Nonlocal weighted sparse unmixing based on global search and parallel optimization

Yongxin Li,[a,b] Wenxing Bao,[a,b,*] Kewen Qu,[a,b] and Xiangfei Shen[a,b]
[a]North Minzu University, School of Computer Science and Engineering, Yinchuan, China
[b]North Minzu University, The Key Laboratory of Images and Graphics Intelligent Processing of State Ethnic Affairs Commission, Yinchuan, China

Abstract. Sparse unmixing (SU) can represent an observed image using pure spectral signatures and corresponding fractional abundance from a large spectral library and is an important technique in hyperspectral unmixing. However, the existing SU algorithms mainly exploit spatial information from a fixed neighborhood system, which is not sufficient. To solve this problem, we propose a nonlocal weighted SU algorithm based on global search (G-NLWSU). By exploring the nonlocal similarity of the hyperspectral image, the weights of pixels are calculated to form a matrix to weight the abundance matrix. Specifically, G-NLWSU first searches for a similar group of each pixel in the global scope then uses singular value decomposition to denoise and finally obtains the weight matrix by considering correlations between similar pixels. To reduce the execution burden of G-NLWSU, we propose a parallel computing version of G-NLWSU, named PG-NLWSU, which integrates compute unified device architecture-based parallel computing into G-NLWSU to accelerate the search for groups of nonlocally similar pixels. Our proposed algorithms shed new light on SU by considering a new exploitation process of spatial information and parallel computing scenario. Experimental results conducted on simulated and real datasets show that PG-NLWSU is superior to comparison algorithms. [DOI: 10.1117/1.JRS.15.016501]

Keywords: hyperspectral imaging; sparse unmixing; nonlocal information; global search; parallel computing.

Paper 200670 received Sep. 8, 2020; accepted for publication Dec. 23, 2020; published online Jan. 11, 2021.

1 Introduction

The rapid development of hyperspectral remote sensing technology has resulted in its wide use, such as in geological prospecting, target detection, and environmental surveillance.[1] However, owing to the low spatial resolution of the sensor, the spatial complexity will lead to the existence of mixed pixels in a hyperspectral image (HSI), i.e., different substances form a pixel.[2] Linear spectral unmixing, also known as a linear mixing model (LMM), is a standard method to solve the problem of mixed pixels, which assumes that the observed spectral signal can be expressed linearly through a set of pure spectral signatures and their corresponding abundance weights.[3]

LMMs can be categorized as either unsupervised- or semisupervised-based unmixing algorithms. Unsupervised-based unmixing methods require endmember extraction and abundance estimation steps,[4,5] which can be classified as geometric-,[6–18] statistic-,[19–24] or spatial-information-based.[25–29] Geometric-based algorithms usually assume that desired pure pixels exist in an image, but they usually fail if the image mixing degree is high. Statistic-based algorithms can elegantly solve problems of a high mixing degree, but they only consider spectral information and ignore correlations between pixels. Spatial-information-based algorithms fuse spatial and spectral information for endmember extraction and have attracted huge attention.

Compared with the unsupervised methods, semisupervised-based unmixing algorithms, also known as sparse unmixing (SU) algorithms, solve the spectral unmixing problem by introducing a large public spectral library. It assumes that the observed spectral signal can be represented by a

*Address all correspondence to Wenxing Bao, baowenxing@nun.edu.cn

linear combination of a few spectral signatures in the spectral library. Therefore, the SU problem is equivalent to searching for an optimal subset from the spectral library to simulate mixed pixels in the scene.[30] However, the number of spectra in the library is much larger than the number of endmembers, implying that an abundance matrix usually has only a few nonzero entries, i.e., it is sparse. Bioucas-Dias and Figueiredo[31] proposed a variable splitting and augmented Lagrangian-based sparse unmixing algorithm (SUnSAL), which uses an alternating direction multiplier method (ADMM) to optimize this constrained sparse regression problem. To enhance the sparsity of a solution, Iordache et al.[32] proposed a new SUnSAL extension by considering collaborative sparsity, i.e., the $\ell_{2,1}$ norm, to simultaneously boost row sparseness of the abundance. Sun et al.[33] proposed an $\ell_p(0 < p < 1)$ norm to be used as a new sparsity regularizer. Cheng et al.[34] introduced weighted ℓ_1 norm minimization to penalize nonzero sparse terms in sparse solutions.

HSI not only contains a large amount of spectral information but also has rich spatial information. Iordache et al.[35] proposed an SU algorithm with variable splitting augmentation Lagrangian and total variation (SUnSAL-TV), where the total variation (TV) regularizer explores spatial neighborhood information. When the TV regularizer is only used, it is called nonnegative least squares TV (NCLS-TV). Zhang et al.[36] proposed a local collaborative SU that combines neighborhood weights with cooperative sparse regression to achieve better unmixing results than SUnSAL-TV. Based on the weighted ℓ_1 strategy, Wang et al.[37,38] proposed a double weighted SU algorithm to enhance the sparsity of abundances from the spectral and spatial domains, and a variant with TV regularization. Wang et al.[39] introduced graph regularization to improve unmixing performance. Zhang et al.[40] proposed a spectraspatial weighted SU model using the neighborhood information of the image in a weight matrix. Zhong et al.[41] proposed a nonlocal TV regularizer for sparse unmixing (NLSU). Feng et al.[42] improved NLSU and extracted fixed spatial information from the original image.

However, the unmixing algorithm that combines spatial information still has two problems: (1) the global spatial information is not considered. Local and nonlocal-based methods find similar pixels from the neighborhood and large local blocks, respectively; (2) the spatial regularization term makes the model more complicated. In this paper, we propose a nonlocal weighted SU model, PG-NLWSU, based on global search and parallel acceleration. Inspired by a nonlocal mean algorithm, PG-NLWSU first searches similar blocks of all pixels from the whole image and uses singular value decomposition (SVD) to denoise them. The explored nonlocal information is used to weight an abundance matrix with the ℓ_1-norm to generate a solution with stronger sparsity. To search similar blocks is generally time consuming. PG-NLWSU uses parallel computing to enhance its performance. We make three primary contributions:

1. We propose a global search strategy to exploit nonlocal similar blocks from whole images.
2. The exploited nonlocal information is used as a weight matrix to penalize the abundance matrix with the ℓ_1 norm.
3. Parallel computing is applied to spatial information extraction to improve the algorithm's efficiency.

Several experiments conducted on synthetic and real hyperspectral datasets indicate that PG-NLWSU significantly improves upon current SU algorithms.

The rest of this article is organized as follows. Section 2 describes SU theory. Section 3 discusses the PG-NLWSU algorithm. Section 4 details experiments on simulated and real datasets. Section 5 discusses the advantages and disadvantages of the algorithm and relates directions of future research.

2 Sparse Unmixing

2.1 *LMM*

The LMM assumes that the mixed pixels in an HSI can be expressed as a linear combination of the endmember spectrum and their corresponding abundances, given as

$$\mathbf{y} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n}, \tag{1}$$

where $\mathbf{y} \in \Re^{L\times 1}$, $\mathbf{M} \in \Re^{L\times q}$, $\boldsymbol{\alpha} \in \Re^{q\times 1}$, and $\mathbf{n} \in \Re^{L\times 1}$ denote the observed mixed pixel spectrum, endmember spectral vectors, abundance vector, and error term, respectively. $\boldsymbol{\alpha}$ generally has two constraints: abundance non-negative constraint (ANC), i.e., $\boldsymbol{\alpha} \geq 0$, and the abundance sum-to-one constraint (ASC), i.e., $\mathbf{1}^T\boldsymbol{\alpha} = 1$.

2.2 *Sparse Unmixing*

SU introduces a large spectral library to the LMM, given by

$$\mathbf{Y} = \mathbf{AX} + \mathbf{N}, \tag{2}$$

where $\mathbf{Y} \in \Re^{L\times p}$ is an HSI with L spectral bands and P pixels, $\mathbf{A} \in \Re^{L\times M}$ is a spectral library with M endmembers, $\mathbf{X} \in \Re^{M\times p}$ is the fractional abundance matrix corresponding to all endmembers in the spectral library, and $\mathbf{N} \in \Re^{L\times M}$ is the error that affects each pixel. Since an HSI is usually composed of only a small number of endmember spectral, the fractional abundance matrix $\mathbf{X}$ is sparse, which can minimize an SU model and regularize the abundance matrix $\mathbf{X}$ with a sparsity regularizer

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{AX} - \mathbf{Y}\|_F^2 + \lambda\|\mathbf{X}\|_0 \quad \text{s.t. } \mathbf{X} \geq 0, \tag{3}$$

where $\|\cdot\|_F$ is the F norm that represents the error between the measured value and real data, $\|\cdot\|_0$ is an ℓ_0 norm sparsity constraint, and λ is a regularization parameter. It is worth noting that owing to the existence of spectral variability in hyperspectral data, only the ANC is normally considered, and the ASC is not considered. In addition, because Eq. (3) is nonconvex, which is hard to solve, the ℓ_1 norm sparsity constraint

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{AX} - \mathbf{Y}\|_F^2 + \lambda\|\mathbf{X}\|_1 \quad \text{s.t. } \mathbf{X} \geq 0, \tag{4}$$

is taken into account.

To improve the sparsity of the ℓ_1 norm, a weighted ℓ_1 norm is proposed:[43]

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{AX} - \mathbf{Y}\|_F^2 + \lambda\|\mathbf{W} \odot \mathbf{X}\|_1 \quad \text{s.t. } \mathbf{X} \geq 0, \tag{5}$$

where $\odot$ represents the point multiplication operation, $\mathbf{W} \in \Re^{M\times p}$ is obtained as

$$\mathbf{W}_{ij}^{k+1} = \frac{1}{|\mathbf{X}_{ij}^k| + \varepsilon}, \tag{6}$$

where ε is an extremely small constant and $\mathbf{X}_{ij}^t$ is the value of the (i, j)'th element of the abundance matrix $\mathbf{X}$ in the t'th iteration.

3 Nonlocal Weighted Sparse Unmixing Method Based on Global Search

3.1 *Nonlocal Similarity Block Based on Global Search*

Nonlocal information, as important spatial information, is widely used in HSI classification,[44,45] denoising,[46] and unmixing.[47] However, nonlocal-based methods tend to use a local search box strategy to extract nonlocal similar pixels,[44–47] which requires a large number of local search boxes. Although this search strategy improves execution efficiency, it fails to find many similar pixels that may be far from a fixed search area. We use a global search strategy to extract similar pixels, which can exploit spatial information well. In particular, abundance images and HSIs have similar spatial characteristics.[48] As shown in Fig. 1, the following operations are performed on the abundance image:

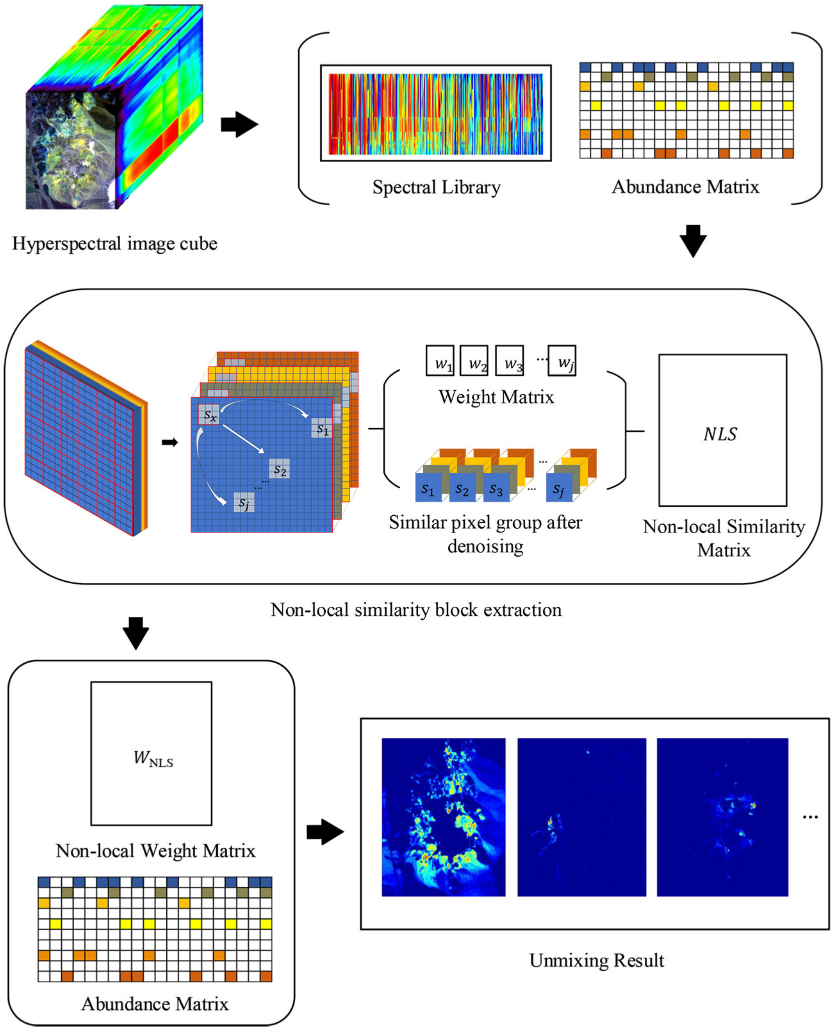


Fig. 1 Flowchart of G-NLWSU.

1. $\mathbf{X}$, for the 8-neighborhood block $\mathbf{S}_x$ centered on x, j neighborhood blocks $\mathbf{S}_1 \sim \mathbf{S}_j$ with the highest similarity were searched in the whole image, with similarity discrimination method
$$\mathrm{dist}(x, y) = (\mathbf{S}_x - \mathbf{S}_y) \otimes \mathbf{g}, \tag{7}$$
where $\mathbf{S}_x$ and $\mathbf{S}_y$ represent the neighborhood block centered on x and y, respectively, $\mathbf{g}$ is a Gaussian kernel function, and $\otimes$ denotes the convolution operation.
2. Extract the center pixels of j similar blocks and reorganize them into a similar pixels group $\mathbf{P}_x = [s_1 \sim s_j]^T$ and then perform a low-rank noise reduction on $\mathbf{P}_x$ to remove noises
$$\mathbf{P}_x = \mathrm{svd}(\mathbf{P}_x), \tag{8}$$
where $\mathrm{svd}(\cdot)$ is the SVD function, and the following operations are performed:
$$[\mathbf{U}, \Sigma, \mathbf{V}^T] = \mathrm{svd}(\mathbf{P}_x), \tag{9}$$
where $\mathbf{U}$ denotes a left singular vector matrix, Σ represents the diagonal matrix, where its the diagonal elements are the singular values in descending order, and $\mathbf{V}$ denotes a right singular vector matrix. The largest k singular values in Σ and their corresponding left and right singular vectors are selected for reconstruction of $\mathbf{P}_x$.

3. Perform a weighted summation on $\mathbf{P}_x$ to obtain the nonlocal similarity value $\mathrm{NLS}(x)$ corresponding to x

$$\mathbf{W}(i) = |x - s_i|^2/h, \tag{10}$$

where the s_i are the center pixels of the i'th similar blocks of the pixel x, and h is a smoothing parameter:

$$\mathrm{NLS}(x) = \sum_{i=1}^{j} \mathbf{W}(i) \times \mathbf{P}_x(i, :), \tag{11}$$

where $\mathbf{P}_x(i, :)$ is the i'th similar pixel of x, $\mathbf{W}(i)$ is the weight corresponding to the i'th similar pixel.

4. Perform the above three steps for all pixels to get the final nonlocal similarity matrix $\mathrm{NLS}(\mathbf{X})$.

3.2 *Nonlocal Weighted Sparse Unmixing Method*

The proposed nonlocal weighted algorithm, G-NLWSU, is described mathematically as

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{AX} - \mathbf{Y}\|_F^2 + \lambda\|\mathbf{W}_{\mathrm{NLS}} \odot \mathbf{X}\|_1 \quad \text{s.t. } \mathbf{X} \geq 0, \tag{12}$$

with weighting matrix

$$\mathbf{W}_{\mathrm{NLS}} = \frac{1}{|\mathrm{NLS}(\mathbf{X}_{ij}^k)| + \varepsilon}, \tag{13}$$

Algorithm 1 Pseudocode of G-NLWSU algorithm.

```
Initialization:
Set $k$, $t = 0$, $\lambda$, $\mu$, $\varepsilon > 0$, $\mathbf{U}^0$, $\mathbf{V}_1^0$, $\mathbf{V}_2^0$, $\mathbf{V}_3^0$, $\mathbf{D}_1^0$, $\mathbf{D}_2^0$, $\mathbf{D}_3^0$
Repeat:
$\mathbf{W}_{\mathrm{NLS}}^k = \frac{1}{|\mathrm{NLS}(\mathbf{X}_{ij}^k)|+\varepsilon}$   (13)
Repeat:
    $\mathbf{U}^{t+1} \leftarrow (\mathbf{A}^T\mathbf{A} + 2\mathbf{I})^{-1}[\mathbf{A}^T(\mathbf{V}_1^t + \mathbf{D}_1^t) + (\mathbf{V}_2^t + \mathbf{D}_2^t) + (\mathbf{V}_3^t + \mathbf{D}_3^t)]$   (25)
    $\mathbf{V}_1^{t+1} \leftarrow \frac{1}{1+\mu}[\mathbf{Y} + \mu(\mathbf{AU}^t - \mathbf{D}_1^t)]$   (27)
    $\mathbf{V}_2^{t+1} \leftarrow \mathrm{soft}\left(\mathbf{U}^{t+1} - \mathbf{D}_2^t, \frac{\lambda}{\mu}\mathbf{W}_{\mathrm{NLS}}^k\right)$   (29)
    $\mathbf{V}_3^{t+1} \leftarrow \max(\mathbf{U}^{t+1} - \mathbf{D}_3^t, 0)$   (32)
Update LaGrange multipliers:
    $\mathbf{D}_1^{t+1} \leftarrow \mathbf{D}_1^t - \mathbf{AU}^{t+1} + \mathbf{V}_1^{t+1}$
    $\mathbf{D}_2^{t+1} \leftarrow \mathbf{D}_2^t - \mathbf{U}^{t+1} + \mathbf{V}_2^{t+1}$
    $\mathbf{D}_3^{t+1} \leftarrow \mathbf{D}_3^t - \mathbf{U}^{t+1} + \mathbf{V}_3^{t+1}$
Update iteration:
    $k \leftarrow k + 1$
    $t \leftarrow t + 1$
until some stopping criterion is satisfied.
```

where $\mathrm{NLS}(\cdot)$ is a nonlocal similarity matrix extracted from the abundance matrix in each iteration, which is calculated by Eq. (11).

Equation (12) can be expressed as

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{AX}-\mathbf{Y}\|_F^2 + \lambda\|\mathbf{W}_{\mathrm{NLS}} \odot \mathbf{X}\|_1 + \imath_{R+}(\mathbf{X}), \tag{14}$$

where $\imath_{R+}(\mathbf{X}) = \sum_{i=1}^{n} \imath_{R+}(\mathbf{X}_i)$ is an index function, $\mathbf{X}_i$ is the i'th column of $\mathbf{X}$, and if $\mathbf{X}_i$ is nonnegative, then $\imath_{R+}(\mathbf{X}_i) = 0$, and otherwise $\imath_{R+}(\mathbf{X}_i) = +\infty$.

The equivalent constrained formula for Eq. (14) is

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{V}_1-\mathbf{Y}\|_F^2 + \lambda\|\mathbf{W}_{\mathrm{NLS}} \odot \mathbf{V}_2\|_1 + \imath_{R+}(\mathbf{V}_3) \quad \text{s.t. } \mathbf{V}_1 = \mathbf{AU}, \quad \mathbf{V}_2 = \mathbf{U}, \quad \mathbf{V}_3 = \mathbf{U}. \tag{15}$$

Algorithm 1 shows the algorithmic pseudocode.

3.3 *Parallel Computing-Based G-NLWSU*

HSI data have extremely high dimensions and complex topographic features, which means they require much computational time to perform an unmixing task. With recent hardware developments, graphics processing units (GPUs) have been used with large-scale data in parallel computing strategies, such as to improve computational efficiency in hyperspectral classification.[49,50]

We introduce a parallel computing-based G-NLWSU method, PG-NLWSU. The computing strategy considers the Compute Unified Device Architecture (CUDA) platform, a general-purpose computing architecture developed by NVIDIA that uses the parallel computing power of a GPU. CUDA controls the GPU to perform calculations through kernel functions. Since its layered structure (grid, block, and thread) is similar to that of three-dimensional images, it can be intuitively applied to the field of images. We use a hybrid programming strategy based on MATLAB and CUDA to accelerate the algorithm.

The data interaction between the host (multicore central processing units, CPUs) and device (GPUs) usually consumes much computational time and storage space. To alleviate these problems, we minimize the number of data transfers between the host and device, load some important calculation tasks to the GPU, return the results to the CPU, and use space preallocation or memory reduction to reduce memory consumption. After finishing calculations in the GPU, the occupied memory is released.

The complexity of $\mathbf{W}_{\mathrm{NLS}}$ in each iteration is $\mathcal{O}(\mathrm{MPS})$, where S is the number of sub-blocks of the image. To improve efficiency, we optimize the algorithm as follows:

1. The data of the matrix $\mathbf{U}$ are transferred to the GPU before the outer loop, and the nonlocal similar block operation is performed by the NLS_kernel function.
2. A grid, composed of thread, blocks, and threads in stages, is configured in the kernel function. This is shown in Fig. 2, where threadnum $=$ [16,16], blocknum $= \left[\frac{\text{width}+16-1}{\text{width}}, \frac{\text{height}+16-1}{\text{height}}\right]$.

The pseudocode of PG-NLWSU can be found in Algorithm 2.

3.4 *Computational Complexity Analysis*

The experiment in this article was performed using MATLAB R2018a and CUDA v10.2, a GTX1060 GPU, and 16 GB of memory.

For G-NLWSU, the most expensive computational step is $\mathbf{W}_{\mathrm{NLS}}$, which has order of complexity $\mathcal{O}(MPS)$, where P is the total number of pixels of the image and S is the number of sub-blocks. For $\mathbf{U}$, let $p_1 = \mathbf{A}^T\mathbf{A} + 2\mathbf{I}$, $p_2 = p_1^{-1}$, $p_3 = \mathbf{A}^T(\mathbf{V}_1^t + \mathbf{D}_1^t)$, $p_4 = p_2 p_3$, with respective complexity $\mathcal{O}(LM^2)$, $\mathcal{O}(M^3)$, $\mathcal{O}(MPL)$, $\mathcal{O}(PM^2)$. It is worth noting that P and S are usually greater than L and M, so the complexity of $\mathbf{U}$ is $\mathcal{O}(MP \cdot \max\{L, M\})$. The complexity of $\mathbf{V}_1^t$ and $\mathbf{D}_1^t$ is $\mathcal{O}(MPL)$, and the complexity of other steps is $\mathcal{O}(MP)$. In PG-NLWSU, the computational complexity of $\mathbf{W}_{\mathrm{NLS}}$ is $\mathcal{O}[MS \cdot (P/C)]$, where C is the parallel computing power of the GPU.

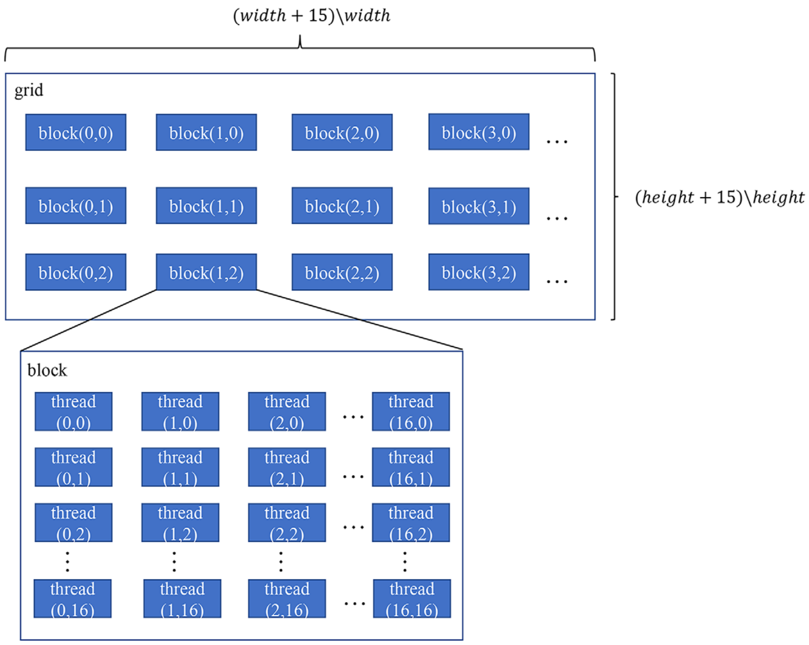


Fig. 2 Grid, blocks, and threads.

Algorithm 2 Pseudocode of PG-NLWSU algorithm

```
Initialization:
    Set $k$, $t = 0$, $\lambda$, $\mu$, $\varepsilon > 0$, $\mathbf{U}^0$, $\mathbf{V}_1^0$, $\mathbf{V}_2^0$, $\mathbf{V}_3^0$, $\mathbf{D}_1^0$, $\mathbf{D}_2^0$, $\mathbf{D}_3^0$
Repeat:
    Step 1. Transfer data $\mathbf{U}$ from host to device.
    Step 2. Execute NLS_kernel to calculate $\mathbf{P}_x$.
    Step 3. Transfer $\mathbf{P}_x$ from device to host.
    Step 4. Compute $\mathbf{P}_x = \mathrm{svd}(\mathbf{P}_x)$ on CPU.
    Step 5. Compute $\mathbf{W}_{\mathrm{NLS}}^k = \frac{1}{|\mathrm{NLS}(\mathbf{X}_{ij}^k)|+\varepsilon}$ on CPU.
Repeat:
     Step 6. Compute $\mathbf{U}^{t+1}$, $\mathbf{V}_1^{t+1}$, $\mathbf{V}_2^{t+1}$, $\mathbf{V}_3^{t+1}$, $\mathbf{D}_1^{t+1}$, $\mathbf{D}_2^{t+1}$, $\mathbf{D}_3^{t+1}$ on CPU.
     Step 7. Update $k$, $t$.
 until some stopping criterion is satisfied.
```

4 Experiments and Analysis

In this section, we evaluated PG-NLWSU on simulated and real datasets.

4.1 *Evaluation Metrics*

We consider four evaluation metrics to clearly compare the performance of the algorithms.

(1) Signal reconstruction error (SRE):

$$\mathrm{SRE(dB)} = 10 \cdot \log_{10}\left[\frac{E(\|\mathbf{X}\|_2^2)}{E(\|\mathbf{X}-\hat{\mathbf{X}}\|_2^2)}\right], \tag{16}$$

where $\mathbf{X}$ is the estimated abundance matrix, $\hat{\mathbf{X}}$ is the true abundance matrix, and $E(\cdot)$ denotes expectation. The larger the SRE, the smaller the error between the estimated and true abundance matrices.

(2) Probability of success ($\mathbf{P_s}$), which is an estimate of the probability that the relative error capability is less than a certain threshold:

$$\mathbf{P_s} \equiv \mathrm{P}\left(\frac{\|\mathbf{X}-\hat{\mathbf{X}}\|^2}{\|\mathbf{X}\|^2} \leq \text{threshold}\right), \tag{17}$$

where $\mathbf{X}$ is the estimation matrix and $\hat{\mathbf{X}}$ is the true matrix. According to previous work in SUnSAL-TV, we assume threshold$\leqslant 3.16\,(5\text{ dB})$. When $\mathbf{P_s} = 1$, then the total relative error power of abundance is less than $1/3.16$. We use $\mathbf{P_s}$ to supplement the stability evaluation index of the algorithm (SRE alone cannot derive the stability of the estimate).

(3) Sparsity:

$$\text{Sparsity} = \frac{s}{mn}, \tag{18}$$

where s is the number of elements in $\mathbf{X}$ greater than 0.005, and mn is the total number of elements in the matrix $\mathbf{X}$. The smaller the Sparsity the higher the sparsity of the matrix.

(4) Root mean square error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij})^2}, \tag{19}$$

where $\mathbf{X}_{ij}$, $\hat{\mathbf{X}}_{ij}$ are the respective element values at position (i, j) in the estimation matrix and real matrix, m and n are the matrix dimensions. The smaller the RMSE, the smaller the error between the measured and true values.

Based on the above descriptions, the evaluation metrics, i.e., SRE and RMSE, are used to evaluate the error of algorithm unmixing results. $\mathbf{P_s}$ was used to evaluate the stability of the algorithm. Sparsity focused on comparing the sparsity of unmixing results.

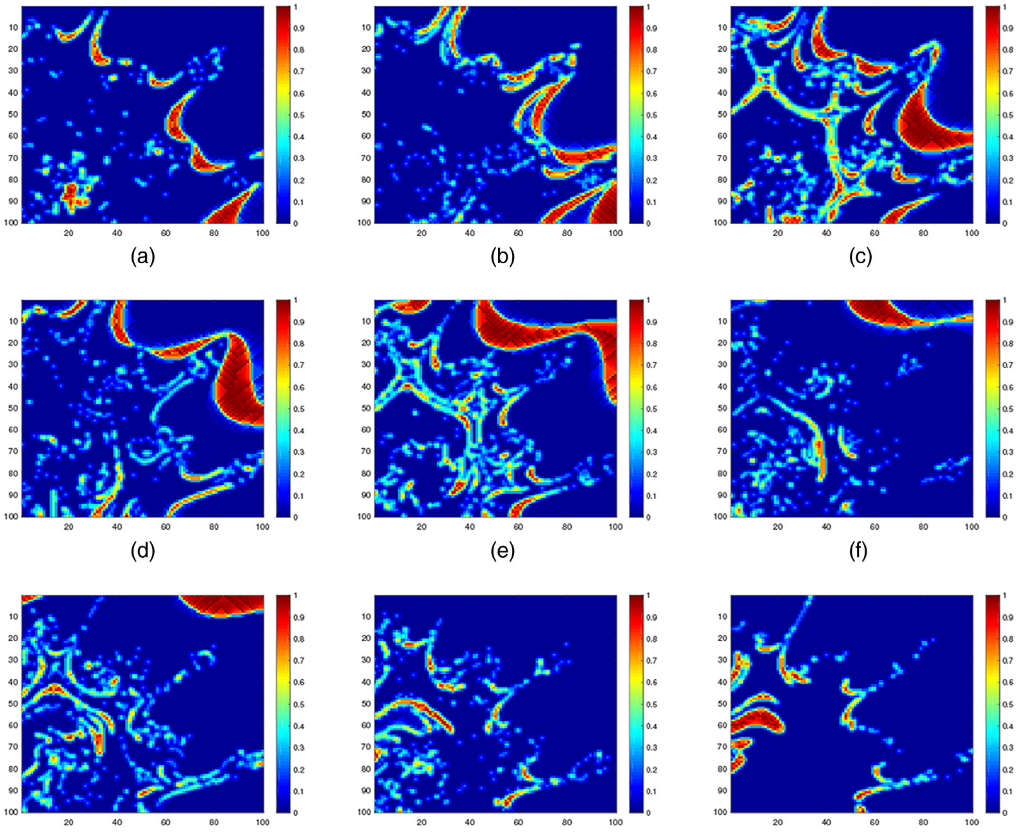
4.2 *Experiment on Simulated Datasets*

PG-NLWSU was tested on two simulated datasets to validate its unmixing performance. To facilitate the comparison, SUnSAL, CLSUnSAL, NCLS-TV, and SUnSAL-TV were run on the simulated dataset to evaluate the algorithm results.

4.2.1 *Simulated datasets*

The spectral library $\mathbf{A} \in \Re^{224\times 240}$ used in the simulated data experiment is a combination of 240 mineral types from the United States Geological Survey (USGS) library. The reflectance values of the spectral characteristics are given in 224 spectral bands and are evenly distributed in the range of 0.4 to 2.5 μm.

1. Simulated data cube 1 (DC1) is composed of nine endmember features randomly selected from spectral library $\mathbf{A}$.[51] It contains 100×100 pixels and 224 bands, with added Gaussian noise (SNR = 30, 40, 50). The ground truth abundances of each endmember are shown in Fig. 3.
2. As shown in Fig. 4, simulated data cube 2 (DC2) is composed of five endmembers, and the abundance matrix is generated using the hyperspectral imagery synthesis (EIAS) toolbox. The abundance distribution of the endmembers in DC2 is relatively uniform and contains much spatial information. Independent and uniformly distributed Gaussian noise (SNR = 30, 40, 50) is added.

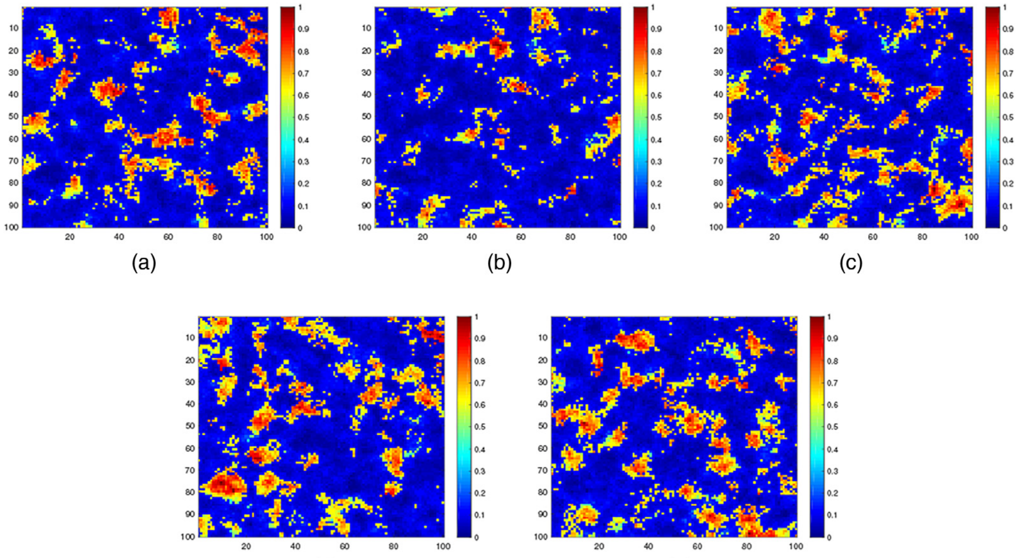


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 3 Real abundances of endmembers in DC1, where (a)–(i) are abundances of endmembers 1 to 9, respectively.



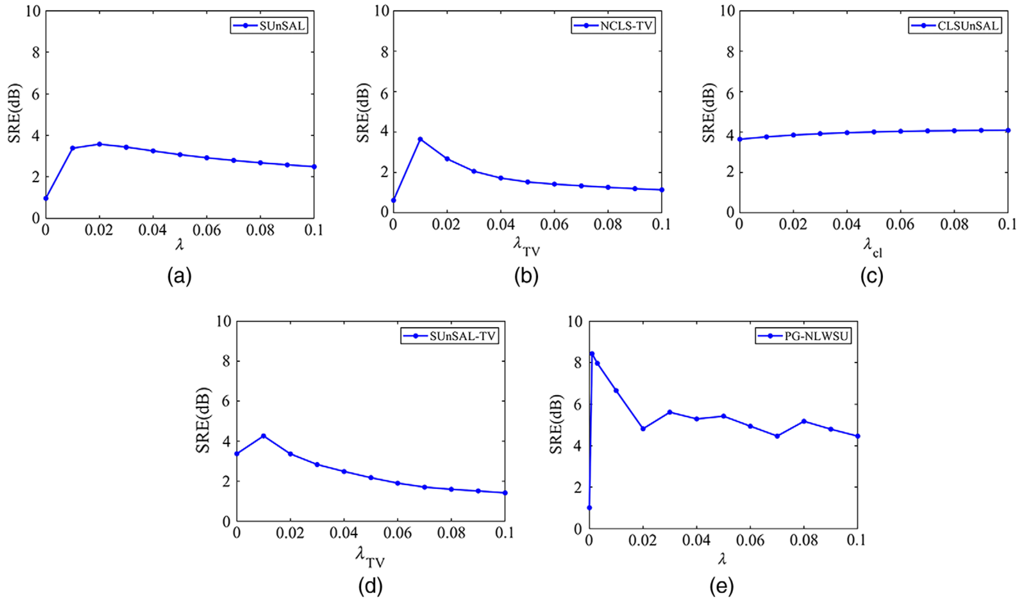
(a) (b) (c)

(d) (e)

Fig. 4 Real abundances of endmembers in DC2, where (a)–(e) are abundances of endmembers 1 to 5, respectively.

4.2.2 *Influence of regularization parameters*

We tested the effect of regularization parameters of different algorithms on the SRE in the range of 0 to 0.1 on DC1. The results are shown in Fig. 5. Among the algorithms, PG-NLWSU had the



(a) (b) (c)

(d) (e)

Fig. 5 Influence of regularization parameters on algorithms: (a) SUnSAL, (b) NCLS-TV, (c) CLSUnSAL, (d) SUnSAL-TV ($\lambda = 0.01$), and (e) PG-NLWSU.

highest accuracy. Comparing the curves of NCLS-TV and SUnSAL-TV, it can be seen that the spatial regularization parameter λ_{TV} has a greater influence than λ on the algorithm accuracy. By comparing SUnSAL, CLSUnSAL, NCLS-TV, and PG-NLWSU, it can be seen that the unmixing algorithm using spatial information more easily obtains the optimal solution.

4.2.3 *Optimization effect analysis*

We compare the speed of nonlocal similarity block extraction (the outer loop) between G-NLWSU and PG-NLWSU on the following data:

(1) DC1: data size $100 \times 100 \times 224$.
(2) DC2: data size $100 \times 100 \times 224$.
(3) Cuprite dataset: data size $250 \times 191 \times 188$.

The results are shown in Table 1.

Speed is defined as the ratio of the running times of the two algorithms. According to the information in the table, as image size and amount of information increase, the advantages of parallel acceleration become more obvious.

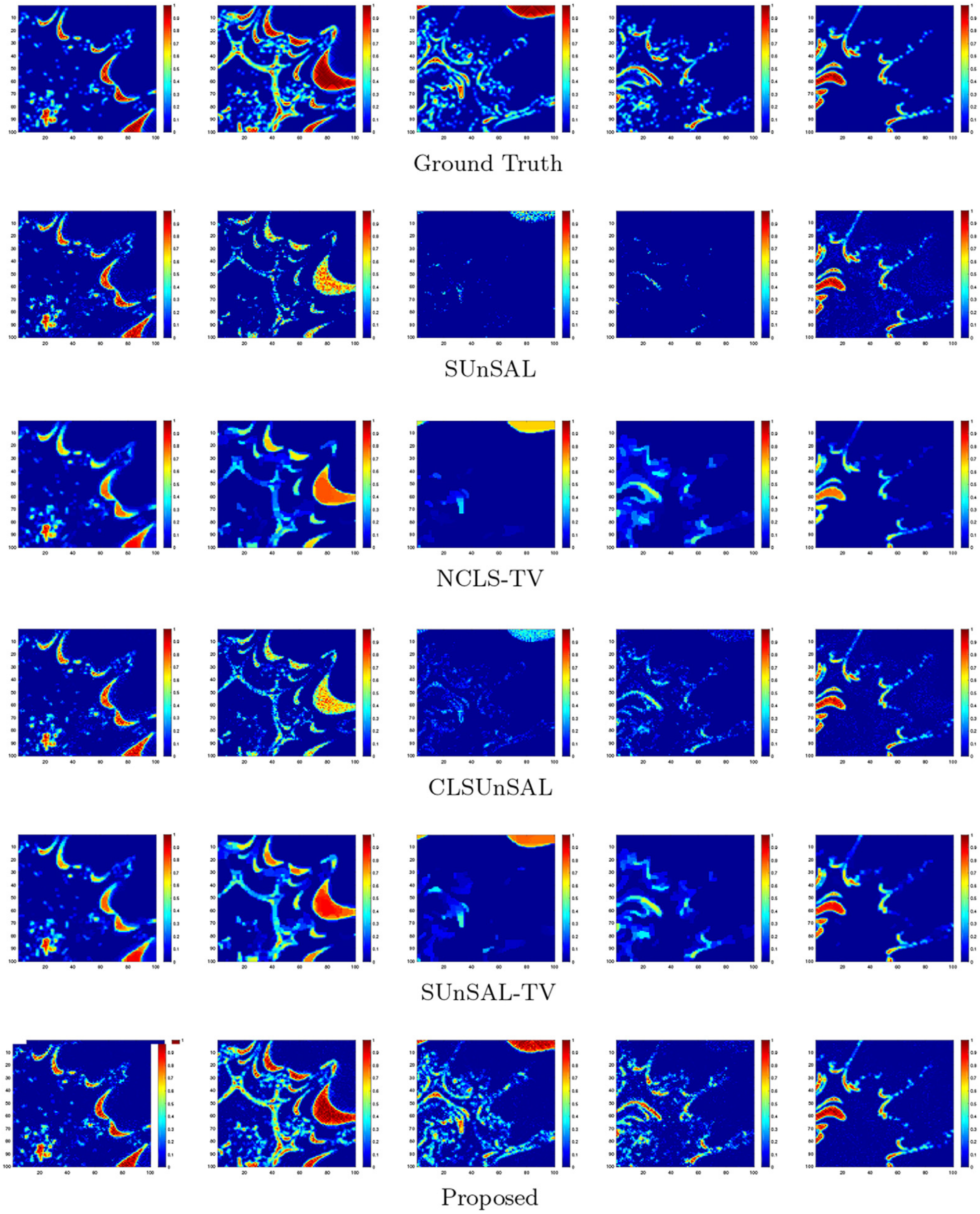
Table 1 Comparison of computing efficiency between G-NLWSU and PG-NLWSU.

Dataset	Algorithm	Time	Speed
DC1	G-NLWSU	384.9466 (s)	12.3871
	PG-NLWSU	31.4859 (s)	
DC2	G-NLWSU	387.5336 (s)	12.2086
	PG-NLWSU	31.7428 (s)	
Cuprite	G-NLWSU	282.1 (m)	26.8667
	PG-NLWSU	10.5 (m)	

4.2.4 *Results on simulated data sets*

We show the experimental results of PG-NLWSU and comparison algorithms on two simulated datasets. Figure 6 shows the unmixing results of different algorithms on endmembers 1, 3, 7, 8, and 9 of DC1 when SNR = 30. Figure 7 shows the unmixing results of all the endmembers on DC2 for different algorithms when SNR = 30. Table 2 shows the optimal parameters of all algorithms when SNR is fixed at 30, 40, and 50 dB on simulated data sets DC1 and DC2. Tables 3 and 4 show the index evaluation results of all algorithms on DC1 and DC2 when the parameters are optimal. In Tables 3 and 4, bold represents the best experimental result.

As can be seen from Figs. 6 and 7, the endmember abundance results estimated by SUnSAL are relatively poor. The result of CLSUnSAL is similar to that of SUnSAL but contains more information. Comparing NCLS-TV and SUnSAL-TV, one can find that TV regularization leads to excessive smoothing of abundance, which is obvious in the results of DC1 endmembers 7 and 8 and DC2 endmember 3. The unmixing effect of PG-NLWSU is significantly better than that of



Ground Truth

SUnSAL

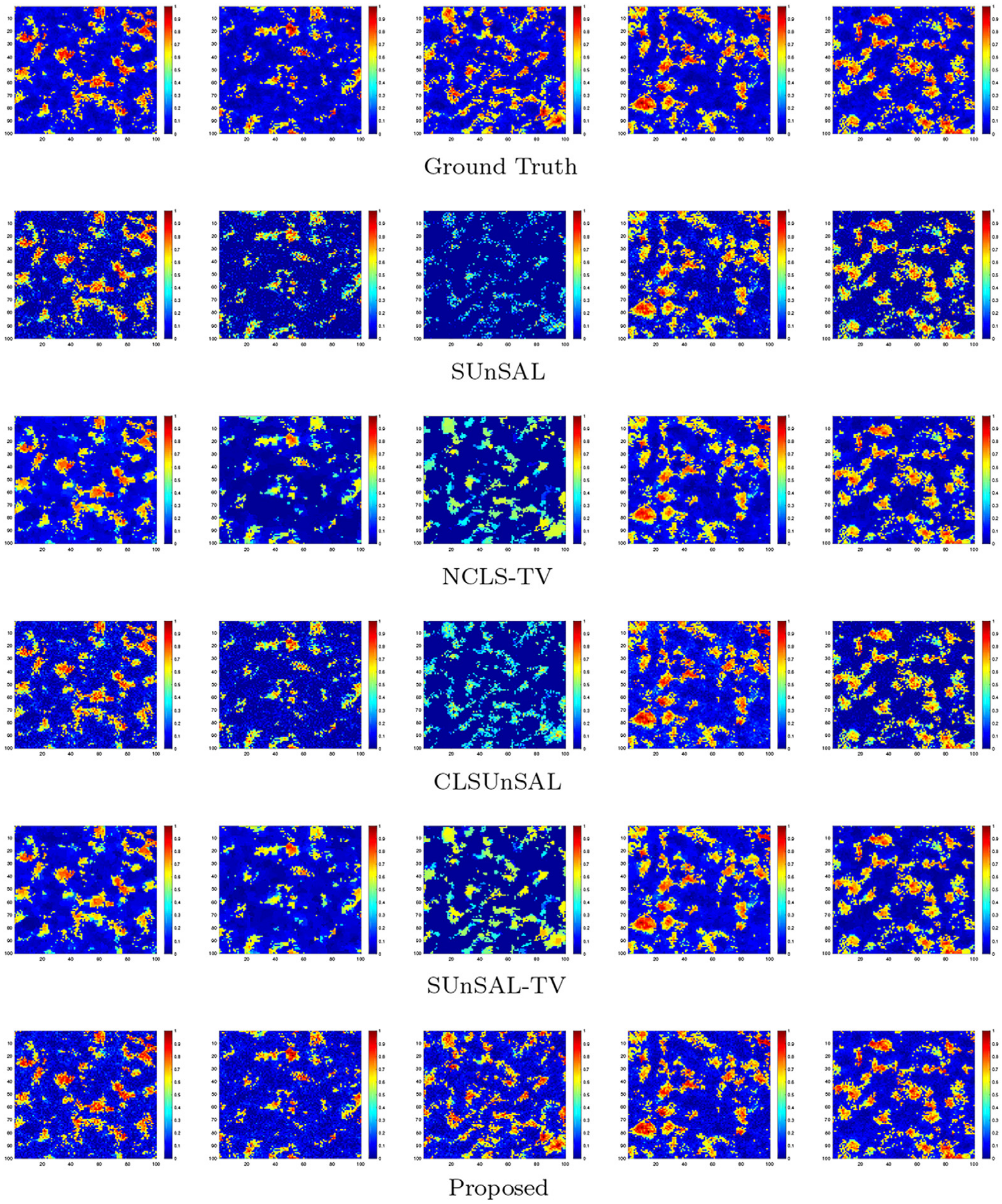
NCLS-TV

CLSUnSAL

SUnSAL-TV

Proposed

Fig. 6 Abundances of endmembers 1, 3, 7, 8, 9 estimated by all algorithms on DC1.



Ground Truth

SUnSAL

NCLS-TV

CLSUnSAL

SUnSAL-TV

Proposed

Fig. 7 Abundances of endmembers 1, 2, 3, 4, 5 estimated by all algorithms on DC2.

other algorithms, indicating that the nonlocal weighting matrix improves the performance of the algorithm.

From the results in Table 3, PG-NLWSU algorithm can provide higher SRE, $\mathbf{P_s}$, sparsity, and RMSE results than that of other comparison algorithms at different noise levels especially for the worse noise scenario such as 30 dB, which means that PG-NLWSU has high robustness to noises.

According to the results in Table 4, the results of SRE, sparsity, and RMSE under all noise intensity show that the PG-NLWSU can generate sparser unmixing accuracy and than other algorithms primarily because DC2 maintains relatively uniform spatial distribution, indicating that the nonlocal weighted model effectively promotes the sparseness of the algorithm. For $\mathbf{P_s}$, PG-NLWSU still captures the best result especially at 30 dB, which implies that the proposed PG-NLWSU algorithm is very stable.

The computational efficiency of all algorithms on DC1 and DC2 is shown in Table 5, and bold represents the best experimental results. SUnSAL takes the least time. NCLS-TV and SUnSAL-TV run slower because they must extract neighborhood information of images, whereas CLSUnSAL and PG-NLWSU run longer, but when the SNR increases, PG-NLWSU shows higher calculational efficiency.

Table 2 Parameter values used by different algorithms in experiments on simulated datasets.

Data cube	Algorithm	SNR		
		30 dB	40 dB	50 dB
DC1	SUnSAL	$\lambda = 5e-2$	$\lambda = 5e-3$	$\lambda = 5e-4$
	NCLS-TV	$\lambda = 5e-3$	$\lambda = 7e-4$	$\lambda = 7e-4$
	CLSUnSAL	$\lambda = 9e-2$	$\lambda = 5e-2$	$\lambda = 5e-3$
	SUnSAL-TV	$\lambda = 5e-3$	$\lambda = 5e-3$	$\lambda = 5e-4$
		$\lambda_{TV} = 5e-3$	$\lambda_{TV} = 7e-4$	$\lambda_{TV} = 7e-4$
	Proposed	$\lambda = 1e-3$	$\lambda = 9e-5$	$\lambda = 9e-5$
DC2	SUnSAL	$\lambda = 5e-3$	$\lambda = 5e-3$	$\lambda = 5e-3$
	NCLS-TV	$\lambda = 5e-3$	$\lambda = 5e-3$	$\lambda = 5e-3$
	CLSUnSAL	$\lambda = 9e-2$	$\lambda = 9e-2$	$\lambda = 9e-2$
	SUnSAL-TV	$\lambda = 5e-3$	$\lambda = 5e-3$	$\lambda = 5e-3$
		$\lambda_{TV} = 5e-3$	$\lambda_{TV} = 5e-3$	$\lambda_{TV} = 5e-4$
	Proposed	$\lambda = 9e-4$	$\lambda = 9e-4$	$\lambda = 9e-4$

Table 3 Evaluation results of different algorithms in experiments on simulated dataset 1.

SNR	Algorithm	SRE	$\mathbf{P_s}$	Sparsity	RMSE
30 dB	SUnSAL	3.0749	0.5986	0.0455	0.3613
	NCLS-TV	4.1514	0.7352	0.0792	0.2820
	CLSUnSAL	4.1061	0.7345	0.0688	0.2850
	SUnSAL-TV	4.9235	0.8284	0.0572	0.2361
	Proposed	**8.3768**	**0.9852**	**0.0207**	**0.1066**
40 dB	SUnSAL	6.6155	0.9390	0.0481	0.1599
	NCLS-TV	6.7969	0.9793	0.0701	0.1534
	CLSUnSAL	7.9331	1.0000	0.0461	0.1181
	SUnSAL-TV	8.0080	0.9950	0.0424	0.1160
	Proposed	**12.9224**	**1.0000**	**0.0183**	**0.0374**
50 dB	SUnSAL	9.0011	0.9997	0.0388	0.0923
	NCLS-TV	9.9019	1.0000	0.0433	0.0750
	CLSUnSAL	11.9637	1.0000	0.0245	0.0467
	SUnSAL-TV	10.1044	0.9993	0.0259	0.0716
	Proposed	**15.3358**	**1.0000**	**0.0146**	**0.0215**

4.3 *Experiment on Real Dataset*

We experimentally evaluate the algorithm on the AVIRIS Cuprite dataset, whose information was collected from a copper mine in Nevada and is widely used to verify spectral unmixing

Table 4 Evaluation results of different algorithms in experiments on simulated dataset 2.

SNR	Algorithm	SRE	$\mathbf{P_s}$	Sparsity	RMSE
30 dB	SUnSAL	3.3858	0.7996	0.0688	0.3244
	NCLS-TV	5.6221	0.9638	0.0755	0.1938
	CLSUnSAL	5.0941	0.9302	0.0602	0.2189
	SUnSAL-TV	6.4853	0.9764	0.0511	0.1589
	Proposed	**9.5163**	**1.0000**	**0.0215**	**0.0791**
40 dB	SUnSAL	7.5352	1.0000	0.0543	0.1248
	NCLS-TV	7.2637	0.9859	0.0514	0.1328
	CLSUnSAL	8.5245	1.0000	0.0464	0.0994
	SUnSAL-TV	8.3807	0.9896	0.0317	0.1027
	Proposed	**11.8972**	**1.0000**	**0.0204**	**0.0457**
50 dB	SUnSAL	11.1740	1.0000	0.0315	0.0540
	NCLS-TV	11.3668	1.0000	0.0403	0.0516
	CLSUnSAL	12.4526	1.0000	0.0228	0.0402
	SUnSAL-TV	12.5586	1.0000	0.0249	0.0392
	Proposed	**17.4999**	**1.0000**	**0.0208**	**0.0126**

Table 5 Time efficiency of various algorithms on simulated datasets (in seconds).

		SNR		
Data cube	Algorithm	30 dB	40 dB	50 dB
DC1	SUnSAL	**4.8210**	**3.8051**	**3.6672**
	NCLS-TV	70.8100	111.4890	138.1508
	CLSUnSAL	146.7633	160.1402	158.1793
	SUnSAL-TV	71.9640	112.7408	151.6289
	Proposed	380.1912	283.4539	167.6324
DC2	SUnSAL	**4.6042**	**3.8688**	**4.1016**
	NCLS-TV	64.4921	60.9866	60.3077
	CLSUnSAL	161.0892	152.7736	154.1891
	SUnSAL-TV	74.3964	75.2871	66.1086
	Proposed	379.0015	206.9122	154.2739

algorithms. Figure 8 shows the mineral distribution map drawn by USGS in 1995. The band range is 0.4 to 2.5 μm. A 250 × 191-pixel subregion with 188 bands is considered, where noisy bands 1 and 2 and water-absorption bands 104 to 113 and 148 to 167 are removed from the original 224 bands. The spectral library $\mathbf{A} \in \Re^{188\times 498}$ also processed and reserved 188 bands of the USGS spectral library.

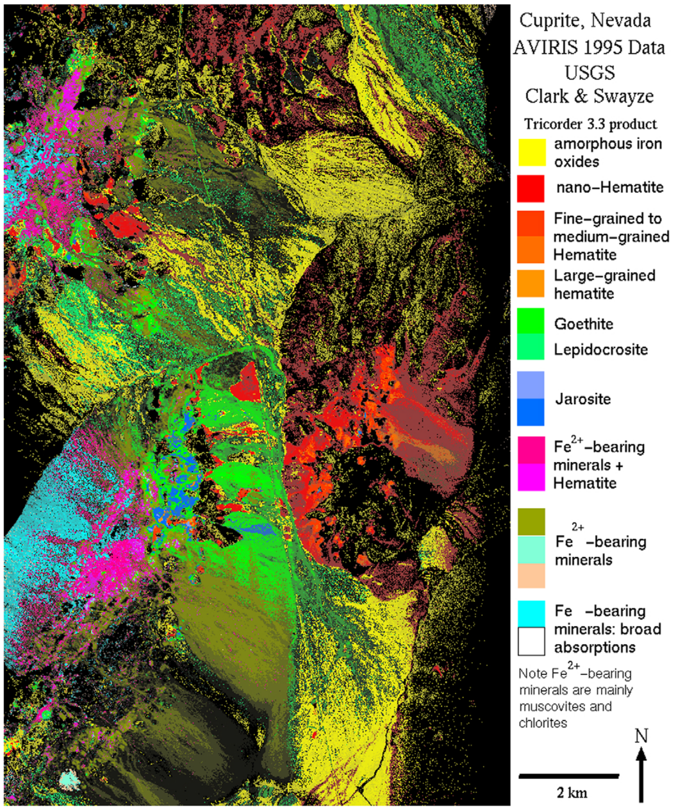

Cuprite, Nevada
AVIRIS 1995 Data
USGS
Clark & Swayze
Tricorder 3.3 product
amorphous iron oxides
nano–Hematite
Fine–grained to medium–grained Hematite
Large–grained hematite
Goethite
Lepidocrosite
Jarosite
Fe^{2+}–bearing minerals + Hematite
Fe^{2+}–bearing minerals
Fe –bearing minerals: broad absorptions
Note Fe^{2+}–bearing minerals are mainly muscovites and chlorites
N
2 km

Fig. 8 USGS uses Tricorder 3.3 software to obtain the mineral distribution map of the Cuprite mining district in Nevada.

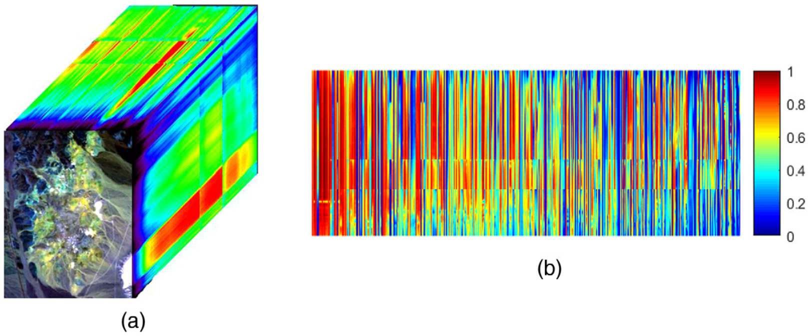
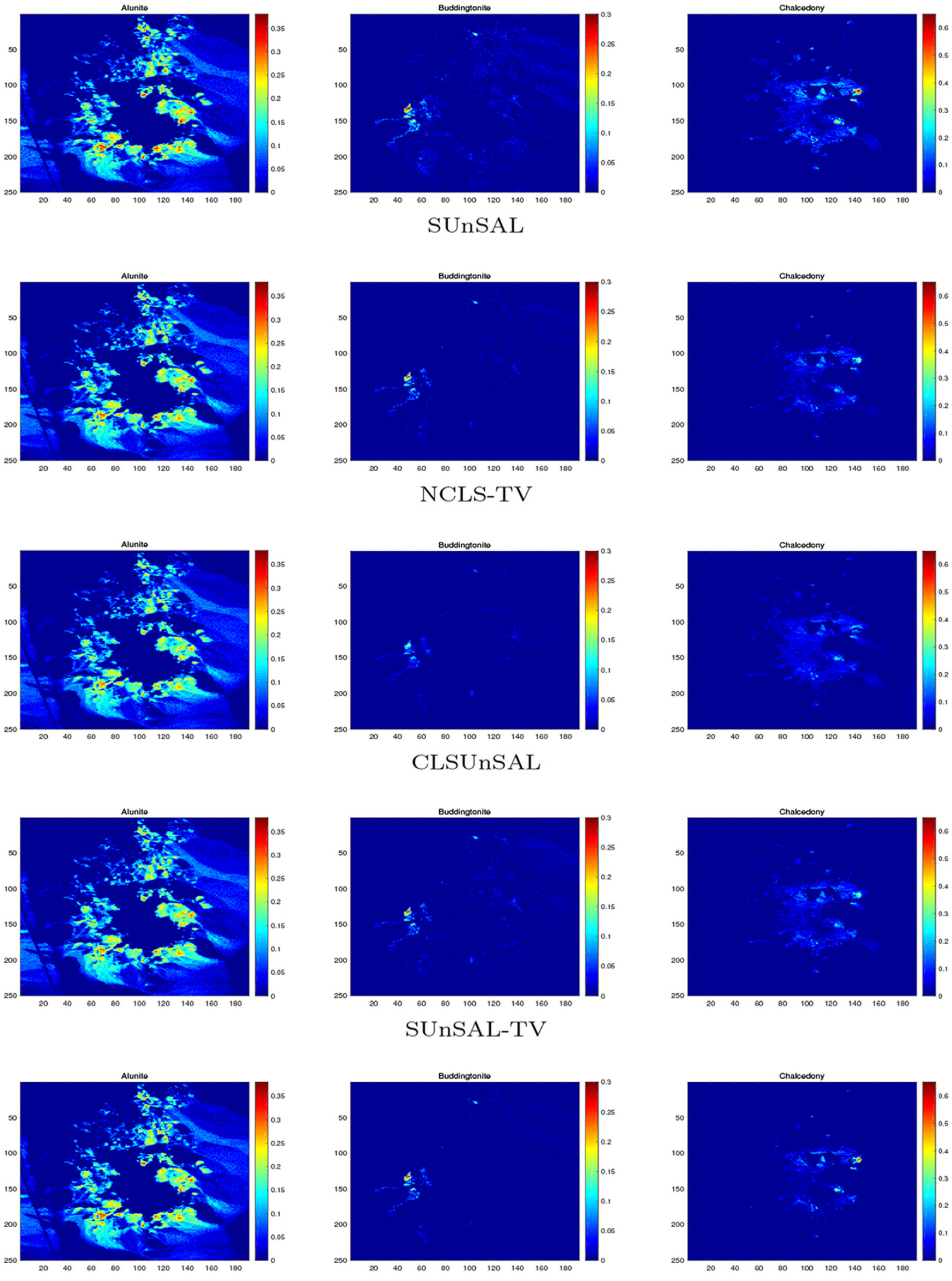


Fig. 9 (a) Hyperspectral cubes in cuprite area and (b) spectral library $\mathbf{A}_1$.

Figure 9 shows the Cuprite data cube and spectral library $\mathbf{A}_1$. Since no real abundance map can be used, the experimental results of this algorithm are compared with those of SUnSAL, CLSUnSAL, NCLS-TV, and SUnSAL-TV, with results as shown in Fig. 10.

Figure 10 compares the estimated abundances of three minerals (alunite, buddingtonite, and chalcedony) in the cuprite area by SUnSAL, NCLS-TV, CLSUnSAL, SUnSAL-TV, and PG-NLWSU. We set the parameters of SUnSAL, NCLS-TV, CLSUnSAL, and PG-NLWSU to $\lambda = 1e-3$, $\lambda_{\mathrm{TV}} = 3e-9$, $\lambda = 9e-2$ and $\lambda = 9e-5$, respectively, and set the parameters of SUnSAL-TV to $\lambda = 1e-3$, $\lambda_{\mathrm{TV}} = 3e-9$. The abundance graph of buddingtonite clearly shows the performance of different algorithms in dealing with noise, whereas chalcedony shows the algorithm's more detailed unmixing ability. It can be seen that the abundance graph of PG-NLWSU performs better than those of the other algorithms.



SUnSAL

NCLS-TV

CLSUnSAL

SUnSAL-TV

Proposed

Fig. 10 Abundance results of different algorithms on Cuprite datasets.

5 Conclusions

SU is a classical method to solve the problem of spectral unmixing. To use a weighted ℓ_1 norm to enhance the sparsity of the abundance matrix is a current research hotspot. We propose a nonlocal weighted sparse unmixing algorithm (PG-NLWSU) based on global search and parallel acceleration. The nonlocal mean is introduced to the sparse decomposition model through a weighting factor, for a new method to utilize the nonlocal information of the image. To fully use this nonlocality, we use a search box of the same size as the image, and parallel computing improves the computing efficiency. Experimental results show that the proposed algorithm performs better in images with abundant nonlocal information. In future work, we will consider the texture information of the image and use it to eliminate the smoothing effect caused by the use of neighborhood or nonlocal information.

6 Appendix A: Model Solution

Equation (15) can be written as

$$\min_{\mathbf{U},\mathbf{V}} \mathrm{g}(\mathbf{V}) \quad \text{s.t. } \mathbf{GU}+\mathbf{BV}=0, \tag{20}$$

where

$$\begin{aligned} \mathbf{V} &= (\mathbf{V_1},\mathbf{V_2},\mathbf{V_3}),\\ g(\mathbf{V}) &= \frac{1}{2}\|\mathbf{V}_1-\mathbf{Y}\|_F^2+\lambda\|\mathbf{W}_{\mathrm{NLS}}\odot\mathbf{V}_2\|_1+\imath_{R+}(\mathbf{V}_3),\\ \mathbf{G} &= \begin{bmatrix}\mathbf{A}\\ \mathbf{I}\\ \mathbf{I}\end{bmatrix} \mathbf{B}=\begin{bmatrix}-\mathbf{I} & 0 & 0\\ 0 & -\mathbf{I} & 0\\ 0 & 0 & -\mathbf{I}\end{bmatrix}. \end{aligned} \tag{21}$$

Equation (20) is solved by the ADMM algorithm, and its augmented LaGrange function is as follows:

$$\mathcal{L}(\mathbf{U},\mathbf{V},\mathbf{D})\equiv \mathrm{g}(\mathbf{U},\mathbf{V})+\frac{\mu}{2}\|\mathbf{GU}+\mathbf{BV}-\mathbf{D}\|_F^2, \tag{22}$$

where $\mu>0$, $\frac{\mathbf{D}}{\mu}$ is the LaGrange multiplier.

Equation (22) is expanded as

$$\begin{aligned} \mathcal{L}(\mathbf{U},\mathbf{V}_1,\mathbf{V}_2,\mathbf{V}_3,\mathbf{D}_1,\mathbf{D}_2,\mathbf{D}_3) = {} & \frac{1}{2}\|\mathbf{V}_1-\mathbf{Y}\|_F^2+\lambda\|\mathbf{W}_{\mathrm{NLS}}\odot\mathbf{V}_2\|_1+\imath_{R+}(\mathbf{V}_3)\\ & +\frac{\mu}{2}\|\mathbf{AU}-\mathbf{V}_1-\mathbf{D}_1\|_F^2+\frac{\mu}{2}\|\mathbf{U}-\mathbf{V}_2-\mathbf{D}_2\|_F^2\\ & +\frac{\mu}{2}\|\mathbf{U}-\mathbf{V}_3-\mathbf{D}_3\|_F^2. \end{aligned} \tag{23}$$

In Eq. (23), the variables need to be separated during the solution process. When solving for a variable (e.g., **U**), other variables default to fixed values (e.g., **V**, **D**). We separately solve the three variables in the above formula, and the optimization function is

$$\mathbf{U}^{t+1}\leftarrow \arg\min_{\mathbf{U}}\frac{\mu}{2}\|\mathbf{AU}-\mathbf{V}_1-\mathbf{D}_1\|_F^2+\frac{\mu}{2}\|\mathbf{U}-\mathbf{V}_2-\mathbf{D}_2\|_F^2+\frac{\mu}{2}\|\mathbf{U}-\mathbf{V}_3-\mathbf{D}_3\|_F^2. \tag{24}$$

The matrix derivation solution is

$$\mathbf{U}^{t+1}\leftarrow(\mathbf{A}^T\mathbf{A}+2\mathbf{I})^{-1}[\mathbf{A}^T(\mathbf{V}_1^t+\mathbf{D}_1^t)+(\mathbf{V}_2^t+\mathbf{D}_2^t)+(\mathbf{V}_3^t+\mathbf{D}_3^t)]. \tag{25}$$

The optimization function of $\mathbf{V}_1$ is

$$\mathbf{V}_1^{t+1}\leftarrow \arg\min_{\mathbf{V}_1}\frac{1}{2}\|\mathbf{V}_1-\mathbf{Y}\|_F^2+\frac{\mu}{2}\|\mathbf{AU}-\mathbf{V}_1-\mathbf{D}_1\|_F^2, \tag{26}$$

its solution is

$$\mathbf{V}_1^{t+1}\leftarrow\frac{1}{1+\mu}[\mathbf{Y}+\mu(\mathbf{AU}^t-\mathbf{D}_1^t)]. \tag{27}$$

The optimization function of $\mathbf{V}_2$ is

$$\mathbf{V}_2^{t+1}\leftarrow \arg\min_{\mathbf{V}_2}\lambda\|\mathbf{W}_{\mathrm{NLS}}\odot\mathbf{V}_2\|_1+\frac{\mu}{2}\|\mathbf{U}-\mathbf{V}_2-\mathbf{D}_2\|_F^2, \tag{28}$$

with solution

$$\mathbf{V}_2^{t+1} \leftarrow \mathrm{soft}\left(\mathbf{U}^t - \mathbf{D}_2^t, \frac{\lambda}{\mu}\mathbf{W}_{\mathrm{NLS}}\right), \tag{29}$$

where soft(a, b) is a soft threshold function, expressed as

$$\mathrm{soft}(a, b) \equiv \mathrm{sign}(a) \max\{|a| - b, 0\}, \tag{30}$$

where sign(a) is a sign function

$$\begin{cases} 1, & a > 0 \\ -1, & a < 0 \end{cases}.$$

The optimization function of $\mathbf{V}_3$ is

$$\mathbf{V}_3^{t+1} \leftarrow \arg\min_{\mathbf{V}_3} \iota_{R+}(\mathbf{V}_3) + \frac{\mu}{2}\|\mathbf{U} - \mathbf{V}_3 - \mathbf{D}_3\|_F^2, \tag{31}$$

since the actual role of ι_{R+} is to remove the nonnegative value of the solution, the solution of $\mathbf{V}_3$ is

$$\mathbf{V}_3^{t+1} \leftarrow \max(\mathbf{U}^t - \mathbf{D}_3^t, 0). \tag{32}$$

Acknowledgments

This work was supported by the Natural Science Foundation of Ningxia Province of China (Project No. 2020AAC02028) and the Innovation Projects for Graduate Students of North Minzu University (Project No. YCX20064). We thank the Image and Intelligence Information Processing Innovation Team of the National Ethnic Affairs Commission of China and the Science Foundation of North Minzu University (Project No. 2020KYQD50) for their support. Disclosures: The authors declare no conflict of interest.

Data, Materials, and Code Availability

The EIAS toolbox used in Sec. 4.2.1 of this article is available at the following web site: http://www.ehu.es/ccwintco/index.php/Hyperspectral_Imagery_Synthesis_tools_for_MATLAB.

References

1. J. M. Bioucas-Dias et al., "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.* **1**(2), 6–36 (2013).
2. N. Keshava and J. F. Mustard, "Spectral unmixing," *IEEE Signal Process. Mag.* **19**(1), 44–57 (2002).
3. A. Plaza et al., "Foreword to the special issue on spectral unmixing of remotely sensed data," *IEEE Trans. Geosci. Remote Sens.* **49**(11), 4103–4110 (2011).
4. Q. Du et al., "End-member extraction for hyperspectral image analysis," *Appl. Opt.* **47**(28), F77–F84 (2008).
5. J. M. Bioucas-Dias et al., "Hyperspectral unmixing overview: geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **5**(2), 354–379 (2012).
6. J. W. Boardman, "Automating spectral unmixing of aviris data using convex geometry concepts," in *Proc. Fourth Annu. Airborne Gcosci. Workshop*, DC, pp. 11–14 (1993).
7. J. W. Boardman, F. A. Kruse, and R. O. Green, "Mapping target signatures via partial unmixing of AVIRIS data," in *Proc. 5th Annu. JPL Airborne Earth Sci. Workshop*, JPL, pp. 23–26 (1995).

8. M. E. Winter, "N-FINDR: an algorithm for fast autonomous spectral endmember determination in hyperspecral data," *Proc. SPIE* **3753**, 266–275 (1999).
9. C. C. Wu, S. Chu, and C. I. Chang, "Sequential N-FINDR algorithms," *Proc. SPIE* **7086**, 70860C (2008).
10. B. Zhang, W. Luo, and X. Jia, "New improvements in parallel implementation of N-FINDR algorithm," *IEEE Trans. Geosci. Remote Sens.* **50**(10), 3648–3659 (2012).
11. J. M. P. Nascimento and J. M. B. Dias, "Does independent component analysis play a role in unmixing hyperspectral data," *IEEE Trans. Geosci. Remote Sens.* **43**(1), 175–187 (2005).
12. C. I. Chang et al., "A new growing method for simplex-based endmember extraction algorithm," *IEEE Trans. Geosci. Remote Sens.* **44**(10), 2804–2819 (2006).
13. J. H. Gruninger, A. J. Ratkowski, and M. L. Hoke, "The sequential maximum angle convex cone (SMACC) endmember model," *Proc. SPIE* **5425**, 1–14 (2004).
14. T. H. Chan et al., "A simplex volume maximization framework for hyperspectral endmember extraction," *IEEE Trans. Geosci. Remote Sens.* **49**(11), 4177–4193 (2011).
15. J. Li and J. M. Bioucas-Dias, "Minimum volume simplex analysis: a fast algorithm to unmix hyperspectral data," in *IGARSS 2008-2008 IEEE Int. Geosci. and Remote Sens. Symp.*, IEEE, Vol. 3, pp. III–250–III–253 (2008).
16. T.-H. Chan et al., "A convex analysis based minimum-volume enclosing simplex algorithm for hyperspectral unmixing," *IEEE Trans. Signal Process.* **57**(11), 4418–4432 (2009).
17. J. Bioucas-Dias, "A variable splitting augmented Lagrangian approach to linear spectral unmixing," in *First Workshop on Hyperspectral Image and Signal Process.: Evol.in Remote Sens.*, pp. 1–4 (2009).
18. L. Miao and H. Qi, "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.* **45**, 765–777 (2007).
19. J. D. Bayliss, J. A. Gualtieri, and R. F. Cromp, "Analyzing hyperspectral data with independent component analysis," *Proc. SPIE* **3240**, 133–143 (1998).
20. C. H. Chen and X. Zhang, "Independent component analysis for remote sensing study," *Proc. SPIE* **3871**, 150–158 (1999).
21. T. M. Tu, "Unsupervised signature extraction and separation in hyperspectral images: a noise-adjusted fast independent component analysis," *Opt. Eng.* **39**(4), 897–906 (2000).
22. J. M. P. Nascimento and J. M. Bioucas-Dias, "Hyperspectral unmixing based on mixtures of Dirichlet components," *IEEE Trans. Geosci. Remote Sens.* **50**(3), 863–878 (2012).
23. S. Moussaoui et al., "Bayesian analysis of spectral mixture data using Markov chain Monte Carlo methods," *Chemometr. Intell. Lab. Syst.* **81**(2), 137–148 (2006).
24. M. Berman et al., "Ice: a statistical approach to identifying endmembers in hyperspectral images," *IEEE Trans. Geosci. Remote Sens.* **42**(10), 2085–2095 (2004).
25. A. Plaza et al., "Spatial/spectral endmember extraction by multidimensional morphological operations," *IEEE Trans. Geosci. Remote Sens.* **40**(9), 2025–2041 (2002).
26. D. M. Rogge et al., "Integration of spatial–spectral information for the improved extraction of endmembers," *Remote Sens. Environ.* **110**(3), 287–303 (2007).
27. G. Martin and A. Plaza, "Region-based spatial preprocessing for endmember extraction and spectral unmixing," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **8**(4), 745–749 (2011).
28. G. Martin, "Spatial-spectral preprocessing prior to endmember identification and unmixing of remotely sensed hyperspectral data," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **5**(2), 380–395 (2012).
29. S. Moussaoui et al., "On the decomposition of mars hyperspectral data by ICA and Bayesian positive source separation," *Neurocomputing* **71**(10–12), 2194–2208 (2007).
30. M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Sparse unmixing of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.* **49**(6), 2014–2039 (2011).
31. J. M. Bioucas-Dias and M. A. T. Figueiredo, "Alternating direction algorithms for constrained sparse regression: application to hyperspectral unmixing," in *IEEE Workshop Hyperspectral Image and Signal Process.: Evol. in Remote Sens.*, IEEE (2010).
32. M.-D. Iordache, J. Bioucas-Dias, and A. Plaza, "Collaborative sparse regression for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.* **52**(1), 341–354 (2013).

33. L. Sun et al., "A novel $\ell_{1/2}$ sparse regression method for hyperspectral unmixing," *Int. J. Remote Sens.* **34**(20), 6983–7001 (2013).
34. Y. Z. Cheng et al., "Reweighted sparse regression for hyperspectral unmixing," in *IEEE Geosci. and Remote Sens. Symp.* (2016).
35. M.-D. Iordache, J. M. Bioucas-Dias, and A. Plaza, "Total variation spatial regularization for sparse hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.* **50**(11), 4484–4502 (2012).
36. S. Zhang et al., "Hyperspectral unmixing based on local collaborative sparse regression," *IEEE Geosci. Remote Sens. Lett.* **13**(5), 631–635 (2016).
37. R. Wang et al., "Double reweighted sparse regression for hyperspectral unmixing," in *IGARSS 2016–2016 IEEE Int. Geosci. Remote Sens. Symp.* (2016).
38. R. Wang et al., "Hyperspectral unmixing using double reweighted sparse regression and total variation," *IEEE Geoence Remote Sens. Lett.* **14**(7), 1146–1150 (2017).
39. S. Wang et al., "Double reweighted sparse regression and graph regularization for hyperspectral unmixing," *Remote Sens.* **10**(7), 1046 (2018).
40. S. Zhang et al., "Spectral-spatial weighted sparse regression for hyperspectral image unmixing," *IEEE Trans. Geosci. Remote Sens.* **56**(6), 3265–3276 (2018).
41. Y. Zhong, R. Feng, and L. Zhang, "Non-local sparse unmixing for hyperspectral remote sensing imagery," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **7**(6), 1889–1909 (2014).
42. R. Feng, Y. Zhong, and L. Zhang, "An improved nonlocal sparse unmixing algorithm for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.* **12**(4), 915–919 (2015).
43. E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.* **14**(5), 877–905 (2007).
44. Z. J. Zhang, Y. Hui, and C. S. Wang, "A non-local means based classification method of hyperspectral image," in *Int. Conf. Optoelectron. and Microelectron. (ICOM)* (2015).
45. J. Wang et al., "Adaptive nonlocal spatial–spectral kernel for hyperspectral imagery classification," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **9**(9), 1–16 (2016).
46. A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit. (CVPR'05)*, IEEE, Vol. 2, pp. 60–65 (2005).
47. L. Sun et al., "Weighted nonlocal low-rank tensor decomposition method for sparse unmixing of hyperspectral images," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **13**, 1174–1188 (2020).
48. 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.* **52**(7), 4404–4423 (2014).
49. Z. Wu et al., "Real-time implementation of the sparse multinomial logistic regression for hyperspectral image classification on gpus," *IEEE Geosci. Remote Sens. Lett.* **12**(7), 1456–1460 (2015).
50. Z. Wu et al., "GPU parallel implementation of spatially adaptive hyperspectral image classification," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **11**(4), 1–13 (2017).
51. X. Wang et al., "Spatial group sparsity regularized nonnegative matrix factorization for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.* **55**(11), 6287–6304 (2017).

Yongxin Li received his BEng degree in computer science and technology from Xinke College of Henan Institute of Science and Technology, Xinxiang, China, in 2018. He is currently pursuing his MS degree at the School of Computer Science and Engineering, North Minzu University, Yinchuan, China. His research interests include hyperspectral image processing.

Wenxing Bao received his BEng degree in industrial automation from Xidian University, Xi'an, China, in 1993, his MSc degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2001, and his PhD in electronic science and technology from Xi'an Jiaotong University, Xi'an, China, in 2006. He is currently a professor. His current research interests include digital image processing, remote sensing image classification, and fusion.

Kewen Qu received his MS degree in computer application technology from North Minzu University, Yinchuan, China, in 2011. He is currently pursuing a PhD at the School of Computers and Information, Hefei University of Technology, Hefei, China. His research interests include remote sensing image processing, statistical learning, and deep learning.

Xiangfei Shen received his BS and MS degrees in computer science and technology from North Minzu University, Yinchuan, China, in 2017 and 2020, respectively. His research interests include hyperspectral image processing, pattern recognition, and machine learning.