspectral images (HSIs) is composed of the same endmembers, which results in a few nonzero rows in the abundance matrix. Recall that a plethora of unmixing algorithms transform a 3-D HSI into a 2-D matrix with vertical priority. The transformation makes matrix computation easier. It is, however, hard to maintain the horizontal spatial information in HSIs in many cases. To make further use of the spatial information of HSIs, in this article, we propose a *bilateral joint-sparse structure* for hyperspectral unmixing in an attempt to exploit the local joint sparsity of the abundance matrix in both the vertical and horizontal directions. In particular, we introduce a permutation matrix to realize the bilateral joint-sparse representation and there is no need to construct the matrix explicitly. Moreover, we propose to simultaneously impose the bilateral joint-sparse structure and low rankness on the abundance and develop a new algorithm named *bilateral joint-sparse and low-rank unmixing*. The proposed algorithm is based on the alternating direction method of multipliers (ADMM) framework and employs a reweighting strategy. The convergence analysis of the proposed algorithm is investigated. Simulated and real experiments show the effectiveness of the proposed algorithm.

Index Terms—hyperspectral images, spectral unmixing, bilateral joint-sparse, low-rank matrix, alternating direction method of multipliers (ADMM), iterative reweighting.

I. INTRODUCTION

SPECTRAL unmixing of hyperspectral images (HSIs) has attracted much attention in different scientific fields [1]–[3]. It is the task of identifying the spectral signatures of distinct materials (*endmembers*) and estimating the fractions (*abundances*) of the materials for each pixel in HSI. The mixing process can be characterized as either linear [1] or nonlinear [4], [5]. In particular, bilinear mixture models have been proposed and used more commonly in practice [6]–[8]. Linear mixture model (LMM) assumes that the measurement spectrum of each pixel is a linear combination of the spectral signatures of the endmembers [3]. Due to its simplicity and tractability, we will adopt the LMM in spectral unmixing for the remainder of this article. Commonly, the abundance vector of a mixed pixel should satisfy the abundance nonnegative

constraint (ANC) and the abundance sum-to-one constraint (ASC) for physical meaning [9].

Nonnegative matrix factorization for spectral unmixing aims to find two nonnegative matrices, one for an endmembers' dictionary and another for fractional abundances, so that their product is equal to the target HSI matrix [10]–[13]. Recently, nonnegative tensor factorization has been proposed for hyperspectral unmixing to further exploit spatial information of HSIs and improve the unmixing performance [14]–[16]. With the available of the spectral dictionaries, sparse unmixing provides an alternative way for spectral unmixing. It assumes that, compared with large-scale dictionaries, only a few of spectral signatures participate in the LMM of each pixel, leading a sparse structure in the abundance matrix [17]. Diverse sparse structures have been exploited in the literature, including the standard ℓ_1-norm regularization [17], the $\ell_{2,1}$-norm regularization [18], and their variants [10], [19]–[27].

Exploiting spatial information in HSIs significantly improves the accuracy of spectral unmixing. Typically, the total variation (TV) regularization imposes the piecewise smooth on each abundance map [28]–[31]. It suppresses the noise and generates smooth abundance maps with preserved edges. In many cases, however, it is overstrict to assume that neighboring pixels have both similar mixing endmembers and similar abundance fractions. The low-rank representation provides another perspective of spatial correlation for spectral unmixing [6], [32]–[34]. In this vein, the highly spatial correlation of mixed pixels is transferred into the linear dependence among their corresponding abundance vectors. Simultaneously imposing the low-rank characteristic with sparsity has offered stimulating results [32]. Moreover, it is shown that the unmixing performance can be further improved by adopting a joint-sparse-blocks structure in the low-rank representation based unmixing [26]. In addition, nonlocal spatial information has been utilized to improve the performance of abundance estimation [35]–[40]. Broadly speaking, simultaneously exploiting spatial information and sparsity provides an effective way for spectral unmixing. Notice that most above-mentioned sparse unmixing algorithms first transfer an HSI, a third-order tensor, into a matrix, and then estimate the abundance matrix under the LMM. Thus, spatial information of HSIs might be lost in the transferring process [14]–[16], [41], [42].

In this article, we propose a *bilateral joint-sparse* regression for hyperspectral unmixing to make further use of the spatial information of HSIs. On the one hand, we consider a local joint sparsity structure of the abundance matrix, which is obtained, as usual, by arranging the HSI to be a matrix along the *vertical* direction. On the other hand, we consider the local

joint sparsity on a reshaped abundance matrix obtained by arranging the HSI along a *horizontal* direction. In this case, we utilize the spatial information along both the vertical and horizontal directions in HSIs. In particular, we introduce a permutation matrix to realize the bilateral joint-sparse representation and there is no need to build this permutation matrix explicitly. In addition, as previously stated, low rankness is a natural character of the abundance matrix. Thus, we propose to simultaneously impose the bilateral joint-sparse structure and low rankness on abundance matrices. We then develop an algorithm called *bilateral joint-sparse and low-rank unmixing* (BiJSpLRU) under the *alternating direction method of multipliers* (ADMM) framework combined with a reweighting strategy. The convergence analysis of BiJSpLRU is investigated. Simulated and real data experiments show that the proposed algorithm outperforms other state-of-the-art sparse unmixing algorithms.

The organization of this article is as follows. Section II introduces a bilateral joint-sparse structure and derives a BiJSpLRU algorithm with convergence results. Then, the performance of the proposed algorithm is demonstrated by both simulated experiments in Section III and a real-data experiment in Section IV. Finally, concluding remarks are given in Section V.

II. BiJSpLRU

In this section, we first briefly review several sparse unmixing models and then propose a bilateral joint-sparse structure and an unmixing model. After that we present an algorithm to solve the proposed model and finally investigate the convergence results of the proposed algorithm.

A. Sparse Unmixing Models

Let $Y \in \mathbb{R}^{l \times n}$ be the observed data matrix with l bands and n pixels and $A \in \mathbb{R}^{l \times m}$ be the spectral dictionary with m endmembers. The LMM assumes that

$$Y = AX + E \tag{1}$$

where $X \in \mathbb{R}^{m \times n}$ is the abundance matrix and $E \in \mathbb{R}^{l \times n}$ is an independent and identically distributed (i.i.d.) zero-mean Gaussian noise matrix. Two physical constraints: ANC and ASC, i.e.,

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

respectively, are often imposed on X [17]. Here we relax the ASC due to the variability of the endmembers, similarly as in [18], [28], [32], see more details in [17]. With large and available spectral dictionaries, many sparse unmixing algorithms have been well studied in the literature. Typically, the sparse unmixing algorithm via variable splitting and augmented Lagrangian (SUnSAL) [17] considers the following model

$$\min_{X} \frac{1}{2}\|Y - AX\|_F^2 + \lambda\|X\|_1 \tag{3}$$
$$\text{s.t. } X \geq 0$$

where $\lambda \geq 0$, $\|X\|_1 = \sum_i \sum_j |x_{ij}|$ is the ℓ_1-norm of X, and x_{ij} is the (i,j)th element of X. Note that SUnSAL exploits

the sparsity in each abundance vector but the collaborative SUnSAL (CLSUnSAL) [18] pays more attention to the joint sparsity, or row sparsity, of the abundance matrix. The CLSUn-SAL model is

$$\min_{X} \frac{1}{2}\|Y - AX\|_F^2 + \lambda\|X\|_{2,1} \tag{4}$$
$$\text{s.t. } X \geq 0$$

where $\|X\|_{2,1} = \sum_i \sqrt{\sum_j |x_{ij}|^2}$ is the $\ell_{2,1}$-norm of X. In addition, many sparse regularization terms, such as the weighted ℓ_1-norm, the ℓ_p-norm, and the $\ell_{p,q}$-norm, are utilized to enhance the sparsity of the abundance [19], [24], [25], [43], [44].

Recently, low-rank unmixing algorithms have attracted more and more attentions. The alternating direction sparse and low-rank unmixing (ADSpLRU) algorithm [32] simultaneously exploits sparsity and low rankness of the abundance, leading to an unmixing model as

$$\min_{X} \frac{1}{2}\|Y - AX\|_F^2 + \lambda\|Z \odot X\|_1 + \tau\|X\|_{b,*} \tag{5}$$
$$\text{s.t. } X \geq 0$$

where $\lambda \geq 0$, $\tau \geq 0$, $\|Z \odot X\|_1 = \sum_{i,j} z_{ij}|x_{ij}|$ is the weighted ℓ_1-norm of X [43], $\|X\|_{b,*} = \sum_i b_i\sigma_i(X)$ is the weighted nuclear norm of X [45], $\sigma_i(X)$ is the ith singular values of X, $Z = [z_{ij}]$ is a nonnegative weighting matrix and $b = [b_1, \cdots, b_r]$ is a nonnegative weighting vector with r being the rank of X.

Consider instead the local joint-sparsity, the joint-sparse-blocks and low-rank unmixing (JSpBLRU) algorithm [26] first partitions the abundance matrix as

$$X = [X_1, X_2, \cdots, X_s] \tag{6}$$

and then imposes the joint sparsity on each column block matrix X_j, where $X_j \in \mathbb{R}^{m \times d_j}$ with $\sum_{j=1}^{s} d_j = n$. Thus, the unmixing model becomes

$$\min_{X} \frac{1}{2}\|Y - AX\|_F^2 + \lambda\sum_{j=1}^{s} \|X_j\|_{w_j,2,1} + \tau\|X\|_{b,*} \tag{7}$$
$$\text{s.t. } X \geq 0$$

where

$$\|X_j\|_{w_j,2,1} = \sum_{i=1}^{m} w_{ij}\|X_j^{[i]}\|_2 \tag{8}$$

$X_j^{[i]}$ is the ith row of the jth block of X and $w_j = [w_{1j}, w_{2j}, \cdots, w_{mj}]$ is a nonnegative weighing vector computed by using a reweighting strategy. It is shown in [26], [32] that exploiting simultaneously sparsity and low rankness provides a new sight on the abundance matrix and the structural sparsity has great potential to achieve better abundance estimation results. In this vein, we aim to further exploit the structural sparsity by utilizing the spatial information in the HSI.

B. Bilateral Joint-Sparse Structure

We now introduce a bilateral joint-sparse structure for hyperspectral unmixing. The key idea is to incorporate more
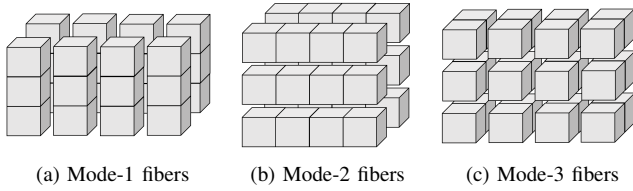
(a) Mode-1 fibers   (b) Mode-2 fibers   (c) Mode-3 fibers

Fig. 1: Fibers of a third-order tensor $\mathcal{Y} \in \mathbb{R}^{3 \times 4 \times 2}$.

(a) Vertical priority   (b) Horizontal priority

Fig. 2: Two ways of vectorization.

spatial information of HSIs into the structural sparsity. Recall that the main task of many sparse unmixing algorithms is to estimate the abundance of a reshaped HSI matrix Y as in (1). That said, a 3-D HSI, also a third-order tensor, is first transformed into a 2-D matrix and then one can estimate the abundance matrix. In fact, in this vein, it is hard to make full use of the spatial information of HSIs.

For example, consider a third-order tensor $\mathcal{Y} \in \mathbb{R}^{3 \times 4 \times 2}$ in [41], where

$$Y_1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, Y_2 = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix} \quad (9)$$

are the two frontal slices of $\mathcal{Y}$. Recall that the mode-n unfolding of a tensor $\mathcal{Y}$, denoted by $Y_{(n)}$, arranges the mode-n fibers to be the columns of the resulting matrix. Here we show the mode-n fibers of $\mathcal{Y}$ in Fig. 1. Then from [41], three mode-n unfoldings of $\mathcal{Y}$ are

$$Y_{(1)} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix}$$

$$Y_{(2)} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{bmatrix} \quad (10)$$

$$Y_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & \cdots & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & \cdots & 21 & 22 & 23 & 24 \end{bmatrix}.$$

If we view $\mathcal{Y}$ as an HSI in hyperspectral unmixing, then the task in [17], [18], [26], [28], [32], to name a few, is to unmix the matrix $Y_{(3)}$. In this vein, to utilize the sparsity of the abundance, SUnSAL and ADSpLRU consider the sparsity of each abundance vector but JSpBLRU emphasizes the joint sparsity of the neighboring abundance vectors. Observe that the mode-3 unfolding $Y_{(3)}$ is obtained by arranging the mode-3 fibers of $\mathcal{Y}$, as in Fig. 1(c), to be the columns of $Y_{(3)}$. Particularly, the vertical direction gets priority over the horizontal direction in the arrangement of the frontal slice Y_1 (also for Y_2), see Fig. 2(a). It follows that the local joint sparsity on the abundance matrix of $Y_{(3)}$ exploits the spatial information only along the vertical direction of $\mathcal{Y}$.

In attempt to further utilize the spatial information of the HSI, we propose to consider the mode-3 matricization of $\mathcal{Y}$ along the *vertical* and *horizontal* directions simultaneously. To this end, besides the construction of $Y_{(3)}$ in (10) with vertical priority, we construct another mode-3 unfolding of $\mathcal{Y}$:

$$Y'_{(3)} = \begin{bmatrix} 1 & 4 & 7 & 10 & \cdots & 3 & 6 & 9 & 12 \\ 13 & 16 & 19 & 22 & \cdots & 15 & 18 & 21 & 24 \end{bmatrix} \quad (11)$$

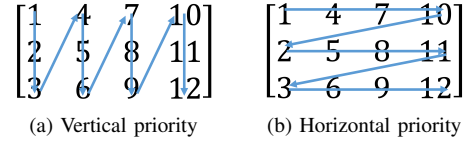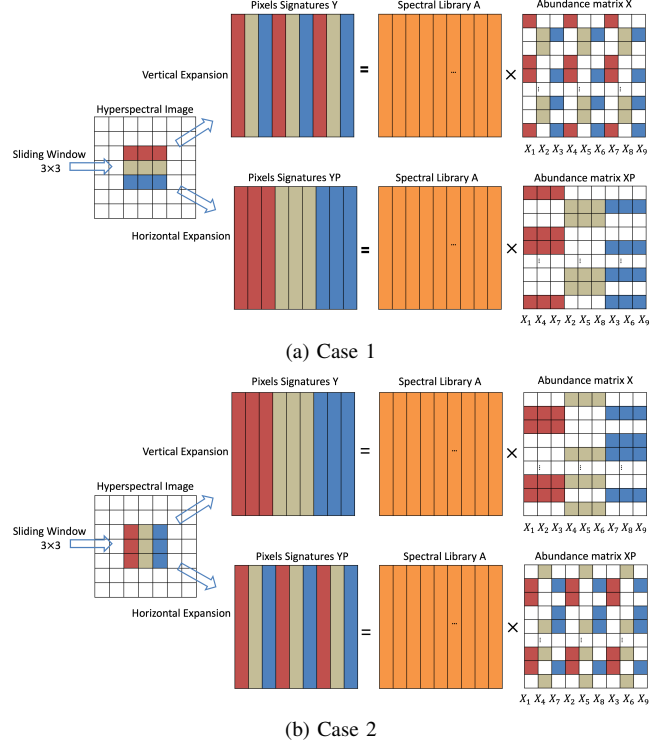(a) Case 1

(b) Case 2

Fig. 3: Vertical and horizontal expansions of the pixels in HSIs for hyperspectral unmixing. Note that there is no need to construct the permutation matrix $P \in \mathbb{R}^{9 \times 9}$ explicitly.

with horizontal priority as in Fig. 2(b). Then we can exploit the local spatial information from both $Y_{(3)}$ and $Y'_{(3)}$.

Following the analysis, for hyperspectral unmixing, we propose to impose the joint-sparse-blocks structure on the abundance matrices of the mode-3 unfoldings along both the vertical and horizontal directions and name it as a *bilateral joint-sparse structure*. To see it more clearly, we show two simple and extreme cases in Fig. 3. We consider nine pixels in the sliding window. Case 1 has a clear spatial characteristic along the horizontal direction and case 2 shows a vertical characteristic. From Fig. 3 we see that vertical and horizontal expansions have their respective advantages. In particular, we can mathematically realize the bilateral matricization via a permutation matrix. Partition the abundance matrix X of the mode-3 unfolding of the pixels with vertical priority as

$$X = [X_1, X_2, \cdots, X_9] \quad (12)$$

where X_i is the ith column of X. Then the abundance matrix with horizontal priority is obtained by permuting the columns of X, that is,

$$XP = [X_1, X_4, X_7, X_2, X_5, X_8, X_3, X_6, X_9] \quad (13)$$

where $\boldsymbol{P} \in \mathbb{R}^{9 \times 9}$ is a permutation matrix [46]. Note that we can reshape the abundance matrix to do the permutation so there is no need to construct the matrix $\boldsymbol{P}$ explicitly, which is time-saving and easy to implement. Then we realize the bilateral joint-sparse structure by simultaneously imposing the joint-sparse-blocks structure on $\boldsymbol{X}$ and $\boldsymbol{XP}$. In this vein, we incorporate both the vertical and horizontal spatial information of HSIs into the sparse structure of the abundance matrix. In practical situations, utilizing bilateral spatial information certainly provides an advantage over either vertical or horizontal expansions, as will be demonstrated later in Sections III and IV.

C. Proposed Model and BiJSpLRU Algorithm

Now we propose an unmixing model by considering both the bilateral joint-sparse structure via the weighted $\ell_{2,1}$-norm and the low-rank character via the weighted nuclear norm. Reconsider the LMM model in (1) and the abundance matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$. Note that $\boldsymbol{X}$ is obtained corresponding to the vertical expansion of the HSI. Let $\boldsymbol{XP} \in \mathbb{R}^{m \times n}$ be the abundance matrix with respect to the horizontal expansion of the HSI, where $\boldsymbol{P} \in \mathbb{R}^{n \times n}$ is the corresponding permutation matrix. We impose the joint-sparse-blocks structure on both $\boldsymbol{X}$ and $\boldsymbol{XP}$. That is, we first partition $\boldsymbol{X}$ and $\boldsymbol{XP}$ as

$$\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_s]$$
$$\boldsymbol{XP} = [(\boldsymbol{XP})_1, (\boldsymbol{XP})_2, \cdots, (\boldsymbol{XP})_s] \tag{14}$$

where $\boldsymbol{X}_j$ and $(\boldsymbol{XP})_j \in \mathbb{R}^{m \times d_j}$ are the jth column blocks of $\boldsymbol{X}$ and $\boldsymbol{XP}$, respectively, for $\sum_{j=1}^{s} d_j = n$. We note that both $\boldsymbol{X}$ and $\boldsymbol{XP}$ share the same partition strategy for model simplicity. Then the proposed optimization problem is

$$\min_{\boldsymbol{X}} \frac{1}{2} \|\boldsymbol{Y} - \boldsymbol{AX}\|_F^2 + \lambda_1 \sum_{j=1}^{s} \|(\boldsymbol{XP})_j\|_{\boldsymbol{W}_{1,j},2,1}$$
$$+ \lambda_2 \sum_{j=1}^{s} \|\boldsymbol{X}_j\|_{\boldsymbol{W}_{2,j},2,1} + \tau \|\boldsymbol{X}\|_{\boldsymbol{W}_3,*} \tag{15}$$

s.t. $\boldsymbol{X} \geq \boldsymbol{0}$

where λ_1, λ_2, and τ are nonnegative regularization parameters, $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, and $\boldsymbol{W}_3$ are nonnegative weighting matrices with

$$\boldsymbol{W}_1 = [\boldsymbol{W}_{1,1}, \cdots, \boldsymbol{W}_{1,s}] \in \mathbb{R}^{m \times s}$$
$$\boldsymbol{W}_2 = [\boldsymbol{W}_{2,1}, \cdots, \boldsymbol{W}_{2,s}] \in \mathbb{R}^{m \times s} \tag{16}$$

and $\boldsymbol{W}_3$ is a diagonal matrix of order m. Particularly,

$$\|(\boldsymbol{XP})_j\|_{\boldsymbol{W}_{1,j},2,1} = \sum_{i=1}^{m} (\boldsymbol{W}_1)_{ij} \|(\boldsymbol{XP})_j^{[i]}\|_2$$
$$\|\boldsymbol{X}_j\|_{\boldsymbol{W}_{2,j},2,1} = \sum_{i=1}^{m} (\boldsymbol{W}_2)_{ij} \|\boldsymbol{X}_j^{[i]}\|_2 \tag{17}$$
$$\|\boldsymbol{X}\|_{\boldsymbol{W}_3,*} = \sum_{i=1}^{m} (\boldsymbol{W}_3)_{ii} \sigma_i(\boldsymbol{X})$$

where, for any matrix $\boldsymbol{T}$, $(\boldsymbol{T})_{ij}$ is the (i,j)th element of $\boldsymbol{T}$. We will use a reweighting strategy to automatically update $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, and $\boldsymbol{W}_3$ at each iteration. Notice that the model in (15) has three regularization parameters so it may be time-consuming

to choose optimal parameters in real applications. In our experience, λ_1 and λ_2 attain the same optimal regularization values for many cases. Therefore for simplicity, we set $\lambda_1 = \lambda_2 = \lambda$ in (15) and the proposed model becomes

$$\min_{\boldsymbol{X}} \frac{1}{2} \|\boldsymbol{Y} - \boldsymbol{AX}\|_F^2 + \lambda \sum_{j=1}^{s} \|(\boldsymbol{XP})_j\|_{\boldsymbol{W}_{1,j},2,1}$$
$$+ \lambda \sum_{j=1}^{s} \|\boldsymbol{X}_j\|_{\boldsymbol{W}_{2,j},2,1} + \tau \|\boldsymbol{X}\|_{\boldsymbol{W}_3,*} \tag{18}$$

s.t. $\boldsymbol{X} \geq \boldsymbol{0}$.

In the following, we show how to solve the proposed model in (18) under the ADMM framework [47]. To this end, we first introduce four variables $\boldsymbol{V}_1$, $\boldsymbol{V}_2$, $\boldsymbol{V}_3$, and $\boldsymbol{V}_4$. Partition $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$ as $\boldsymbol{XP}$ and $\boldsymbol{X}$ in (14), respectively,

$$\boldsymbol{V}_1 = [\boldsymbol{V}_{1,1}, \boldsymbol{V}_{1,2}, \cdots, \boldsymbol{V}_{1,s}]$$
$$\boldsymbol{V}_2 = [\boldsymbol{V}_{2,1}, \boldsymbol{V}_{2,2}, \cdots, \boldsymbol{V}_{2,s}]. \tag{19}$$

Then using variable replacement, we equivalently transform (18) into the following model

$$\min_{\boldsymbol{X}, \boldsymbol{V}_1, \cdots, \boldsymbol{V}_4} \frac{1}{2} \|\boldsymbol{Y} - \boldsymbol{AX}\|_F^2 + \lambda \sum_{i=1}^{s} \|\boldsymbol{V}_{1,j}\|_{\boldsymbol{W}_{1,j},2,1}$$
$$+ \lambda \sum_{j=1}^{s} \|\boldsymbol{V}_{2,j}\|_{\boldsymbol{W}_{2,j},2,1} + \tau \|\boldsymbol{V}_3\|_{\boldsymbol{W}_3,*} + \iota_{\mathbb{R}^+}(\boldsymbol{V}_4) \tag{20}$$

s.t. $\boldsymbol{XP} = \boldsymbol{V}_1$, $\boldsymbol{X} = \boldsymbol{V}_2$, $\boldsymbol{X} = \boldsymbol{V}_3$, $\boldsymbol{X} = \boldsymbol{V}_4$

where $\iota_{\mathbb{R}^+}(x)$ is the indicator function, i.e., $\iota_{\mathbb{R}^+}(x) = 0$ if x is nonnegative and $\iota_{\mathbb{R}^+}(x) = +\infty$ otherwise. Let

$$f(\boldsymbol{X}) = \frac{1}{2} \|\boldsymbol{Y} - \boldsymbol{AX}\|_F^2$$
$$g(\boldsymbol{V}, \boldsymbol{W}) = \lambda \sum_{j=1}^{s} \|\boldsymbol{V}_{1,j}\|_{\boldsymbol{W}_{1,j},2,1} + \lambda \sum_{j=1}^{s} \|\boldsymbol{V}_{2,j}\|_{\boldsymbol{W}_{2,j},2,1} \tag{21}$$
$$+ \tau \|\boldsymbol{V}_3\|_{\boldsymbol{W}_3,*} + \iota_{\mathbb{R}^+}(\boldsymbol{V}_4)$$

and define

$$\boldsymbol{V} = [\boldsymbol{V}_1, \boldsymbol{V}_2, \boldsymbol{V}_3, \boldsymbol{V}_4] \in \mathbb{R}^{m \times 4n}$$
$$\boldsymbol{G} = [\boldsymbol{P}, \boldsymbol{I}, \boldsymbol{I}, \boldsymbol{I}] \in \mathbb{R}^{n \times 4n} \tag{22}$$
$$\boldsymbol{W} = [\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3] \in \mathbb{R}^{m \times (2s+m)}$$

where $\boldsymbol{I}$ is an n-by-n identity matrix. Clearly (20) becomes

$$\min_{\boldsymbol{X}, \boldsymbol{V}} \quad f(\boldsymbol{X}) + g(\boldsymbol{V}, \boldsymbol{W})$$
$$\text{s.t. } \boldsymbol{XG} = \boldsymbol{V}. \tag{23}$$

We note that a reweighting strategy is employed to update $\boldsymbol{W}$ in each iterate.

Define the augmented Lagrangian function

$$\mathcal{L}(\boldsymbol{X}, \boldsymbol{V}, \boldsymbol{W}, \boldsymbol{\Lambda}) = f(\boldsymbol{X}) + g(\boldsymbol{V}, \boldsymbol{W})$$
$$+ \langle \boldsymbol{\Lambda}, \boldsymbol{XG} - \boldsymbol{V} \rangle + \frac{\mu}{2} \|\boldsymbol{XG} - \boldsymbol{V}\|_F^2 \tag{24}$$

where $\boldsymbol{\Lambda} = [\boldsymbol{\Lambda}_1, \boldsymbol{\Lambda}_2, \boldsymbol{\Lambda}_3, \boldsymbol{\Lambda}_4]$ is the Lagrange multiplier, $\mu > 0$, and $\langle \boldsymbol{R}, \boldsymbol{T} \rangle = \text{Tr}(\boldsymbol{R}^T \boldsymbol{T})$ is the standard inner product between matrices, where $\boldsymbol{R}, \boldsymbol{T} \in \mathbb{R}^{n_1 \times n_2}$. Partition $\boldsymbol{X}$ and $\boldsymbol{XP}$

as in (14) and

$$\Lambda_1 = [\Lambda_{1,1}, \cdots, \Lambda_{1,s}] \\ \Lambda_2 = [\Lambda_{2,1}, \cdots, \Lambda_{2,s}]. \quad (25)$$

Then the ADMM framework with a reweighting strategy is derived

$$\begin{cases} \boldsymbol{X}^{k+1} = \underset{\boldsymbol{X}}{\operatorname{argmin}} \, \mathcal{L}(\boldsymbol{X}, \boldsymbol{V}^k, \boldsymbol{W}^k, \boldsymbol{\Lambda}^k) \\ (\boldsymbol{W}_1^{k+1})_{ij} = \dfrac{1}{\|((\boldsymbol{X}^{k+1}\boldsymbol{P})_j + \frac{1}{\mu}\boldsymbol{\Lambda}_{1,j}^k)^{[i]}\|_2 + \epsilon} \\ (\boldsymbol{W}_2^{k+1})_{ij} = \dfrac{1}{\|(\boldsymbol{X}_j^{k+1} + \frac{1}{\mu}\boldsymbol{\Lambda}_{2,j}^k)^{[i]}\|_2 + \epsilon} \\ (\boldsymbol{W}_3^{k+1})_{ii} = \dfrac{1}{\sigma_i(\boldsymbol{X}^{k+1} + \frac{1}{\mu}\boldsymbol{\Lambda}_3^k) + \epsilon} \\ \boldsymbol{V}^{k+1} = \underset{\boldsymbol{V}}{\operatorname{argmin}} \, \mathcal{L}(\boldsymbol{X}^{k+1}, \boldsymbol{V}, \boldsymbol{W}^{k+1}, \boldsymbol{\Lambda}^k) \\ \boldsymbol{\Lambda}^{k+1} = \boldsymbol{\Lambda}^k + \mu(\boldsymbol{X}^{k+1}\boldsymbol{G} - \boldsymbol{V}^{k+1}) \end{cases} \quad (26)$$

for $i = 1, \ldots, m$, $j = 1, \ldots, s$, where $\sigma_i(\boldsymbol{X}^{k+1} + \frac{1}{\mu}\boldsymbol{\Lambda}_3^k)$ is the ith singular value of $\boldsymbol{X}^{k+1} + \frac{1}{\mu}\boldsymbol{\Lambda}_3^k$ and $\epsilon = 10^{-16}$ is a small constant added to avoid singularities. Now we solve each subproblem.

• *Update $\boldsymbol{X}$*. For $\boldsymbol{X}$-subproblem, it is to solve

$$\min_{\boldsymbol{X}} \frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{AX}\|_F^2 + \langle \boldsymbol{\Lambda}^k, \boldsymbol{XG} - \boldsymbol{V}^k \rangle + \frac{\mu}{2}\|\boldsymbol{XG} - \boldsymbol{V}^k\|_F^2. \quad (27)$$

Clearly, the optimality condition is

$$\boldsymbol{0} = \boldsymbol{A}^T(\boldsymbol{AX}^{k+1} - \boldsymbol{Y}) + \boldsymbol{\Lambda}^k \boldsymbol{G}^T + \mu(\boldsymbol{X}^{k+1}\boldsymbol{G} - \boldsymbol{V}^k)\boldsymbol{G}^T. \quad (28)$$

Recall that $\boldsymbol{P}$ is a permutation matrix with $\boldsymbol{PP}^T = \boldsymbol{I}$ and so $\boldsymbol{GG}^T = 4\boldsymbol{I}$, then we have

$$\boldsymbol{X}^{k+1} = (\boldsymbol{A}^T\boldsymbol{A} + 4\mu\hat{\boldsymbol{I}})^{-1}(\boldsymbol{A}^T\boldsymbol{Y} + (\mu\boldsymbol{V}^k - \boldsymbol{\Lambda}^k)\boldsymbol{G}^T) \quad (29)$$

where $\hat{\boldsymbol{I}}$ is an m-by-m identity matrix.

• *Update $\boldsymbol{W}$*. With obtained $\boldsymbol{X}^{k+1}$, we update $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, and $\boldsymbol{W}_3$ as in (26), which will be used in the following $\boldsymbol{V}$-subproblem. We note that the weighting coefficients in $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ are computed, similarly as in [26], to enhance the row sparsity of $\boldsymbol{V}_1$ and $\boldsymbol{V}_2$, respectively. The weighting coefficients in $\boldsymbol{W}_3$ are used to enhance the sparsity of singular values of $\boldsymbol{V}_3$, similarly as in [26], [32], [45], [48].

• *Update $\boldsymbol{V}$*. Clearly, the minimization problem about $\boldsymbol{V}$ can be decoupled and so we solve the $\boldsymbol{V}_i$-subproblem, $i = 1, \ldots, 4$, separately.

○ *Update $\boldsymbol{V}_1$*. For $\boldsymbol{V}_1$-subproblem, after dropping constant terms, we obtain

$$\min_{\boldsymbol{V}_1} \lambda \sum_{j=1}^s \|\boldsymbol{V}_{1,j}\|_{\boldsymbol{W}_{1,j}^{k+1},2,1} + \langle \boldsymbol{\Lambda}_1^k, -\boldsymbol{V}_1 \rangle + \frac{\mu}{2}\|\boldsymbol{X}^{k+1}\boldsymbol{P} - \boldsymbol{V}_1\|_F^2. \quad (30)$$

Partition $\boldsymbol{XP}$, $\boldsymbol{V}_1$, and $\boldsymbol{\Lambda}_1$ as in (14), (19), and (25), respectively. For each block of $\boldsymbol{V}_{1,j}$, we obtain

$$\boldsymbol{V}_{1,j}^{k+1} = \underset{\boldsymbol{V}_{1,j}}{\operatorname{argmin}} \, \lambda\|\boldsymbol{V}_{1,j}\|_{\boldsymbol{W}_{1,j}^{k+1},2,1} + \frac{\mu}{2}\|(\boldsymbol{X}^{k+1}\boldsymbol{P})_j - \boldsymbol{V}_{1,j} + \frac{1}{\mu}\boldsymbol{\Lambda}_{1,j}^k\|_F^2. \quad (31)$$

Define the vect-soft operator as

$$\text{vect-soft}_\alpha(\boldsymbol{x}) = \boldsymbol{x}\frac{\max\{\|\boldsymbol{x}\|_2 - \alpha, 0\}}{\max\{\|\boldsymbol{x}\|_2 - \alpha, 0\} + \alpha} \quad (32)$$

for $\boldsymbol{x} \in \mathbb{R}^n$ and $\alpha > 0$. Then from [49] we obtain that each subproblem in (31) has a unique solution and its ith row can be obtained by

$$(\boldsymbol{V}_{1,j}^{k+1})^{[i]} = \text{vect-soft}_{\frac{\lambda}{\mu}(\boldsymbol{W}_1^{k+1})_{ij}}(((\boldsymbol{X}^{k+1}\boldsymbol{P})_j + \frac{1}{\mu}\boldsymbol{\Lambda}_{1,j}^k)^{[i]}) \quad (33)$$

where $((\boldsymbol{X}^{k+1}\boldsymbol{P})_j + \frac{1}{\mu}\boldsymbol{\Lambda}_{1,j}^k)^{[i]}$ is the ith row of the jth column block of $\boldsymbol{X}^{k+1}\boldsymbol{P} + \frac{1}{\mu}\boldsymbol{\Lambda}_1^k$, for $i = 1, \cdots, m$ and $j = 1, \cdots, s$.

○ *Update $\boldsymbol{V}_2$*. The $\boldsymbol{V}_2$-subproblem is

$$\min_{\boldsymbol{V}_2} \lambda \sum_{j=1}^s \|\boldsymbol{V}_{2,j}\|_{\boldsymbol{W}_{2,j}^{k+1},2,1} + \frac{\mu}{2}\|\boldsymbol{X}^{k+1} - \boldsymbol{V}_2 + \frac{1}{\mu}\boldsymbol{\Lambda}_2^k\|_F^2 \quad (34)$$

which is similar to the $\boldsymbol{V}_1$-subproblem. Partition $\boldsymbol{\Lambda}_2$ as in (25), then similarly, the ith row of $\boldsymbol{V}_{2,j}^{k+1}$ can be obtained by

$$(\boldsymbol{V}_{2,j}^{k+1})^{[i]} = \text{vect-soft}_{\frac{\lambda}{\mu}(\boldsymbol{W}_2^{k+1})_{ij}}((\boldsymbol{X}_j^{k+1} + \frac{1}{\mu}\boldsymbol{\Lambda}_{2,j}^k)^{[i]}) \quad (35)$$

for $i = 1, \cdots, m$ and $j = 1, \cdots, s$.

○ *Update $\boldsymbol{V}_3$*. Let $\boldsymbol{X} = \tilde{\boldsymbol{U}}Diag(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_r)\tilde{\boldsymbol{V}}^T$ be the singular value decomposition (SVD) of $\boldsymbol{X}$ and $\tilde{\sigma}_i$ is the ith singular value of $\boldsymbol{X}$. Define the weighted singular value thresholding operator $\mathbf{SVT}$ on $\boldsymbol{X}$ as

$$\mathbf{SVT}_{S,\beta}(\boldsymbol{X}) \\ = \tilde{\boldsymbol{U}}Diag\left((\tilde{\sigma}_1 - \beta s_{11})_+, \ldots, (\tilde{\sigma}_r - \beta s_{rr})_+\right)\tilde{\boldsymbol{V}}^T \quad (36)$$

where $(\,\cdot\,)_+ = \max(\,\cdot, 0)$, $\boldsymbol{S} = Diag(s_{11}, \ldots, s_{rr})$ is a nonnegative weighting diagonal matrix, and β is a positive parameter. Then consider the $\boldsymbol{V}_3$-subproblem

$$\min_{\boldsymbol{V}_3} \tau\|\boldsymbol{V}_3\|_{\boldsymbol{W}_3^{k+1},*} + \frac{\mu}{2}\|\boldsymbol{X}^{k+1} - \boldsymbol{V}_3 + \frac{1}{\mu}\boldsymbol{\Lambda}_3^k\|_F^2 \quad (37)$$

we obtain its closed-form solution

$$\boldsymbol{V}_3^{k+1} = \mathbf{SVT}_{\boldsymbol{W}_3^{k+1},\frac{\tau}{\mu}}(\boldsymbol{X}^{k+1} + \frac{1}{\mu}\boldsymbol{\Lambda}_3^k). \quad (38)$$

○ *Update $\boldsymbol{V}_4$*. Consider the $\boldsymbol{V}_4$-subproblem

$$\boldsymbol{V}_4^{k+1} = \underset{\boldsymbol{V}_4}{\operatorname{argmin}} \, \iota_{\mathbb{R}^+}(\boldsymbol{V}_4) + \frac{\mu}{2}\|\boldsymbol{X}^{k+1} - \boldsymbol{V}_4 + \frac{1}{\mu}\boldsymbol{\Lambda}_4^k\|_F^2. \quad (39)$$

It is easy to obtain that

$$\boldsymbol{V}_4^{k+1} = \max(\boldsymbol{X}^{k+1} + \frac{1}{\mu}\boldsymbol{\Lambda}_4^k, \boldsymbol{0}). \quad (40)$$

• *Update multipliers*. The update rules of multipliers are as follows

$$\begin{cases} \boldsymbol{\Lambda}_1^{k+1} = \boldsymbol{\Lambda}_1^k + \mu(\boldsymbol{X}^{k+1}\boldsymbol{P} - \boldsymbol{V}_1^{k+1}) \\ \boldsymbol{\Lambda}_2^{k+1} = \boldsymbol{\Lambda}_2^k + \mu(\boldsymbol{X}^{k+1} - \boldsymbol{V}_2^{k+1}) \\ \boldsymbol{\Lambda}_3^{k+1} = \boldsymbol{\Lambda}_3^k + \mu(\boldsymbol{X}^{k+1} - \boldsymbol{V}_3^{k+1}) \\ \boldsymbol{\Lambda}_4^{k+1} = \boldsymbol{\Lambda}_4^k + \mu(\boldsymbol{X}^{k+1} - \boldsymbol{V}_4^{k+1}). \end{cases} \quad (41)$$

To clarify, we summarize the proposed BiJSpLRU algorithm in the following.

Algorithm: Pseudocode of the BiJSpLRU Algorithm

Input: Y and A
Select parameters: μ, τ, λ, and maximum iterations
Initialization: V^0, Λ^0, and set $k = 0$
Repeat:
   Compute X^{k+1} by (29)
   Compute W_1^{k+1}, W_2^{k+1}, and W_3^{k+1} by (26)
   Compute the ith row of the jth block of V_1^{k+1} by (33)
   Compute the ith row of the jth block of V_2^{k+1} by (35)
   Compute V_3^{k+1} by (38)
   Compute V_4^{k+1} by (40)
   Update Λ_i^{k+1} by (41), for $i = 1, \cdots, 4$.
until some stopping criterion is satisfied
Output: $\hat{X} = X^{k+1}$

We remark that both JSpBLUR and BiJSpBLRU have the same computational complexity. Compared with JSpBLUR, however, BiJSpBLRU mainly has one extra $\ell_{2,1}$-norm subproblem, i.e., V_1-subproblem.

D. Convergence of BiJSpLRU

There are many convergence results about ADMM iterates in the literature, such as [50]–[53]. Following the analysis in [54], we now consider the residual and objective convergence of BiJSpLRU under the following assumption.

Assumption 1. *The unaugmented Lagrangian*

$$\mathcal{L}_0(X, V, W, \Lambda) = f(X) + g(V, W) + \langle \Lambda, XG - V \rangle \quad (42)$$

has a saddle point. That is, there exist $(X^\star, V^\star, W^\star, \Lambda^\star)$, for which

$$\mathcal{L}_0(X^\star, V^\star, W^\star, \Lambda) \leq \mathcal{L}_0(X^\star, V^\star, W^\star, \Lambda^\star)$$
$$\leq \mathcal{L}_0(X, V, W, \Lambda^\star) \quad (43)$$

holds for all X, V, W, Λ.

From Assumption 1, we get that $\mathcal{L}_0(X^\star, V^\star, W^\star, \Lambda^\star)$ is finite, $(X^\star, V^\star, W^\star)$ is a solution to (23), and $X^\star G - V^\star = 0$ with $f(X^\star) < \infty$ and $g(V^\star, W^\star) < \infty$. Let $p^k = f(X^k) + g(V^k, W^k)$, $r^k = X^k G - V^k$, and $p^\star = f(X^\star) + g(V^\star, W^\star)$. Then we have the following convergence results of the sequence $(X^k, V^k, W^k, \Lambda^k)$ generated by (26).

Theorem 1. *Let $(X^k, V^k, W^k, \Lambda^k)$ be the sequence generated by (26). Under Assumption 1 and let $W^{k+1} \leq W^\star$, then the sequence $(X^k, V^k, W^k, \Lambda^k)$ satisfies the following:*

(1) Residual convergence. The residual error $r^k \to 0$ as $k \to \infty$.

(2) Objective convergence. The objective function of the iterates approaches the optimal value: $f(X^k) + g(V^k, W^k) \to p^\star$ as $k \to \infty$.

Proof. The proof can be found in the Appendix. $\square$

Finally, we remark that in a similar way, one can obtain the convergence results of ADSpLRU and JSpBLRU, which are also based on ADMM and a reweighting strategy.

III. EXPERIMENTS ON SIMULATED DATA

In this section, we test the performance of SUnSAL [17], CLSUnSAL [18], SUnSAL-TV[1] [28], ADSpLRU [32], and JSpBLRU [26], and BiJSpLRU [2] on simulated data. Our tests were done by using MATLAB R2019a on a desktop with 3.60 GHz Intel Core i9 and 32 GB memory.

A. Experiment Setting

In our experiments, two spectral dictionaries are used:

1) $A_1 \in \mathbb{R}^{224 \times 240}$: A dictionary of minerals extracted from the U.S. Geological Survey (USGS). The reflectance values are given in 224 spectral bands with uniform distribution in the interval of 0.4–2.5 μm.

2) $A_2 \in \mathbb{R}^{99 \times 120}$: It consists of 120 materials and 99 spectral bands. It is from the National Aeronautics and Space Administration Johnson Space Center Spacecraft Materials Spectral Database.

We use the root-mean-square error (RMSE) and the signal-to-reconstruction error (SRE) to measure the performance of the estimated abundance matrices. They are defined as,

$$\text{RMSE} = \sqrt{\frac{1}{mn} \sum_{i=1}^{n} \|\hat{x}_i - x_i\|_2^2}$$
$$\text{SRE (dB)} = 10 \log_{10} \left(\frac{\sum_{i=1}^{n} \|x_i\|_2^2}{\sum_{i=1}^{n} \|\hat{x}_i - x_i\|_2^2} \right) \quad (44)$$

respectively, where $\hat{x}_i$ and x_i are the estimated and true abundance vectors of the ith pixel. Generally speaking, lower RMSEs and higher SREs give better estimations. For all six algorithms, regularization parameters are chosen to get maximum SREs. Similarly as in [26], we set $d = 3$ for both JSpBLRU and BiJSpLRU in the experiments. That said, for n pixels in an HSI, let $s = \lfloor n/d \rfloor$ be the largest integer no greater than n/d, then each of the first $s - 1$ blocks in (14) contains d columns and the last block contains the remains. We select optimal regularization parameters for low-rank unmixing algorithms: ADSpLRU, JSpBLRU, and BiJSpLRU, as follows:

$$\lambda \in \{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$$
$$\tau \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100\}.$$

We select optimal regularization parameters for SUnSAL, CLSUnSAL and SUnSAL-TV from the following sequence:

$$\{0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\}.$$

In addition, we select μ from $\{0.001, 0.01, 0.1, 1\}$. All possible combinations are considered.

For BiJSpLRU, recall the primal residual r^k and define the dual residual s^k at the kth iteration as

$$r^k = X^k G - V^k$$
$$s^k = \mu(V^k - V^{k-1})G^T. \quad (45)$$

---

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

[2]The MATLAB codes of JSpBLRU and BiJSpLRU are available at https://huangjie-uestc.github.io/
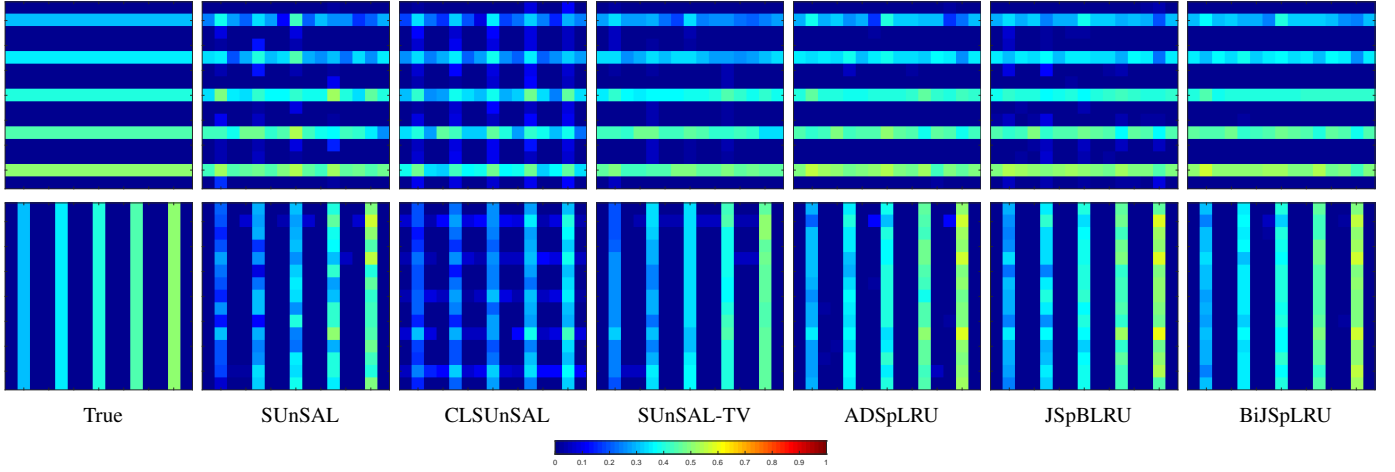
Fig. 4: True and estimated abundance maps for Example 1 with SNR = 30 dB for endmembers (top row) #1 and (bottom row) #2.
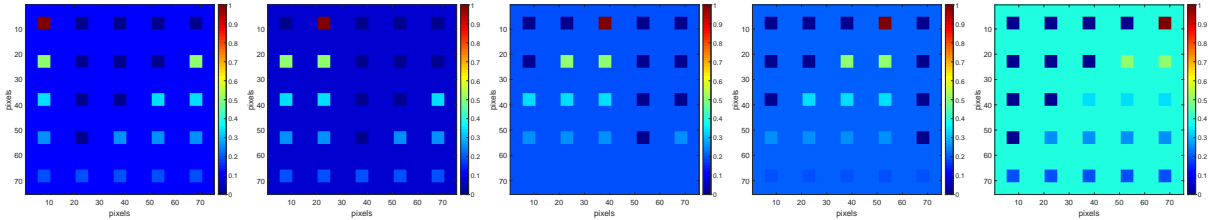

Fig. 5: True abundance maps of selected endmembers for Example 2. From left to right: Endmembers #1–#5.
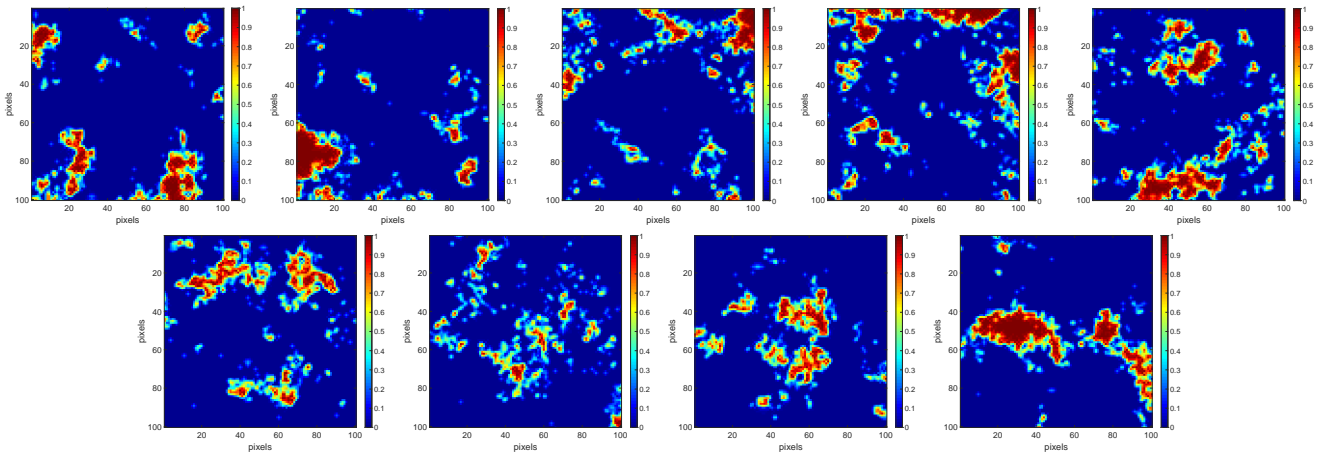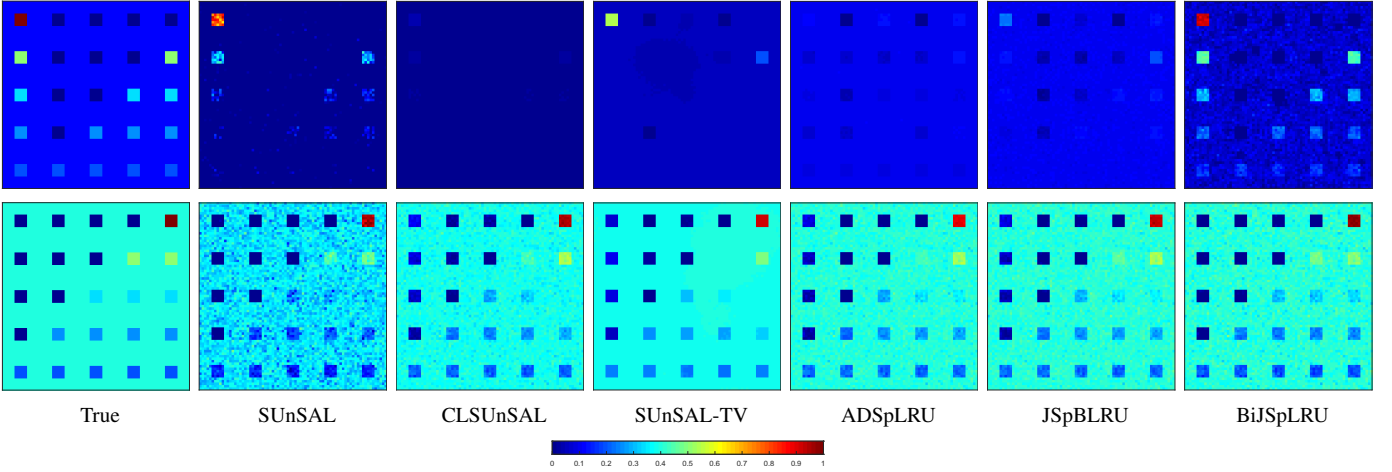

Fig. 6: True abundance maps for selected endmembers for Example 3. Top row: endmembers #1–#5. Bottom row: endmembers #6–#9.

TABLE I: Average of SRE (dB) and RMSE after 20 runs by different unmixing algorithms for Example 1 with SNR = 30 dB.

Criteria	SUnSAL	CLSUnSAL	SUnSAL-TV	ADSpLRU	JSpBLRU	BiJSpLRU
SRE	6.58	7.70	8.45	19.11	19.94	**21.17**
RMSE	0.0205	0.0180	0.0165	0.0048	0.0044	**0.0038**

Fig. 7: True and Estimated abundance maps by different unmixing algorithms for Example 2 with SNR = 30 dB for endmembers (top row) #1 and (bottom row) #5.

When both termination conditions

$$\|\boldsymbol{r}^k\|_F \leq \zeta, \quad \|\boldsymbol{s}^k\|_F \leq \zeta \tag{46}$$

are satisfied or the number of iterations reaches 300, the iterations are stopped. Here we set $\zeta = \sqrt{(3m+l)n}\zeta^{\mathrm{rel}}$ as in [26], where $\zeta^{\mathrm{rel}} = 5 \times 10^{-6}$.

B. Comparison With Different Algorithms

In the following, we show the performance of BiJSpLRU and other unmixing algorithms on three simulated examples.

Example 1: In this experiment, we consider a toy problem to highlight the significance of the proposed approach, i.e., the bilateral joint-sparse structure. We use the spectral library $\boldsymbol{A}_1$ and randomly select five signatures from $\boldsymbol{A}_1$ to generate a 15×15-pixel data cube according to LMM. There are five abundance maps, two of which are shown in the first column in Fig. 4 and the rest three are randomly generated with the ASC. Note that the true abundance maps in Fig. 4 show a clear spatial characteristic along vertical or horizontal directions. The generated data cube is contaminated by white Gaussian i.i.d. noise with signal-to-noise ratio (SNR) of 30 dB.

Example 2: This example shows the performance of BiJS-pLRU on a widely used simulation data set, which has been used in [26], [28], [34]. The spectral library $\boldsymbol{A}_1$ is used. Five signatures are randomly chosen from $\boldsymbol{A}_1$ and then used to generate a 75×75-pixel data cube according to LMM. Fig. 5 shows corresponding true abundance maps of the five selected endmembers. The generated data cube is contaminated by white Gaussian i.i.d. noise with SNR = 20, 30, and 40 dB, respectively.

Example 3: We use the spectral library $\boldsymbol{A}_2$ in this example. The data set contains 100×100 pixels and is generated according to the LMM by nine randomly selected spectral signatures from $\boldsymbol{A}_2$. The corresponding true abundance maps are shown in Fig. 6. Then, the true data cube is contaminated by white Gaussian i.i.d. noise with the same SNR values adopted for Example 2.

Results and Discussion: Fig. 4 shows the estimated abundance maps by different unmixing algorithms for Example 1.

TABLE II: SRE (dB) and RMSE by different unmixing algorithms for Examples 2 and 3.

Example 2						
Algorithm	SNR = 20 dB		SNR = 30 dB		SNR = 40 dB	
	SRE	RMSE	SRE	RMSE	SRE	RMSE
SUnSAL	2.21	0.0268	5.61	0.0181	11.16	0.0096
CLSUnSAL	6.26	0.0168	9.88	0.0111	16.94	0.0049
SUnSAL-TV	8.76	0.0126	12.71	0.0080	22.92	0.0025
ADSpLRU	6.51	0.0163	14.96	0.0062	32.45	0.0008
JSpBLRU	9.40	0.0117	15.99	0.0055	32.92	0.0008
BiJSpLRU	**9.44**	**0.0116**	**17.24**	**0.0047**	**33.29**	**0.0007**
Example 3						
Algorithm	SNR = 20 dB		SNR = 30 dB		SNR = 40 dB	
	SRE	RMSE	SRE	RMSE	SRE	RMSE
SUnSAL	3.37	0.0492	8.22	0.0281	13.51	0.0153
CLSUnSAL	2.39	0.0550	5.63	0.0379	11.07	0.0203
SUnSAL-TV	8.21	0.0281	13.43	0.0154	19.75	0.0075
ADSpLRU	7.34	0.0311	17.00	0.0102	26.40	0.0035
JSpBLRU	9.98	0.0230	18.82	0.0083	28.44	0.0027
BiJSpLRU	**12.12**	**0.0180**	**20.43**	**0.0069**	**29.70**	**0.0024**

From this figure, we see that sparse unmixing algorithms SUnSAL and CLSUnSAL maintain the overall vertical and horizontal spatial information, but with many low abundance values supposing to be zero. By exploiting spatial information, SUnSAL-TV provides better estimations. For low-rank unmixing algorithms, JSpBLRU utilizes the vertical spatial information so it provides more accurate abundance estimates for endmember #2 compared with ADSpLRU. For endmember #1, however, the estimated abundance map of JSpBLRU also contains many low abundance values supposing to be zero. By simultaneously exploiting vertical and horizontal spatial information, the proposed BiJSpLRU algorithm provides accurate estimations for both endmembers #1 and #2. For further comparison, Table I shows the average SREs and RMSEs after 20 runs of different algorithms for Example 1. It is clear that BiJSpLRU gives higher SRE and lower RMSE values, which is consistent with the visual observations from Fig. 4.

Fig. 7 shows the estimated abundances maps for endmembers #1 and #5 obtained by different unmixing algorithms for Example 2. Abundance maps for other endmembers show
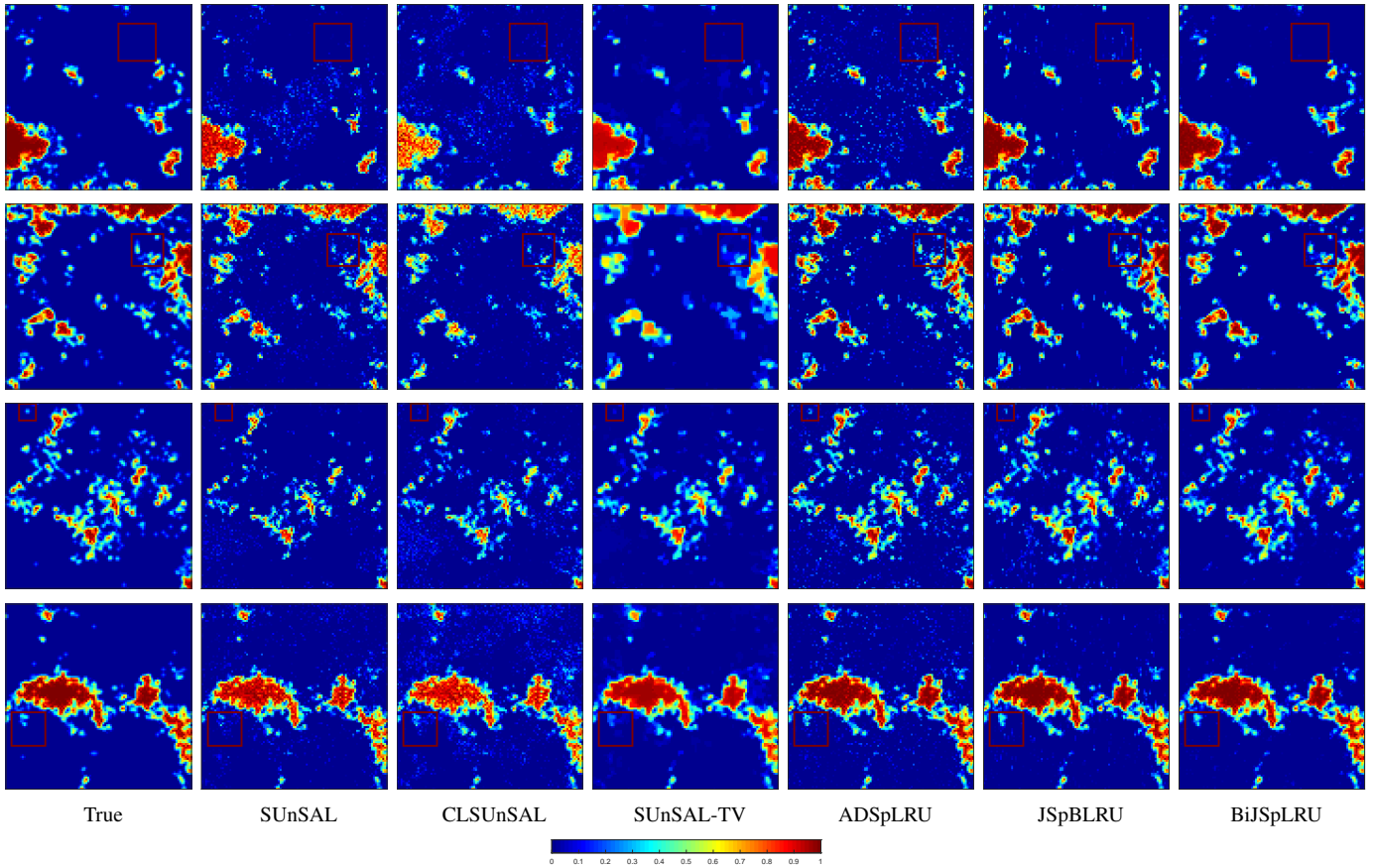
Fig. 8: True and estimated abundance maps by different unmixing algorithms for Example 3 with SNR = 30 dB for (from top to bottom) endmembers #2, #4, #7, and #9.
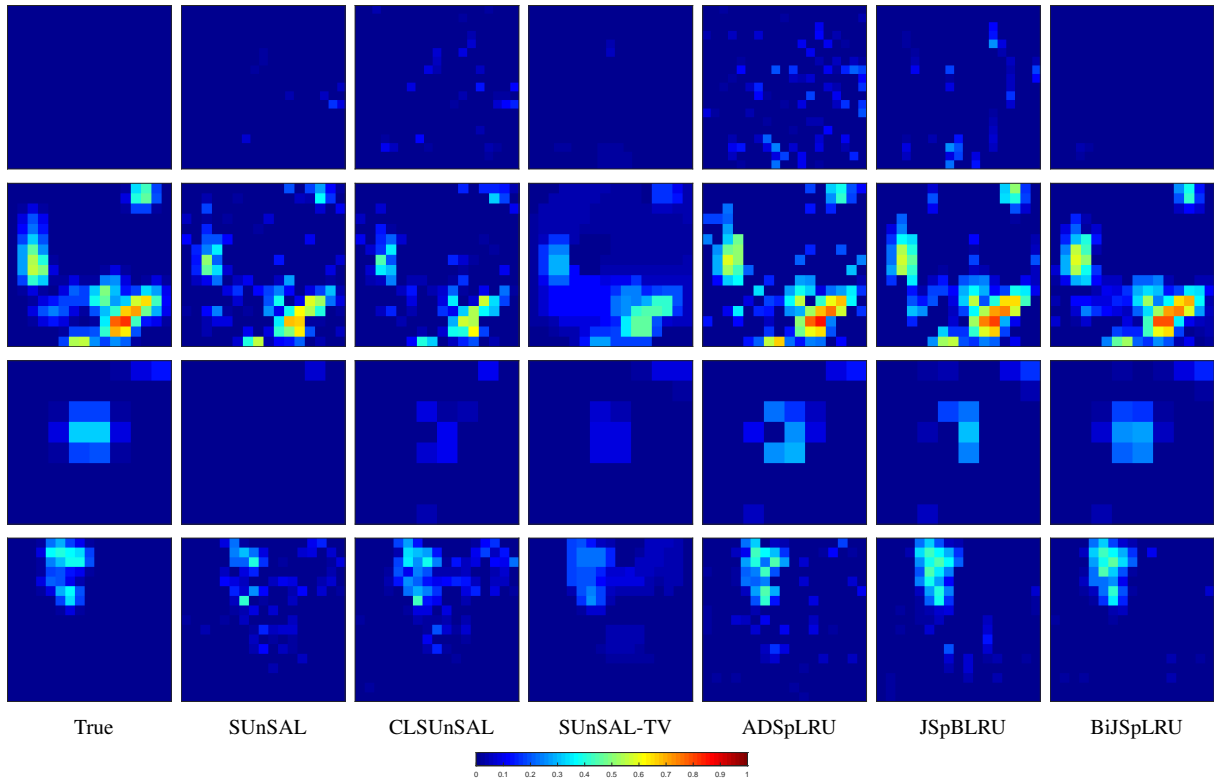

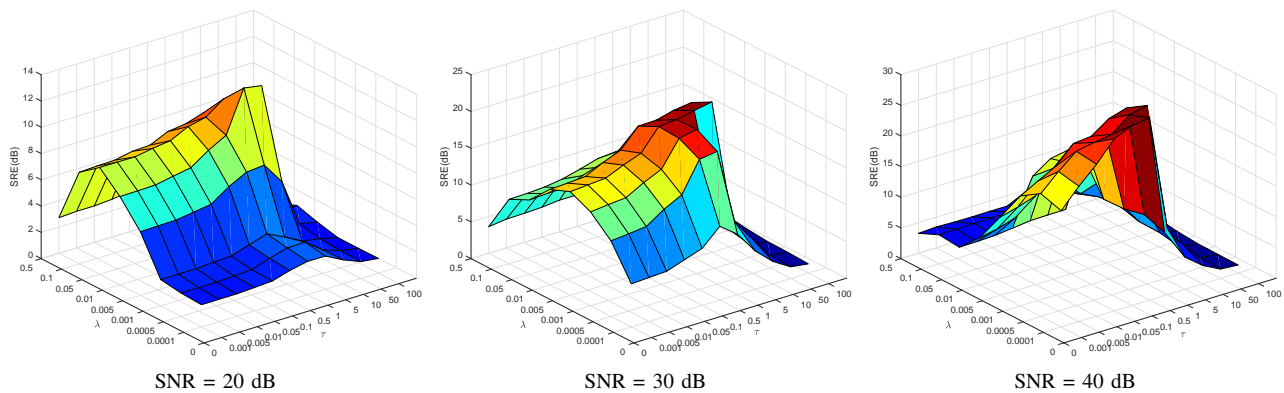Fig. 9: Detailed regions in red boxes from Fig. 8 for (from top to bottom) endmembers #2, #4, #7, and #9.

Fig. 10: SRE (dB) as a function of regularization parameters λ and τ in BiJSpLRU for Example 3.
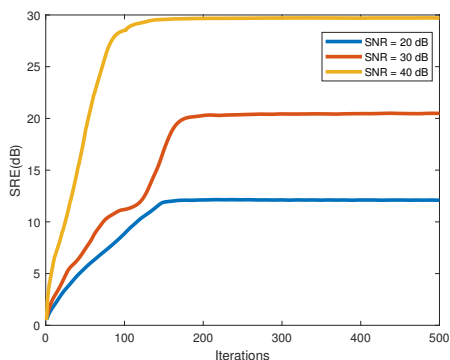

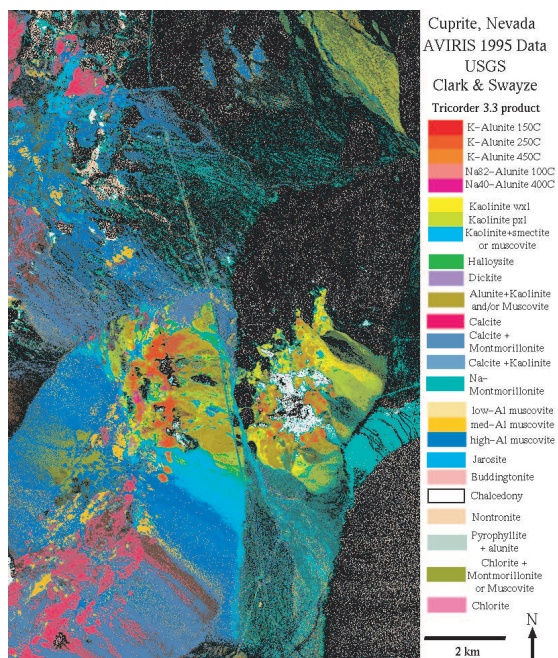Fig. 11: Plot of SRE (dB) against iteration by BiJSpLRU for Example 3.

Fig. 12: The USGS map showing the location of different minerals in the Cuprite mining district in Nevada.

a similar conclusion, so we omit here for space considerations. Compared with SUnSAL and CLSUnSAL, SUnSAL-TV maintains the piecewise smooth behavior in the backgrounds, especially for endmember #5. But the abundance map of SUnSAL-TV is over-smooth for endmember #1. Low-rank unmixing algorithms delineate more square regions for endmember #1. Among them, BiJSpLRU produces most square blocks. It can be expected since BiJSpLRU utilizes more spatial information than ADSpLRU and JSpBLRU, especially for pixels in the boundaries.

The abundance maps estimated by different algorithms for Example 3 are shown in Fig. 8. Particularly, the regions in red boxes are zoomed in and displayed in Fig. 9. SUnSAL and CLSUnSAL estimate abundance maps with low accuracy. Note that each of the two algorithms has only one regularization parameter so it is time-saving to select optimal parameter values compared with other algorithms. In addition, they are fast since there is no need to implement the TV or SVD at each iteration. SUnSAL-TV provides piecewise smooth estimations, in which many details are lost. Clearly, the backgrounds of the abundance maps estimated by JSpBLRU are more clear than those by ADSpLRU. However, when we zoom in the abundance maps of JSpBLRU, we can see much banded vertical noise in the backgrounds. The reason is, in part, that JSpBLRU utilizes the local joint-sparsity structure along the vertical direction of the HSI. BiJSpLRU addresses this problem by utilizing both vertical and horizontal spatial information. We can see that BiJSpLRU provides accurate abundance maps with clearer backgrounds. Finally, Table II shows the SRE and RMSE values by different unmixing algorithms for Examples 2 and 3. We see that BiJSpLRU provides the optimal results, which is consistent with the observations from Figs. 7-9.

C. Parameters Discussion

Now we discuss the influence of the regularization parameters λ and τ and the number of iterations of BiJSpLRU. Recall that in the simulated experiments, we test all possible combinations of parameters for BiJSpLRU and choose the optimal ones for best SREs. Fig. 10 plots the SRE values as a function of parameters λ and τ in the model (18) for Example 3. We can see that both optimal parameters decrease
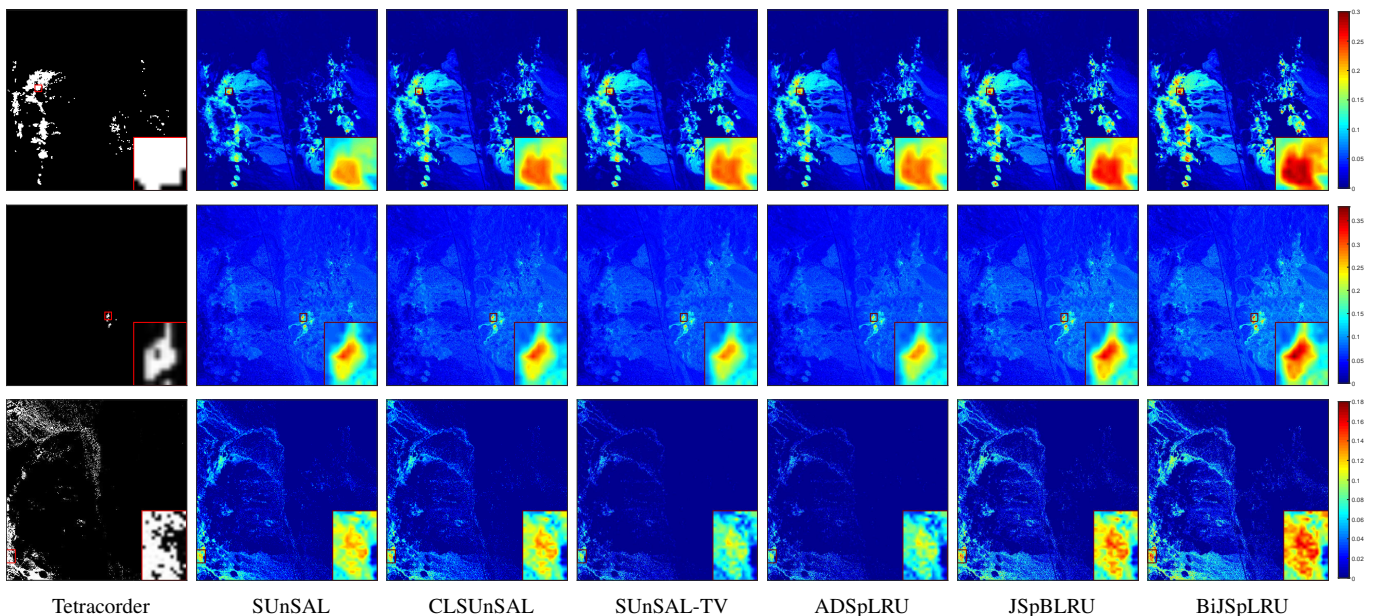
Fig. 13: Estimated abundance maps for (top row) Alunite, (middle row) Buddingtonite, and (bottom row) Muscovite by different unmixing algorithms. The demarcated area is enlarged in the right bottom corner for better visualization.

as SNR increases. Optimal τ is far greater than optimal λ, which shows the effectiveness of the low-rank regularizer.

Fig. 11 plots the SRE values against iteration for Example 3. Clearly for BiJSpLRU, it is enough to set the maximum number of iterations to be 300. In addition, we see from Fig. 11 that for different noise levels, the convergence behavior of BiJSpLRU becomes stable as the iteration increases.

IV. EXPERIMENTS ON REAL DATA

In this test, we show the performance of BiJSpLRU on the well-known Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Cuprite data set. Fig. 12 shows a mineral map produced in 1995 by USGS, in which different minerals were mapped by a Tetracorder software product [55]. We use a subscene of the Cuprite data with 350×350 pixels and 188 spectral bands. The spectral dictionary A of size 188×240 is generated from the USGS spectral library. Similarly as in [17], [18], [28], we set regularization parameters $\lambda = 0.001$ for SUnSAL, $\lambda = 0.01$ for CLSUnSAL, and $\lambda = \lambda_{TV} = 0.001$ for SUnSAL-TV. Similarly as in [26] and [32], we set $\lambda = \tau = 0.001$ for ADSpLRU, JSpBLRU, and BiJSpLRU.

Fig. 13 shows the estimated abundance maps obtained by different unmixing algorithms. Due to the low noise of the real data, all the algorithms produce overall similar results. From the enlarged area in the right bottom corner in Fig. 13, however, we can see that BiJSpLRU gives comparable or higher estimates in the regions considered as respective materials. It shows again the effectiveness of BiJSpLRU by simultaneously utilizing the bilateral joint sparsity structure and the low rankness of the abundance matrix.

V. CONCLUSION

In this paper, we have proposed a bilateral joint-sparse structure to utilize the vertical and horizontal spatial information of HSIs. In particular, we realize the bilateral joint-sparse structure by introducing a permutation matrix, which is no need to be constructed explicitly. Then we have proposed an unmixing model by imposing the bilateral joint-sparse structure and the low-rank property on the abundance matrix, via the weighted $\ell_{2,1}$-norm and the weighted nuclear norm, respectively. We solve the proposed model under the ADMM framework with an iterative reweighting strategy, leading to an algorithm called BiJSpLRU. The residual and objective convergence results of BiJSpLRU have been provided. The simulated and real-data experiments show the effectiveness of the proposed algorithm, compared with other state-of-the-art unmixing algorithms.

The proposed bilateral joint-sparse structure has the potential to contribute other image processing problems via sparse representation, such as (hyperspectral) image restoration, super-resolution, blind hyperspectral unmixing, etc. In addition, the article provides a way to obtain theoretical convergence results of algorithms which are similarly based on the ADMM framework but with a reweighted strategy.

APPENDIX

In order to prove Theorem 1 about the convergence of the sequence $(X^k, V^k, W^k, \Lambda^k)$ by (26), we first give three lemmas.

Lemma 1. *Let $(X^k, V^k, W^k, \Lambda^k)$ be the sequence generated by* (26). *Then under Assumption 1, we have*

$$p^\star - p^{k+1} \leq \langle \Lambda^\star, r^{k+1} \rangle. \tag{47}$$

Proof. Recall from (43), we have

$$\mathcal{L}_0(X^\star, V^\star, W^\star, \Lambda^\star) \leq \mathcal{L}_0(X^{k+1}, V^{k+1}, W^{k+1}, \Lambda^\star). \tag{48}$$

That is

$$f(X^\star) + g(V^\star, W^\star) + \langle \Lambda^\star, X^\star G - V^\star \rangle$$
$$\leq f(X^{k+1}) + g(V^{k+1}, W^{k+1}) + \langle \Lambda^\star, X^{k+1}G - V^{k+1} \rangle. \quad (49)$$

Notice that $X^\star G - V^\star = 0$, then $p^\star \leq p^{k+1} + \langle \Lambda^\star, r^{k+1} \rangle$. $\quad\square$

Lemma 2. *Under the assumptions as in Lemma 1, if $W^{k+1} \leq W^\star$, then*

$$p^{k+1} - p^\star \leq -\langle \Lambda^{k+1}, r^{k+1} \rangle$$
$$+ \mu \langle V^{k+1} - V^k, -r^{k+1} - (V^{k+1} - V^\star) \rangle. \quad (50)$$

Proof. From (26), we see that X^{k+1} minimizes $\mathcal{L}(X, V^k, W^k, \Lambda^k)$, or equivalently miminizes

$$f(X) + \langle \Lambda^k, XG - V^k \rangle + \frac{\mu}{2} \|XG - V^k\|_F^2. \quad (51)$$

Since f is differentiable, the optimality condition is

$$0 = \nabla f(X^{k+1}) + \Lambda^k G^T + \mu (X^{k+1}G - V^k)G^T. \quad (52)$$

Recall that $\Lambda^{k+1} = \Lambda^k + \mu(X^{k+1}G - V^{k+1})$, then we have

$$0 = \nabla f(X^{k+1}) + (\Lambda^k + \mu(X^{k+1}G - V^{k+1}))G^T$$
$$+ \mu(V^{k+1} - V^k)G^T \quad (53)$$
$$= \nabla f(X^{k+1}) + (\Lambda^{k+1} + \mu(V^{k+1} - V^k))G^T$$

which implies that X^{k+1} minimizes

$$f(X) + \langle \Lambda^{k+1} + \mu(V^{k+1} - V^k), XG \rangle. \quad (54)$$

Thus we have

$$f(X^{k+1}) + \langle \Lambda^{k+1} + \mu(V^{k+1} - V^k), X^{k+1}G \rangle$$
$$\leq f(X^\star) + \langle \Lambda^{k+1} + \mu(V^{k+1} - V^k), X^\star G \rangle. \quad (55)$$

Similarly, V^{k+1} minimizes $\mathcal{L}(X^{k+1}, V, W^{k+1}, \Lambda^k)$. Let

$$h(V) = g(V, W^{k+1}), \quad (56)$$

then V^{k+1} equivalently minimizes

$$h(V) + \langle \Lambda^k, X^{k+1}G - V \rangle + \frac{\mu}{2} \|X^{k+1}G - V\|_F^2. \quad (57)$$

We remark that the minimization problem about V_3 can be transformed into a convex minimization problem about its singular values; see [48] for more details. Since h is subdifferentiable, the optimality condition for (57) is

$$0 \in \partial h(V^{k+1}) - \Lambda^k + \mu(V^{k+1} - X^{k+1}G) = \partial h(V^{k+1}) - \Lambda^{k+1}. \quad (58)$$

It implies that V^{k+1} minimizes

$$h(V) - \langle \Lambda^{k+1}, V \rangle = g(V, W^{k+1}) - \langle \Lambda^{k+1}, V \rangle. \quad (59)$$

It follows that

$$g(V^{k+1}, W^{k+1}) - \langle \Lambda^{k+1}, V^{k+1} \rangle \leq g(V^\star, W^{k+1}) - \langle \Lambda^{k+1}, V^\star \rangle. \quad (60)$$

Under the assumption of $W^{k+1} \leq W^\star$, then from the definition of g, we have $g(V^\star, W^{k+1}) \leq g(V^\star, W^\star)$ and

$$g(V^{k+1}, W^{k+1}) - \langle \Lambda^{k+1}, V^{k+1} \rangle \leq g(V^\star, W^\star) - \langle \Lambda^{k+1}, V^\star \rangle. \quad (61)$$

Now together with (55) and (61), we get

$$f(X^{k+1}) + \langle \Lambda^{k+1} + \mu(V^{k+1} - V^k), X^{k+1}G \rangle$$
$$+ g(V^{k+1}, W^{k+1}) - \langle \Lambda^{k+1}, V^{k+1} \rangle$$
$$\leq f(X^\star) + \langle \Lambda^{k+1} + \mu(V^{k+1} - V^k), X^\star G \rangle \quad (62)$$
$$+ g(V^\star, W^\star) - \langle \Lambda^{k+1}, V^\star \rangle.$$

That is,

$$p^{k+1} + \langle \Lambda^{k+1}, r^{k+1} \rangle + \mu \langle V^{k+1} - V^k, X^{k+1}G \rangle$$
$$\leq p^\star + \mu \langle V^{k+1} - V^k, X^\star G \rangle \quad (63)$$

following that

$$p^{k+1} - p^\star$$
$$\leq -\langle \Lambda^{k+1}, r^{k+1} \rangle + \mu \langle V^{k+1} - V^k, (X^\star - X^{k+1})G \rangle,$$
$$= -\langle \Lambda^{k+1}, r^{k+1} \rangle$$
$$+ \mu \langle V^{k+1} - V^k, V^\star - V^{k+1} - (X^{k+1}G - V^{k+1}) \rangle,$$
$$= -\langle \Lambda^{k+1}, r^{k+1} \rangle + \mu \langle V^{k+1} - V^k, -r^{k+1} - (V^{k+1} - V^\star) \rangle \quad (64)$$

which conclude the proof. $\quad\square$

Lemma 3. *Let $(X^k, V^k, W^k, \Lambda^k)$ be the sequence generated by (26). Let*

$$t^k = \frac{1}{\mu} \|\Lambda^k - \Lambda^\star\|_F^2 + \mu \|V^k - V^\star\|_F^2 \quad (65)$$

then under the assumptions as in Lemmas 1 and 2, we have

$$t^k - t^{k+1} \geq \mu \|r^{k+1} + (V^{k+1} - V^k)\|_F^2. \quad (66)$$

Proof. Together with (50) and (47), we obtain that

$$0 \leq \langle -\Lambda^{k+1} + \Lambda^\star, r^{k+1} \rangle + \mu \langle V^{k+1} - V^k, -r^{k+1} - (V^{k+1} - V^\star) \rangle \quad (67)$$

which implies that

$$2\langle \Lambda^{k+1} - \Lambda^\star, r^{k+1} \rangle + 2\mu \langle V^{k+1} - V^k, r^{k+1} \rangle$$
$$+ 2\mu \langle V^{k+1} - V^k, V^{k+1} - V^\star \rangle \leq 0. \quad (68)$$

Let

$$c_1 = 2\langle \Lambda^{k+1} - \Lambda^\star, r^{k+1} \rangle$$
$$c_2 = 2\mu \langle V^{k+1} - V^k, r^{k+1} \rangle + 2\mu \langle V^{k+1} - V^k, V^{k+1} - V^\star \rangle. \quad (69)$$

Then (68) gives that $c_1 + c_2 \leq 0$.

We first simplify c_1. Recall that $\Lambda^{k+1} = \Lambda^k + \mu r^{k+1}$ and $r^{k+1} = \frac{1}{\mu}(\Lambda^{k+1} - \Lambda^k)$, then we get

$$c_1 = 2\langle \Lambda^{k+1} - \Lambda^\star, r^{k+1} \rangle = 2\langle \Lambda^k + \mu r^{k+1} - \Lambda^\star, r^{k+1} \rangle$$
$$= 2\langle \Lambda^k - \Lambda^\star, r^{k+1} \rangle + \mu \|r^{k+1}\|_F^2 + \mu \|r^{k+1}\|_F^2$$
$$= 2\langle \Lambda^k - \Lambda^\star, \frac{1}{\mu}(\Lambda^{k+1} - \Lambda^k) \rangle + \frac{1}{\mu} \|\Lambda^{k+1} - \Lambda^k\|_F^2$$
$$+ \mu \|r^{k+1}\|_F^2 \quad (70)$$
$$= \frac{1}{\mu} \left(2\langle \Lambda^k - \Lambda^\star, \Lambda^{k+1} - \Lambda^k \rangle + \|\Lambda^{k+1} - \Lambda^k\|_F^2 \right)$$
$$+ \mu \|r^{k+1}\|_F^2$$
$$= \frac{1}{\mu} \left(\|\Lambda^{k+1} - \Lambda^\star\|_F^2 - \|\Lambda^k - \Lambda^\star\|_F^2 \right) + \mu \|r^{k+1}\|_F^2.$$

We now rewrite $\mu\|\boldsymbol{r}^{k+1}\|_F^2 + c_2$, i.e.,

$$
\begin{aligned}
&\mu\|\boldsymbol{r}^{k+1}\|_F^2 + c_2 \\
&= \mu\|\boldsymbol{r}^{k+1}\|_F^2 + 2\mu\langle \boldsymbol{V}^{k+1} - \boldsymbol{V}^k, \boldsymbol{r}^{k+1}\rangle \\
&\quad + 2\mu\langle \boldsymbol{V}^{k+1} - \boldsymbol{V}^k, \boldsymbol{V}^{k+1} - \boldsymbol{V}^\star\rangle \\
&= \mu\|\boldsymbol{r}^{k+1}\|_F^2 + 2\mu\langle \boldsymbol{V}^{k+1} - \boldsymbol{V}^k, \boldsymbol{r}^{k+1}\rangle \\
&\quad + 2\mu\langle \boldsymbol{V}^{k+1} - \boldsymbol{V}^k, \boldsymbol{V}^{k+1} - \boldsymbol{V}^k + \boldsymbol{V}^k - \boldsymbol{V}^\star\rangle \quad (71) \\
&= \mu\|\boldsymbol{r}^{k+1} + (\boldsymbol{V}^{k+1} - \boldsymbol{V}^k)\|_F^2 + \mu\|\boldsymbol{V}^{k+1} - \boldsymbol{V}^k\|_F^2 \\
&\quad + 2\mu\langle \boldsymbol{V}^{k+1} - \boldsymbol{V}^k, \boldsymbol{V}^k - \boldsymbol{V}^\star\rangle \\
&= \mu\|\boldsymbol{r}^{k+1} + (\boldsymbol{V}^{k+1} - \boldsymbol{V}^k)\|_F^2 \\
&\quad + \mu\left(\|\boldsymbol{V}^{k+1} - \boldsymbol{V}^\star\|_F^2 - \|\boldsymbol{V}^k - \boldsymbol{V}^\star\|_F^2\right).
\end{aligned}
$$

Together with (70) and (71), we have

$$
\begin{aligned}
&c_1 + c_2 \\
&= \frac{1}{\mu}\left(\|\boldsymbol{\Lambda}^{k+1} - \boldsymbol{\Lambda}^\star\|_F^2 - \|\boldsymbol{\Lambda}^k - \boldsymbol{\Lambda}^\star\|_F^2\right) \\
&\quad + \mu\left(\|\boldsymbol{V}^{k+1} - \boldsymbol{V}^\star\|_F^2 - \|\boldsymbol{V}^k - \boldsymbol{V}^\star\|_F^2\right) \quad (72) \\
&\quad + \mu\|\boldsymbol{r}^{k+1} + (\boldsymbol{V}^{k+1} - \boldsymbol{V}^k)\|_F^2 \\
&= t^{k+1} - t^k + \mu\|\boldsymbol{r}^{k+1} + (\boldsymbol{V}^{k+1} - \boldsymbol{V}^k)\|_F^2 \le 0.
\end{aligned}
$$

$\square$

We now obtain the proof of Theorem 1.

Proof of Theorem 1. (1) Lemma 3 says that $\{t^k\}$ in (65) is monotonically decreasing. As $\{t^k\}$ is bounded from below, $\{t^k\}$ converges, which implies that $\{\boldsymbol{\Lambda}^k\}$ and $\{\boldsymbol{V}^k\}$ converge. Iterating the inequality in (66) gives that

$$
t^0 \ge \mu \sum_{k=0}^{\infty} \|\boldsymbol{r}^{k+1} + (\boldsymbol{V}^{k+1} - \boldsymbol{V}^k)\|_F^2 \quad (73)
$$

which implies that $\boldsymbol{r}^{k+1} + (\boldsymbol{V}^{k+1} - \boldsymbol{V}^k) \to \boldsymbol{0}$, as $k \to \infty$. Since $\{\boldsymbol{V}^k\}$ converges, then we have $\boldsymbol{r}^k \to \boldsymbol{0}$ as $k \to \infty$. It follows that $\{\boldsymbol{X}^k\}$ converges.

(2) Recall from (1) that $\{\boldsymbol{\Lambda}^k\}$ and $\{\boldsymbol{V}^k\}$ converge and $\boldsymbol{r}^k \to \boldsymbol{0}$, then Lemmas 1 and 2 give $p^k \to p^\star$, as $k \to \infty$. $\square$

REFERENCES

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

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

[3] W.-K Ma, J. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A. 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, 2014.

[4] N. Dobigeon, J-Y Tourneret, C. Richard, J. C. M. Bermudez, S. McLaughlin, and A. O. Hero, "Nonlinear unmixing of hyperspectral images: Models and algorithms," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 82–94, 2014.

[5] R. R. Close, P. D. Gader, and J. Wilson, "Hyperspectral unmixing using macroscopic and microscopic mixture models," *J. Appl. Remote Sens.*, vol. 8, no. 1, pp. 1–16, 2014.

[6] Q. Qu, N. Nasrabadi, and T. 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, 2014.

[7] W. Luo, L. Gao, R. Zhang, A. Marinoni, and B. Zhang, "Bilinear normal mixing model for spectral unmixing," *IET Image Process.*, vol. 13, no. 2, pp. 344–354, 2019.

[8] Y. Su, X. Xu, J. Li, H. Qi, P. Gamba, and A. Plaza, "Deep autoencoders with multitask learning for bilinear hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–15, 2020.

[9] D. 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, 2001.

[10] Y. Qian, S. Jia, J. Zhou, and A. Robles-Kelly, "Hyperspectral unmixing via $L_{1/2}$ sparsity-constrained nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4282–4297, 2011.

[11] 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, 2011.

[12] F. Zhu and P. Honeine, "Biobjective nonnegative matrix factorization: Linear versus kernel-based models," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 7, pp. 4012–4022, 2016.

[13] 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, 2017.

[14] Y. Qian, F. Xiong, S. Zeng, J. Zhou, and Y. Y. Tang, "Matrix-vector nonnegative tensor factorization for blind unmixing for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 3, pp. 1776–1792, 2017.

[15] 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, 2019.

[16] B. Feng and J. Wang, "Constrained nonnegative tensor factorization for spectral unmixing of hyperspectral images: A case study of urban impervious surface extraction," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 4, pp. 583–587, 2019.

[17] 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, 2011.

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

[19] J. Sigurdsson, M. O. Ulfarsson, and J. R. Sveinsson, "Hyperspectral unmixing with ℓ_q regularization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 6793–6806, 2014.

[20] Y. Xu, F. Fang, and G. Zhang, "Similarity-guided and ℓ_p-regularized sparse unmixing of hyperspectral data," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2311–2315, 2015.

[21] 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, 2015.

[22] 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, 2016.

[23] S. Zhang, J. Li, Z. Wu, and A. Plaza, "Spatial discontinuity-weighted sparse unmixing of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 10, no. 6, pp. 5767–5779, 2018.

[24] Z. Shi, T. Shi, M. Zhou, and X. Xu, "Collaborative sparse hyperspectral unmixing using ℓ_0 norm," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5495–5508, 2018.

[25] Y. E. Salehani, S. Gazor, and M. Cheriet, "Sparse hyperspectral unmixing via heuristic ℓ_p-norm approach," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 4, pp. 1191–1202, 2018.

[26] 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, 2019.

[27] J.-J. Wang, T.-Z. Huang, J. Huang, H.-X. Dou, L.-J. Deng, and X.-L. Zhao, "Row-sparsity spectral unmixing via total variation," *IEEE*

J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 12, no. 12, pp. 5009–5022, 2019.

[28] 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, 2012.

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

[30] X.-L. Zhao, W. Wang, T.-Y. Zeng, T.-Z. Huang, and M. Ng, "Total variation structured total least squares method for image restoration," *SIAM J. Sci. Comput.*, vol. 35, pp. 1304–1320, 2013.

[31] 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.

[32] P. Giampouras, K. Themelis, A. Rontogiannis, and K. 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, 2016.

[33] R. Wang, W. Liao, H. Li, H. Zhang, and A. Pižurica, "Hyperspectral unmixing by reweighted low rank and total variation," in *Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, Los Angeles, California, USA, Aug. 21-24, 2016.

[34] J. Yang, Y.-Q. Zhao, J. Chan, and S. 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, 2016.

[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, 2014.

[36] R. Feng, Y. Zhong, and L. Zhang, "An improved nonlocal sparse unmixing algorithm for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 4, pp. 915–919, 2015.

[37] R. Feng, Y. Zhong, and L. Zhang, "Adaptive non-local Euclidean medians sparse unmixing for hyperspectral imagery," *ISPRS J. Photogram. Remote Sens.*, vol. 97, pp. 9–24, 2014.

[38] 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, 2017.

[39] L. Sun, F. Wu, T. Zhan, W. Liu, J. Wang, and B. Jeon, "Weighted nonlocal low-rank tensor decomposition method for sparse unmixing of hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 1174–1188, 2020.

[40] J. Huang, T.-Z. Huang, X.-L. Zhao, and L.-J. Deng, "Nonlocal tensor-based sparse hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–15, 2020.

[41] G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[42] H. A. L Kiers, "Towards a standardized notation and terminology in multiway analysis," *J. Chemometr.*, vol. 14, no. 3, pp. 105–122, 2000.

[43] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5-6, pp. 877–905, 2008.

[44] X. Fu, W.-K. Ma, J. Bioucas-Dias, and T.-H. Chan, "Semiblind hyperspectral unmixing in the presence of spectral library mismatches," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5171–5184, 2016.

[45] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 2862–2869.

[46] R. A. Horn and C. R. Johnson, *Matrix Analysis, 2nd Edition*, Cambridge University Press, New York, USA, 2013.

[47] J. Bioucas-Dias and M. A. T. Figueiredo, "Alternating direction algo rithms for constrained sparse regression: Application to hyperspectral unmixing," in *Proc. 2nd Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, pp. 1–4, 2010.

[48] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its applications to low level vision," *Int. J. Comput. Vis.*, vol. 121, no. 2, pp. 183–208, 2017.

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

[50] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, Elsevier, Amsterdam, North-Holland, 1983.

[51] P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM J. Control Optim*, vol. 29, pp. 119–138, 1991.

[52] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, pp. 293–318, 1992.

[53] M. Fukushima, "Application of the alternating direction method of multipliers to separable convex programming problems," *Comput. Optim. Appl.*, vol. 1, pp. 93–111, 1992.

[54] 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, 2011.

[55] R. Clark *et al.*, "Imaging spectroscopy: Earth and planetary remote sensing with the USGS Tetracorder and expert systems," *J. Geophys. Res.*, vol. 108, no. E12, pp. 5131–5135, 2003.