

Tensor Low-Rank Constraint and l_0 Total Variation for Hyperspectral Image Mixed Noise Removal

Minghua Wang, *Student Member, IEEE* Qiang Wang, *Member, IEEE* and Jocelyn Chanussot, *Fellow, IEEE*

Abstract—Several methods based on Total Variation (TV) have been proposed for Hyperspectral Image (HSI) denoising. However, the TV terms of these methods just use various l_1 norms and penalize image gradient magnitudes, having a negative influence on the preprocessing of HSI denoising and further HSI classification task. In this paper, a novel l_0 Total Variation (l_0 TV) is first introduced and analyzed for the HSI noise removal framework to preserve more information for classification. We propose a novel Tensor low-rank constraint and l_0 Total Variation (TLR- l_0 TV) model in this paper. l_0 TV directly controls the number of non-zero gradients and focuses on recovering the sharp image edges. The spectral-spatial information among all bands is exploited uniformly for removing mixed noise, which facilitates the subsequent classification after denoising. Including the Weighted Sum of Weighted Nuclear Norm (WSWNN) and the Weighted Sum of Weighted Tensor Nuclear Norm (WSWTNN), we propose two TLR- l_0 TV-based algorithms, namely WSWNN- l_0 TV and WSWTNN- l_0 TV. The Alternating Direction Method of Multipliers (ADMM) and the Augmented Lagrange Multiplier (ALM) are employed to solve the l_0 TV model and TLR- l_0 TV model, respectively. In both simulated and real data, the proposed models achieve superior performances in mixed noise removal of HSI. Especially, HSI classification accuracy is improved more effectively after denoising by the proposed TLR- l_0 TV method.

keywords—Hyperspectral Image (HSI), mixed noise, tensor LR constraint, l_0 TV, ADMM, ALM.

I. INTRODUCTION

Due to the spectral diversity of information, hyperspectral images (HSIs) are now widely used in various applications such as precision agriculture, monitoring of the environment, defense and security, urban planning or planetary and space exploration [1]–[3]. Unfortunately, because of some different noise sources like sensor systems and external environment, the observed HSIs are corrupted by mixed noise, including Gaussian noise, salt and pepper noise, stripes or dead-line noise. This degrades the quality of hyperspectral images and further limits the precision of subsequent HSI applications, including classification [4], [5], unmixing [6], [7], feature learning [8], [9] and target detection [10], [11]. Therefore, HSI denoising appears as an essential preprocessing step to improve image quality. For HSI denoising, it is critical to make good use of image prior knowledge, containing spatial spectral smoothness and low-rank property.

M. Wang is with Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, PR China and also with Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-lab, 38000 Grenoble, France (Email: minghuawang1993@163.com).

Q. Wang is with Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, PR China (Email: wangqiang@hit.edu.cn).

J. Chanussot is with Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, Grenoble, 38000, France (Email: jocelyn.chanussot@grenoble-inp.fr).

To exploit spatial and spectral smoothness, image denoising models based on total variation have been designed. The TV model was firstly applied for gray-level image denoising [12], [13] and color image denoising [14], [15]. Strong *et al.* [12] developed a spatially adaptive TV (SATV) minimization to solve the gray-level image denoising problem and Blomgren *et al.* [14] proposed a new TV for vector-valued data to restore color images. Different from these kinds of images, a hyperspectral image is a 3-D tensor data and has more abundant spectral bands. Yuan *et al.* [16] employed a spectral-spatial adaptive TV (SSAHTV) model where both the spectral and the spatial noise differences are considered for HSI denoising. However, SSAHTV only removes Gaussian noise while failing to retain important image details. Chang *et al.* [17] proposed an anisotropic spectral-spatial total variation regularization to enhance the smoothness along three dimensions, but this TV model only removed stripes and random noise. For efficient mixed noise removal, Aggarwal and Majumdar [18] utilized 2-D TV along the spatial dimension and 1-D TV along the spectral dimension and proposed a spatio-spectral total variation (SSTV) model. But the denoised results by SSTV have artifacts since SSTV just focuses on local spatio-spectral differences. These TV methods capture local spatial and spectral information but ignore another important HSI typical characteristic, namely the assumption of Low Rank (LR) property.

As another powerful technique for image processing, LR-based methods have shown great potential in the field of HSI denoising. Each spectral band is rearranged into a vector and a clean 3-D HSI data can be treated as a 2-D matrix [19]–[23]. Due to the spectral correlation of HSI, this matrix can be represented by a linear combination of finite pure endmembers and hence has the LR property. Zhang *et al.* [19] formulated the HSI denoising problem into an LR Matrix Recovery (LRMR) framework and applied the "Go Decomposition" algorithm to solve the LRMR problem. Xie *et al.* [20] introduced a non-LR model, i.e., weighted Schatten p -norm to get a better approximation of the original LR assumption. Instead of a traditional nuclear norm, Chen *et al.* [21] used a tighter LR approximation and presented a nonconvex LRMR model. Fan *et al.* [22] employed a bi-nuclear quasi-norm to constrain the LR characteristic of HSI and proposed a bilinear LR matrix factorization (BLRMF) HSI denoising method. Moreover, the deep learning-based methods have revealed in handling HSI restoration problems [24]–[26]. Lin *et al.* [24] combined a convolutional neural network with the non-negative matrix factorization framework. However, this model mainly rearranges a 3-D HSI tensor into a 2-D matrix, which unfortunately leads

to the loss of the inherent tensor structure information. Yuan *et al.* [25] employed an HSI Spatial-Spectral Deep Residual Convolutional Neural Network (HSID-CNN) to build a non-linear end-to-end mapping between the noisy and denoised images. Zhang *et al.* [26] proposed a spatial-spectral gradient network (SSGN), which simultaneously extracted the structure directionality of sparse noise and obtained the additional spectral information of HSIs. Consequently, many tensor low-rank (TLR) based methods have been proposed. Renard *et al.* [27] proposed a LR Tensor Approximation (LRTA) model by using Tucker decomposition and Liu *et al.* [28] proposed a Parallel Factor Analysis (PARAFAC) decomposition model. Inspired by Tensor Nuclear Norm (TNN) based Tensor Robust Principal Component Analysis (TRPCA) [29], Fan *et al.* [30] reformulated the HSI denoising task as an LR Tensor Recovery (LRTR) problem. Xue *et al.* [31] proposed a non-convex tensor rank minimization (NTRM) model by using the non-convex logarithmic surrogate function of the singular values to approximate the tensor rank. To implicitly describe the intrinsic prior, Xue *et al.* [32] used both local and global sparsity prior of a tensor and presented an enhanced sparsity prior model. Zheng *et al.* [33] discussed the LR property along three directions and proposed the Weighted Sum of Tensor Nuclear Norm (WSTNN) model for HSI sparse noise removal. Although these approaches consider the LR prior knowledge of HSI, they are not effective to suppress mixed noise since they ignore both the spatial and the spectral information.

Recent studies of HSI denoising focus on the combination of various types of priors. Xue *et al.* [34] incorporated the spectral low-rankness into sparse representation to characterize the spatial structure. Chang *et al.* [35] designed a unidirectional LR tensor recovery model to utilize the LR constraint along the Nonlocal Self-Similarity (NSS) direction and the hyper-Laplacian prior. Xue *et al.* [36] explored two characteristics for HSI: the Global Correlation across Spectrum (GCS) and NSS over space, and further proposed a nonlocal LR regularized CANDECOMP/PARAFAC (CP) tensor decomposition (NLR-CPTD). He *et al.* [37] proposed a new paradigm to combine the spatial NSS and global spectral LR property, named non-local meets global (NGmeet) method. Nevertheless, NSS-based methods usually establish matrix or tensor groups by matching Non-local Full Band Patches (NLFBP). When the spectral band number increases, the size of NLFBP will become larger, leading to a heavier computation burden. He *et al.* [38] proposed a band-by-band total variation regularized LR Matrix Factorization (LRTV) method and Wu *et al.* [39] combined the band-by-band TV regularization with the Weighted Nuclear Norm Minimization (WNNM) model. But this band-by-band TV regularization causes spectral distortions. To exploit the spatial-spectral information simultaneously, Wang *et al.* [40] proposed a novel LR constraint and spatial-spectral total variation regularization model (LSSTV) and Fan *et al.* [41] integrated this kind of SSTV into LR tensor factorization. Another 3-D anisotropic total variation (3DATV) is defined by l_1 TV along three directions of HSI. He *et al.* [42] combined this 3DATV with the local LR matrix and proposed an LLRSSTV model for the HSI mixed noise removal. Consequently, Wang *et al.* [43] injected this 3DATV

regularization into a Tucker decomposition framework. In [44], we developed the weighted sum of the weighted nuclear norm (WSWNN) and the weighted sum of the weighted tensor nuclear norm (WSWTNN), and combined these models with an SSTV term for HSI mixed noise removal. Nevertheless, this model has two shortcomings: 1) SSTV focuses on local spatial and spectral information, leading to artifacts in the denoised results. Although SSTV is incorporated into WSWNN and WSWTNN frameworks, SSTV only relies on the l_1 norm that calculates large gradient magnitudes, hurting image smoothness. 2) The modeling just considers how to remove noise by this combination. HSI denoising is often considered as a preprocessing step for further HSI classification. In addition to mixed noise removal, more information such as sharp image edges for classification should be preserved in the noise reduction processing.

However, the above TV norms rely on the l_1 norm, which only penalizes large gradient magnitudes and further may blur the real image edges. Simultaneously, researchers only consider how to remove noise, but ignore HSI denoising as a preprocessing step to improve the further classification. To sharpen image edges, a new l_0 gradient minimization has been proposed for image smoothing [45]. Actually, l_1 TV norm is a relaxation of the l_0 gradient which directly calculates the number of non-zero image gradients. This work has been used in image denoising [45], image deblurring [46], and image inpainting [47]. Xu *et al.* [46] proposed an effective method based on a generalized l_0 sparse expression for motion deblurring. Due to a significant fraction of non-zero pixels of depth images, Xue *et al.* [47] extended the above l_0 gradient into a low gradient and converted this regularization into an LR framework. We developed the l_0 gradient to explore both the spatial and the spectral structure information and combined it with the LR Tucker decomposition model [48]. However, the degrees of smoothness of these methods are controlled by a parameter, which is given by users and has no physical meaning. To alleviate this limitation, Ono [49] proposed a novel l_0 gradient projection, which directly used a parameter to control the degree of smoothness. This parameter has a clear meaning: it is the l_0 gradient value of the output image. Therefore, in this paper, we extend the l_0 TV model to cope with an HSI denoising problem and preserve more information for classification. Nevertheless, we find it hard to distinguish sparse noise when this kind of noise locate on the image edges. For mixed noise removal, we proposed a novel Tensor low-rank constraint and l_0 gradient (TLR- l_0 TV) model. Compared with existing techniques, the main contributions of this study are summarized as follows:

- (1) We first introduce l_0 TV into HSI denoising and analyze the significance of l_0 TV for HSI. This minimization controls the number of non-zero image gradients directly and builds a physical relationship with the l_0 gradient value of the denoised result, which leads to the great benefit of removing Gaussian noise and preserving image edges.
- (2) Based on the global low-rank property of HSI, we integrate Tensor Low-rank (TLR) constraint into the l_0 TV framework to suppress the sparse noise located on image edges. The novel WSWNN- l_0 TV and WSWTNN- l_0 TV

models do not only remove noise, but also provide more information for the subsequent classification task.

- (3) We develop the Alternating Direction Method of Multipliers (ADMM) and the Augmented Lagrange Multiplier (ALM) to solve the l_0 TV model and the TLR- l_0 TV model, respectively. Experiments on simulated and real datasets demonstrate the superiority of TLR- l_0 TV. In particular, the classification accuracy of the denoised results by TLR- l_0 TV is improved significantly.

The remainder of this paper is organized as follows. Some notations and preliminaries of tensors are introduced in Sec.II. Then, the introduction and the study of l_0 gradient minimization for HSI denoising is given in Sec.III. The TLR- l_0 TV is described in Sec.IV. The experimental results and analysis are provided in Sec.V. Finally, the conclusions are drawn in Sec.VI.

II. NOTATIONS AND PRELIMINARIES

In this section, we introduce some notations and preliminaries. For clear description, the notations are list in Table I.

TABLE I: The notations used in the paper

Notation	Description
x	scalars
\mathbf{x}	vectors
\mathbf{X}	matrices
$\mathcal{X} \in \mathbb{R}^{h \times v \times z}$	tensors with 3-modes
$\mathcal{X}_{h_i, v_i, z_i}$	the (h_i, v_i, z_i) -element of \mathcal{X}
$\mathcal{X}(i, :, :)$, $\mathcal{X}(:, i, :)$ and $\mathcal{X}(:, :, i)$	the i^{th} horizontal, lateral and frontal slices
$\ \mathcal{X}\ _1 = \sum_{h_i, v_i, z_i} \mathcal{X}_{h_i, v_i, z_i} $	l_1 norm
$\ \mathcal{X}\ _F = \sqrt{\sum_{h_i, v_i, z_i} \mathcal{X}_{h_i, v_i, z_i} ^2}$	Frobenius norm
$\sigma_i(\mathbf{X})$	the singular values of matrix \mathbf{X}
$\ \mathbf{X}\ _* = \sum_i \sigma_i(\mathbf{X})$	nuclear norm
$\ \mathbf{x}\ _2 = \sqrt{\sum_i \mathbf{x}_i ^2}$	l_2 norm
$\hat{\mathcal{X}} = \text{fft}(\mathcal{X}, [], 3)$	Fourier transformation of \mathcal{X} along mode-3

Definition 1 (Mixed $l_{1,0}$ pseudo-norm [49]): Given a vector $\mathbf{y} \in \mathbb{R}^m$ and index sets $\theta_1, \dots, \theta_i, \dots, \theta_n (1 \leq n \leq m)$ that satisfies

- Each θ_i is a subset of $1, \dots, m$,
- $\theta_i \cap \theta_l = \emptyset$ for any $i \neq l$,
- $\cup_{i=1}^n \theta_i = 1, \dots, m$,

the mixed $l_{1,0}$ pseudo-norm of \mathbf{y} is defined as:

$$\|\mathbf{y}\|_{1,0}^\theta = \|(\|\mathbf{y}_{\theta_1}\|_1, \dots, \|\mathbf{y}_{\theta_i}\|_1, \dots, \|\mathbf{y}_{\theta_n}\|_1)\|_0, \quad (1)$$

where \mathbf{y}_{θ_i} denotes a sub-vector of \mathbf{y} with its entries specified by θ_i and $\|\cdot\|_0$ calculates the number of the non-zero entries in (\cdot) .

Definition 2 (Indicator function $I(\mathbf{y})$ of $l_{1,0}$ mixed pseudo-norm [49]):

$$I_{\|\mathcal{B}\cdot\|_{1,0}^\theta}(\mathbf{y}) = \begin{cases} 0, & \|\mathcal{B}\mathbf{y}\|_{1,0}^\theta \leq \gamma, \\ \infty, & \text{otherwise,} \end{cases} \quad (2)$$

where \mathcal{B} is a given operator and γ is a given parameter. In this paper, \mathcal{B} is a diagonal tensor with 0 or 1 entries and γ is a desired degree of l_0 TV imposed on denoised results.

Proposition 1 (Projection onto $l_{1,0}$ mixed pseudo-norm ball with binary mask [49]): Set $\mathbf{y} \in \mathbb{R}^m$ as a known vector, set γ as a non-negative integer. Let \mathbf{W} be a known diagonal binary matrix, and let $\theta_1, \dots, \theta_i, \dots, \theta_n (1 \leq n \leq m)$ be index sets satisfying the conditions from **definition 1**. Without loss of generality, $\mathbf{W}\mathbf{y} = (\mathbf{y}_{\theta_1}^T \dots \mathbf{y}_{\theta_n}^T)^T$ are assumed. $\mathbf{y}_{\theta_1} \dots \mathbf{y}_{\theta_n}$ are sorted in descending order according to $\|\mathbf{y}_{\theta_{\vec{1}}}\|_2 > \|\mathbf{y}_{\theta_{\vec{2}}}\|_2 > \dots > \|\mathbf{y}_{\theta_{\vec{n}}}\|_2$, where $\mathbf{y}_{\theta_{\vec{1}}} \dots \mathbf{y}_{\theta_{\vec{n}}}$ are obtained with the new order, and the original index sets i have \vec{i} that corresponds to it one by one.

For the following problem

$$\min_{\mathbf{y}^*} \|\mathbf{y}^* - \mathbf{y}\|^2 \text{ s.t. } \|\mathbf{W}\mathbf{y}^*\|_{1,0}^\theta \leq \gamma, \quad (3)$$

one of the optimal solution is given by:

$$\mathbf{y}^* = \begin{cases} \mathbf{y}, & \|\mathbf{W}\mathbf{y}\|_{1,0}^\theta \leq \gamma, \\ (\tilde{\mathbf{y}}_{\theta_{\vec{1}}}^T \dots \tilde{\mathbf{y}}_{\theta_{\vec{n}}}^T)^T + (\mathbf{I} - \mathbf{W})\mathbf{y}, & \|\mathbf{W}\mathbf{y}\|_{1,0}^\theta > \gamma, \end{cases} \quad (4)$$

where

$$\tilde{\mathbf{y}}_{\theta_{\vec{i}}}^T = \begin{cases} \mathbf{y}_{\theta_i}^T, & \text{if } i \in \{1, \dots, \gamma\}, \\ 0, & \text{if } i \in \{\gamma + 1, \dots, n\}. \end{cases} \quad (5)$$

Example: Let $\mathbf{y} := (1, 5, -2, 9, 6, -3) \in \mathbb{R}^6$, $\gamma = 1$, $\mathbf{W} := \text{diag}(1, 1, 1, 0, 1, 1)$, and $n = 3$, so index sets include $\theta_1 := \{1, 2\}$, $\theta_2 := \{3, 4\}$, $\theta_3 := \{5, 6\}$. Then, $\mathbf{W}\mathbf{y} = (1, 5, -2, 0, 6, -3)$ and subvectors of \mathbf{y} are obtained: $\mathbf{y}_{\theta_1} = [1, 5]$, $\mathbf{y}_{\theta_2} = [-2, 0]$, $\mathbf{y}_{\theta_3} = [6, -3]$. Since $\|\mathbf{y}_{\theta_1}\|_2 = \sqrt{26}$, $\|\mathbf{y}_{\theta_2}\|_2 = 2$, $\|\mathbf{y}_{\theta_3}\|_2 = 3\sqrt{5}$, we have $\|\mathbf{y}_{\theta_3}\|_2 > \|\mathbf{y}_{\theta_2}\|_2 > \|\mathbf{y}_{\theta_1}\|_2$ and sort them in descending order. then in descending order. $\vec{1} = 3$, $\vec{2} = 1$, $\vec{3} = 2$ and $\mathbf{y}_{\theta_{\vec{1}}} = \mathbf{y}_{\theta_3} = [6, -3]$, $\mathbf{y}_{\theta_{\vec{2}}} = \mathbf{y}_{\theta_1} = [1, 5]$, $\mathbf{y}_{\theta_{\vec{3}}} = \mathbf{y}_{\theta_2} = [-2, 0]$ are obtained. Due to $\|\mathbf{W}\mathbf{y}\|_{1,0}^\theta > \gamma$, the optimal solution is solution is $(\tilde{\mathbf{y}}_{\theta_{\vec{1}}}^T \dots \tilde{\mathbf{y}}_{\theta_{\vec{n}}}^T)^T + (\mathbf{I} - \mathbf{W})\mathbf{y}$ and $\gamma = 1$, thus $\mathbf{y}_{\theta_{\vec{1}}} = \mathbf{y}_{\theta_3} = [6, -3]$ is selected and its corresponding original subvector is $\mathbf{y}^* = (0, 0, 0, 0, 6, -3) + (0, 0, 0, 8, 0, 0) = (0, 0, 0, 8, 6, -3)$.

Definition 3 (Two tensors product [50]): Given two 3-way tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, the result $\mathcal{C} \in \mathbb{R}^{n_1 \times n_4 \times n_3}$ of the product of \mathcal{A} and \mathcal{B} is:

$$\mathcal{C}(i, k, :) = \mathcal{A} * \mathcal{B} = \sum_{j=1}^{n_2} \mathcal{A}(i, j, :) * \mathcal{B}(j, k, :), \quad (6)$$

where $\mathcal{A}(i, j, :)$ and $\mathcal{B}(j, k, :)$ denote the $(i, j, :)$ -th tubal scalar of \mathcal{A} and $(j, k, :)$ -th tubal scalar of \mathcal{B} , respectively. They can be regarded as two n_3 -tuples \mathbf{p} and $\mathbf{q} \in \mathbb{R}^{1 \times 1 \times n_3}$. The multiplication between \mathbf{p} and \mathbf{q} represents

$$\mathbf{o}(i) = \mathbf{p} * \mathbf{q} = \sum_{l=0}^{n_3-1} \mathbf{p}(l) \mathbf{q}((i-l) \bmod (n_3)) \quad (7)$$

where $i = 0, 1, \dots, n_3 - 1$ and mod is a cotermious operator.

Definition 4 (Conjugate transpose of a tensor): Assuming $\mathcal{X} \in \mathbb{R}^{h \times v \times z}$ is a three-way tensor, its conjugate transpose is

the tensor $\text{conj}(\mathcal{X}) = \mathcal{X}^* \in \mathbb{R}^{v \times h \times z}$ obtained by conjugate transposing each of the front slices and reversing the order of transposed frontal 2 through z .

Definition 5 (Identity tensor): The identity tensor $\mathcal{I} \in \mathbb{R}^{h \times v \times z}$ is the tensor with its first frontal slice being the identity matrix and other frontal slices being all zeros.

Definition 6 (Orthogonal tensor): A three-way tensor \mathcal{Q} is orthogonal if it satisfies $\mathcal{Q}^* * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^* = \mathcal{I}$.

Definition 7 (F-diagonal tensor): A three-way tensor \mathcal{S} is f-diagonal if its slices are all diagonal matrices.

Definition 8 (Mode- k matricization/Unfolding [51], [52]): This operator noted $\text{unfold}(\mathcal{X}, k)$ reorders the elements of a tensor \mathcal{X} into a matrix $\mathbf{X}_{(k)}$. Its three unfoldings are, $\mathbf{X}_{(1)} \in \mathbb{R}^{h \times v \times z}$, $\mathbf{X}_{(2)} \in \mathbb{R}^{v \times z \times h}$ and $\mathbf{X}_{(3)} \in \mathbb{R}^{z \times h \times v}$. Inversely, $\text{fold}(\mathbf{X})$ denotes the folding of the matrix into a tensor.

Definition 9 (Tensor Trace Norm [53]): For a 3-way HSI tensor, its tensor trace norm or weighted sum of the nuclear norm (WSNN) of the mode- k unfolding matrix is:

$$\|\mathcal{X}\|_{\text{SNN}} := \sum_{k=1}^3 \alpha_k \|\mathbf{X}_{(k)}\|_* \quad (8)$$

where weights α_k satisfy $\alpha_k \geq 0$ ($k = 1, 2, 3$) and $\sum_{k=1}^3 \alpha_k = 1$.

Definition 10 (Tensor Nuclear Norm (TNN) [54]): For a tensor $\mathcal{X} \in \mathbb{R}^{h \times v \times z}$, its TNN can be expressed as the sum of nuclear norm of all the frontal slices of $\hat{\mathcal{X}}$:

$$\|\mathcal{X}\|_* := \sum_{k=1}^z \|\hat{\mathcal{X}}(:, :, k)\|_*. \quad (9)$$

Definition 11 (Mode- k permutation of a tensor [33]): For a three tensor $\mathcal{X} \in \mathbb{R}^{h \times v \times z}$, this operator noted by $\mathcal{X}^k = \text{permutation}(\mathcal{X}, k)$ changes its permutation order of (h, v, z) with k times, i.e., $\mathcal{X}^1 \in \mathbb{R}^{v \times z \times h}$, $\mathcal{X}^2 \in \mathbb{R}^{z \times h \times v}$, $\mathcal{X}^3 \in \mathbb{R}^{h \times v \times z}$. The inverse operator is defined as $\mathcal{X} = \text{ipermutation}(\mathcal{X}^k, k)$.

Definition 12 (The weighted sum of Tensor Trace Norm (WSTNN) [33]): For a 3-way HSI tensor, its weighted sum of the tensor nuclear norm (TNN) of each mode- k unfolding tensor is defined as:

$$\|\mathcal{X}\|_{\text{WSTNN}} := \sum_{k=1}^3 \alpha_k \|\mathcal{X}^k\|_* \quad (10)$$

where weights α_k satisfy $\alpha_k \geq 0$ ($k = 1, 2, 3$) and $\sum_{k=1}^3 \alpha_k = 1$.

Definition 13 (Mode- k_1, k_2 unfolding of a tensor [33]): For a m -way tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_m}$, this operator is denoted by $\mathcal{X}_{(k_1, k_2)} \in \mathbb{R}^{n_{k_1} \times n_{k_2} \times \prod_{i \neq k_1, k_2} n_i}$, whose frontal slices are the lexicographic ordering of the mode- k_1, k_2 slice of \mathcal{X} .

III. HYPERSPECTRAL l_0 TOTAL VARIATION MODEL

A. l_0 Gradient model

l_0 Gradient model was first proposed by Xu et al. [46] for natural image denoising, smoothing, enhancement and extraction because of its powerful edge-preserving property. For a 2D image $\mathbf{X} \in \mathbb{R}^{m \times n}$, the l_0 Gradient of \mathbf{X} is defined as:

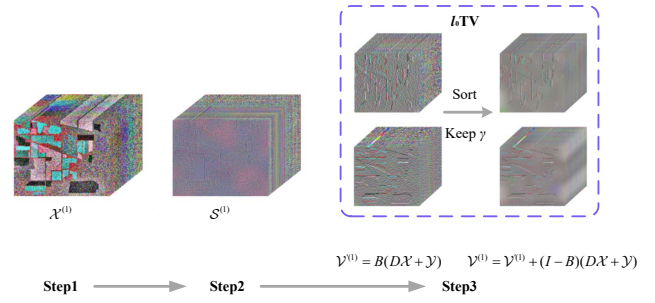


Fig. 1: First iteration result of Algorithm 1 for removing Gaussian noise.

$$\begin{aligned} l_0\text{TV}(\mathbf{X}) &= \sum C(\|D_h \mathbf{X}\|_1 + \|D_v \mathbf{X}\|_1) \\ &= \sum_i^h \sum_j^v C(|\mathbf{X}_{i+1,j} - \mathbf{X}_{i,j}| + |\mathbf{X}_{i,j+1} - \mathbf{X}_{i,j}|) \end{aligned} \quad (11)$$

where D_h and D_v are linear difference operators along image horizontal and vertical directions. $C(X)$ is a binary function used for counting the number of non-zeros image gradients and $C(X) = 1$, if $x \neq 0$; $C(X) = 0$, otherwise.

An observed 3D hyperspectral image $\mathcal{T} \in \mathbb{R}^{h \times v \times z}$ is a corrupted observation of a noise free HSI \mathcal{X} with mixed noises including Gaussian noise \mathcal{N} and sparse noise \mathcal{S} , which leads to the following degradation model:

$$\mathcal{T} = \mathcal{X} + \mathcal{S} + \mathcal{N}. \quad (12)$$

The goal of hyperspectral image denoising is to get an uncorrupted result \mathcal{X} for further HSI analysis. With the definition of $l_0\text{TV}$ (11), the HSI denoising problem is expressed as

$$\min_{\mathcal{X}, \mathcal{S}} \|\mathcal{T} - \mathcal{X} - \mathcal{S}\|_F^2 + \lambda_1 \|\mathcal{S}\|_1 + \lambda_2 \sum_{k=1}^z l_0\text{TV}(\mathcal{X}^k), \quad (13)$$

where z is the number of hyperspectral bands and \mathcal{X}^k denotes the k -th band of the hyperspectral image. $\sum_{k=1}^z l_0\text{TV}(\mathcal{X}^k)$ is a band-by-band model, which is the simplest way of extending $l_0\text{TV}$ to hyperspectral images. Each band is separately denoised and its spatial smoothness is promoted, but the intrinsic spectral correlation is ignored. Simultaneously, the same regularization parameter λ_2 results in equal regularization strength for all bands. Since each spectral band has a different noise intensity, it is difficult to select a suitable value of λ_2 is hard to select. Large values of λ_2 cause over-smoothing in the low-noise-intensity bands while small ones lead to poor noise removal in the high-noise-intensity bands.

B. $l_0\text{TV}$ model

The l_0 gradient is firstly applied to the HSI and its new formulation is given by using **definition 1**. Parameter γ denotes the l_0 gradient value of the output image itself and replaces the above-mentioned parameter λ_2 . l_0 Hyperspectral Total Variation ($l_0\text{TV}$) model is defined as

$$l_0\text{TV}(\mathcal{X}) = \sum_i^h \sum_j^v C(\sum_k^z (|\mathcal{X}_{i+1,j,k} - \mathcal{X}_{i,j,k}| + |\mathcal{X}_{i,j+1,k} - \mathcal{X}_{i,j,k}|)), \quad (14)$$

where boundary values of gradients are defined as: $\mathcal{X}_{i,j+1,k} - \mathcal{X}_{i,j,k} = 0$, if $i = h$, and $\mathcal{X}_{i+1,j,k} - \mathcal{X}_{i,j,k} = 0$, if $j = v$. $l_0\text{TV}(\mathcal{X})$ actually counts the non-zeros gradients in the spatial dimension with the assistance of spectral information, which greatly benefits for finding the spatial edges of the hyperspectral image. With **definition 1**, another formulation of $l_0\text{TV}$ becomes:

$$l_0\text{TV}(\mathcal{X}) = \|BD\mathcal{X}\|_{1,0}^\theta, \quad (15)$$

where operator D is an operator to calculate both horizontal and vertical differences. Operator B is an operator that forces boundary values of gradients to be zero when $i = h$ and $j = v$. Let us see a simple example. **Example 1:** A tensor $\mathcal{X} \in \mathbb{R}^{3 \times 3 \times 2}$ and its $BD\mathcal{X} \in \mathbb{R}^{3 \times 3 \times 2 \times 2}$ can be expressed as:

$$\mathcal{X}(:, :, 1) = \begin{bmatrix} \mathcal{X}_{111} & \mathcal{X}_{121} & \mathcal{X}_{131} \\ \mathcal{X}_{211} & \mathcal{X}_{221} & \mathcal{X}_{231} \\ \mathcal{X}_{311} & \mathcal{X}_{321} & \mathcal{X}_{331} \end{bmatrix} \quad (16)$$

$$\mathcal{X}(:, :, 2) = \begin{bmatrix} \mathcal{X}_{112} & \mathcal{X}_{122} & \mathcal{X}_{132} \\ \mathcal{X}_{212} & \mathcal{X}_{222} & \mathcal{X}_{232} \\ \mathcal{X}_{312} & \mathcal{X}_{322} & \mathcal{X}_{332} \end{bmatrix} \quad (17)$$

$$(BD\mathcal{X})(:, :, 1, 1) = \begin{bmatrix} \mathcal{X}_{121} - \mathcal{X}_{111} & \mathcal{X}_{131} - \mathcal{X}_{121} & 0 \\ \mathcal{X}_{221} - \mathcal{X}_{211} & \mathcal{X}_{231} - \mathcal{X}_{221} & 0 \\ \mathcal{X}_{321} - \mathcal{X}_{311} & \mathcal{X}_{331} - \mathcal{X}_{321} & 0 \end{bmatrix} \quad (18)$$

$$(BD\mathcal{X})(:, :, 2, 2) = \begin{bmatrix} \mathcal{X}_{122} - \mathcal{X}_{112} & \mathcal{X}_{132} - \mathcal{X}_{122} & 0 \\ \mathcal{X}_{222} - \mathcal{X}_{212} & \mathcal{X}_{232} - \mathcal{X}_{222} & 0 \\ \mathcal{X}_{322} - \mathcal{X}_{312} & \mathcal{X}_{332} - \mathcal{X}_{322} & 0 \end{bmatrix} \quad (19)$$

$$(BD\mathcal{X})(:, :, 1, 1) = \begin{bmatrix} \mathcal{X}_{211} - \mathcal{X}_{111} & \mathcal{X}_{221} - \mathcal{X}_{121} & \mathcal{X}_{231} - \mathcal{X}_{131} \\ \mathcal{X}_{311} - \mathcal{X}_{211} & \mathcal{X}_{321} - \mathcal{X}_{221} & \mathcal{X}_{331} - \mathcal{X}_{231} \\ 0 & 0 & 0 \end{bmatrix} \quad (20)$$

$$(BD\mathcal{X})(:, :, 2, 2) = \begin{bmatrix} \mathcal{X}_{212} - \mathcal{X}_{112} & \mathcal{X}_{222} - \mathcal{X}_{122} & \mathcal{X}_{232} - \mathcal{X}_{132} \\ \mathcal{X}_{312} - \mathcal{X}_{212} & \mathcal{X}_{322} - \mathcal{X}_{222} & \mathcal{X}_{332} - \mathcal{X}_{232} \\ 0 & 0 & 0 \end{bmatrix} \quad (21)$$

With **definition 1**, here $n = 9$ and each \mathbf{y}_{θ_i} is obtained by mode-1,2 unfolding $(BD\mathcal{X})_{(1,2)} \in \mathbb{R}^{9 \times 2 \times 2}$, for example, $\mathbf{y}_{\theta_1} = [\mathcal{X}_{121} - \mathcal{X}_{111}, \mathcal{X}_{122} - \mathcal{X}_{112}, \mathcal{X}_{211} - \mathcal{X}_{111}, \mathcal{X}_{212} - \mathcal{X}_{112}]^T$. It is easily seen that $l_0\text{TV}(\mathcal{X})$ is equal to $\|BD\mathcal{X}\|_{1,0}^\theta$,

This model counts the number of pixels along the horizontal and vertical direction with the assistance of spectral dimension, which actually quantifies the spectral-spatial flatness of the hyperspectral image. The above-mentioned parameter λ_2 is a weight controlling the significance of l_0 regularization, but λ_2 is hard to be determined since it has no physical meaning and has no relation to the flatness of \mathcal{X} . Therefore, the l_0 gradient value γ of \mathcal{X} is introduced to replace λ_2 and directly controls the flatness of the denoised image. The hyperspectral image denoising model based on $l_0\text{TV}$ is rewritten as:

$$\begin{aligned} \min_{\mathcal{X}, \mathcal{S}} \quad & \|\mathcal{T} - \mathcal{X} - \mathcal{S}\|_F^2 + \lambda \|\mathcal{S}\|_1, \\ \text{s.t.} \quad & \|BD\mathcal{X}\|_{1,0}^\theta \leq \gamma. \end{aligned} \quad (22)$$

C. Optimization Procedure

To apply ADMM method [55] for the aforementioned problem and introduce auxiliary variable \mathcal{V} , we introduce definition 2 to reformulate (22) as follows:

$$\begin{aligned} \min_{\mathcal{X}, \mathcal{S}, \mathcal{V}} \quad & \|\mathcal{T} - \mathcal{X} - \mathcal{S}\|_F^2 + \lambda \|\mathcal{S}\|_1 + I_{\|B \cdot\|_{1,0}^\theta}(\mathcal{V}), \\ \text{s.t.} \quad & \mathcal{V} = D\mathcal{X}. \end{aligned} \quad (23)$$

The aforementioned problem can be expressed as an unconstrained optimization problem:

$$\begin{aligned} \min_{\mathcal{X}, \mathcal{S}, \mathcal{V}} \quad & \|\mathcal{T} - \mathcal{X} - \mathcal{S}\|_F^2 + \lambda \|\mathcal{S}\|_1 + I_{\|B \cdot\|_{1,0}^\theta}(\mathcal{V}) \\ & + \beta \|\mathcal{V} - D\mathcal{X} - \mathcal{Y}\|_F^2 \end{aligned} \quad (24)$$

where β is a regularization parameter and Eq. (24) is divided into three sub-problems:

$$1) \min_{\mathcal{X}} \|\mathcal{T} - \mathcal{X} - \mathcal{S}\|_F^2 + \beta \|\mathcal{V} - D\mathcal{X} - \mathcal{Y}\|_F^2, \quad (25)$$

$$2) \min_{\mathcal{S}} \|\mathcal{T} - \mathcal{X} - \mathcal{S}\|_F^2 + \lambda \|\mathcal{S}\|_1, \quad (26)$$

$$3) \min_{\mathcal{V}} I_{\|B \cdot\|_{1,0}^\theta}(\mathcal{V}) + \beta \|\mathcal{V} - D\mathcal{X} - \mathcal{Y}\|_F^2. \quad (27)$$

Step 1: Since the function of \mathcal{X} is strictly quadratic, gradient descent can be used to find a global solution. Set the partial derivative of the function of \mathcal{X} as zero, and yield the following linear equation:

$$(1 + \beta D^* D)\mathcal{X} = \mathcal{T} - \mathcal{S} + \beta D^T \mathcal{V} - \mathcal{Y}, \quad (28)$$

where D^* indicates the adjoint operator of D . Here, Fast Fourier Transform (FFT) is applied for speedup and the solution is calculated as

$$\mathcal{X} = \text{ifft}\left(\frac{\text{fft}(\mathcal{T} - \mathcal{S} + \beta D^T \mathcal{V} - \mathcal{Y})}{1 + \beta |\text{fft}(D^* D)|^2}\right), \quad (29)$$

where fft and ifft are fast fourier transform matrix and its inverse matrix, respectively. $|\cdot|^2$ denotes element-wise square.

Step 2: Eq. (26) is rewritten as:

$$\min_{\mathcal{S}} \lambda \|\mathcal{S}\|_1 + \|\mathcal{S} - (\mathcal{T} - \mathcal{X})\|_F^2, \quad (30)$$

By introducing the soft-thresholding operator, the solution is obtained as:

$$\mathcal{S} = R_{\lambda/2}[\mathcal{T} - \mathcal{X}] \quad (31)$$

where $R_{\frac{\lambda}{2}}(\cdot)$ is the soft-thresholding operator defined as follows:

$$R_{\Delta}(x) = \begin{cases} x - \Delta, & x > \Delta \\ x + \Delta, & x < -\Delta \\ 0, & \text{otherwise.} \end{cases}$$

Step 3: Actually, the sub-problem of \mathcal{V} is identical to the

following constrained minimization problem:

$$\begin{aligned} \min_{\mathcal{V}} \quad & \|\mathcal{V} - D\mathcal{X} - \mathcal{Y}\|_F^2 \\ \text{s.t.} \quad & \|B\mathcal{V}\|_{1,0}^\theta \leq \gamma, \end{aligned} \quad (32)$$

where the parameter β is omitted because it has no influence on the optimization. Eq. (32) is solved by **Proposition 1**. The detail of the l_0 TV method is summarized in Algorithm 1. To more intuitively illustrate the effect of l_0 TV, the first iteration of Algorithm 1 for removing the Gaussian noise (SNR=10~20 dB) is given in Fig. 1. $\mathcal{V}^{(1)}, \dots, \mathcal{V}_{\theta_\gamma}^{(1)}$ is sorted in descending order and $\mathcal{V}_{\theta_{\gamma+1}}^{(1)} = 0, \dots, \mathcal{V}_{\theta_n}^{(1)} = 0$ are set as zero vectors. A scalar $\eta \in [1, 1.5]$ is introduced for gradually increasing the value of β , which leads to ADMM stability for non-convex optimization. This can be supported by recent ADMM studies for non-convex cases [56], [57]. The results by ADMM converges to a stationary point under appropriate conditions. Similar strategies are also extended in existing image processing optimization algorithms [30], [31], [34], [43], [45].

Algorithm 1 l_0 TV for HSI mixed denoising

Input: $\mathcal{T}, \lambda, \gamma$

- 1: **Initialization:** $\mathcal{X} = \mathcal{S} = 0, \mathcal{V} = \mathcal{Y} = 0, \text{maxiter} = 200, i = 0, \epsilon = 10^{-4} * hv, \eta \in [1, 1.5]$
- 2: **while** $0 \leq i \leq \text{maxiter}$ or $\|BD\mathcal{X}\|_{1,0}^\theta - \gamma| > \epsilon$
- 3: $i = i + 1;$
- 4: update \mathcal{X} by solving Eq. (29):
 $\mathcal{X} = \text{ifft}(\frac{\text{fft}(\mathcal{T} - \mathcal{S} + \beta D^T \mathcal{V} - \mathcal{Y})}{1 + \beta \|\text{fft}(D)\|^2})$
- 5: update \mathcal{S} by solving Eq. (31):
 $\mathcal{S} = R_{\lambda/2}[\mathcal{T} - \mathcal{X}]$
- 6: update \mathcal{V} by solving Eq. (32):
- 7: $\mathcal{V}' = B(D\mathcal{X} + \mathcal{Y})$,
- 8: Sort $\mathcal{V}'_{\theta_1}, \dots, \mathcal{V}'_{\theta_n}$ in descending order in terms of l_2 norms, let $\mathcal{V}'_{\theta_{\gamma+1}} = 0, \dots, \mathcal{V}'_{\theta_n} = 0$ and then get a new \mathcal{V}' ,
- 9: $\mathcal{V} = \mathcal{V}' + (I - B)(D\mathcal{X} + \mathcal{Y})$
- 10: update the multipliers
 $\mathcal{Y} = \mathcal{Y} + (D\mathcal{X} - \mathcal{V})$
- 11: $\beta = \eta * \beta$
- 12: **end while**

Output: \mathcal{X} .

IV. TENSOR LOW-RANK CONSTRAINT AND l_0 HYPERSPECTRAL TOTAL VARIATION

In this section, the mathematical formulation of our proposed method is presented systematically and its framework is illustrated in Fig. 2.

A. TLR- l_0 TV model

As previously stated, by combining l_0 TV model and tensor low-rank constraint, we propose a novel hyperspectral mixed

denoising model, termed as TLR- l_0 TV, which is formulated as:

$$\begin{aligned} \min_{\mathcal{X}, \mathcal{S}} \quad & \sum_{k=1}^m \alpha_k E_k(\mathcal{X})_\omega + \lambda \|\mathcal{S}\|_1 + \mu \|\mathcal{T} - \mathcal{X} - \mathcal{S}\|_F^2, \\ \text{s.t.} \quad & \|BD\mathcal{X}\|_{1,0}^\theta \leq \gamma, \end{aligned} \quad (33)$$

where λ and μ are two positive regularization parameters. Instead of a simple 2-D low-rank matrix or a 3-D tensor, the TLR model $\sum_{k=1}^m \alpha_k E_k(\mathcal{X})$ uses tensor low-rank property for obtaining uncorrupted part from the noisy hyperspectral image. The functions $E_k(\mathcal{X})_\omega$ are set to be $\|\mathbf{X}_{(k)}\|_{\omega,*}$ in WSWNN- l_0 TV-based method and $\|\mathcal{X}^k\|_{\omega,*}$ in WSWTNN- l_0 TV-based method. For the second term $\|\mathcal{S}\|_1$, sparse noises including salt and pepper noise, dead-line noise are detected by l_1 norm. The Frobenius norm $\|\mathcal{T} - \mathcal{X} - \mathcal{S}\|_F^2$ is used to suppress Gaussian noise. As mentioned in Section III-B, $\|BD\mathcal{X}\|_{1,0}^\theta \leq \gamma$ is another expression of l_0 TV(\mathcal{X}), which keeps more image structure details and further remove Gaussian noise.

B. Optimization Procedure

The Augmented Lagrange Multiplier (ALM) [58] framework is developed to solve problem (33). By introducing auxiliary variables \mathcal{V} and \mathcal{M}_k with $k = 1, 2, 3$, Eq. (33) is rewritten into the following equivalent minimization:

$$\begin{aligned} \min_{\mathcal{X}, \mathcal{M}_k, \mathcal{S}, \mathcal{V}} \quad & \sum_{k=1}^m \alpha_k E_k(\mathcal{M})_\omega + \lambda \|\mathcal{S}\|_1 + I_{\|B \cdot\|_{1,0}^\theta}(\mathcal{V}) \\ & + \mu \|\mathcal{T} - \mathcal{X} - \mathcal{S}\|_F^2, \\ \text{s.t.} \quad & \mathcal{M}_1 = \mathcal{M}_2 = \mathcal{M}_3 = \mathcal{X}, \mathcal{V} = D\mathcal{X}. \end{aligned} \quad (34)$$

With the ALM framework, Eq. (34) is transformed into its corresponding Lagrangian function:

$$\begin{aligned} \min_{\mathcal{X}, \mathcal{M}_k, \mathcal{S}, \mathcal{V}} \quad & \sum_{k=1}^m \alpha_k E_k(\mathcal{M}_k)_\omega + \lambda \|\mathcal{S}\|_1 + \mu \|\mathcal{T} - \mathcal{X} - \mathcal{S}\|_F^2 + I_{\|B \cdot\|_{1,0}^\theta}(\mathcal{V}) \\ & + \sum_{k=1}^3 \langle \mathcal{Y}_{1k}, \mathcal{X}_k - \mathcal{M}_k \rangle + \frac{\beta}{2} \|\mathcal{X}_k - \mathcal{M}_k\|_F^2 \\ & + \langle \mathcal{Y}_2, D\mathcal{X} - \mathcal{V} \rangle + \frac{\beta}{2} \|D\mathcal{X} - \mathcal{V}\|_F^2. \end{aligned} \quad (35)$$

Eq. (35) can be divided into four sub-problems and be solved by the following steps:

Step 1: The $\mathcal{M}_k, k = 1, 2, 3$ -related sub-problem is given as:

$$\min_{\mathcal{M}_k} \frac{\alpha_k}{\beta} E_k(\mathcal{M}_k)_\omega + \frac{1}{2} \|\mathcal{M}_k - (\mathcal{X} + \frac{\mathcal{Y}_{1k}}{\beta})\|_F^2, \quad (36)$$

For WSWNN form, Eq.(36) can be rewritten as:

$$\min_{\mathcal{M}_k} \frac{\alpha_k}{\beta} \|\mathbf{M}_{(k)}\|_{\omega,*} + \frac{1}{2} \|\mathcal{M}_k - (\mathcal{X} + \frac{\mathcal{Y}_{1k}}{\beta})\|_F^2. \quad (37)$$

Eq. (37) is a WNNM problem, which is solved by the following several steps [59]. (1) Calculate the singular value matrix Σ of **SVD**($\text{unfold}(\mathcal{X} + \frac{\mathcal{Y}_{1k}}{\beta}, k)$) = $[\mathbf{U}, \Sigma, \mathbf{V}]$. (2) Apply the weighted shrinking operator $\mathbf{R}_{\frac{\alpha_k}{\beta}}(\mathcal{X} + \mathcal{Y}_1^k) = \text{diag}((\Sigma(i, i) - \frac{\alpha_k}{\beta} \omega_i)_+)$ to the above singular value matrix Σ , where ω_i is the weight calculated by $\omega_i = \frac{1}{\Sigma(i, i) + \epsilon}$, $\epsilon > 0$

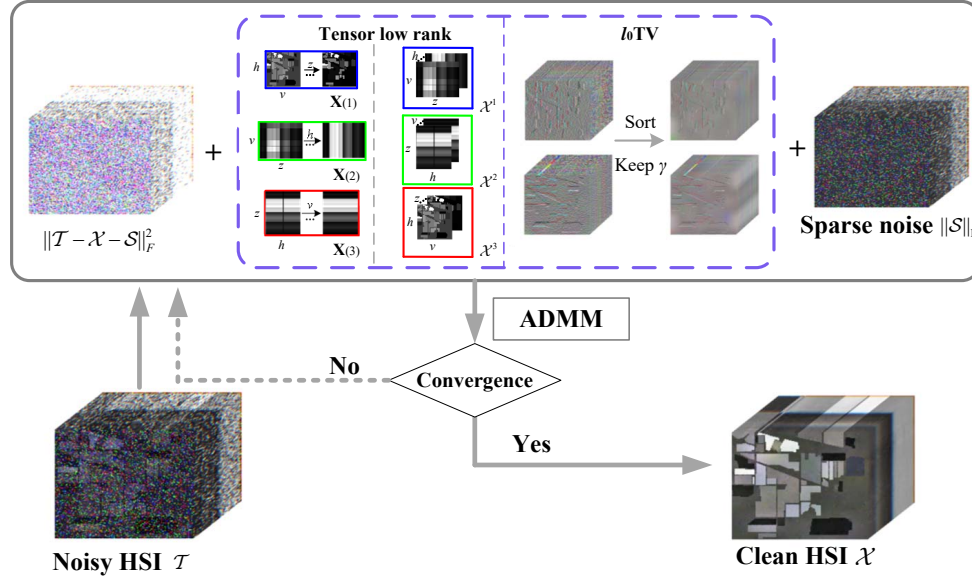


Fig. 2: The flow chart of the proposed TLR- l_0 TV model.

Algorithm 2 Weighted tensor singular value thresholding

Input: $\mathcal{L}_k \in \mathbb{R}^{h \times v \times z}, \frac{\alpha_k}{\beta}$

- 1: $\hat{\mathcal{L}}_k = \text{fft}(\mathcal{L}_k, [], 3)$;
- 2: **for** $i = 0, 1, \dots, \lfloor \frac{z+1}{2} \rfloor$
- 3: $[\hat{\mathcal{U}}(:, :, i), \hat{\mathcal{S}}(:, :, i), \hat{\mathcal{V}}(:, :, i)] = \text{SVD}(\hat{\mathcal{L}}_k(:, :, i))$;
- 4: $\omega = \frac{1}{\text{diag}(\hat{\mathcal{S}}(:, :, i)) + \epsilon}$;
- 5: $\hat{\mathcal{S}}_{new}(:, :, i) = \max(\hat{\mathcal{S}}(:, :, i) - \frac{\alpha_k}{\beta} \omega, 0)$;
- 6: **end for**
- 7: **for** $i = \lfloor \frac{z+1}{2} \rfloor + 1, \dots, z$
- 8: $\hat{\mathcal{U}}(:, :, i) = \text{conj}(\hat{\mathcal{U}}(:, :, z - i + 2))$;
- 9: $\hat{\mathcal{S}}_{new}(:, :, i) = \text{conj}(\hat{\mathcal{S}}_{new}(:, :, i))$;
- 10: $\hat{\mathcal{V}}(:, :, i) = \text{conj}(\hat{\mathcal{V}}(:, :, z - i + 2))$;
- 11: **end for**
- 12: $\mathcal{U} = \text{ifft}(\hat{\mathcal{U}}, [], 3)$, $\mathcal{S}_{new} = \text{ifft}(\hat{\mathcal{S}}_{new}, [], 3)$, $\mathcal{V} = \text{ifft}(\hat{\mathcal{V}}, [], 3)$;
- 13: $\mathcal{M}_k^k = \mathcal{U} * \mathcal{S}_{new} * \mathcal{V}^*$;
- 14: $\mathcal{M}_k = \text{ipermutation}(\mathcal{M}_k^k, k)$;

Output: \mathcal{M}_k .

is a small positive value. (3) The estimated $\mathbf{M}_{(k)}$ and \mathcal{M}_k are constructed by:

$$\mathbf{M}_{(k)} = \mathbf{U} \mathbf{R}^{\frac{\alpha_k}{\beta}} (\mathcal{X} + \frac{\mathcal{Y}_{1k}}{\beta}) \mathbf{V}^T, \quad \mathcal{M}_k = \text{fold}(\mathbf{M}_{(k)}, k). \quad (38)$$

For WSWTNN form, Eq.(36) can be transformed as:

$$\min_{\mathcal{M}_k} \frac{\alpha_k}{\beta} \|\mathcal{M}_k^k\|_{\omega, *} + \frac{1}{2} \|\mathcal{M}_k - (\mathcal{X} + \frac{\mathcal{Y}_{1k}}{\beta})\|_F^2. \quad (39)$$

Eq. (39) is the mode- k permutation of tensor \mathcal{M}_k . Let $\mathcal{L}_k = \mathcal{X} + \frac{\mathcal{Y}_{1k}}{\beta}$. The weighted tensor singular value thresholding method can be used to solve Eq. (39). The details are shown in Algorithm 2.

Step 2: The \mathcal{S} -related sub-problem is rewritten as:

Algorithm 3 TLR- l_0 TV for HSI mixed denoising

Input: $\mathcal{T}, \alpha_k, \lambda, \gamma, \mu, \beta$

- 1: **Initialization:** $\mathcal{X} = \mathcal{S} = 0, \mathcal{M}_k = \mathcal{Y}_{1k} = 0, \mathcal{V} = \mathcal{Y}_2 = 0$, $\text{maxiter} = 200, i = 0, \epsilon = 10^{-4} * hv, \eta \in [1, 1.5]$
- 2: **while** $0 \leq i \leq \text{maxiter}$ or $||BD\mathcal{X}'||_{1,0}^0 - \gamma| > \epsilon$
- 3: $i = i + 1$;
- 4: update \mathcal{M}_k by solving Eq. (36):
- 5: For the WSWNN problem:
 $\mathbf{M}_{(k)} = \mathbf{U} \text{diag}((\Sigma(i, i) - \omega_i)_+) \mathbf{V}^T, \mathcal{M}_k = \text{fold}(\mathbf{M}_{(k)})$;
- 6: Solve the WSWTNN problem by Algorithm 2,
- 7: update \mathcal{S} by solving Eq. (40):
 $\mathcal{S} = R_{\lambda/2\beta}[\mathcal{T} - \mathcal{X}]$,
- 8: update \mathcal{V} by solving Eq. (41):
 $\mathcal{V}' = B(D\mathcal{X} + \frac{\mathcal{Y}_2}{\beta})$,
- 9: Sort $\mathcal{V}'_{\theta_1}, \dots, \mathcal{V}'_{\theta_n}$ in descending order in terms of l_2 norms, let $\mathcal{V}'_{\theta_{\gamma+1}} = 0, \dots, \mathcal{V}'_{\theta_n} = 0$ and then get a new \mathcal{V}' ,
- 10: $\mathcal{V} = \mathcal{V}' + (I - B)(D\mathcal{X} + \frac{\mathcal{Y}_2}{\beta})$,
- 11: update \mathcal{X} by solving Eq. (44):
 $\mathcal{X} = \text{ifft}(\frac{\text{fft}(\mu(\mathcal{T} - \mathcal{S}) + \beta \sum_{k=1}^3 (\mathcal{M}_k - \frac{\mathcal{Y}_{1k}}{\beta} + \beta D^* (\mathcal{V} - \frac{\mathcal{Y}_2}{\beta})))}{\mu + 3\beta + \beta |\text{fft}(D^* D)|^2})$,
- 12: update the multipliers
 $\mathcal{Y}_1^k = \mathcal{Y}_1^k + \beta(\mathcal{X}_k - \mathcal{M}_k)$,
 $\mathcal{Y}_2 = \mathcal{Y}_2 + \beta(D\mathcal{X} - \mathcal{V})$,
- 13: $\gamma = \eta * \gamma$,
- 14: **end while**
- 15: **Output:** \mathcal{X} .

$$\begin{aligned} & \underset{\mathcal{S}}{\text{argmin}} \mu \|\mathcal{T} - \mathcal{X} - \mathcal{S}\|_F^2 + \lambda \|\mathcal{S}\|_1 \\ & = R_{\frac{\lambda}{2\mu}}[\mathcal{T} - \mathcal{X}]. \end{aligned} \quad (40)$$

Step 3: The \mathcal{V} -related sub-problem has the following formu-

TABLE II: The added mixed noises in six cases.

Noise Case	PSNR(dB)
Case 1	Gaussian (SNR = 10 ~ 20 dB)
Case 2	Gaussian (SNR = 10 ~ 20 dB)+salt and pepper(10%) in all bands
Case 3	Gaussian (SNR = 10 ~ 20 dB)+salt and pepper(20%) in all bands + four deadlines in each of four random bands
Case 4	Gaussian (mean SNR = 12.25 dB, and noise variance is calculated by the Eq.(49))
Case 5	Gaussian (mean SNR = 12.25 dB, and noise variance is calculated by the Eq.(49)) +salt and pepper(10%) in all bands
Case 6	Gaussian (mean SNR = 12.25 dB, and noise variance is calculated by the Eq.(49))+salt and pepper(20%) in all bands + eight deadlines in each of eight random bands
Case 7	Gaussian (zero-mean, and noise variance is different for different bands and is selected from 0.1~0.2 randomly)
Case 8	Gaussian (zero-mean, and noise variance is different for different bands and is selected from 0.1~0.2 randomly) +salt and pepper(selected from 0 to 20% randomly) in all bands
Case 9	Gaussian (zero-mean, and noise variance is different for different bands and is selected from 0.1~0.2 randomly) +salt and pepper(selected from 0 to 20% randomly) in all bands + dead-lines in 61~70 bands (with the number of stripes randomly selected from 1 to 10 and the width of the dead-lines randomly generated from 1 to 3)

lation:

$$\underset{\mathcal{V}}{\operatorname{argmin}} I_{\|B \cdot\|_{1,0}^q}(\mathcal{V}) + \mu \|\mathcal{V} - D\mathcal{X} - \frac{\mathcal{Y}_2}{\beta}\|_F^2. \quad (41)$$

As mentioned in section III-C, the solution of Eq. (41) is also obtained by **Proposition 1**.

Step 4: The \mathcal{X} -related sub-problem is reformulated as:

$$\begin{aligned} \min_{\mathcal{X}} \mu \|\mathcal{T} - \mathcal{X} - \mathcal{S}\|_F^2 + \beta \sum_{k=1}^3 \|\mathcal{M}_k - \mathcal{X} - \frac{\mathcal{Y}_{1k}}{\beta}\|_F^2 \\ + \beta \|\mathcal{V} - D\mathcal{X} - \frac{\mathcal{Y}_2}{\beta}\|_F^2 \end{aligned} \quad (42)$$

Eq. (42) can be solved by using the following normal equation:

$$(\mu + 3\beta + \beta D^* D)\mathcal{X} = \mu(\mathcal{T} - \mathcal{S}) + \beta \sum_{k=1}^3 (\mathcal{M}_k - \frac{\mathcal{Y}_{1k}}{\beta}) + \beta D^* (\mathcal{V} - \frac{\mathcal{Y}_2}{\beta}) \quad (43)$$

By FFT, the final solution \mathcal{X} is expressed as:

$$\mathcal{X} = \operatorname{ifft}\left(\frac{\operatorname{fft}(\mu(\mathcal{T} - \mathcal{S}) + \beta \sum_{k=1}^3 (\mathcal{M}_k - \frac{\mathcal{Y}_{1k}}{\beta} + \beta D^* (\mathcal{V} - \frac{\mathcal{Y}_2}{\beta})))}{\mu + 3\beta + \beta |\operatorname{fft}(D^* D)|^2}\right). \quad (44)$$

The aforementioned description of the proposed TLR- l_0 TV model via ADMM is summarized as Algorithm 3.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, both simulated and real data experiments on HSI denoising are conducted to verify the performances of the proposed TLR- l_0 TV. Here, six different state-of-art HSI denoising methods are selected for comparison, i.e., Low-Rank Matrix Recovery (LRMR) [19], TV regularized Low-Rank method (LRTV) [38], Spatio-Spectral Total Variation (SSTV) [18], low-rank Tensor Approximation (LRTA) [17], TV regularized low-rank Tensor Decomposition (LRTDTV) [13], Non-local Meets Global (NGmeet) [37], HSI Spatial-Spectral Deep Residual Convolutional Neural Network (HSID-CNN) [25], and Multidirectional Low-Rank Modeling and SpatialSpectral Total Variation (MLR-SSTV) [44].

A. Simulated HSI data Experiments

1)Data description and noise Configuration: The Indian synthetic (Indian-s) data is generated by USGS splib06 using

the ground truth of the Indian Pines dataset, which is also considered in [19], [43], [60]. The Indian-s data consists of 17 pure endnumbers extracted from USGS splib06. The whole data contains 145×145 pixels and 175 bands. To qualitatively evaluate the denoised results, we adopt four evaluation metrics including the Mean Peak Signal to Noise Ratio (MPSNR), Mean Structural Similarity Index (MSSIM), the Erreur Relative Globale Adimensionnelle de Synthèse (ERGAS):

$$\text{MPSNR} = \frac{1}{z} \sum_{i=1}^z \text{PSNR}(\mathcal{X}(:, :, i), \bar{\mathcal{X}}(:, :, i)), \quad (45)$$

$$\text{MSSIM} = \frac{1}{z} \sum_{i=1}^z \text{SSIM}(\mathcal{X}(:, :, i), \bar{\mathcal{X}}(:, :, i)). \quad (46)$$

$$\text{ERGAS} = \sqrt{\frac{1}{z} \sum_{i=1}^z \frac{\text{mse}(\mathcal{X}(:, :, i), \bar{\mathcal{X}}(:, :, i))}{\text{Mean}_2(\mathcal{X}(:, :, i))}}, \quad (47)$$

and the mean spectral angle distance (MSAD) of all the spectral signatures between noisy and denoised images is calculated as follows and its unit is degree (i. e. deg.) [40], [61], [62]:

$$\text{MSAD} = \frac{1}{hv} \sum_{i=1}^{hv} \frac{180}{\pi} \times \arccos \frac{(\mathbf{X}_{(3)}(i, :))^T \cdot (\bar{\mathbf{X}}_{(3)}(i, :))}{\|\mathbf{X}_{(3)}(i, :)\| \cdot \|\bar{\mathbf{X}}_{(3)}(i, :)\|}, \quad (48)$$

where $\mathcal{X}(:, :, i)$ and $\bar{\mathcal{X}}(:, :, i)$ denote the i th band of the reference and denoised images, and $\mathbf{X}_{(3)}(i, :)$ and $\bar{\mathbf{X}}_{(3)}(i, :)$ are the i th spectral signatures of two images. The operator $\text{Mean}_2(\mathbf{A})$ computes the mean of the values in matrix \mathbf{A} .

To simulate noisy HSI data, Gaussian noise, salt and pepper noise, dead-line noise and strips are added to the Indian-s data. The details of noise cases are given in the Table II and for the CASE 4~6, the variance σ_z^2 of Gaussian noise is defined as [49]

$$\sigma_z^2 = \delta^2 \frac{\exp\{-(z - Z/2)^2 / (2\xi^2)\}}{\sum_{z=1}^Z \exp\{-(z - Z/2)^2 / (2\xi^2)\}} \quad (49)$$

where δ and ξ are two adjustable parameters and the mean noise SNR is set as 12.34 dB.

2)Comparison with Other Methods: TLR- l_0 TV methods are compared with the other HSI denoising methods on simulated Indian-s data. Table III presents the results of mixed noise removal by all the compared methods, where the best ones are labeled in bold and the second-best ones are underlined.

TABLE III: Quantitative results of all the methods under different mixed noise cases for Indian-s dataset.

Noise Case	Index	Noisy	Indian-s Dataset									
			LRMR	LRTV	SSTV	LRTA	LRTDTV	NGmeets	HSID-CNN	MLR-SSTV	WSWNN- l_0 TV	WSWTNN- l_0 TV
Case 1	MPSNR	26.34	41.50	44.39	42.11	28.95	45.70	40.51	37.87	43.50	<u>48.36</u>	50.97
	MSSIM	0.5125	0.9554	0.9895	0.9706	0.6185	0.9947	0.9906	0.9271	0.9841	<u>0.9983</u>	0.9989
	ERAS	11.4981	1.7790	1.7970	1.6352	8.5190	1.4964	2.1202	3.7806	1.4389	<u>0.9809</u>	0.6559
	MSAD	11.1090	1.8373	1.6352	1.7533	7.7411	1.3759	2.3569	2.5400	1.5326	<u>0.8849</u>	0.6057
Case 2	MPSNR	14.10	40.18	43.25	40.76	26.41	43.91	29.86	28.97	42.97	<u>47.30</u>	48.76
	MSSIM	0.1416	0.9362	0.9868	0.9602	0.7243	0.9905	0.9537	0.7143	0.9784	<u>0.9979</u>	0.9982
	ERAS	46.9504	11.1352	17.8865	3.5510	3.6063	<u>3.2904</u>	8.3894	9.2989	4.4786	<u>3.4705</u>	2.9667
	MSAD	34.4532	2.1259	1.7792	2.0229	9.3239	1.5426	5.6182	7.6165	4.5443	<u>0.9289</u>	0.7438
Case 3	MPSNR	11.42	39.15	40.51	39.78	22.85	42.80	24.52	24.16	42.05	<u>45.97</u>	46.55
	MSSIM	0.0742	0.9332	0.9827	0.9565	0.6539	0.9893	0.8858	0.5528	0.9734	<u>0.9970</u>	0.9971
	ERAS	63.0108	2.3057	1.9672	2.1835	18.5049	1.5383	16.4128	16.3973	1.6359	<u>1.1617</u>	1.0183
	MSAD	39.9088	2.3737	2.3184	2.4094	11.9728	1.5287	16.4128	11.0080	1.7363	<u>1.1658</u>	1.0323
Case 4	MPSNR	23.51	38.84	41.52	39.17	28.03	42.17	39.58	35.56	40.90	<u>46.04</u>	47.67
	MSSIM	0.4147	0.9081	0.9667	0.9682	0.5869	0.9663	0.9883	0.8608	0.9584	<u>0.9970</u>	0.9974
	ERAS	16.0971	2.5499	2.4073	2.3333	9.3715	1.7867	2.2857	4.4169	1.8937	<u>1.1253</u>	0.8939
	MSAD	14.6163	2.4827	2.0545	2.3549	8.3010	1.6282	2.5483	3.2425	1.9617	<u>1.1318</u>	0.9018
Case 5	MPSNR	13.62	37.91	39.72	38.14	26.33	40.83	29.36	28.29	39.82	<u>45.45</u>	46.79
	MSSIM	0.1264	0.8965	0.9647	0.9075	0.7354	0.9553	0.9501	0.6911	0.9394	<u>0.9965</u>	0.9967
	ERAS	16.0887	2.8033	2.9360	2.6097	10.9828	2.2039	8.5472	9.5594	2.1487	<u>1.1940</u>	0.9934
	MSAD	36.5602	2.7355	2.4864	2.6529	9.0311	1.9092	5.8010	7.8467	2.1635	<u>1.2042</u>	0.9925
Case 6	MPSNR	11.13	36.68	38.51	37.04	22.67	39.04	23.97	23.64	38.73	<u>44.41</u>	44.98
	MSSIM	0.0681	0.8757	0.9444	0.9143	0.6879	0.9623	0.8642	0.5291	0.9272	<u>0.9959</u>	0.9965
	ERAS	63.7664	3.2670	4.6684	2.9260	18.3834	3.1572	16.7900	17.1243	2.4363	<u>1.4017</u>	1.2770
	MSAD	35.5602	3.1039	3.0536	3.0973	11.9516	2.5337	9.6684	11.3056	2.4338	<u>1.3089</u>	1.1878
Case 7	MPSNR	16.07	34.66	37.69	32.04	27.96	38.16	40.06	33.24	39.40	<u>40.88</u>	42.27
	MSSIM	0.1499	0.8819	0.9412	0.7656	0.6828	0.9594	0.9880	0.8671	0.9680	<u>0.9926</u>	0.9927
	ERAS	36.0327	3.7659	4.2930	5.9355	8.7951	2.7124	2.1434	5.3771	2.2891	<u>2.1258</u>	1.6963
	MSAD	32.9956	4.2082	3.3816	6.6732	9.7845	2.8561	2.4015	4.8156	2.5384	<u>2.3278</u>	1.8892
Case 8	MPSNR	12.93	33.86	36.54	31.32	23.99	36.83	27.02	25.87	37.51	<u>40.45</u>	41.29
	MSSIM	0.0956	0.8631	0.9383	0.7403	0.6173	0.9703	0.9073	0.6307	0.9614	<u>0.9920</u>	0.9913
	ERAS	54.4352	4.1185	4.7792	6.3556	17.4364	3.1778	14.4709	15.2208	3.0007	<u>2.1868</u>	1.8802
	MSAD	36.5381	4.5191	3.4203	7.0768	11.8623	3.1017	8.1556	9.6958	2.9030	<u>2.3947</u>	2.0886
Case 9	MPSNR	12.87	33.78	36.16	31.22	23.15	36.18	26.79	25.63	37.30	<u>40.24</u>	40.81
	MSSIM	0.0951	0.8615	0.9267	0.7388	0.5400	0.9528	0.9035	0.6250	0.9678	<u>0.9918</u>	0.9907
	ERAS	54.5384	4.1608	5.3751	6.3824	18.4054	3.4232	14.6071	15.3156	3.5446	<u>2.2145</u>	1.9825
	MSAD	36.5962	4.5275	3.6142	7.1007	12.9351	3.4077	8.2190	9.8136	3.0214	<u>2.4002</u>	2.1698

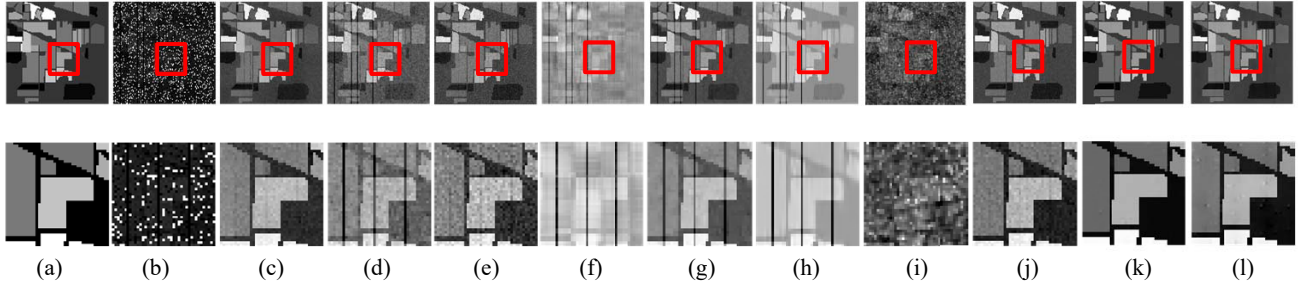


Fig. 3: Denoised results by all the compared methods: (a) original band 15, (b) simulated noise band of case 6, (c) LRMR, (d) LRTV, (e) SSTV, (f) LRTA, (g) LRTDTV, (h) NGmeet, (i) HSID-CNN, (j) MLR-SSTV, (k) WSWNN- l_0 TV, (l) WSWTNN- l_0 TV.

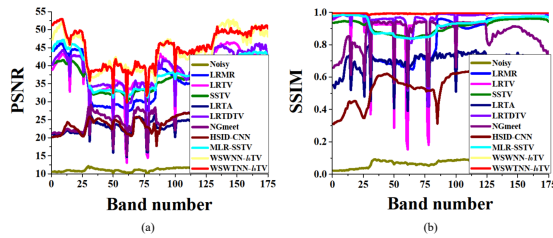


Fig. 4: PSNR and SSIM values of each band in Case 6: (a) PSNR, (b) SSIM.

Overall, our proposed TLR- l_0 TV methods achieve the highest MPSNR, MSSIM, ERAS, and MSAD values. LRTA fails to

remove mixed noise and has the worst performance among all compared methods. NGmeet and HSID-CNN get higher metric values than LRTA, especially in Gaussian noise cases. But these two methods also hardly have good performances for mixed noise removal since sparse noise is ignored in their modelings. LRMR rearranges an HSI data into a 2-D matrix and loses its effectiveness in cases 7 ~ 9. Similarly, the quantitative evaluation results of SSTV in cases 7 ~ 9 are also worse than its other cases. Obviously, pure LR, TLR, or TV methods appear unsuitable for the complicated Gaussian noise distribution like cases 7 ~ 9 since they only consider one type of HSI prior knowledge. LRTV combines low-rank matrix property with a band-by-band TV regularization, which leads to better results than LRTA, LRMR, and SSTV. LRTDTV

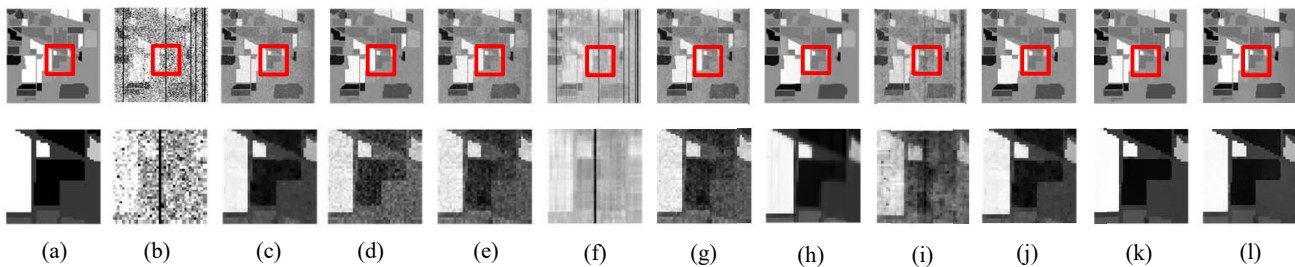


Fig. 5: Denoised results by all the compared methods: (a) original band 65, (b) simulated noise band of case 9, (c) LRMR, (d) LRTV, (e) SSTV, (f) LRTA, (g) LRTDTV, (h) NGmeet, (i) HSID-CNN, (j) MLR-SSTV, (k) WSWNN- l_0 TV, (l) WSWTNN- l_0 TV.

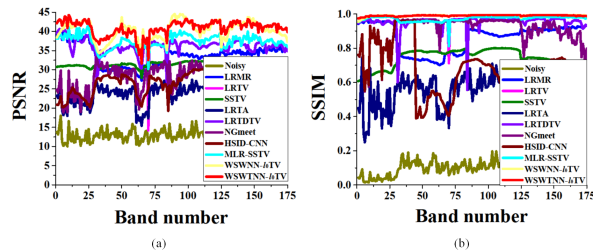


Fig. 6: PSNR and SSIM values of each band in Case 9: (a) PSNR, (b) SSIM.

explores the low-rank tensor property and inserts a 3DATV regularization into a low-rank Tucker decomposition model. Therefore, LRTDTV achieves better metric scores than the first four approaches. Due to the introduction of the l_1 TV norm, the performance of MLR-SSTV remains limited on the recovery of the image texture structure. However, our proposed models obtain the best quantitative evaluation scores and this improvement demonstrates the contributions of the tensor low-rank and the l_0 TV prior knowledge. Especially, l_0 TV regularization is excellent for removing Gaussian noise, as the results of cases 1, 4 and 7 show.

In Fig.3 and Fig. 5, the performances of seven competing methods are further visually compared under two mixed noise cases 6 and 9. The denoised results by LRMR, LRTV and LRTA still remain a lot of Gaussian noise and dead-line noise. The results by NGmeet and HSID-CNN contain unexpectedly spectral distortion since they can hardly cope with the mixed noise cases. SSTV, LRTDTV, and MLR-SSTV perform comparatively better, but their results contain some evident artifacts, hindering further HSI analysis work like HSI classification. The proposed TLR- l_0 TV models remove most of the mixed noise and preserve clear image details like sharp image edges.

Fig. 4 and Fig. 6 give the PSNR and SSIM values of each band in cases 6 and 9, respectively. The PSNR and SSIM values of LRTA shown in blue are lower than the other models. The curves of LRMR and LRTV vary similarly, and they perform worse suddenly when they encounter the heavy dead-line noises. In addition, many serious fluctuations exist in the curves of NGmeet and HSID-CNN, especially for those bands with salt and pepper noise and dead-lines. Due to the considerations of the l_1 TV norms, LRTDTV and

MLR-SSTV get much higher PSNR and SSIM values than the aforementioned algorithms. Our proposed WSWNN- l_0 TV and WSWTNN- l_0 TV shown in yellow and red provide the best performances among all the compared methods for almost every band.

3)Discussion: The proposed methods achieve excellent performances on the Indian-s dataset. This dataset has exact image edges and clear texture, which is beneficial for demonstrating the superiority of our methods. A sub-image of the Pavia City dataset of size $128 \times 128 \times 80$ is adapted as another simulated testing data. Fig. 7 shows the visual results of band 65 in HSI Pavia City under case 6. It is easily seen that the texture information and image edges are not as clear as those in Indian-s data. The methods without spatial smoothness or sparsity constraint fail to completely get rid of all mixed noise. HSID-CNN, MLR-SSTV, and TLR- l_0 TV are capable of better removing mixed noise, whereas MLR-SSTV and TLR- l_0 TV preserve the image details and characterize the smooth structure better than SSTV since these methods consider two types of constraints. The different image properties of these two datasets lead to the different available improvement spaces of PSNR values. The proposed method is more suitable for handling the HS dataset with more classification classes and image edges. Therefore, in the next section, we select three real datasets with classification maps and design the classification experiments.

4)Effectiveness of l_0 TV: We prepare two simulated experiments to verify the effectiveness of l_0 TV and TLR constraint, respectively. As Fig. 8 (b) shows, the signal-to-noise ratio (SNR) value varies from 10 to 20 dB randomly for each band, and the mean SNR value of all bands is 15.16 dB. l_0 TV improves the denoising result more than band-by-band TV and SSTV. The result by band-by-band TV has spatial over-smoothing, and SSTV causes artifacts. This is the fatal flaw of the SSTV method for the consequent HIS work classification. Even combined with the MLR models, SSTV still causes artifacts in some degree (Fig. 5 (j)), which hinders the effectiveness of the MLR-SSTV models. In contrast, l_0 TV keeps sharp image edges without artifacts. Simultaneously, the over-smoothing of the denoised result by l_0 TV is avoided since spectral information of all bands is also considered.

l_0 TV leads to a better-denoised result in the Gaussian noise case. However, it does not always perform better than the others. As shown in Fig. 9, the Gaussian noise (SNR=10~20 dB), salt and pepper noise (5% randomly in all bands), and

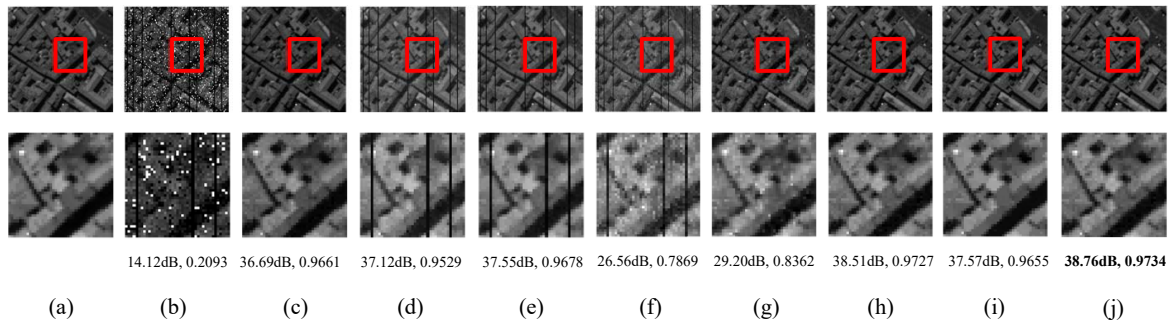


Fig. 7: Denoised results with their (Left) MPSNR and (Right) MSSIM: (a) original band 65, (b) simulated noise band of case 6, (c) LRM, (d) LRTV, (e) LRTDTV, (f) NGmeet, (g) HSID-CNN, (h) MLR-SSTV, (i) WSWNN- l_0 TV, (j) WSWTNN- l_0 TV.

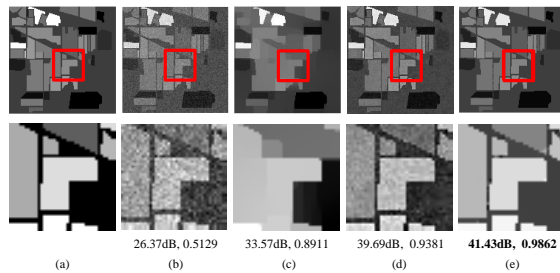


Fig. 8: An example of the hyperspectral denoising results with their PSNR (left) and SSIM (right). (a)Original band 1, (b) noisy band 1 with a Gaussian noise (SNR=10~20dB), (c) band-by-band TV, (d) SSTV, (e) l_0 TV.

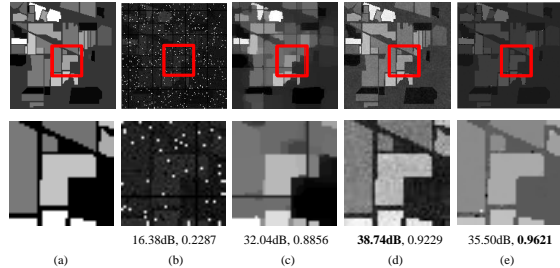


Fig. 9: An example of the hyperspectral denoising results with their PSNR (left) and SSIM (right). (a)Original band 15, (b) noisy band 15 with the mixed noise, (c) band-by-band TV, (d) SSTV, (e) l_0 TV.

dead-line noise (4 dead-lines in each of 4 random bands) are added into the first original hyperspectral image. The degradation in image quality for denoised results with more noises is intuitively observed from their PSNR values. Nevertheless, l_0 TV suffers a more severe decay. This TV focuses on finding the non-zeros gradients for preserving the image edges, but it is excessively sensitive to image edges. Nevertheless, as we can see in Fig. 9 (e), image structures are recovered by l_0 TV with the best SSIM value, but some sparse noises on the edges are retained by mistake. If these sparse noises can be removed, the PSNR values will be improved significantly. l_0 TV is not accurate enough to characterize the global spectral correlation of hyperspectral images. To solve this problem, the global LR property is exploited. The WSWNN and WSWTNN constraints are beneficial to get rid of sparse noise and hence

are incorporated into an l_0 TV model.

5)Effectiveness of TLR constraint: To intuitively describe the significance of TLR constraints, we compare: 1)WSNN, 2) WSTNN, 3) WSWNN, 4) WSWTNN, 5) l_0 TV, 6) WSWNN- l_0 TV, 7) WSWTNN- l_0 TV in the case of mixed noise (Gaussian noise (SNR = 10~20 dB), salt and pepper noise (5%), and four dead-lines randomly in four bands). In Fig. 10, the results restored by these methods are compared with the original HSI, and the regions marked with the red frame are zoomed. There still exists more or less dead-lines in the results of WSNN and WSTNN. Due to the weight nuclear norm introduced, WSWNN and WSWTNN have better performances to suppress the dead-lines. But we still need to enhance the spatial smoothness in the results of WSWNN and WSWTNN, which also promotes us to inject l_0 TV into the TLR denoising framework. Therefore, WSWNN- l_0 TV and WSWTNN- l_0 TV are superior to the other methods.

B. Real-world HSI data Experiments

In this section, three real-world HSI datasets are carried out i.e., Salinas, Indian Pines and Pavia University datasets. The above-compared methods are still implemented. As original reference HSIs are unavailable, we visually observe the denoised results and do classification experiments to further demonstrate the denoised performances.

1)Salinas: This scene was acquired by the AVIRIS sensor over Salinas Valley, California, and was characterized by high spatial resolution (3.7-meter pixels) with the data size of $512 \times 217 \times 224$. As Fig. 11 (a) shows, noisy images are corrupted by Gaussian noise. The results of this data set are presented in Fig. 11. LRM, SSTV and LRTA fail to get rid of all Gaussian noise. LRTV, LRTDTV, and MLR-SSTV achieve better results, but they more or less lose the local details of HSI images. Although NGmeet and HSID-CNN remove Gaussian noise effectively, it loses lots of useful structure information. For more clear explanations, we provide a classification experiment consequently. The proposed WSWNN- l_0 TV and WSWTNN- l_0 TV methods remove various noises completely and preserve clear image edges that have a positive influence on further HSI classification work. Here, the Support Vector Machine (SVM) [63] algorithm is utilized to conduct the supervised HSI classification. Salinas ground-truth contains 16 classes including vegetables, bare soils and

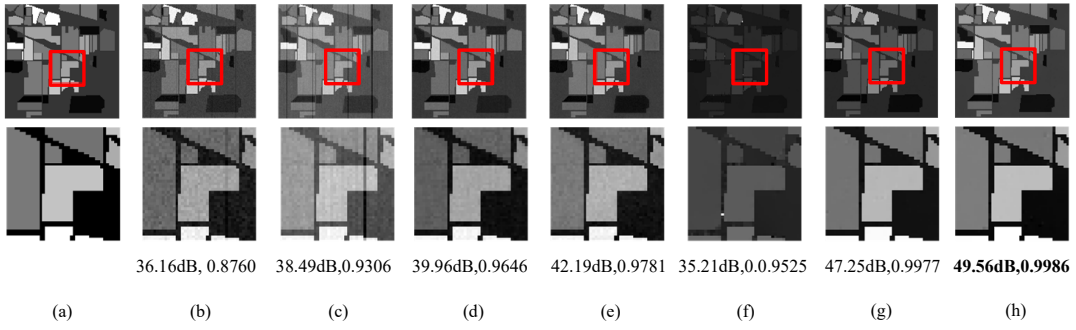


Fig. 10: Resulting HSIs with their (Left) MPSNR and (Right) MSSIM. (a) Original band 15, (b) WSN, (c) WSTNN, (d) WSWNN, (e) WSWTNN, (f) l_0 TV, (g) WSWNN- l_0 TV, (h) WSWTNN- l_0 TV.

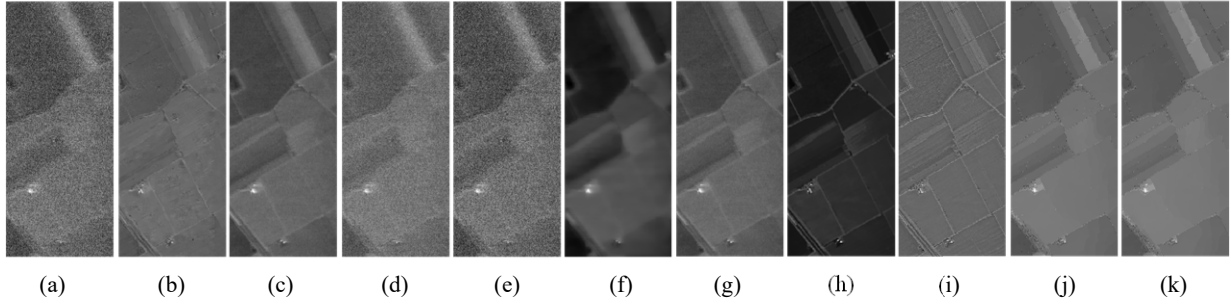


Fig. 11: Denoising results on the Salinas data. (a) Noisy band 1, (b) LRMR, (c) LRTV, (d) SSTV, (e) LRTA, (f) LRTDTV, (g) NGmeet, (h) HSID-CNN, (i) MLR-SSTV, (j) WSWNN- l_0 TV, (k) WSWTNN- l_0 TV.

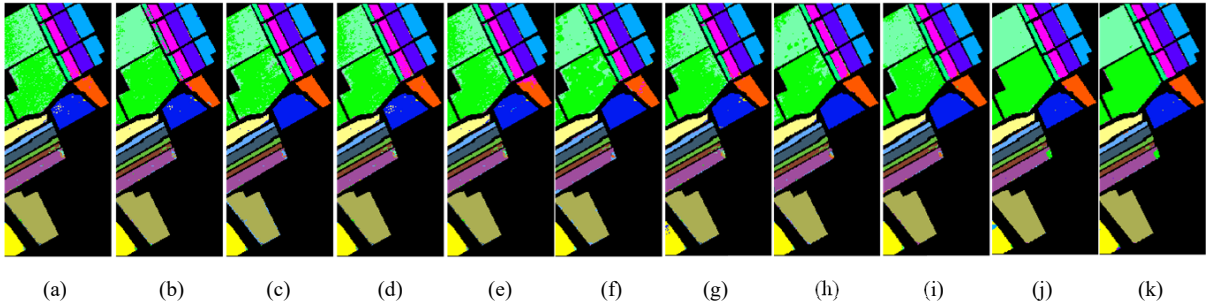


Fig. 12: Classification results by all the compared methods on the Salinas data. (a) Original, (b) LRMR, (c) LRTV, (d) SSTV, (e) LRTA, (f) LRTDTV, (g) NGmeet, (h) HSID-CNN, (i) MLR-SSTV, (j) WSWNN- l_0 TV, (k) WSWTNN- l_0 TV.

TABLE IV: Classification results on the Salinas data by different methods.

Class	Train/Test	Noisy	LRMR	LRTV	SSTV	LRTA	LRTDTV	NGmeets	HSID-CNN	MLR-SSTV	WSWNN- l_0 TV	WSWTNN- l_0 TV
Brocoli-green-weeds-1	21/2009	0.9869	<u>0.9935</u>	0.9536	0.9874	0.9779	0.9743	0.9894	0.9914	0.9899	0.9990	0.9990
Brocoli-green-weeds-2	38/3726	0.9707	0.9813	<u>0.9967</u>	0.9713	0.9881	0.9957	0.9843	0.9970	0.9970	0.9957	0.9957
Fallow	20/1976	0.7161	0.9847	0.8955	0.9923	0.8543	0.9371	0.9920	0.9938	0.9964	0.9923	<u>0.9928</u>
Fallow-rough-plow	14/1394	0.9659	0.6072	0.9730	0.9645	<u>0.9790</u>	0.9775	0.9406	0.9616	0.9754	1.0000	1.0000
Fallow-smooth	27/2678	0.9653	0.9917	0.9863	0.9604	0.9845	0.9476	0.9698	0.9196	0.9740	0.9898	<u>0.9906</u>
Stubble	40/3959	0.9946	0.9941	0.9874	0.9946	0.9941	0.9939	0.9903	0.9903	0.9946	<u>0.9964</u>	0.9969
Celery	36/3579	0.9938	0.9842	0.9413	0.9938	0.9938	0.9927	<u>0.9929</u>	0.9907	0.9938	0.9910	0.9915
Grapes-untrained	113/11271	0.8328	0.9383	0.8487	0.8833	0.8756	0.8653	0.8856	0.9020	0.9383	<u>0.9837</u>	0.9842
Soil-vinyard-develop	63/6203	0.9754	0.9783	0.9757	0.9757	0.9741	0.9748	0.9694	0.9731	0.9853	<u>0.9969</u>	0.9980
Corn-senesced-green-weeds	33/3278	0.9248	0.9273	0.9226	0.9267	0.9254	0.9168	0.9180	0.9072	0.9414	<u>0.9356</u>	0.9359
Lettuce-romaine-4wk	11/1068	0.9054	0.9198	0.9891	0.9082	0.9073	0.9111	0.9016	0.9016	0.9328	0.9655	<u>0.9489</u>
Lettuce-romaine-5wk	20/1927	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.9984	1.0000	<u>0.9948</u>	0.9756	1.0000
Lettuce-romaine-6wk	10/916	0.9757	0.9757	0.9757	0.9945	0.9757	0.9978	0.9790	<u>0.9956</u>	0.9845	0.9823	0.9823
Lettuce-romaine-7wk	11/1070	0.9150	<u>0.9169</u>	0.9968	0.9150	0.9150	0.8980	0.9112	0.9037	0.9169	0.9075	0.9093
Vinyard-untrained	77/7268	0.6371	0.9426	0.7326	0.6511	0.8587	0.8562	0.7081	0.8228	0.8767	<u>0.9846</u>	0.9899
Vinyard-vertical-trellis	19/1807	0.9732	0.9782	0.9783	<u>0.9748</u>	0.9732	0.9435	0.9843	0.9793	0.9871	0.9234	0.9692
kappa	-	0.8860	0.9479	0.9479	0.9002	0.8849	0.9251	0.9074	0.9282	0.9532	<u>0.9807</u>	0.9836
OA	-	0.8978	0.9532	0.9237	0.9105	0.9314	0.9326	0.9169	0.9356	0.9580	<u>0.9827</u>	0.9852
AA	-	0.9376	0.9437	0.8911	0.9422	0.8969	0.9513	0.9426	0.9521	0.9673	<u>0.9770</u>	0.9802

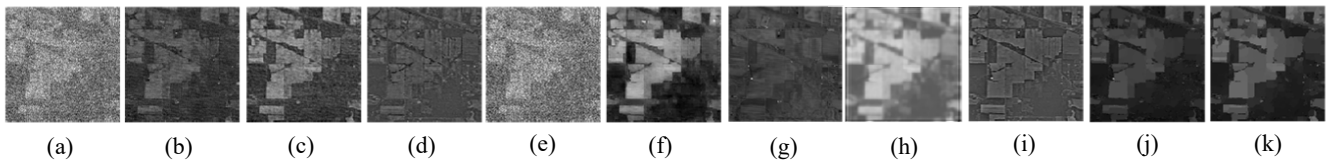


Fig. 13: Denoised results by all the compared methods on the Indian Pine data: (a) original band 220 (b) LRMR, (c) LRTV, (d) SSTV, (e) LRTA, (f) LRTDTV, (g) NGmeet, (h) HSID-CNN, (i) MLR-SSTV, (j) WSWNN- l_0 TV, (k) WSWTNN- l_0 TV.

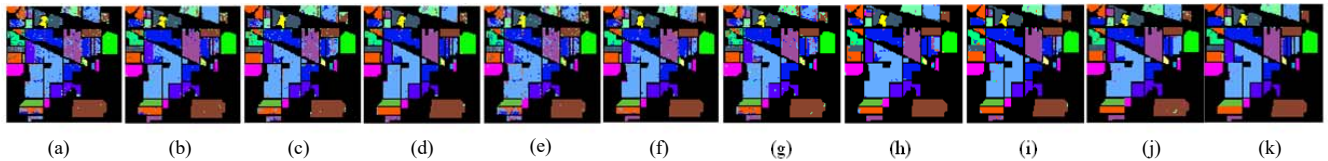


Fig. 14: Classification results by all the compared methods in the Indian Pine data: (a) original (b) BM4D, (c) LRMR, (d) LRTV, (e) SSTV, (f) LRTDTV, (g) NGmeet, (h) HSID-CNN, (i) MLR-SSTV, (j) WSWNN- l_0 TV, (k) WSWTNN- l_0 TV.

TABLE V: Classification results on Indian Pines data by different methods.

Class	Train/Test	Noisy	LRMR	LRTV	SSTV	LRTA	LRTDTV	NGmeets	HSID-CNN	MLR-SSTV	WSWNN- l_0 TV	WSWTNN- l_0 TV
Alfalfa	5/41	0.6341	0.7317	0.6829	<u>0.7804</u>	0.9512	0.6829	0.8537	0.6097	0.9268	0.9512	0.9512
Corn-notill	143/1285	0.8202	0.9105	0.8287	0.9167	0.8249	0.8708	0.8451	0.9012	0.9362	0.9619	<u>0.9315</u>
Corn-mintill	83/747	0.7161	0.8955	0.8888	0.9518	0.7175	0.8969	0.8661	0.9183	<u>0.9732</u>	0.9491	0.9746
Corn	24/213	0.8028	0.9248	0.8309	0.9154	0.8122	<u>0.9201</u>	0.7746	0.9390	0.9061	0.9108	0.9531
Grass-pasture	49/434	0.9377	0.9354	0.9700	0.9723	0.9378	0.9746	0.9539	0.9608	<u>0.9816</u>	<u>0.9816</u>	0.9862
Grass-trees	73/657	0.9497	0.9619	0.9786	0.9863	0.9482	0.9878	0.9848	0.9924	<u>0.9939</u>	0.9985	0.9985
Grass-pasture-mowed	3/25	0.6400	0.6400	0.7600	<u>0.8400</u>	0.6400	0.7200	0.6400	0.7600	0.9200	1.0000	1.0000
Hay-windrowed	48/430	0.9906	0.9976	1.0000	<u>0.9953</u>	0.9907	0.9930	1.0000	0.9930	1.0000	1.0000	1.0000
Oats	2/18	0.7222	0.8333	1.0000	<u>0.9444</u>	0.7778	1.0000	0.8333	1.0000	1.0000	1.0000	1.0000
Soybean-notill	98/874	0.7059	0.8821	0.7276	0.9405	0.7037	0.9165	0.8352	0.8890	0.8982	<u>0.9302</u>	0.9153
Soybean-mintill	246/2209	0.8415	0.9388	0.8632	0.9601	0.8483	0.9157	0.8818	0.9457	0.9479	<u>0.9728</u>	0.9810
Soybean-clean	60/533	0.8480	0.9324	0.9643	<u>0.9699</u>	0.7880	0.9437	0.9099	0.9250	0.9662	0.9681	0.9775
Wheat	21/184	<u>0.9945</u>	<u>0.9945</u>	<u>0.9945</u>	<u>0.9945</u>	<u>0.9945</u>	0.9891	0.9946	0.9837	0.9946	0.9946	0.9946
Woods	127/1138	<u>0.9516</u>	0.9455	0.9472	0.9674	0.9468	0.9964	0.9587	<u>0.9947</u>	0.9912	0.9622	0.9886
Buildings-Grass-Trees-Drives	39/347	0.6311	0.7809	0.8040	0.8587	0.8040	0.8703	0.6686	0.9885	0.9654	0.8732	<u>0.9395</u>
Stone-Steel-Towers	10/83	0.9518	0.9518	1.0000	0.9638	0.9759	<u>0.9879</u>	0.9636	1.0000	1.0000	0.9518	1.0000
kappa	-	0.8197	0.9131	0.8675	0.9451	0.8273	0.9114	0.8748	0.9396	0.9526	<u>0.9528</u>	0.9643
OA	-	0.8423	0.9237	0.8835	0.9518	0.8484	0.9223	0.8900	0.9438	0.9583	<u>0.9611</u>	0.9686
AA	-	0.8212	0.8911	0.8901	0.9349	0.8191	0.9114	0.8728	0.9250	<u>0.9626</u>	0.9620	0.9745

vineyard fields, etc. For each class, approximately 1% of the labeled samples are chosen randomly for training and the rest 99% are tested. The number of training and test samples for each class is shown in the second column of Table IV.

Fig. 12 shows the classification results on the Salinas dataset intuitively. The specific classification accuracy results of each class and kappa coefficient, Overall Accuracy (OA), and Average accuracy (AA) are also given in Table IV. It is easy to find that the classification results have been more or less improved after the HSI denoising process. An obvious fragmentary phenomenon is presented in the classification results of LRTV, SSTV, LRTA, NGmeet, and HSID-CNN, which can be observed in the green regions of figures. LRMR, LRTDTV, MLR-SSTV, and the proposed methods suppress the fragmentary effect in most regions of the HSI, whereas WSWTNN- l_0 TV gets the highest metric values with kappa, OA, and AA of 0.9839, 0.9852 and 0.9802, respectively, among all the results of seven comparing methods.

2) Indian Pine: This scene was acquired by an AVIRIS sensor over the Indian Pines test site in North-western Indiana. It consists of 145×145 pixels and 224 spectral reflectance bands. As Fig. 13 (a) shows, some bands are contaminated by

mixed noise including heavy Gaussian noise and sparse noise, which has a bad effect on the next processing step like HSI classification. In Fig. 13, LRTA obtains the worst result. For the LRMR and LRTV, the dead-line noises remain. Although SSTV and LRTDTV recover the images better than LRMR and LRTV, they unavoidably degrade the edges. NGmeet and HSID-CNN more or less remove some noise, but they lose their utility and cause some degradation of the image texture structure. To further compare the performances of the above methods, we also apply the well-known SVM classifier into the supervised classification experiment. As the first column of Table V shows, 16 different classes are employed to test the classification accuracy. The random 10% of samples from each class generate the training sets. The classification results before and after denoising can be found in Fig. 14 and Table V. Before denoising, the kappa, OA and AA of the original image are just 0.8197, 0.8423 and 0.8212, respectively. The accuracy of each class and three metrics are improved after using seven denoising methods. The evaluation results of LRTV, LRTA, and NGmeet rise less and the over smoothing of LRTDTV also causes similar metric values. With the help of TLR representation and TV regularizers, MLR-SSTV, WSWNN-

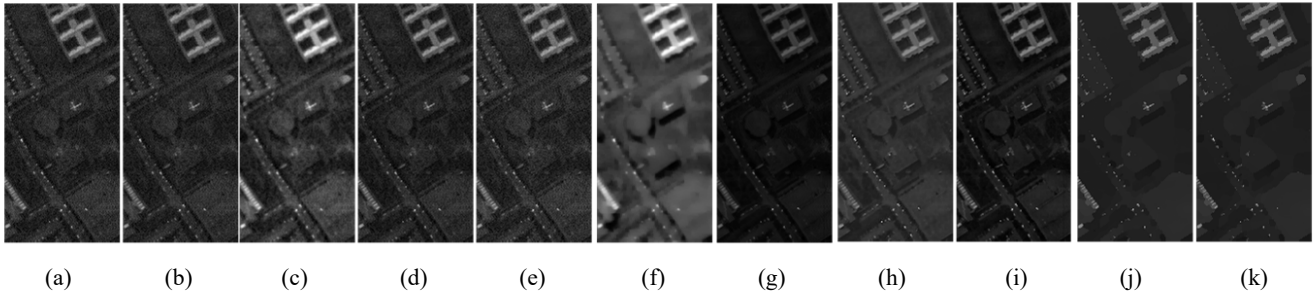


Fig. 15: Denoising results on the Pavia University data set. (a) Noisy band 1, (b) LRMR, (c) LRTV, (d) SSTV, (e) LRTA, (f) LRTDTV, (g) NGmeet, (h) HSD-CNN, (i) MLR-SSTV, (j) WSWNN- l_0 TV, (k) WSWTNN- l_0 TV.

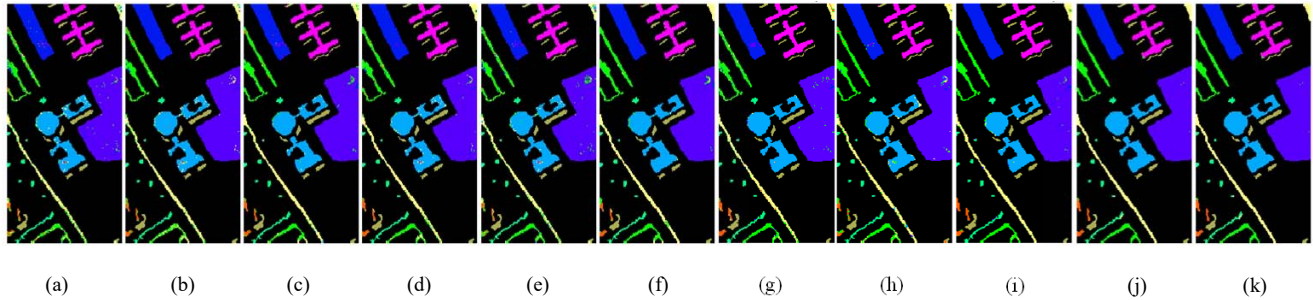


Fig. 16: Classification results by all the compared methods on Pavia University data set. (a) Original, (b) LRMR, (c) LRTV, (d) SSTV, (e) LRTA, (f) LRTDTV, (g) NGmeet, (h) HSD-CNN, (i) MLR-SSTV, (j) WSWNN- l_0 TV, (k) WSWTNN- l_0 TV.

TABLE VI: Classification results on Pavia University data by different methods.

Class	Train/Test	Noisy	LRMR	LRTV	SSTV	LRTA	LRTDTV	NGmeet	HSD-CNN	MLR-SSTV	WSWNN- l_0 TV	WSWTNN- l_0 TV
Asphalt	53/521	0.7073	0.8590	0.9252	0.8761	0.7564	0.9423	0.8697	0.8761	0.9701	<u>0.9936</u>	1.0000
Meadows	96/957	0.9396	<u>0.9919</u>	0.9419	0.9826	0.9419	0.9837	0.9640	0.9756	1.0000	1.0000	1.0000
Gravel	11/110	0.6162	0.7071	0.7980	<u>0.8384</u>	0.4242	0.7172	0.8687	0.5253	0.9091	<u>0.9394</u>	0.9394
Trees	23/226	0.9754	0.9754	<u>0.9803</u>	0.9704	0.9754	0.9901	<u>0.9803</u>	0.9212	0.9754	0.9507	0.9507
Painted metal sheets	70/698	0.9936	<u>0.9936</u>	0.9968	0.9936	0.9936	0.9952	0.9952	0.9952	0.9952	0.9904	0.9904
Bare Soil	305/3042	0.9708	0.9828	0.9799	0.9850	0.9675	<u>0.9865</u>	0.9839	0.9825	0.9883	0.9989	0.9989
Bitumen	133/1330	0.9282	0.9332	0.9699	0.9457	0.9407	<u>0.9774</u>	0.9816	0.9465	0.9699	1.0000	1.0000
Self-Blocking Bricks	63/630	0.7249	0.7919	0.9153	0.8624	0.8360	0.9665	0.9136	0.9242	0.9735	1.0000	<u>0.9989</u>
Shadows	43/428	0.9818	0.9922	0.9766	0.9896	0.9818	0.9948	0.9948	0.9844	<u>0.9974</u>	1.0000	1.0000
kappa	-	0.8998	0.9364	0.9541	0.9492	0.9719	0.9251	0.9593	0.9457	0.9796	<u>0.9950</u>	0.9954
OA	-	0.9209	0.9498	0.9638	0.9598	0.9314	0.9777	0.9679	0.9572	0.9839	<u>0.9961</u>	0.9964
AA	-	0.8708	0.9141	0.9427	0.9382	0.8686	0.9504	0.9502	0.9484	0.9754	<u>0.9859</u>	0.9864

l_0 TV and WSWTNN- l_0 TV produce better results, whereas the highest metric values with kappa, OA and AA are 0.9643, 0.9686, and 0.9745, respectively for WSWTNN- l_0 TV.

3)Pavia University: This dataset is collected by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor during a flight campaign over Pavia University, northern Italy. The size of the selected sub-image is $250 \times 121 \times 103$ and Image ground truths differentiate 9 classes. As Fig. 15 (a) shows, the original image contains Gaussian noise and a small amount of sparse noise. Fig. 15 and 16 give the denoised and classification results by different methods. The number of training and test samples for each class are shown in Table VI. For this image, about 10% of the labeled data are used as training samples. The classification results have been improved after denoising methods used. LRMR, SSTV, LRTA, and NGmeet obtain similar results since they just use either the low-rank property or the spatial-spectral TV. LRTV, LRTDTV, and MLR-SSTV exploit these two types of

prior knowledge, but LRTDTV and MLR-SSTV based on the LR tensor models perform better than LRTV. The proposed TLR- l_0 TV methods produce superior results than the above approaches. In particular, classification accuracies of most classes are more than 0.99 for WSWTNN- l_0 TV.

In real-word data and classification experiments, our proposed methods are efficient to remove mixed Gaussian noise and sparse noise since tensor low-rank prior knowledge is considered comprehensively. Simultaneously, edges of the denoised image are preserved more completely and classification results on three different data sets are significantly improved due to the utilization of l_0 TV regularization.

VI. CONCLUSION

In this paper, we have introduced and analyzed l_0 TV for HSI denoising. Different from l_1 norms, l_0 TV establishes a physical relationship with the l_0 gradient values of the

- [37] W. He, Q. Yao, C. Li, N. Yokoya, and Q. Zhao, "Non-local meets global: An integrated paradigm for hyperspectral denoising," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 6861–6870.
- [38] W. He, H. Zhang, L. Zhang, and H. Shen, "Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration," *IEEE Trans. on Geosci. Remote Sens.*, vol. 54, no. 1, pp. 178–188, Jan. 2016.
- [39] Z. Wu, Q. Wang, Z. Wu, and Y. Shen, "Total variation-regularized weighted nuclear norm minimization for hyperspectral image mixed denoising," *J. Electron. Imag.*, vol. 25, no. 1, 2016.
- [40] 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, 2018.
- [41] 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, 2018.
- [42] 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. Top. Appl. Earth Obs. Remote Sens.*, vol. 11, no. 3, pp. 713–729, 2018.
- [43] Y. Wang, J. Peng, Q. Zhao, Y. Leung, X. Zhao, and D. Meng, "Hyperspectral image restoration via total variation regularized low-rank tensor decomposition," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 11, no. 4, pp. 1227–1243, Apr. 2018.
- [44] M. Wang, Q. Wang, J. Chanussot, and D. Li, "Hyperspectral image mixed noise removal based on multidirectional low-rank modeling and spatial-spectral total variation," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 488–507, 2021.
- [45] X. Li, C. L. Y. Xu, and J. Jia, "Image smoothing via l0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 174:1–174:12, Dec. 2011.
- [46] L. Xu, S. Zheng, and J. Jia, "Unnatural l0 sparse representation for natural image deblurring," in *Pro. IEEE conf. comput. vis. pattern recognit. (CVPR)*, 2013, pp. 1107–1114.
- [47] H. Xue, S. Zhang, and D. Cai, "Depth image inpainting: Improving low rank matrix completion with low gradient regularization," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4311–4320, 2017.
- [48] M. Wang, Q. Wang, and J. Chanussot, "l0 gradient regularized low-rank tensor model for hyperspectral image denoising," in *10th Workshop Hyperspectral Imag. Signal Process., Evol. Remote Sens. (WHISPERS)*, 2019, pp. 1–6.
- [49] S. Ono, "L0 gradient projection," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1554–1564, 2017.
- [50] Z. Zhang, D. Liu, S. Aeron, and A. Vetro, "An online tensor robust PCA algorithm for sequential 2D data," in *IEEE Int. Conf. Acoust. Speech Signal Process.*, Mar. 2016, pp. 2434–2438.
- [51] T. Kolda and B. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
- [52] R. Zhao, Q. Wang, J. Fu, and L. Ren, "Exploiting block-sparsity for hyperspectral kronecker compressive sensing: A tensor-based bayesian method," *IEEE Trans. Image Process.*, vol. 29, pp. 1654–1668, 2020.
- [53] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.
- [54] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 5249–5257.
- [55] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends[®] Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.
- [56] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM J. Optim.*, vol. 26, no. 1, pp. 337–364, 2016.
- [57] G. Li and T. K. Pong, "Global convergence of splitting methods for nonconvex composite optimization," *SIAM J. Optim.*, vol. 25, no. 4, pp. 2434–2460, 2015.
- [58] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.
- [59] M. Wang, Q. Wang, J. Chanussot, and D. Hong, "l0-l1 hybrid total variation regularization and its applications on hyperspectral image mixed noise removal and compressed sensing," *IEEE Trans. Geosci. Remote Sens.*, 2021.
- [60] Z. Wu, Q. Wang, J. Jin, and Y. Shen, "Structure tensor total variation-regularized weighted nuclear norm minimization for hyperspectral image mixed denoising," *Signal Process.*, vol. 131, pp. 202–219, 2017.
- [61] W. He, H. Zhang, L. Zhang, and H. Shen, "Hyperspectral image denoising via noise-adjusted iterative low-rank matrix approximation," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 8, no. 6, pp. 3050–3061, 2015.
- [62] B. Pan, Z. Shi, and X. Xu, "Hierarchical guidance filtering-based ensemble classification for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 4177–4189, 2017.
- [63] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, pp. 1778–1790, Aug. 2004.