Hyperspectral Image Denoising Using Group Low-Rank and Spatial-Spectral Total Variation

TANER INCE [ID], (Member, IEEE)
Electrical and Electronics Engineering Department, University of Gaziantep, 27310 Gaziantep, Turkey

e-mail: tanerince@gantep.edu.tr

ABSTRACT Hyperspectral images (HSIs) are frequently corrupted by various types of noise, such as Gaussian noise, impulse noise, stripes, and deadlines due to the atmospheric conditions or imperfect hyperspectral imaging sensors. These types of noise, which are also called mixed noise, severely degrade the HSI and limit the performance of post-processing operations, such as classification, unmixing, target recognition, and so on. The patch-based low-rank and sparse based approaches have shown their ability to remove these types of noise to some extent. In order to remove the mixed noise further, total variation (TV)-based methods are utilized to denoise HSI. In this paper, we propose a group low-rank and spatial-spectral TV (GLSSTV) to denoise HSI. Here, the advantage is twofold. First, group low-rank exploits the local similarity inside patches and non-local similarity between patches which brings extra structural information. Second, SSTV helps in removing Gaussian and sparse noise using the spatial and spectral smoothness of HSI. The extensive simulations show that GLSSTV is effective in removing mixed noise both quantitatively and qualitatively and it outperforms the state-of-the-art low-rank and TV-based methods.

INDEX TERMS Denoising, hyperspectral image (HSI), mixed noise, group low-rank, spatial-spectral total variation (SSTV).

I. INTRODUCTION

Hyperspectral imaging is the measurement of light spectrum over a large number of narrow wavelengths reflected from objects. High spectral resolution in HSI provides higher success rate in the identification of the substances compared to the classical imaging methods. Therefore, hyperspectral imaging is used in various fields of science such as remote sensing, astronomy, mineralogy and fluorescence microscopy. HSI suffers many types of noise that are introduced in imaging process due to several factors such as atmospheric effects and imperfect hyperspectral sensors. The types of noise which are frequently encountered are Gaussian noise, impulse noise, stripes, which are termed as mixed noise, seriously corrupts the hyperspectral data and affects the success of post-processing operations such as classification [1], unmixing [2] and target recognition [3].

The two-dimensional (2D) gray-scale image denoising methods such as K-SVD [4], BM3D [5], NCSR [6] or other

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

well-known methods can be utilized to denoise HSI by considering the each band of HSI as a 2D gray-scale image. However, band-wise denoising methods do not consider the spectral correlations between each band of the HSI, which is the most crucial property of HSI. In literature, many HSI denoising methods have been proposed [7]–[17]. In [7], Chen and Qian propose a principal component analysis (PCA) formulation to remove noise from low-energy PCA output channels which is believed to contain large amount of noise. [8] employs the sparse representation and low-rank penalty for redundancy and correlation (RAC). The sparse representation captures the local RAC in spectral domain and global RAC in spatial dimension and low-rank penalty captures the global RAC in spectral dimension. Lu *et al.* [9] propose a spectral-spatial adaptive sparse representation which explores the correlated spectral and spatial information. Besides this, low-rank approximation based methods which employ robust principal component analysis (RPCA) have been proposed for HSI denoising [18], [19]. The RPCA has attracted great attention due to the success of recovering the subspace structures possibly with outliers. It is used in many diverse

applications such as image analysis, web data ranking and computer vision. In the RPCA model, the lowest rank representation of data among all possible solutions is obtained and outliers are simultaneously removed by introducing a sparse error matrix to the model. Since HSI has a low-rank structure and it inevitably contains noise in the form of sparse and Gaussian noise, RPCA based methods are utilized in many studies for HSI denoising and restoration [10], [12], [14], [15]. Zhang *et al.* [10] divide the HSI into overlapping patches and solve low-rank matrix recovery (LRMR) for each patch and aggregates the patches to obtain the clean HSI. LRMR adopts the Go Decomposition (GoDec) algorithm [20] to obtain the low-rank approximation of the patches. The noise-adjusted iterative low-rank matrix approximation (NAILRMA) for Gaussian noise and noise-adjusted iterative low-rank matrix recovery (NAILRMR) for mixed noise are proposed in [12] that considers the noise level in different bands of the HSI. Furthermore, nonconvex low-rank approximation based denoising methods are also considered in the literature. The weighted Schatten *p*-norm low-rank matrix approximation (WSN-LRMA) [14] utilizes an iterative approach to restore the HSI. Recently, nonconvex low-rank matrix approximation (NonLRMA) [15] is proposed for HSI denoising which approximates the rank of HSI in iterative manner.

Besides this, in order to increase the success of denoising results, spatial and spectral information are investigated simultaneously in various studies. The low-rank spectral non-local approach is used in [11] which includes the low-rank representation of precleaned image patches and the application of spectral non-local method to restore the image. Reference [16] proposes a noise reduction algorithm using hybrid spatial-spectral wavelet shrinkage that works in spectral derivative domain and considers the dissimilarity in spatial and spectral dimensions and the algorithm is applied to denoise HSI. Furthermore, group low-rank representation (GLRR) is proposed in [21] to exploit the local similarity in a patch and non-local similarity across patches in a group simultaneously. GLRR which uses group low-rank and sparse based denoising scheme denoises each group individually using LRMR. The low-rank representation with spectral difference space (LRRSDS) [22] denoises HSI by utilizing the LRR of the spectral difference image. Fan *et al.* [23] propose a denoising method based on superpixel segmentation and low-rank representation (SS-LRR) to capture the spatial information more effectively.

Furthermore, total variation [24], which is an effective method in gray-scale image denoising, is used in many studies to remove noise from HSI [13], [25]–[29]. Yuan *et al.* propose a spatial-spectral adaptive TV (SSAHTV) model by considering the smoothness in spatial and spectral views to improve the denoising result [25]. Reference [13] employs a TV regularized low-rank matrix approximation (LRTV) to restore the HSI in which TV is applied in spatial dimensions. Spatial-spectral TV (SSTV) model is introduced in [26] to

denoise in spatial and spectral dimensions. The low-rank constraint SSTV (LSSTV) denoising method is considered in [29] to utilize the spatial and spectral smoothness and spectral correlation, simultaneously which is an effective method in removing the mixed noise from HSI. The spatial-spectral total variation regularized local low-rank matrix recovery (LLRSSTV) is proposed in [27]. LLRSSTV utilizes patch based low-rank approximation and SSTV simultaneously to remove the mixed noise efficiently. Reference [28] applies weighted total variation regularized low-rank model (LRWTV) for HSI restoration to preserve the spatial structure. The spectral-spatial weighted TV captures the spatial and spectral information of HSI to preserve the details. In [30], spectral difference-induced total variation and low-rank approximation (SDTVLA) is introduced to remove the structured noise such as structured stripes and deadlines. Sun *et. al* propose a cross TV regularized unidirectional low-rank tensor approximation (CrTVLRT) method [31] to explore the spectral-spatial correlation and non-local self-similarity simultaneously.

Patch based HSI denoising methods by arranging the 3D subcube to matrix form for removing mixed noise cannot exploit the full advantage of spatial and spectral information. Therefore, tensor based methods are proposed for HSI denoising [32], [33]. Huang *et al.* propose a group sparse and low-rank tensor decomposition (GSLRTD) method [32] to exploit the spatial and spectral information more effectively. First, 3D HSI is divided into overlapping 3D tensor cubes, then these cubes are clustered using *k*-means algorithm to form a group tensor. Each group tensor is denoised by SLRTD and clean HSI is obtained by aggregating all 3D cubes. Fan *et. al* propose a spatial-spectral TV regularized low-rank tensor factorization (SSTV-LRTF) method for mixed noise removal [33], which prevents the loss of multiway structural information in HSI.

Inspired by the GLRR [21] and SSTV based methods [25]–[29], [33] proposed in the literature, we propose a group low-rank approximation with spatial-spectral total variation (GLSSTV) method to denoise the HSI in the presence of mixed noise. Here, group low-rank representation exploits the local similarity within a patch and non-local similarity across patches, which brings extra structural information to help the reconstruction of corrupted patches and SSTV eliminates the noise further by utilizing the smoothness in spatial and spectral dimensions simultaneously. Basically, GLSSTV is related to aforementioned works. However, there are differences between GLSSTV and the others. GLRR [21] denoises each group individually to recover the corrupted patches but mixed noise cannot be removed completely. LLRSSTV [27], which is a combination of patch-based restoration and SSTV, do not consider the non-local similarity across patches. Furthermore, LSSTV [29] utilizes SSTV and low-rank approximation simultaneously without considering the patch-based denoising framework.

The simulated experiments on simulated and real datasets indicate that the proposed denoising scheme is effective

in removing mixed noise from HSI and outperforms the state-of-the-art methods proposed in the literature.

The rest of the paper is organized as follows. Section II gives notations and necessary background for problem formulation. Section III describes the proposed method. The simulated and real data experiments are given in Section IV. Finally, Section V concludes the paper. Some suggestions and future works are given in this section.

II. NOTATIONS AND BACKGROUND

Throughout the paper, we denote the hyperspectral cubes as Euler script letters, e.g., $\mathcal{A}$. Matrices are denotes as boldface capital letters e.g., $\mathbf{A}$, vectors are denoted as boldface lowercase letters, e.g., $\mathbf{a}$, and scalars are denoted by lowercase letters e.g. a. Some norms are used for tensors, matrix and vectors. Let $a_{i_1, i_2, \ldots, i_m}$ denote the $(i_1, i_2, \ldots, i_m)$-element of $\mathcal{A}$. We denote the l_1 norm as $\|\mathcal{A}\|_1 = \sum_{i_1, i_2, \ldots, i_m} |a_{i_1, i_2, \ldots, i_m}|$, the Frobenius norm as $\|\mathcal{A}\|_F = (\sum_{i_1, i_2, \ldots, i_m} |a_{i_1, i_2, \ldots, i_m}|^2)^{1/2}$. These norms reduces to matrix or vector norms if $\mathcal{A}$ is a matrix or vector. The nuclear norm of $\mathbf{A}$ is denoted as $\|\mathbf{A}\|_*$ which is defined as $\|\mathbf{A}\|_* = \sum_i \sigma_i$ where σ_i is the ith singular value of $\mathbf{A}$.

A. PATCH BASED LOW-RANK MATRIX RECOVERY MODEL

Hyperspectral images inevitably contain noise in the form of additive and sparse noise which includes Gaussian noise, impulse noise, stripes and deadlines. Suppose that the observation model for HSI is given as

$$\mathcal{Y} = \mathcal{X} + \mathcal{S} + \mathcal{N} \tag{1}$$

where $\mathcal{Y}$, $\mathcal{X}$, $\mathcal{S}$ and $\mathcal{N}$ denote the noisy HSI, clean HSI, sparse noise and Gaussian noise, respectively. The size of each term is $m \times n \times p$. It is known that there exist high spectral correlation between the bands of HSI, (RPCA) [34], [35] can be utilized in the patch-based denoising framework [10], [12]. The observation model for a subcube of size $M \times M \times p$ centered at location (i, j) can be written as

$$\mathcal{Y}_{(i,j)} = \mathcal{X}_{(i,j)} + \mathcal{S}_{(i,j)} + \mathcal{N}_{(i,j)} \tag{2}$$

then the observation model is converted to a 2D Casorati matrix form which means that each band of subcube is converted to $M^2 \times 1$ vector and then each vector is stacked column-wise to obtain the Casorati matrix. The observation model for subcube centered at location (i, j) in matrix form can be written as

$$\mathbf{Y}_{(i,j)} = \mathbf{X}_{(i,j)} + \mathbf{S}_{(i,j)} + \mathbf{N}_{(i,j)} \tag{3}$$

where $\mathbf{Y}_{(i,j)}$, $\mathbf{X}_{(i,j)}$, $\mathbf{S}_{(i,j)}$, and $\mathbf{N}_{(i,j)}$ are the Casorati matrices of $\mathcal{Y}_{(i,j)}$, $\mathcal{X}_{(i,j)}$, $\mathcal{S}_{(i,j)}$ and $\mathcal{N}_{(i,j)}$, respectively. The size of each term is $M^2 \times p$. The rank-constrained RPCA formulation is proposed to obtain clean HSI $\mathbf{X}$, which is formulated as:

$$\min_{\mathbf{X}_{(i,j)}, \mathbf{S}_{(i,j)}} \|\mathbf{X}_{(i,j)}\|_* + \lambda \|\mathbf{S}_{(i,j)}\|_1 \tag{4}$$

$$s.t. , \ \|\mathbf{Y}_{(i,j)} - \mathbf{X}_{(i,j)} - \mathbf{S}_{(i,j)}\|_F^2 \leq \epsilon$$
$$\text{rank}(\mathbf{X}_{(i,j)}) \leq r$$

where λ is the regularization parameter and r is defined as the upper rank value of $\mathbf{X}_{(i,j)}$. After all patches are restored, patches are aggregated to obtain the clean HSI.

B. SPATIAL-SPECTRAL TOTAL VARIATION MODEL

The RPCA based denoising methods are effective in removing sparse noise from the data. However, Gaussian noise in the HSI data cannot be removed completely. Furthermore, if the sparse noise has a structured characteristics then sparse noise can be regarded as low-rank part which means that the sparse noise cannot be eliminated efficiently. Therefore, using the spatial and spectral smoothness property of HSI, TV is applied either band-by-band [13] or SSTV is applied in spatial and spectral dimensions simultaneously [26], [27], [29] to remove the Gaussian noise and sparse noise efficiently. SSTV [26] model is formulated as

$$\|\mathcal{X}\|_{\text{SSTV}} = \|D_h \mathcal{X} D\|_1 + \|D_v \mathcal{X} D\|_1 \tag{5}$$

D_h and D_v are horizontal and vertical two dimensional difference operators applied on spatial dimensions and D is the one dimensional difference operator applied on spectral dimension. SSTV is very effective in removing Gaussian and sparse noise from hyperspectral data. However, it does not utilize the spectral correlation between the bands of hyperspectral data. LSSTV [29] utilizes the spectral correlation and spatial and spectral smoothness of HSI by using the low-rank approximation and SSTV simultaneously. It has been shown that superior performance is achieved in removing the Gaussian and sparse noise. LLRSSTV [27] employs patch-based low-rank approximation and SSTV simultaneously.

III. PROPOSED METHOD

A. GROUP LOW-RANK APPROXIMATION WITH SPATIAL-SPECTRAL TOTAL VARIATION

We propose a novel denoising scheme that will employ group low-rank approximation and SSTV regularization simultaneously, which we call group low-rank approximation with SSTV regularization (GLSSTV). Group low-rank representation will provide the local similarity inside a patch and nonlocal similarity between patches. SSTV is utilized to remove the sparse noise and Gaussian noise by considering the spatial and spectral smoothness of HSI.

First, we define an operator $\mathcal{R}_{(q)}(\cdot)$ that extracts a subcube of size $M \times M \times p$ centered at location (i, j) and reshapes into a patch matrix of size $M^2 \times p$. The total number of patches is $K = \lfloor \frac{m-s}{s} \rfloor \times \lfloor \frac{n-s}{s} \rfloor$ where $\lfloor \star \rfloor$ rounds $\star$ down to the nearest integer. Therefore, a patch $\mathbf{Y}_{(q)}$ for $(q = 1, \ldots, K)$ can be represented as

$$\mathbf{Y}_{(q)} = \mathcal{R}_{(q)}(\mathcal{Y}) \quad q = 1, \ldots, K \tag{6}$$

Then, we define a similarity matrix $\mathbf{D}$ where each element of $\mathbf{D}$ is calculated based on Euclidean distance between patches such that

$$\mathbf{D}_{qv} = \|\mathbf{Y}_{(q)} - \mathbf{Y}_{(v)}\|_F \quad q, v = 1, \ldots, K \tag{7}$$

We sort each row of $\mathbf{D}$ in descending order and collect the (i, j) indices of the k patches that have the lowest Euclidean distance in each row of $\mathbf{D}$. The (i, j) indices of the k patches in each group are stored in sets $\Lambda_q = \{\Lambda_{q_1}, \ldots, \Lambda_{q_k}\}$ for $(q = 1, \ldots, K)$ which will be used in the reconstruction stage of the proposed formulation.

At this point, we define an operator $\mathcal{T}_{\Lambda_q}$ to create a group for each patch $\mathbf{Y}_{(q)}$ for $(q = 1, \ldots, K)$ such that

$$\mathcal{T}_{\Lambda_q}(\mathcal{Y}) = \begin{bmatrix} \mathbf{Y}_{(\Lambda_{q_1})} \\ \mathbf{Y}_{(\Lambda_{q_2})} \\ \vdots \\ \mathbf{Y}_{(\Lambda_{q_k})} \end{bmatrix}_{kM^2 \times p} \quad q = 1, \ldots, K \quad (8)$$

Basically, $\mathcal{T}_{\Lambda_q}$ is a two-step operator. First, it extracts K subcubes of size $M \times M \times p$ centered at location (i, j) and then convert them to Casorati matrix form using operator $\mathcal{R}_{(q)}(\cdot)$. Second, K groups are created using the indices provided by Λ_q for $(q = 1, \ldots, K)$.

Based on operator $\mathcal{T}_{\Lambda_q}$, our proposed rank constrained group low-rank and SSTV model for HSI denoising as follows.

$$\min_{\mathcal{X}, \mathcal{S}} \sum_{q=1}^{K} \left(\|\mathcal{T}_{\Lambda_q}(\mathcal{X})\|_* \right) + \lambda \|\mathcal{S}\|_1 + \tau \|\mathcal{X}\|_{\text{SSTV}}$$
$$+ \beta \|\mathcal{Y} - \mathcal{X} - \mathcal{S}\|_F^2$$
$$s.t. \operatorname{rank}(\mathcal{T}_{\Lambda_q}(\mathcal{X})) \leq r; \quad q = 1, \ldots, K \quad (9)$$

Here, $\sum_{q=1}^{K} \left(\|\mathcal{T}_{\Lambda_q}(\mathcal{X})\|_* \right)$ calculates the group nuclear norm for each group, $\lambda \|\mathcal{S}\|_1$ accounts for the sparse noise in the data, $\tau \|\mathcal{X}\|_{\text{SSTV}}$ considers the spatial and spectral smoothness of the HSI and $\beta \|\mathcal{Y} - \mathcal{X} - \mathcal{S}\|_F^2$ balances the tradeoff between noisy and clean data. Replacing the SSTV model in (5) to (9) leads to following optimization problem.

$$\min_{\mathcal{X}, \mathcal{S}} \sum_{q=1}^{K} \left(\|\mathcal{T}_{\Lambda_q}(\mathcal{X})\|_* \right) + \lambda \|\mathcal{S}\|_1 + + \beta \|\mathcal{Y} - \mathcal{X} - \mathcal{S}\|_F^2$$
$$+ \tau \|D_h \mathcal{X} D\|_1 + \tau \|D_v \mathcal{X} D\|_1$$
$$s.t. \operatorname{rank}(\mathcal{T}_{\Lambda_q}(\mathcal{X})) \leq r; \quad q = 1, \ldots, K \quad (10)$$

We employ Alternating Direction Method of Multipliers (ADMM) [36] to solve (10). By including auxiliary variables, we can write

$$\min_{\mathcal{J}, \mathcal{X}, \mathcal{S}} \sum_{q=1}^{K} \left(\|\mathcal{T}_{\Lambda_q}(\mathcal{J})\|_* \right) + \beta \|\mathcal{Y} - \mathcal{X} - \mathcal{S}\|_F^2$$
$$+ \lambda \|\mathcal{S}\|_1 + \tau \|\mathcal{P}\|_1 + \tau \|\mathcal{Q}\|_1$$
$$+ \mu \|\mathcal{P} - D_h \mathcal{X} D - \mathcal{B}_1\|_F^2$$
$$+ \mu \|\mathcal{Q} - D_v \mathcal{X} D - \mathcal{B}_2\|_F^2 + \mu \|\mathcal{J} - \mathcal{X} - \mathcal{B}_3\|_F^2$$
$$s.t. \operatorname{rank}(\mathcal{T}_{\Lambda_q}(\mathcal{J})) \leq r; \quad q = 1, \ldots, K \quad (11)$$

where $\mathcal{B}_1$, $\mathcal{B}_2$ and $\mathcal{B}_3$ are Lagrangian multipliers and $\mathcal{J}$, $\mathcal{P}$ and $\mathcal{Q}$ are the auxiliary variables to decouple the variables. We can write the individual problems for decoupled variables as:

$$\mathcal{J}: \quad \underset{\operatorname{rank}(\mathcal{T}_{\Lambda_q}(\mathcal{J})) \leq r}{\arg\min} \sum_{q=1}^{K} \|\mathcal{T}_{\Lambda_q}(\mathcal{J})\|_* \quad (12)$$
$$+ \mu \|\mathcal{J} - \mathcal{X} - \mathcal{B}_3\|_F^2$$

$$\mathcal{P}: \quad \underset{\mathcal{P}}{\arg\min} \, \tau \|\mathcal{P}\|_1 + \mu \|\mathcal{P} - D_h \mathcal{X} D - \mathcal{B}_1\|_F^2 \quad (13)$$

$$\mathcal{Q}: \quad \underset{\mathcal{Q}}{\arg\min} \, \tau \|\mathcal{Q}\|_1 + \mu \|\mathcal{Q} - D_v \mathcal{X} D - \mathcal{B}_2\|_F^2 \quad (14)$$

$$\mathcal{S}: \quad \underset{\mathcal{S}}{\arg\min} \, \lambda \|\mathcal{S}\|_1 + \beta \|\mathcal{Y} - \mathcal{X} - \mathcal{S}\|_F^2 \quad (15)$$

$$\mathcal{X}: \quad \underset{\mathcal{X}}{\arg\min} \, \beta \|\mathcal{Y} - \mathcal{X} - \mathcal{S}\|_F^2 + \mu \|\mathcal{J} - \mathcal{X} - \mathcal{B}_3\|_F^2$$
$$+ \mu \|\mathcal{P} - D_h \mathcal{X} D - \mathcal{B}_1\|_F^2 + \mu \|\mathcal{Q} - D_v \mathcal{X} D - \mathcal{B}_2\|_F^2 \quad (16)$$

For problem (12), the optimization problem can be divided into K independent optimization problems. We can write

$$\underset{\operatorname{rank}(\mathcal{T}_{\Lambda_q}(\mathcal{J})) \leq r}{\arg\min} \|\mathcal{T}_{\Lambda_q}(\mathcal{J})\|_* + \mu \|\mathcal{T}_{\Lambda_q}(\mathcal{J})$$
$$- \mathcal{T}_{\Lambda_q}(\mathcal{X} + \mathcal{B}_3)\|_F^2 \quad (17)$$

for each group $(q = 1, \ldots, K)$.

Solution to (17) can be obtained using the following lemma.

Lemma 1 [37]: Consider the singular value decomposition (SVD) of matrix $\mathbf{Q} \in \mathbb{R}^{mn \times p}$ of rank r

$$\mathbf{Q} = \mathbf{U} \Sigma \mathbf{V}^*, \quad \Sigma = \operatorname{diag}(\{\sigma_i\}_{1 \leq i \leq r}) \quad (18)$$

where $\mathbf{U}$ and $\mathbf{V}$ are $mn \times r$ and $p \times r$ matrices. Then singular value shrinkage operator obeys

$$\mathcal{D}_\rho(\mathbf{R}) = \arg\min_{\mathbf{Q}} \frac{1}{2} \|\mathbf{Q} - \mathbf{R}\|_F^2 + \rho \|\mathbf{Q}\|_* \quad (19)$$

where $\mathcal{D}_\rho(\mathbf{R}) = \mathbf{U} \mathcal{D}_\rho(\Sigma) \mathbf{V}^*$ and $\mathcal{D}_\rho$ is defined as $\mathcal{D}_\rho = \operatorname{diag}(\{\sigma_i - \rho\}_+)$

Therefore, the solution to (17) for each group can be obtained directly using Lemma 1.

$$\mathcal{T}_{\Lambda_q}(\mathcal{J}) = \mathcal{D}_{1/2\mu}(\mathcal{T}_{\Lambda_q}(\mathcal{X} + \mathcal{B}_3)) \quad (20)$$

In order to take the consideration of $\operatorname{rank}(\mathcal{T}_{\Lambda_q}(\mathcal{J})) \leq r$, we set $\sigma(i) = 0$ for $i > r$ for each group.

Then, we reconstruct $\mathcal{J}$ from K groups using the indices provided by Λ_q such that

$$\mathcal{J} = \mathcal{T}_{\Lambda_q}^T(\mathcal{T}_{\Lambda_q}(\mathcal{J})) \quad (21)$$

where $\mathcal{T}_{\Lambda_q}^T(\cdot)$ is inverse operator of $\mathcal{T}_{\Lambda_q}(\cdot)$.

$\mathcal{T}_{\Lambda_q}^T(\cdot)$ is a two-step averaging operation. We know that $\mathcal{T}_{\Lambda_q}(\mathcal{J})$ creates groups for each patch $\mathbf{J}_{(q)}$ such that

$$\mathcal{T}_{\Lambda_q}(\mathcal{J}) = \begin{bmatrix} \mathbf{J}_{(\Lambda_{q_1})} \\ \mathbf{J}_{(\Lambda_{q_2})} \\ \vdots \\ \mathbf{J}_{(\Lambda_{q_k})} \end{bmatrix}_{kM^2 \times p} \quad q = 1, \ldots, K \quad (22)$$

and also a patch $\mathbf{J}_{(q)}$ in a group can be included in C groups, there will be multiple patches for $\mathbf{J}_{(q)}$ denoted as $\{\mathbf{J}_{(q)}^{(l_1)}, \ldots, \mathbf{J}_{(q)}^{(l_C)}\}$ where $(l_1, \ldots, l_C)$ is the group indices for $\mathbf{J}_{(q)}$.

In the first step, we average the patches with the same indices using

$$\mathbf{J}_{(q)} = \frac{1}{C} \sum_{t=1}^{C} \mathbf{J}_{(q)}^{(l_t)} \qquad q = 1, \ldots, K \qquad (23)$$

In the second step, $\mathcal{J}$ is calculated by averaging the overlapped patches as

$$\mathcal{J} = \sum_{q=1}^{K} \mathcal{R}_{(q)}^{T}(\mathbf{J}_{(q)}). / \left(\sum_{q=1}^{K} \mathcal{R}_{(q)}^{T} \mathcal{R}_{(q)} \right) \qquad (24)$$

where $\mathcal{R}_{(q)}^{T}(\cdot)$ is inverse operator of $\mathcal{R}_{(q)}(\cdot)$ and it converts $\mathbf{J}_{(q)}$ to subcube $\mathcal{J}_{(q)}$ and forms an $m \times n \times p$ image where the values of $\mathcal{J}_{(q)}$ are put in the appropriate position in $m \times n \times p$ image and the rest of the image is set to zero. $\sum_{q=1}^{K} \mathcal{R}_{(q)}^{T} \mathcal{R}_{(q)}$ is an $m \times n \times p$ matrix that averages the overlapped subcubes formed by $\sum_{q=1}^{K} \mathcal{R}_{(q)}^{T}(\mathbf{J}_{(q)})$.

The solutions to subproblems (13), (14) and (15) can be obtained as

$$\mathcal{P} = \mathfrak{R}_{\tau/2\mu}(D_h \mathcal{X} D + \mathcal{B}_1) \qquad (25)$$

$$\mathcal{Q} = \mathfrak{R}_{\tau/2\mu}(D_v \mathcal{X} D + \mathcal{B}_2) \qquad (26)$$

$$\mathcal{S} = \mathfrak{R}_{\lambda/2\beta}(\mathcal{Y} - \mathcal{X}) \qquad (27)$$

where the soft-thresholding operator $\mathfrak{R}(\cdot)$ is defined as:

$$\mathfrak{R}_{\tau/2\mu}(o) = \begin{cases} o - \tau/2\mu, & \text{if } o > \tau/2\mu \\ o + \tau/2\mu, & \text{if } o < \tau/2\mu. \end{cases} \qquad (28)$$

Next, we obtain the vectorized form of subproblem (16) to solve for $\mathcal{X}$

$$x : \arg\min_{x} \beta \| y - x - s \|_2^2 + \mu \| j - x - b_3 \|_2^2$$
$$+ \mu \| p - \nabla_h x - b_1 \|_2^2 + \mu \| q - \nabla_v x - b_2 \|_2^2 \qquad (29)$$

where $\nabla_h = D^T \otimes D_h$ and $\nabla_v = D^T \otimes D_v$. Here, we use the property of Kronecker product $D_h \mathcal{X} D = (D^T \otimes D_h)x$ and $D_v \mathcal{X} D = (D^T \otimes D_v)x$.

Differentiating (29) with respect to x leads to following linear system of equations

$$(\beta + \mu + \mu \nabla_h^T \nabla_h + \mu \nabla_v^T \nabla_v)x$$
$$= \beta(y - s) + \mu(j - b_3) + \mu(p - b_1) + \mu(q - b_2) \qquad (30)$$

which can be solved using LSQR [38]. Then, $\mathcal{X}$ can be obtained by converting vectorized form of x to the tensor form.

The Lagrange multipliers $\mathcal{B}_1$, $\mathcal{B}_2$ and $\mathcal{B}_3$ are updated as

$$\mathcal{B}_1 = \mathcal{B}_1 + D_h \mathcal{X} D - \mathcal{P} \qquad (31)$$

$$\mathcal{B}_2 = \mathcal{B}_2 + D_v \mathcal{X} D - \mathcal{Q} \qquad (32)$$

$$\mathcal{B}_3 = \mathcal{B}_3 + \mathcal{X} - \mathcal{J} \qquad (33)$$

We summarize the proposed GLSSTV in Algorithm 1.

Algorithm 1 Algorithm for GLSSTV

Input: $\mathcal{Y}, M, s, \lambda, \beta, \tau, \mu, r, \epsilon, k$
Output: $\mathcal{X}$

   Initialization: $\mathcal{X}^{(0)} = \mathcal{J}^{(0)} = \mathcal{S}^{(0)} = 0, \mathcal{P}^{(0)} = \mathcal{Q}^{(0)} = 0, \mathcal{B}_1^{(0)} = \mathcal{B}_2^{(0)} = \mathcal{B}_3^{(0)} = 0, x^{(0)} = 0$, maxiter = 50, $\epsilon = 10^{-5}$
1: **Step 1:** Get $\mathcal{T}_{\Lambda_q}$ for $(q = 1, \ldots, K)$ using (8).
2: **Step2:**
3: **for** $i = 1$ to maxiter **do**
4:     **for** $q = 1$ to K **do**
5:         $\mathcal{T}_{\Lambda_q}(\mathcal{J}^{(i)}) = \mathcal{D}_{1/2\mu}(\mathcal{T}_{\Lambda_q}(\mathcal{X}^{(i-1)} + \mathcal{B}_3^{(i-1)}))$
6:     **end for**
7:     $\mathcal{J}^{(i)} = \mathcal{T}_{\Lambda_q}^{T}(\mathcal{T}_{\Lambda_q}(\mathcal{J}^{(i)}))$
8:     $\mathcal{P}^{(i)} = \mathfrak{R}_{\tau/2\mu}(D_h \mathcal{X}^{(i-1)} D + \mathcal{B}_1^{(i-1)})$
9:     $\mathcal{Q}^{(i)} = \mathfrak{R}_{\tau/2\mu}(D_v \mathcal{X}^{(i-1)} D + \mathcal{B}_2^{(i-1)})$
10:     $\mathcal{S}^{(i)} = \mathfrak{R}_{\lambda/2\beta}(\mathcal{Y} - \mathcal{X}^{(i-1)})$
11:     Solve (30) for $x^{(i)}$ using LSQR [38]
12:     Arrange $x^{(i)}$ to obtain $\mathcal{X}^{(i)}$
13:     $\mathcal{B}_1^{(i)} = \mathcal{B}_1^{(i-1)} + D_h \mathcal{X}^{(i)} D - \mathcal{P}^{(i)}$
14:     $\mathcal{B}_2^{(i)} = \mathcal{B}_2^{(i-1)} + D_v \mathcal{X}^{(i)} D - \mathcal{Q}^{(i)}$
15:     $\mathcal{B}_3^{(i)} = \mathcal{B}_3^{(i-1)} + \mathcal{X}^{(i)} - \mathcal{J}^{(i)}$
16:     **if** $\| x^{(i)} - x^{(i-1)} \|_2 < \epsilon$ **then**
17:         **break**
18:     **end if**
19: **end for**
20: **return** $\mathcal{X}$

B. COMPUTATIONAL COMPLEXITY

The computational complexity of GLSSTV is based on SVD calculations, soft-thresholding operations and least square solution. If we select a step size of s and a block size of M for a HSI with dimensions $m \times n \times p$, there will be $K = \lfloor \frac{m-s}{s} \rfloor \times \lfloor \frac{n-s}{s} \rfloor$ groups and k patches in each group. The size of each group will be $kM^2 \times p$. At each iteration of GLSSTV, SVD calculation in (20) requires $\mathcal{O}(KkM^2p^2)$ flops assuming that $kM^2 > p$ and soft-thresholding operation needs $\mathcal{O}(KkM^2p)$. The soft-thresholding operators in (25), (26) and (27) require $\mathcal{O}(3mnp)$ flops and the least square solution in (30) requires $\mathcal{O}(3mn + 5p)$ flops using LSQR method [38]. Updating the Lagrange multipliers in (31), (32) and (33) needs $\mathcal{O}(6mnp)$ flops. Therefore, overall complexity of GLSSTV at each iteration is $\mathcal{O}(KkM^2p^2 + KkM^2p + 9mnp + 3mn + 5p)$. The main computational cost of GLSSTV comes from the SVD calculation for each group which dominates the computational complexity of GLSSTV. Moreover, the computational complexity of GLSSTV is higher compared to the computational complexities of the algorithms under comparison. We report the computation times of each algorithm in real data experiments in Section IV-H.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we perform several simulated data and real data experiments to verify the effectiveness of GLSSTV for HSI denoising. We compare GLSSTV with state-of-the-art HSI mixed denoising methods proposed in the literature.

We select nine HSI denoising algorithms. These are LRMR [10] , NAILRMR [12], GLRR [21], TVL1 [39], SSTV [26], LRTV [13], LSSTV [29], LLRSSTV [27]. LRMR and NAILRMR are well-known patch-based HSI denoising methods which utilizes RPCA. GLRR solves group low-rank RPCA. TVL1 uses 3D total variation to restore video sequences. LRTV is a TV regularized low-rank matrix factorization algorithm in which TV is applied each band of HSI individually. SSTV combines spatial and spectral total variation simultaneously. LSSTV combines SSTV with low-rank constraint. LLRSSTV denoises HSI using local low-rank patch-based RPCA with SSTV. The codes of LRMR, NAILRMR, LRTV, LLRSSTV, TVL1 and SSTV are provided by authors. We implemented the algorithms GLRR an LSSTV, since they are not available. The optimal parameters of each of the algorithm is adjusted as it is explained in their original papers.
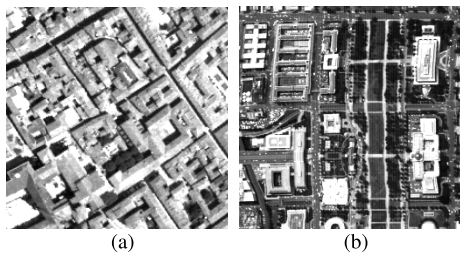
FIGURE 1. (a) Pavia city dataset (b) Washington DC Mall dataset.

A. EXPERIMENTS ON SIMULATED DATA

We use two simulated datasets in the simulated data experiments. The first one is Pavia city[1] which was collected by the reflective optics system imaging spectrometer (ROSIS-03). The size of this dataset is $1400 \times 512 \times 102$. As first bands of Pavia dataset are noisy, we removed the first 22 bands. We selected a subscene of size $200 \times 200 \times 80$ which is shown in Fig. 1(a). The second simulated dataset is Washington DC Mall[2] dataset which was collected by the hyperspectral digital imagery collection experiment (HYDICE) sensor. It has a size of $1208 \times 307 \times 191$. We selected a subscene of $256 \times 256 \times 191$ which is shown in Fig. 1(b). Before the simulations, the gray values of each band were normalized to the range of $[0, 1]$. After denoising process, the gray values of each band were stretched to original range. The peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [40] and the mean spectral angle distance (MSAD) are used to evaluate the quality of the denoising results of each algorithm. The mean values of PSNR and SSIM are denoted as MPSNR

[1]Pavia scenes were provided by Prof. Paolo Gamba from the Telecommunications and Remote Sensing Laboratory, Pavia university (Italy) which can be downloaded at http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.

[2]Available at Purdue University Research Repository (https://engineering.purdue.edu/biehl/MultiSpec/hyperspectral.html)

and MSSIM. These metrics are calculated as:

$$\text{PSNR}_l(\hat{\mathcal{X}}_l, \mathcal{X}_l) = 10 \log \frac{mn}{\sum_{i=1}^{m} \sum_{j=1}^{n} [\hat{\mathcal{X}}_l(i,j) - \mathcal{X}_l(i,j)]^2}$$

$$\text{SSIM}_l(\hat{\mathcal{X}}_l, \mathcal{X}_l) = \frac{(2\mu_{\mathcal{X}_l}\mu_{\hat{\mathcal{X}}_l} + C_1)(2\sigma_{\hat{\mathcal{X}}_l\mathcal{X}_l} + C_2)}{(\mu_{\mathcal{X}_l}^2 + \mu_{\hat{\mathcal{X}}_l}^2 + C_1)(\sigma_{\mathcal{X}_l}^2 + \sigma_{\hat{\mathcal{X}}_l}^2 + C_2)}$$

$$\text{MPSNR} = \frac{1}{p} \sum_{l=1}^{p} \text{PSNR}_l(\hat{\mathcal{X}}_l, \mathcal{X}_l)$$

$$\text{MSSIM} = \frac{1}{p} \sum_{l=1}^{p} \text{SSIM}_l(\hat{\mathcal{X}}_l, \mathcal{X}_l) \quad (34)$$

Here, $\mathcal{X}_l$ and $\hat{\mathcal{X}}_l$ denotes the original and restored HSI in lth band. $\mu_{\mathcal{X}_l}$ and $\mu_{\hat{\mathcal{X}}_l}$ are the mean intensity values of $\mathcal{X}_l$ and $\hat{\mathcal{X}}_l$. $\sigma_{\mathcal{X}_l}^2$ and $\sigma_{\hat{\mathcal{X}}_l}^2$ are the variances of $\mathcal{X}_l$ and $\hat{\mathcal{X}}_l$, respectively. $\sigma_{\hat{\mathcal{X}}_l \mathcal{X}_l}$ is the covariance between $\mathcal{X}_l$ and $\hat{\mathcal{X}}_l$. MSAD is calculated as

$$\text{MSAD} = \frac{1}{mn} \sum_{i=1}^{mn} \frac{180}{\pi} \times \arccos \frac{(\mathcal{X}^i)^T . \hat{\mathcal{X}}^i}{\|\mathcal{X}^i\| . \|\hat{\mathcal{X}}^i\|} \quad (35)$$

where $\mathcal{X}^i$ and $\hat{\mathcal{X}}^i$ denote the ith spectral signatures of clean and denoised HSI, respectively.

In simulations, we added three types of noise to Pavia city dataset and Washington DC Mall dataset.

1) Zero-mean Gaussian noise was added to all bands. Each band has different noise intensities. The standard deviation of the Gaussian noise of each band is selected randomly from 0 to 0.2. The mean SNR value of all bands for Pavia city and Washington DC Mall are 5.66 dB and 7.05 dB, respectively.
2) Impulse noise was added to all bands. The percentage of impulse noise is selected randomly from 0 to 0.2
3) Stripes were simulated on 30% of the bands which were selected randomly. The number of stripes of each selected band ranges from 3 to 15 lines.

B. PARAMETER DETERMINATION

The parameters of the GLSSTV need to be carefully tuned to obtain good denoising results. The parameter λ controls the sparsity of the sparse noise $\mathcal{S}$, τ adjusts the spatial-spectral smoothness of the reconstructed $\mathcal{X}$. A large value of τ will oversmooth the image and too small value of τ will not exploit the spatial and spectral smoothness of HSI and the noise cannot be removed efficiently. β controls the quality of the reconstruction between noisy image $\mathcal{Y}$ and reconstructed image $\mathcal{X}$. μ is the penalty parameter that effects the convergence of the algorithm. In the experiments, we fixed the block size and step size as $M = 20$ and $s = 10$. The rank values for Pavia dataset and Washington DC Mall dataset are fixed to $r = 3$ and $r = 5$, respectively.

In the first experiment, we analyze the influence of the parameters λ and τ when other parameters of GLSSTV are fixed as $\beta = 5$ and $\mu = 1$. GLSSTV is solved by varying the parameters λ and τ between 0.1 and 0.6 in steps of 0.1.
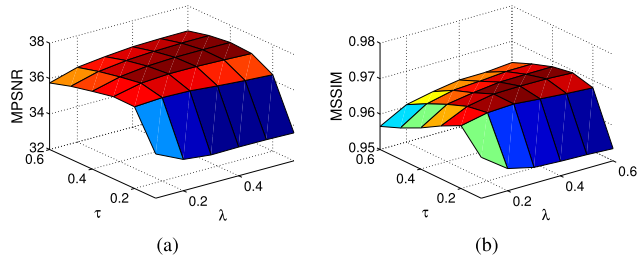
FIGURE 2. MPSNR and MSSIM values of GLSSTV for Pavia city subscene by varying parameters λ and τ when $\beta = 5$ and $k = 4$. (a) MPSNR. (b) MSSIM.
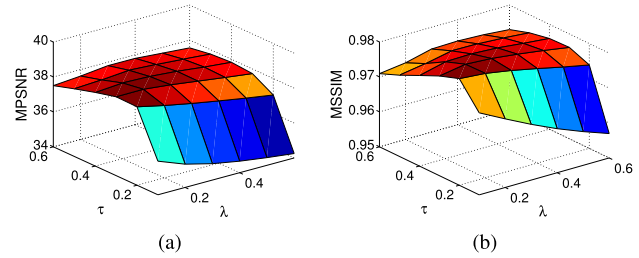
FIGURE 3. MPSNR and MSSIM values of GLSSTV for Washington DC Mall subscene by varying parameters λ and τ when $\beta = 5$ and $k = 4$. (a) MPSNR. (b) MSSIM.
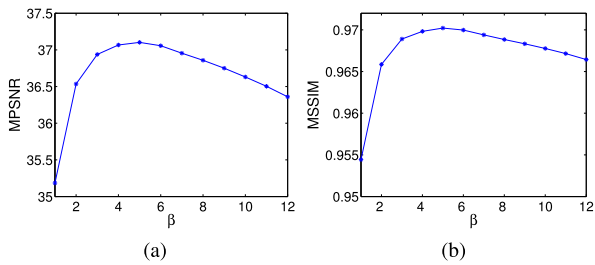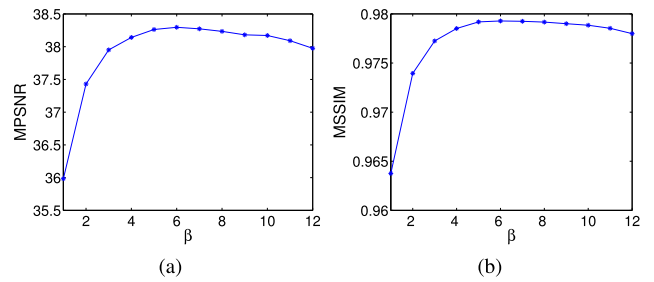

FIGURE 4. MPSNR and MSSIM values of GLSSTV for Pavia city subscene by varying parameter β when $\lambda = 0.3$, $\tau = 0.4$ and $k = 4$. (a) MPSNR. (b) MSSIM.

FIGURE 5. MPSNR and MSSIM values of GLSSTV for Washington DC Mall subscene by varying parameter β when $\lambda = 0.3$, $\tau = 0.4$ and $k = 4$. (a) MPSNR. (b) MSSIM.
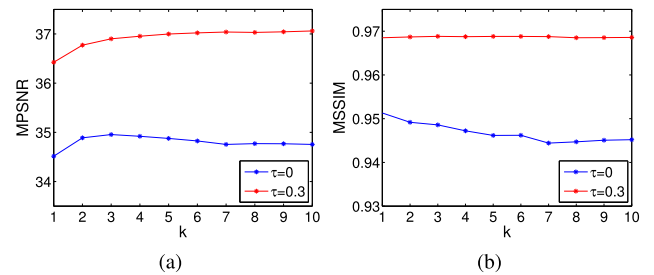
FIGURE 6. MPSNR and MSSIM values of GLSSTV for Pavia city subscene by varying parameter k. (a) MPSNR. (b) MSSIM.
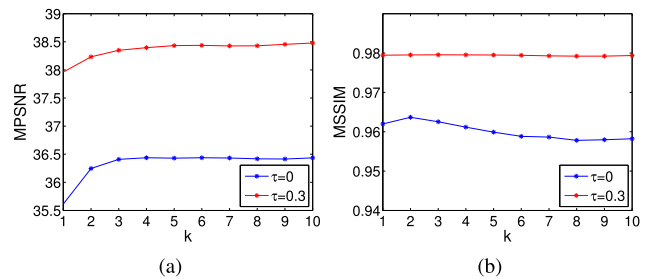
FIGURE 7. MPSNR and MSSIM values of GLSSTV for Washington DC Mall subscene by varying parameter k. (a) MPSNR. (b) MSSIM.

FIGURE 8. MPSNR versus iteration number for GLSSTV with the two datasets in the simulated experiments. (a) The Pavia city subscene. (b) The Washington DC Mall subscene.

Figs. 2 and 3 show the MPSNR and MSSIM plots of Pavia dataset and Washington DC Mall dataset, respectively. We can see that $\lambda = 0.3$ and $\tau = 0.4$ gives good MPSNR and MSSIM values for both datasets.

In the second experiment, we analyze the effect of β. In this case, we fixed the parameters as $\lambda = 0.3$ and $\tau = 0.4$. We solve GLSSTV for β values between 1 and 12 in steps of 1. Figs. 4 and 5 show the MPSNR and MSSIM plots. We can observe that $\beta = 5$ gives maximum MSPNR and MSSIM values for both datasets. When β is greater than 5, MPSNR and MSSIM values tend to decrease for both datasets.

## C. EFFECT OF NON-LOCAL LOW-RANK AND SSTV REGULARIZATION

GLSSTV includes SSTV regularization and non-local low-rank regularization. We perform experiments to see the effect of SSTV and non-local low-rank regularization separately. First, we solve GLSSTV by varying k from 1 to 10 and setting
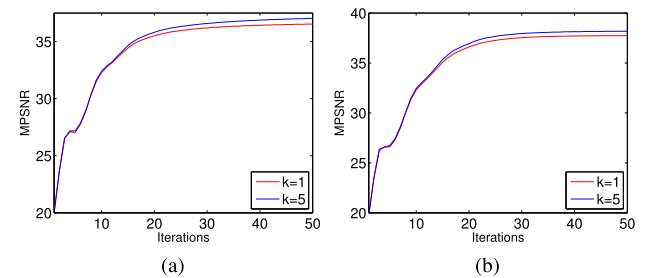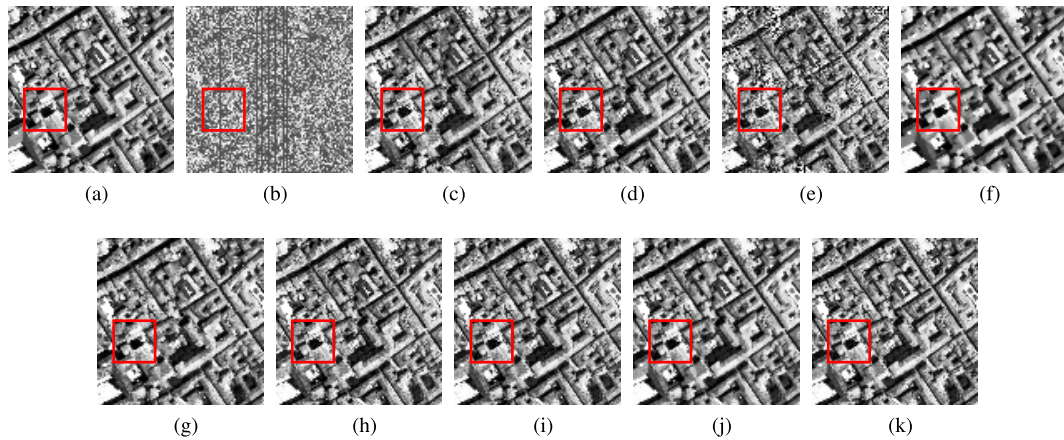
$\tau = 0$ when all other parameters are fixed. This means that SSTV regularization is removed in the formulation of GLSSTV. Second, in order to see the effect of SSTV regularization, GLSSTV is solved for $\tau = 0.3$ by varying k from 1 to 10. Figs. 6 and 7 show the MPSNR and MSSIM plots

FIGURE 9. Denoising results of band 15 of Pavia city subscene using different denoising methods. (a) Original image. (b) Noisy image (11.04 dB). The denoising results of (c) LRMR (29.75 dB). (d) GLRR (31.23 dB). (e) NAILRMR (25.47 dB). (f) LRTV (28.18 dB). (g) TVL1 (33.56 dB). (h) SSTV (33.46 dB). (i) LSSTV (35.47 dB). (j) LLRSSTV (32.64 dB). (k) GLSSTV (37.01 dB).
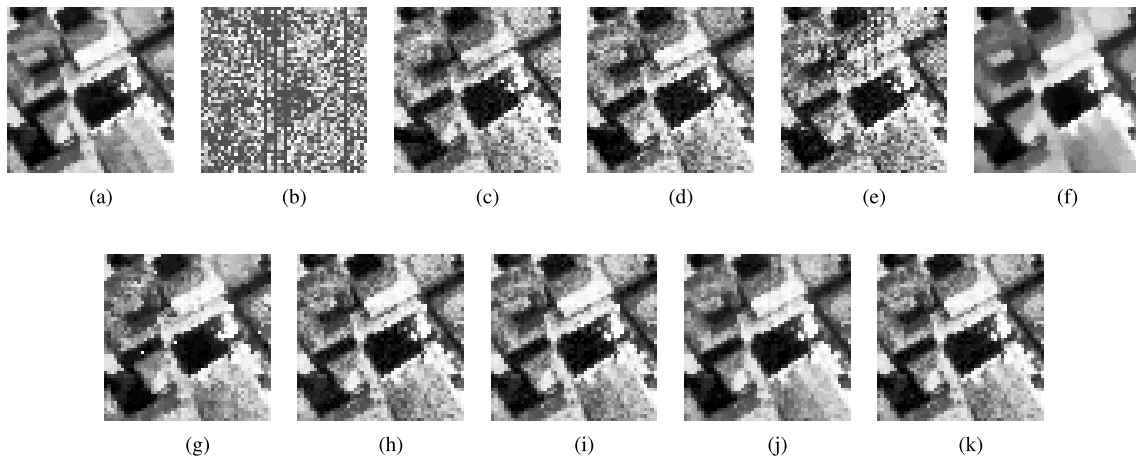
FIGURE 10. Magnified results of the region marked with red rectangle in Fig. 9. (a) Original. (b) Noisy. (c) LRMR. (d) GLRR. (e) NAILRMR. (f) LRTV. (g) TVL1. (h) SSTV. (i) LSSTV. (j) LLRSSTV. (k) GLSSTV.

when k is varied for $\tau = 0$ and $\tau = 0.3$. We can observe that when k increases, MPSNR and MSSIM decrease after some k value for $\tau = 0$. This is the expected result since the similarity within a group may not be guaranteed when k is large. However, when SSTV regularization is included in GLSSTV formulation with $\tau = 0.3$, MPSNR increases and remains almost constant with increasing values of k. Also MSSIM increase slightly and remain almost constant for increasing values of k. Therefore, we can select $k = 4$ for GLSSTV to obtain good denoising results. In order to prove the convergence of the proposed GLSSTV algorithm, we give the MPSNR results versus iterations for Pavia and Washington DC Mall dataset in Fig. 8, respectively. It can be observed that after 40 iterations GLSSTV become stable.

D. VISUAL COMPARISON

Fig. 9 shows the denoising results of band 15 of Pavia city subcene and magnified results of the area marked with red

rectangle in Fig. 9 is shown in Fig. 10. This band is corrupted by three types of noise including Gaussian noise, impulse noise and stripes. It can be observed from Fig. 10 that LRMR and GLRR are able to remove the sparse noise however NAILRMR fails to remove stripes. TVL1 is also not able to remove mixed noise. LRTV removes mixed noise but it smooths the details in the image. LSSTV performs better than SSTV as it can be seen from Figs. 10(h) and 10(i). LLRSSTV removes sparse and Gaussian noise perfectly but stripes are not removed very well. The results of GLSSTV is best in this Pavia subscene which is also validated by the PSNR results given in caption of Fig. 9.

The denoising results of the algorithms for band 6 of Washington DC Mall subscene are shown in Fig. 11 and the magnified results of the region marked with red rectangle in Fig. 11 are shown in Fig. 12. Visually, LRMR, NAILRMR and GLRR are able to remove the mixed noise. LRTV removes mixed noise whereas it smooths the details
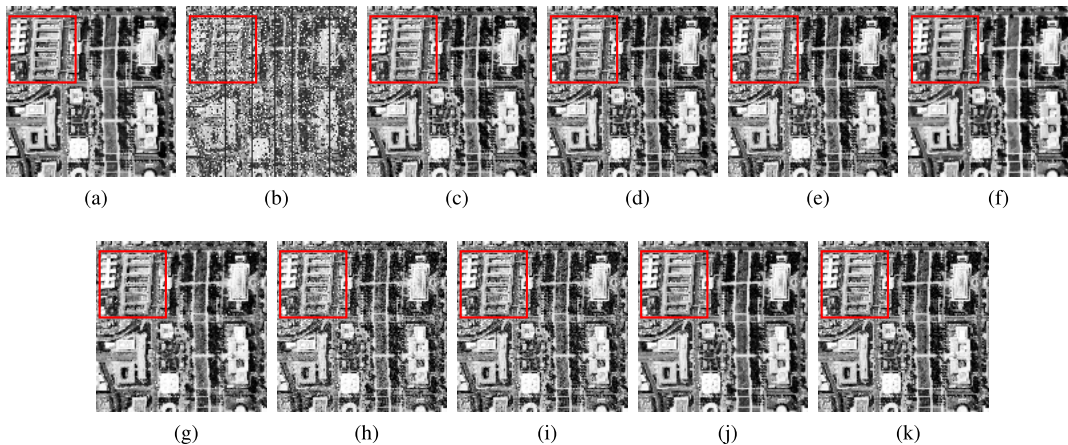
FIGURE 11. Denoising results of band 6 of Washington DC Mall subscene using different denoising methods. (a) Original image. (b) Noisy image (11.32 dB). The denoising results of (c) LRMR (34.50 dB). (d) GLRR (34.39 dB). (e) NAILRMR (34.69 dB). (f) LRTV (33.33 dB). (g) TVL1 (32.57 db). (h) SSTV (33.48 dB). (i) LSSTV (35.01 dB). (j) LLRSSTV (35.74 dB). (k) GLSSTV (37.1 dB).
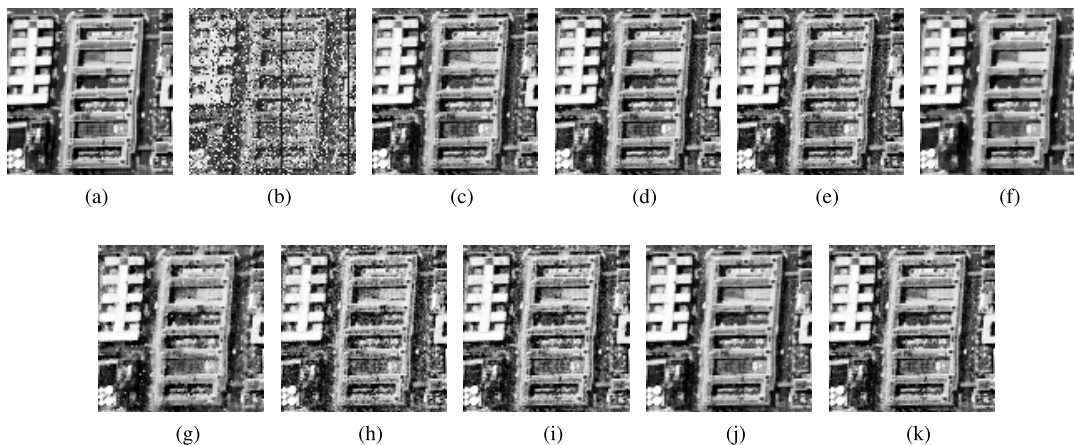
FIGURE 12. Magnified results of Fig. 11. (a) Original. (b) Noisy. (c) LRMR. (d) GLRR. (e) NAILRMR. (f) LRTV. (g) TVL1. (h) SSTV. (i) LSSTV. (j) LLRSSTV. (k) GLSSTV.

TABLE 1. MPSNR and MSSIM values of the denoising results in the simulated experiment. Boldface means the best and underline means the second best.

| Data	Evaluation Index	LRMR	NAILRMR	GLRR	LRTV	TVL1	SSTV	LSSTV	LLRSSTV	GLSSTV
	MPSNR	33.39	32.49	33.80	33.88	32.25	34.86	<u>36.64</u>	34.70	**37.13**
Pavia	MSSIM	0.9386	0.9188	0.9467	0.9367	0.9115	0.9499	<u>0.9655</u>	0.9627	**0.9702**
	MSAD	5.66	6.38	5.36	5.11	6.40	4.76	<u>3.92</u>	4.10	**3.68**
	MPSNR	34.42	34.03	35.37	34.44	31.85	34.85	36.61	<u>36.95</u>	**38.39**
Washington DC	MSSIM	0.9553	0.9458	0.9648	0.9558	0.9110	0.9517	0.9673	<u>0.9742</u>	**0.9795**
	MSAD	5.48	5.70	4.85	5.13	7.25	5.26	4.25	<u>3.78</u>	**3.37**

in the image. TVL1 removes the sparse noise to some extent but it does not remove it completely. LSSTV is better than SSTV in removing noise because of an addition of a low-rank constraint to SSTV formulation. LLRSSTV and GLSSTV perform best and preserve the details in the image as shown in Figs. 12(j) and 12(k).

E. QUANTITATIVE EVALUATION

Table 1 reports the quantitative evaluation results of all algorithms under comparison for Pavia and Washington DC Mall datasets. The best results of MPSNR, MSSIM and MSAD values are given in bold and the values with the second highest values are underlined. It is clear that GLSSTV
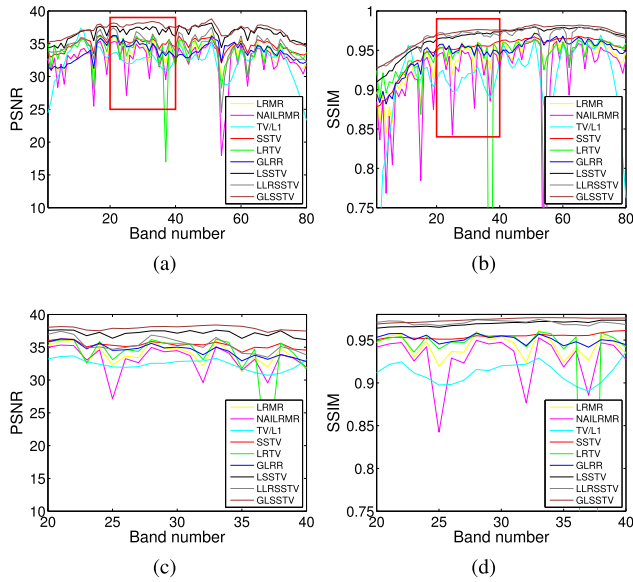
FIGURE 13. (a) and (b) are PSNR and SSIM values of each band achieved by different denoising methods in the experiment with the Pavia city subscene, respectively. (c) and (d) are PSNR and SSIM values of the bands between 20 and 40, respectively (marked by red rectangle in (a) and (b)).
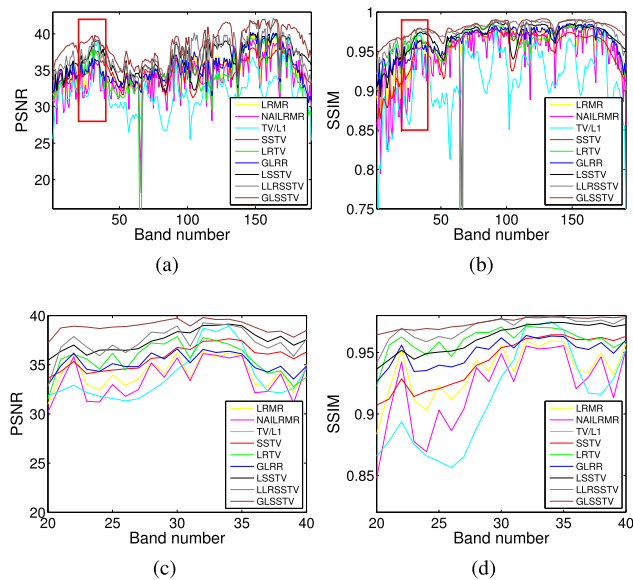
FIGURE 14. (a) and (b) are PSNR and SSIM values of each band achieved by different denoising methods in the experiment with the Washington DC Mall subscene. (c) and (d) are PSNR and SSIM values of the bands between 20 and 40, respectively. (marked by red rectangle in (a) and (b)).

outperforms the other methods in terms of MPSNR, MSSIM and MSAD values. PSNR and SSIM values of each band for Pavia and Washington Mall dataset are shown in Figs. 13 and 14, respectively. The results can be clearly seen in the magnified results of the region marked with red rectangle for corresponding figures. It can be deduced that GLSSTV performs better in most of the bands. Furthermore, we show the spectrum of the individual pixels for

two datasets. Figs. 15 and 16 show the spectrum of the pixels after denoising results. We can observe that, GLSSTV approximates the spectrum of pixel better than the other algorithms under comparison.

F. EXPERIMENTS ON REAL DATA
In this section, we perform several real data experiments to validate the effectiveness of GLSSTV.

1) AVIRIS INDIAN PINES DataSet
AVIRIS Indian Pines dataset[3] was acquired by the NASA airborne visible/infrared imaging spectrometer (AVIRIS) instrument over the Indian Pines test site in Northwestern Indiana in 1992. It has 220 spectral bands with spatial size of 220×220. Some bands of Indian Pines dataset are corrupted by atmosphere and water absorption. In the experiment, we use all of the bands of the Indian Pines dataset. Fig. 17 shows the band 220 of the Indian Pines dataset and magnified image of region marked with red rectangle in Fig. 17 is shown in Fig. 18. It can be clearly seen that LRMR, NAILRMR, GLRR and TVL1 are not able to remove the noise and do not preserve the details in the image. LRTV removes the noise but details are lost in the restored image. LLRSSTV is also good at removing noise but the details are not restored very well. LSSTV is better than SSTV in removing noise and both methods preserve the details in the image. GLSSTV removes the noise and performs the best in preserving the details in the image.

2) HYDICE URBAN DataSet
In the second real data experiment, we used Hyperspectral Digital Imagery Collection Experiment (HYDICE) Urban dataset[4] for comparison purposes. It has a size of $307 \times 307 \times 210$. We selected a subimage of size $200 \times 200 \times 210$. Fig. 19 shows the denoising results of band 139 of the Urban dataset. Band 139 contains sparse noise and stripes. Only low-rank based methods such as LRMR, NAILRMR and GLRR are not good at removing stripes. LRTV removes the sparse noise and stripes but also removes the details in the image. SSTV and LSSTV removes sparse noise and stripes to some extent but they do not completely remove the stripes. LLRSSTV and GLSSTV performs best in removing the sparse noise and stripes.

Moreover, we show the vertical mean profiles of band 132 in Fig. 23. LRMR, NAILRMR and GLRR are not good at suppressing the rapid fluctuations whereas TVL1 suppresses the fluctuations but do not preserve the structure. LRTV, SSTV and LSSTV also are not good at suppressing the fluctuations. LLRSSTV and GLSSTV performs similar and performs best in suppressing the fluctuations.

[3]http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes
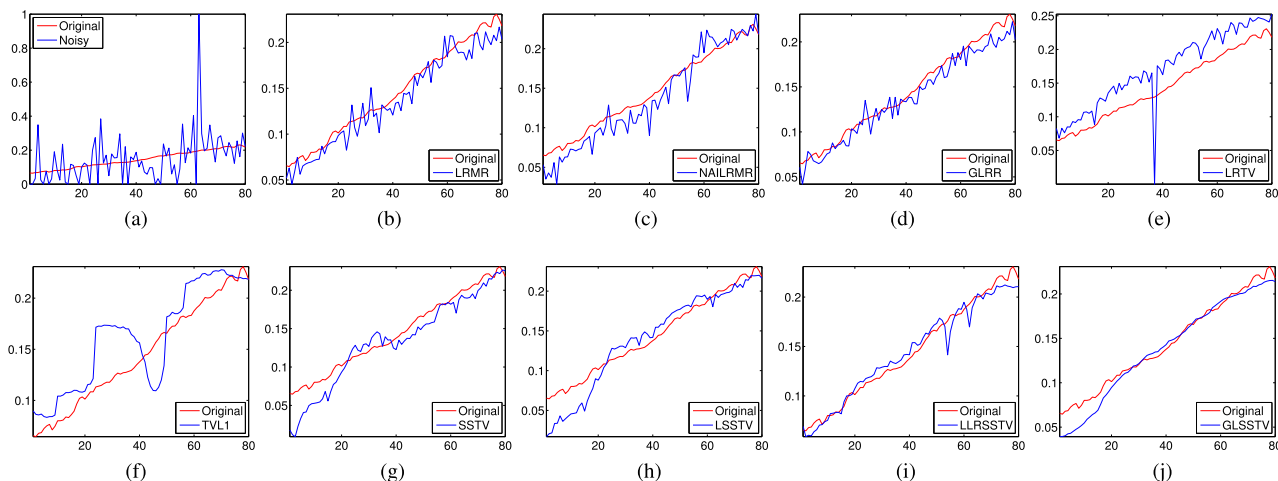[4]http://www.erdc.usace.army.mil/Media/Fact-Sheets/Fact-Sheet-ArticleView/Article/610433/hypercube/

FIGURE 15. Spectrum of noisy and denoised pixel using different methods for Pavia dataset at location (65,70).
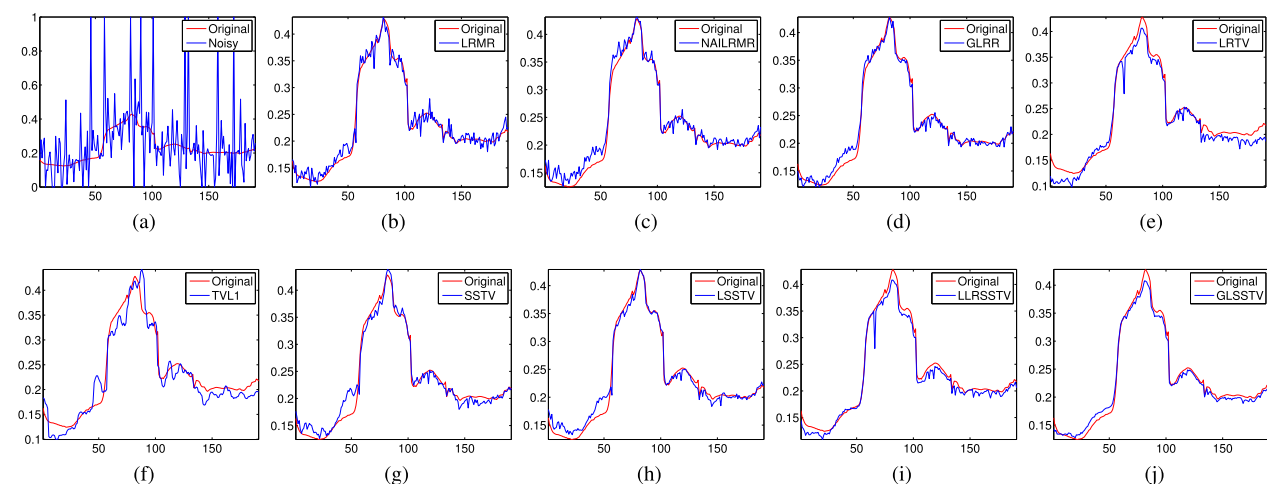

FIGURE 16. Spectrum of noisy and denoised pixel using different methods for Washington DC Mall dataset at location (189,126).
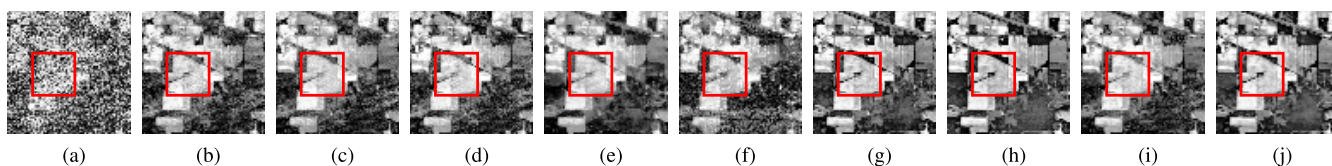

FIGURE 17. Denoising results of Indian Pines dataset for band 220 using different denoising methods. (a) Original image. The denoising results of (b) LRMR. (c) NAILRMR. (d) GLRR. (e) LRTV. (f) TVL1. (g) SSTV. (h) LSSTV. (i) LLRSSTV. (j) GLSSTV.

FIGURE 18. Magnified results of Fig. 17. (a) Original image. The denoising results of (b) LRMR. (c) NAILRMR. (d) GLRR. (e) LRTV. (f) TVL1. (g) SSTV. (h) LSSTV. (i) LLRSSTV. (j) GLSSTV.

3) EO-1 HYPERION DataSet

Third set of real experiments are conducted on EO-1 Hyperion DataSet.[5] It has a size of $1000 \times 400 \times 242$. The water

---

[5]http://www.gscloud.cn/

absorption bands are removed and we selected a subimage of size $200 \times 200 \times 166$. The Hyperion dataset is mainly corrupted by stripes, deadlines and Gaussian noise. Fig. 21 shows the denoising results of band 132 of Hyperion dataset. Magnified region marked with red in Fig. 21 is shown

FIGURE 19. Denoising results of HYDICE Urban dataset for band 139 using different denoising methods. (a) Original image. The denoising results of (b) LRMR. (c) NAILRMR. (d) GLRR. (e) LRTV. (f) TVL1. (g) SSTV. (h) LSSTV. (i) LLRSSTV. (j) GLSSTV.
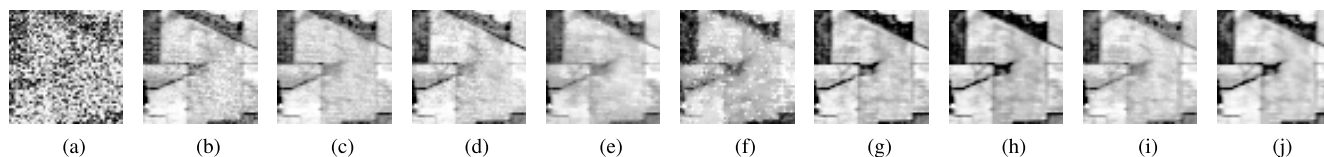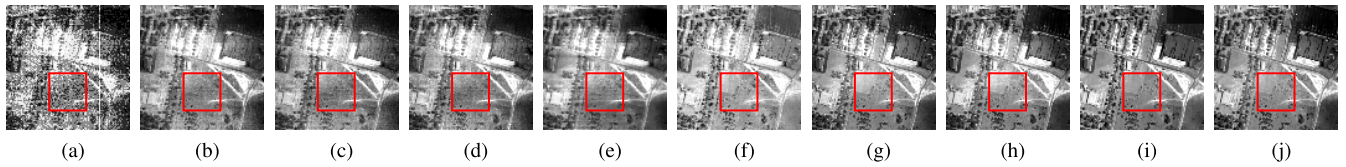

FIGURE 20. Magnified results of Fig. 19. (a) Original image. The denoising results of (b) LRMR. (c) NAILRMR. (d) GLRR. (e) LRTV. (f) TVL1. (g) SSTV. (h) LSSTV. (i) LLRSSTV. (j) GLSSTV.
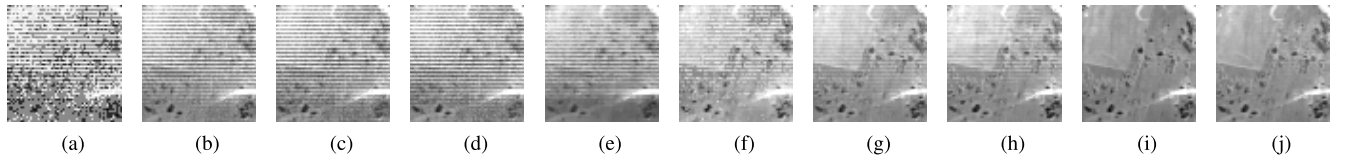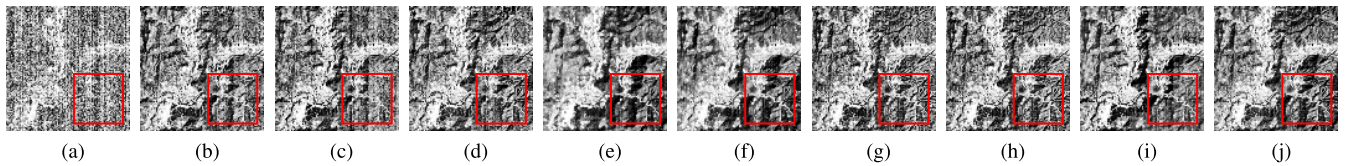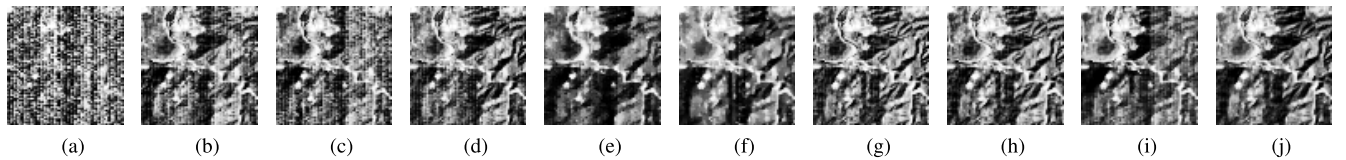

FIGURE 21. Denoising results of EO-1 Hyperion dataset for band 132 using different denoising methods. (a) Original image. The denoising results of (b) LRMR. (c) NAILRMR. (d) GLRR. (e) LRTV. (f) TVL1. (g) SSTV. (h) LSSTV. (i) LLRSSTV. (j) GLSSTV.
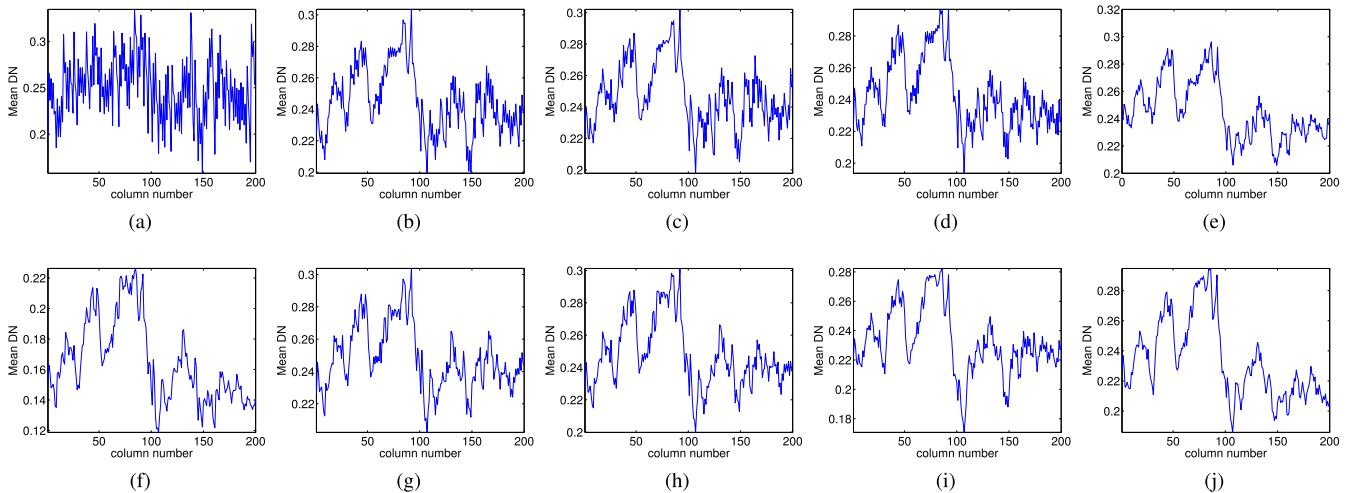

FIGURE 22. Magnified results of Fig. 21. (a) Original image. The denoising results of (b) LRMR. (c) NAILRMR. (d) GLRR. (e) LRTV. (f) TVL1. (g) SSTV. (h) LSSTV. (i) LLRSSTV. (j) GLSSTV.

FIGURE 23. Vertical mean profiles for EO-1 Hyperion dataset for band 132 obtained after denoising using different methods. (a) Original. (b) LRMR. (c) NAILRMR. (d) GLRR. (e) LRTV. (f) TVL1. (g) SSTV. (h) LSSTV. (i) LLRSSTV. (j) GLSSTV.

in Fig. 22. SSTV, LSSTV and GLSSTV remove the Gaussian noise and stripes while preserving the details in the image. LLRSSTV is also good at removing sparse and Gaussian

noise but some stripes are still left in the denoised image. The other compared methods are not very good at removing mixed noise from the image.

TABLE 2. Classification accuracy results (mean accuracy (%)± standard deviation) of the indian pines dataset using different denoising methods.

Class Name	Train/Test	LRMR	NAILRMR	GLRR	SSTV	LSSTV	LLRSSTV	GLSSTV
Alfalfa	6/48	43.96±9.44	46.46±8.45	36.67±12.77	75.83±16.85	69.17±17.70	52.08±10.67	**83.54**±13.90
Corn-notill	144/1290	77.32±3.65	78.87±2.98	73.72±2.75	**87.78**±1.90	81.40±2.34	73.60±3.42	87.58±2.42
Corn-mintill	84/750	77.16±5.82	79.17±3.90	72.63±3.92	86.72±1.85	85.25±3.39	63.52±5.63	**86.92**±3.23
Corn	24/214	88.00±3.26	92.10±3.33	84.71±6.87	88.48±7.0	88.62±3.50	78.90±5.85	**91.48**±2.86
Grass-pasture	50/447	80.27±5.60	79.44±4.33	71.19±5.03	**86.69**±4.15	83.76±3.69	72.62±4.30	82.26±4.59
Grass-tree	75/672	74.48±3.81	80.45±3.28	66.03±2.80	**91.16**±3.06	85.65±2.73	70.51±2.87	88.76±3.04
Grass-pasture -mowed	3/23	49.13±12.81	38.70±13.20	41.30±11.60	62.11±17.29	59.57±2.73	40.87±17.77	**68.70**±20.68
Hay-windrowed	49/440	97.61±13.30	97.66±16.1	94.36±1.51	**94.27**±1.67	93.91±3.48	88.80±4.24	93.66±2.76
Oats	2/18	28.33±21.19	37.22±28.57	36.11±19.11	50.56±24.63	46.11±23.87	36.11±19.11	**52.22**±21.31
Soybeans-notill	97/871	75.20±3.38	78.99±3.09	73.86±2.13	84.03±2.55	79.92±2.87	73.12±3.52	**86.42**±3.09
Soybeans-mintill	247/2221	87.45±1.65	89.72±2.00	82.01±2.12	90.24±1.51	86.73±2.90	81.93±1.38	**91.09**±1.87
Soybeans-cleantill	62/552	74.24±3.01	74.75±3.08	67.36±4.02	80.09±3.17	78.15±3.93	61.74±2.49	**80.40**±3.09
Wheat	22/190	50.42±10.39	39.47±9.64	52.68±8.76	67.16±9.65	68.58±5.95	32.05±5.80	**70.00**±6.86
Woods	130/1164	91.38±3.39	92.76±1.95	87.77±2.79	93.75±2.21	92.96±2.61	89.74±2.85	**94.33**±2.08
Building-Grass -Trees-Drives	38/342	66.49±6.77	65.82±8.02	65.09±7.24	75.47±6.35	71.23±6.32	63.45±7.24	**77.75**±6.60
Stone-Steel -Towers	10/85	63.18±5.95	62.94±6.73	56.94±9.22	72.82±8.35	**78.35**±6.05	52.71±8.84	74.00±8.27
Overall Accuracy	–	80.82±0.82	82.53±0.6	76.30±1.14	87.53±0.88	84.42±0.87	74.94±1.04	**87.90**±0.81
Average Accuracy	–	70.29±2.31	70.91±2.49	66.40±2.17	80.48±2.46	78.08±2.15	64.48±2.02	**81.82**±2.44
Kappa	–	78.12±0.95	80.07±0.69	72.94±1.31	85.80±1.0	82.28±0.98	71.45±1.18	**86.21**±0.92

TABLE 3. Computation times of the algorithms for real data experiments. Boldface means the best and underline means the second best.

Data	LRMR	NAILRMR	GLRR	LRTV	TVL1	SSTV	LSSTV	LLRSSTV	GLSSTV
Indian Pines	**21.17**	195.66	122.55	99.86	23.91	254.48	306.47	173.87	828.23
Urban	63.20	221.58	242.51	160.37	**46.24**	465.56	559.79	266.08	1540.42
Hyperion	38.01	166.63	185.54	116.81	**35.10**	367.37	434.23	243.20	1189.02

G. CLASSIFICATION EXPERIMENT

Classification accuracy results are another performance measure for HSI denoising algorithms. We investigated the classification accuracy results using support vector machine (SVM) classifier [1] with cross-validation for all denoising results on Indian Pines dataset. Indian pines dataset contains 10249 samples and including 16 classes. We selected 10% of samples randomly from each class for training and remaining ones are selected as test data. We evaluated the Overall Accuracy (OA), Average Accuracy (AA), Kappa values and class accuracies. We repeat the classification process ten times by selecting different training and test samples for each trial and average the results. We give only the classification accuracy results of SSTV and patch-based denosing methods. LRTV and TVL1 are excluded in the classification experiments, since they perform poorly in classification experiment. Table 2 shows the classification accuracy results for each of the denoising results. It can be observed that classification accuracy results are increased after denoising process. GLSSTV achieves best classification accuracy results in terms of OA, AA and Kappa.

H. COMPUTATION TIME

There are several parameters that affect the computation time of GLSSTV. These are block size M, step size s and number of patches k. Selecting a small s causes high value of K, which leads to high computation time. Also, a large value of M will increase the SVD computation time. In addition, k affects the computation time. When k increases, the size of the group increases. Therefore, it leads to high computation time.

In the simulations, we performed the experiments on a workstation with a 3.1 GHz Intel 4 core Xeon processor and 16 GB memory using MATLAB. The computation times of the real data experiments are reported in Table 3. It can be seen that GLSSTV has a longer computation times compared to other methods due to the non-local low-rank approach used in the algorithm. Computation of the low-rank approximation of each group takes the main computational time. Since SVD for each group is calculated separately, this computation can be calculated using parallel computation to reduce the computation time on a computer having multi-core support.

V. CONCLUSION

In this paper, we have proposed a novel HSI denoising method for mixed noise removal using rank constrained group low-rank approximation and SSTV. Group low-rank approximation exploits the local similarity inside a patch and non-local similarity across patches. Therefore, additional

structural information is exploited in HSI, which helps the reconstruction of corrupted patches effectively. Moreover, SSTV removes Gaussian and sparse noise further by utilizing the spatial and spectral smoothness of HSI. Experiments on simulated datasets show that the proposed method is effective in HSI denoising and outperforms the state-of-the-art algorithms proposed in the literature in terms of PSNR and SSIM. Furthermore, in real data experiments, the proposed method reduces the mixed noise by retaining the fine details in the image.

GLSSTV performs low-rank approximation based on 2D matrix form of the 3D subcubes extracted from HSI which ignores the spatial information inside a patch. Therefore, as a future work, we will employ SSTV regularized group low-rank tensor approximation to use the spatial information effectively.

REFERENCES

[1] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

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

[3] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Sparse representation for target detection in hyperspectral imagery," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 3, pp. 629–640, Jun. 2011.

[4] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[6] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.

[7] G. Chen and S.-E. Qian, "Denoising of hyperspectral imagery using principal component analysis and wavelet shrinkage," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 3, pp. 973–980, Mar. 2011.

[8] Y.-Q. Zhao and J. Yang, "Hyperspectral image denoising via sparse representation and low-rank constraint," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 1, pp. 296–308, Jan. 2015.

[9] T. Lu, S. Li, L. Fang, Y. Ma, and J. A. Benediktsson, "Spectral–spatial adaptive sparse representation for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 373–385, Jan. 2016.

[10] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan, "Hyperspectral image restoration using low-rank matrix recovery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4729–4743, Aug. 2014.

[11] R. Zhu, M. Dong, and J. H. Xue, "Spectral nonlocal restoration of hyperspectral images with low-rank property," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 3062–3067, Jun. 2015.

[12] W. He, H. Zhang, L. Zhang, and H. Shen, "Hyperspectral image denoising via noise-adjusted iterative low-rank matrix approximation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 3050–3061, Jun. 2015.

[13] W. He, H. Zhang, L. Zhang, and H. Shen, "Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 178–188, Jan. 2016.

[14] Y. Xie, Y. Qu, D. Tao, W. Wu, Q. Yuan, and W. Zhang, "Hyperspectral image restoration via iteratively regularized weighted schatten *p*-norm minimization," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4642–4659, Aug. 2016.

[15] Y. Chen, Y. Guo, Y. Wang, D. Wang, C. Peng, and G. He, "Denoising of hyperspectral images using nonconvex low rank matrix approximation," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5366–5380, Sep. 2017.

[16] H. Othman and S.-E. Qian, "Noise reduction of hyperspectral imagery using hybrid spatial-spectral derivative-domain wavelet shrinkage," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 2, pp. 397–408, Feb. 2006.

[17] Q. Yuan, L. Zhang, and H. Shen, "Hyperspectral image denoising with a spatial–spectral view fusion strategy," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 5, pp. 2314–2325, May 2014.

[18] J. Chen and J. Yang, "Robust subspace segmentation via low-rank representation," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1432–1445, Aug. 2014.

[19] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[20] T. Zhou and D. Tao, "Godec: Randomized low-rank sparse matrix decomposition in noisy case," in *Proc. 28th Int. Conf. Mach. Learn.(ICML)*, L. Getoor and T. Scheffer, Eds. New York, NY, USA: ACM, Jun. 2011, pp. 33–40.

[21] M. Wang, J. Yu, J.-H. Xue, and W. Sun, "Denoising of hyperspectral images using group low-rank representation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4420–4427, Sep. 2016.

[22] L. Sun, B. Jeon, Y. Zheng, and Z. Wu, "Hyperspectral image restoration using low-rank representation on spectral difference image," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 7, pp. 1151–1155, Jul. 2017.

[23] F. Fan, Y. Ma, C. Li, X. Mei, J. Huang, and J. Ma, "Hyperspectral image denoising with superpixel segmentation and low-rank representation," *Inf. Sci.*, vols. 397–398, pp. 48–68, Aug. 2017.

[24] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D, Nonlinear Phenomena*, vol. 60, nos. 1–4, pp. 259–268, 1992.

[25] Q. Yuan, L. Zhang, and H. Shen, "Hyperspectral image denoising employing a spectral–spatial adaptive total variation model," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3660–3677, Oct. 2012.

[26] H. K. Aggarwal and A. Majumdar, "Hyperspectral image denoising using spatio-spectral total variation," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 442–446, Mar. 2016.

[27] W. He, H. Zhang, H. Shen, and L. Zhang, "Hyperspectral image denoising using local low-rank matrix recovery and global spatial–spectral total variation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 3, pp. 713–729, Mar. 2018.

[28] H. Liu, P. Sun, Q. Du, Z. Wu, and Z. Wei, "Hyperspectral image restoration based on low-rank recovery with a local neighborhood weighted spectral-spatial total variation model," *IEEE Trans. Geosci. Remote Sens.*, to be published.

[29] Q. Wang, Z. Wu, J. Jin, T. Wang, and Y. Shen, "Low rank constraint and spatial spectral total variation for hyperspectral image mixed denoising," *Signal Process.*, vol. 142, pp. 11–26, Jan. 2018.

[30] L. Sun, T. Zhan, Z. Wu, L. Xiao, and B. Jeon, "Hyperspectral mixed denoising via spectral difference-induced total variation and low-rank approximation," *Remote Sens.*, vol. 10, no. 12, p. 1956, 2018. [Online]. Available: http://www.mdpi.com/2072-4292/10/12/1956

[31] L. Sun, B. Jeon, Z. Wu, and L. Xiao, "Hyperspectral denoising via cross total variation-regularized unidirectional nonlocal low-rank tensor approximation," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 2900–2904.

[32] Z. Huang, S. Li, L. Fang, H. Li, and J. A. Benediktsson, "Hyperspectral image denoising with group sparse and low-rank tensor decomposition," *IEEE Access*, vol. 6, pp. 1380–1390, Dec. 2017.

[33] H. Fan, C. Li, Y. Guo, G. Kuang, and J. Ma, "Spatial–spectral total variation regularized low-rank tensor decomposition for hyperspectral image denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 6196–6213, Oct. 2018.

[34] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, May 2011, Art. no. 11.

[35] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2009, pp. 2080–2088.

[36] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," in *Foundations and Trends in Machine Learning*, vol. 3, no. 1. Hanover, MA, USA: Now, 2011, pp. 1–122.

[37] J. Cai, E. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.

[38] C. C. Paige and M. A. Saunders, "LSQR: An algorithm for sparse linear equations and sparse least squares," *ACM Trans. Math. Softw.*, vol. 8, no. 1, pp. 43–71, Mar. 1982.

[39] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An augmented Lagrangian method for total variation video restoration," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3097–3111, Nov. 2011.

[40] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

TANER INCE (M'05) received the B.Sc., M.Sc., and Ph.D. degrees in electrical and electronics engineering from the University of Gaziantep, Gaziantep, Turkey, in 2003, 2006, and 2012, respectively. He was a Visiting Researcher with the University of California at Los Angeles, Los Angeles, CA, USA, from 2013 to 2014, supported by the International Research Fellowship Program (2219) of The Scientific and Technological Research Council of Turkey (TUBITAK).

From 2004 to 2012, he was a Research Assistant with the University of Gaziantep, where he is currently an Assistant Professor. His research interests include compressed sensing, sparse representation, and machine learning with applications to remote sensing image analysis.

● ● ●