Hyperspectral Image Restoration Using Low-Rank Tensor Recovery

Article *in* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing · July 2017

DOI: 10.1109/JSTARS.2017.2714338

5 authors, including:

Yunjin Chen
Graz University of Technology
28 PUBLICATIONS **339** CITATIONS
SEE PROFILE

Yulan Guo
National University of Defense Technology
54 PUBLICATIONS **639** CITATIONS
SEE PROFILE

Hongyan Zhang
Wuhan University
55 PUBLICATIONS **937** CITATIONS
SEE PROFILE

Gangyao Kuang
National University of Defense Technology
140 PUBLICATIONS **849** CITATIONS
SEE PROFILE

Hyperspectral Image Restoration Using Low-Rank Tensor Recovery

Haiyan Fan, Yunjin Chen, Yulan Guo, Hongyan Zhang, *Senior Member, IEEE*,
and Gangyao Kuang, *Senior Member, IEEE*

***Abstract*—This paper studies the hyperspectral image (HSI) denoising problem under the assumption that the signal is low in rank. In this paper, a mixture of Gaussian noise and sparse noise is considered. The sparse noise includes stripes, impulse noise, and dead pixels. The denoising task is formulated as a low-rank tensor recovery (LRTR) problem from Gaussian noise and sparse noise. Traditional low-rank tensor decomposition methods are generally NP-hard to compute. Besides, these tensor decomposition based methods are sensitive to sparse noise. In contrast, the proposed LRTR method can preserve the global structure of HSIs and simultaneously remove Gaussian noise and sparse noise.The proposed method is based on a new tensor singular value decomposition and tensor nuclear norm. The NP-hard tensor recovery task is well accomplished by polynomial time algorithms. The convergence of the algorithm and the parameter settings are also described in detail. Preliminary numerical experiments have demonstrated that the proposed method is effective for low-rank tensor recovery from Gaussian noise and sparse noise. Experimental results also show that the proposed LRTR method outperforms other denoising algorithms on real corrupted hyperspectral data.**

***Index Terms*—Alternating direction method of multiplier (ADMM), gaussian noise, low-rank tensor, recovery, sparse noise, tensor singular value decomposition (t-SVD).**

I. Introduction

A HYPERSPECTRAL image (HSI) is a three-dimensional data cube, in which the first and second dimensions correspond to the spatial domain and the third dimension corresponds to the spectral dimension. Due to the rich spectral information, HSIs have drawn a lot of attention in various application fields [1]–[2]. Nevertheless, HSIs collected in practice often suffer from various annoying degradations, e.g., noise contamination, stripe corruption, and missing data, due to the sensor, photon effects, and calibration error. For real-world HSIs, there usually exists a combination of several types of noise, e.g., Gaussian noise, impulse noise, dead pixels or lines, and stripes [3]–[4]. The non-Gaussian noise (including impulse noise, dead pixels or lines, stripes, etc.) has a sparsity property. So the mixture of impulse noise, dead pixels or lines, stripes is considered as sparse noise. The degradation of HSIs caused by various types of noise hinders the effectiveness of subsequent HSI processing tasks, e.g., spectral signature unmixing [5]–[6], segmentation [7], matching [8], and classification [9]–[10]. Therefore, HSI denoising is a critical preprocessing step for HSI applications [11].

In last few decades, a large number of denoising methods have been proposed. 3-D transform-domain collaborative filtering [12] and nonlocal algorithm [13] consider each band of HSI as a 2-D image. These methods introduce a loss of interspectral information as the correlation between neighboring spectral bands is not considered. To use spectral information, HSI noise reduction algorithms combining spatial and spectral information are proposed, including a hybrid spatial–spectral derivative-domain wavelet shrinkage noise reduction approach [14], a bivariate wavelet shrinkage based method [15], and a spectral–spatial adaptive total variation (TV) model [16]. Another popular approach is based on principle component analysis (PCA), which assumes that high-dimensional hyperspectral data underlie a low-dimensional intrinsic space. HSI is denoised using the inverse transform of the first few principal components (PCs). However, some useful information may be lost in the denoised image. In [3], a low-rank matrix recovery method was adopted to simultaneously remove Gaussian noise, impulse noise, and stripes. To handle the problem that noise intensity is different in different bands, a noise-adjusted iterative low-rank matrix approximation approach was proposed in [17]. He *et al.* [18] unified the TV regularization and low-rank matrix factorization for HSI denoising. Wang *et al.* [19] proposed a group low-rank representation (LLR)denoising method for the reconstruction of corrupted HSIs. Fan *et al.* [20] proposed an superpixel segmentation (SS) LRR denoising method in which the SS and LRR were combined for HSI denoising. As a powerful statistical image modeling technique, sparse representation has been successfully used in image denoising [21]. Zhao *et al.* [22] proposed to combine sparse representation and low-rank constraint for Gaussian noise removal. Yang *et al.* [23] extended the work and unified denoising and unmixing in a sparse representation framework. A nonnegative sparse matrix factorization method was proposed for HSI denoising in [24]. Zhu *et al.* [25] proposed a low-rank spectral nonlocal approach to

Manuscript received December 24, 2016; revised March 21, 2017 and May 7, 2017; accepted June 7, 2017. *(Corresponding author: Haiyan Fan.)*

H. Fan, Y. Guo, and G. Kuang are with the National University of Defense Technology, Changsha 410073, China (e-mail: hy_fan@yeah.net; yulan.guo@nudt.edu.cn; kuangyeats@hotmail.com).

Y. Chen is with the ULSee Inc., Hangzhou, 310020,, China (e-mail: chenyunjin_nudt@hotmail.com).

H. Zhang is with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing and the Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430072, China (e-mail: zhanghongyan@whu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2017.2714338

simultaneously remove mixed noise. Sparse and low-rank penalties were combined in [26] for the removal of sparse noise. Besides, Ma *et al.* [27] proposed a low-rank matrix approximation to image matching and has achieved very promising results. Although these methods have considered the spectral correlationships, they mainly convert high-dimensional HSI data into 2-D data by vectorizing the data in each band. This strategy will introduce loss of useful multiorder structure information.

In multilinear algebra, HSI data cube can be considered as a three-order tensor. Therefore, spatial-spectral information can be simultaneously handled by a tensor decomposition based algorithm. Two kinds of tensor decomposition algorithms are usually used in the literature, namely Tucker decomposition and parallel factor analysis (PARAFAC) decomposition. Tucker decomposition based denoising methods include the lower rank tensor approximation (LRTA) [28], the genetic kernel Tucker decomposition [29], and the multidimensional Wiener filtering method [30]. The PARAFAC decomposition based denoising methods include the PARAFAC model [31] and the rank-1 tensor decomposition method [32]. Under the limitation of prior knowledge, the aforementioned tensor algebra methods are implicitly developed for additive white Gaussian noise. Several 3-D denoising methods, such as VBM3D [33] and BM4D [34], can also be used for HSI noise removal. 3-D sparse coding was exploited to denoise HSI [35], which fully explored the spectral information by extracting different patches.

In recent years, robust tensor recovery and completion plays an important role in multilinear data analysis. Tensor decompositions are robust to outliers and gross corruptions [36]. Tensor decomposition resembles PC analysis for matrices, and the robust PC analysis (RPCA) [37] is robust to outliers and corrupted observations. More recently, Zhang *et al.* [38] proposed the tensor tubal rank using a new tensor decomposition scheme [39], which is referred as tensor singular value decomposition (t-SVD). t-SVD is based on a new definition of tensor-tensor product, which has many similar properties as its matrix counterpart. Based on the computable t-SVD, the tensor nuclear norm [40] was used to replace the tubal rank for low-rank tensor recovery (LRTR) from incomplete/corrupted tensors. Lu *et al.* [41] studied the tensor RPCA to recover the low tubal rank component and sparse component from noisy observations using convex optimization. Unlike the traditional HSI denoising methods, our proposed method is based on the prior knowledge in practice. First, the clean HSI data have the underlying low-rank tensor property, even though the actual HSI data may not be due to outliers and non-Gaussian noise [28]. The nuclear norm is used as a convex surrogate function for the low-rank tensor. In this paper, we propose an HSI denoising technique using LRTR. Our model is based on the aforementioned t-SVD and its induced tensor tubal rank and tensor nuclear norm. It can simultaneously remove Gaussian noise, impulse noise, dead pixels, dead lines, and stripes. Our approach can be solved using polynomial-time algorithms, e.g., the alternating direction method of multiplier (ADMM). In this paper, the ADMM algorithm is used to achieve LRTR by heuristically solving a convex relaxation problem. Specifically, the tensor nuclear norm, l_1-norm, and l_2-norm are used to induce low-rank component, sparse component and Gaussian noise term.

A. *Paper Contribution*

The contributions of this paper are threefold.

1) The robust tensor recovery from Gaussian noise and sparse noise is well accomplished by integrating the tensor nuclear norm, the l_1-norm and the l_2-norm in a unified convex relaxation framework.
2) ADMM is used to solve the optimization problem. The optimal range for the penalty parameter and the convergence analysis of the ADMM-based algorithm are given.
3) Numerical experiments on synthetic datasets confirm that the proposed method can achieve the recovery of low-rank tensors corrupted by Gaussian noise and sparse noise. When applied to corrupted HSI restoration, experiments show that the proposed method outperforms other denoising algorithms.

B. *Paper Organization*

The rest of this paper is structured as follows. Section II introduces some notations and preliminaries for tensors. In Section III, we describe the proposed model in detail. Section IV presents the ADMM method to solve the optimization problem. Convergence analysis and stopping criterion of the proposed ADMM-based algorithm are also given. In Section V, experiments are conducted to demonstrate the effectiveness of the proposed method. The final section gives concluding remarks.

II. Notation and Preliminaries

The order of a tensor is equal to the number of its dimensions (also known as ways or modes). Throughout this paper, tensors are denoted as capitalized calligraphic letters, e.g., $\mathcal{A}$. Matrices are denoted as capitalized boldface letters, e.g., $\mathbf{A}$. Vectors are denoted as boldface letters, e.g., $\mathbf{a}$. Scalars are denoted as lowercase letters, e.g., a. The field of real number is denoted as $\mathbb{R}$. For an m-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$, we denote its $(i_1, i_2, \ldots, i_m)$-element as $\mathcal{A}_{i_1 i_2 \cdots i_m}$ or $a_{i_1, i_2, \ldots, i_m}$. For a three-order tensor $\mathcal{A}$, $\mathcal{A}(i,:,:)$, $\mathcal{A}(:,i,:)$, and $\mathcal{A}(:,:,i)$ are used to represent the ith horizontal, lateral, and frontal slice. The frontal slice $\mathcal{A}(:,:,i)$ can also be denoted as $\mathbf{A}^{(i)}$.

Several norms for vector, matrix, and tensor are used. The l_1-norm is calculated as $\|\mathcal{A}\|_1 = \sum_{i_1,i_2,\ldots,i_m} |a_{i_1,i_2,\ldots,i_m}|$ and the Frobenius norm is calculated as $\|\mathcal{A}\|_F = \sqrt{\sum_{i_1,i_2,\ldots,i_m} |a_{i_1,i_2,\ldots,i_m}|^2}$. The matrix nuclear norm is $\|\mathbf{A}\|_* = \sum_i \sigma_i(\mathbf{A})$.

For a three-order tensor $\mathcal{A}$, let $\hat{\mathcal{A}}$ be the discrete Fourier transform (DFT) along the third dimension of $\mathcal{A}$, i.e., $\hat{\mathcal{A}} = \mathtt{fft}(\mathcal{A}, [], 3)$. Similarly, $\mathcal{A}$ can be calculated from $\hat{\mathcal{A}}$ by $\mathtt{ifft}(\hat{\mathcal{A}}, [], 3)$. Let $\bar{\mathbf{A}} \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$ be the block diagonal matrix with each block on diagonal as the frontal slice $\hat{\mathbf{A}}^{(i)}$ of $\hat{\mathcal{A}}$ [42], i.e.,

$$\begin{pmatrix} \hat{\mathbf{A}}^{(1)} & & & \\ & \hat{\mathbf{A}}^{(2)} & & \\ & & \ddots & \\ & & & \hat{\mathbf{A}}^{(n_3)} \end{pmatrix}$$

The *t*-product [39], [41] is defined on a block circular matrix. For a three-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its block circular matrix $\mathrm{bcirc}(\mathcal{A})$ has a size of $n_1 n_3 \times n_2 n_3$ and is defined as

$$\mathrm{bcirc}(\mathcal{A}) = \begin{pmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(2)} & \cdots & \mathbf{A}^{(n_3)} \\ \mathbf{A}^{(2)} & \mathbf{A}^{(1)} & \cdots & \mathbf{A}^{(n_3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{(n_3)} & \mathbf{A}^{(n_3-1)} & \cdots & \mathbf{A}^{(1)} \end{pmatrix}. \tag{1}$$

Two other operators, `unfold` and `fold` are, respectively, defined as

$$\mathtt{unfold}(\mathcal{A}) = \begin{pmatrix} \mathbf{A}^{(1)} \\ \mathbf{A}^{(2)} \\ \vdots \\ \mathbf{A}^{(n_3)} \end{pmatrix} \tag{2}$$

$$\mathtt{fold}(\mathtt{unfold}(\mathcal{A})) = \mathcal{A}. \tag{3}$$

Given two third-order 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 *t*-product of $\mathcal{A}$ and $\mathcal{B}$ is defined as

$$\mathcal{A} * \mathcal{B} = \mathtt{fold}(\mathrm{bcirc}(\mathcal{A}) \cdot \mathtt{unfold}(\mathcal{B})). \tag{4}$$

Theorem 1: Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the t-SVD factorization of tensor $\mathcal{A}$ is

$$\mathcal{A} = \mathcal{Q} * \mathcal{S} * \mathcal{V}^* \tag{5}$$

where $\mathcal{Q} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal. $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an *f*-diagonal tensor and $\mathcal{V}^*$ is the conjugate tensor of $\mathcal{V}$ obtained by conjugately transposing each frontal slice and then reversing the order of transposed frontal slices 2 through n_3 [41].

t-SVD cannot be directly computed in the original domain as matrix SVD does. However, t-SVD can be efficiently computed based on the matrix SVD in the Fourier domain. This is based on a key property that the block circulant matrix can be mapped to a block diagonal matrix in the Fourier domain, i.e.,

$$(\mathbf{F}_{n_3} \otimes \mathbf{I}_{n_1}) \cdot \mathrm{bcirc}(\mathcal{A}) \cdot (\mathbf{F}_{n_3}^{-1} \otimes \mathbf{I}_{n_2}) = \bar{\mathbf{A}} \tag{6}$$

where $\mathbf{F}_{n_3}$ denotes the $n_3 \times n_3$ DFT matrix and $\otimes$ denotes the Kronecker product. Using the relationship between the circular convolution and the DFT, we can perform t-SVD via matrix SVD on each frontal slice in the Fourier domain, i.e.,

$$[\hat{\mathcal{Q}}(:,:,i)), \hat{\mathcal{S}}(:,:,i)), \hat{\mathcal{V}}^*(:,:,i))] = \mathtt{SVD}(\hat{\mathcal{A}}(:,:,i) \tag{7}$$

for $k = 1, \ldots, n_3$. Then, we can obtain the t-SVD of tensor $\mathcal{A}$ by

$$\mathcal{Q} = \mathtt{ifft}(\hat{\mathcal{Q}}, [], 3) \tag{8}$$

$$\mathcal{S} = \mathtt{ifft}(\hat{\mathcal{S}}, [], 3) \tag{9}$$

$$\mathcal{V} = \mathtt{ifft}(\hat{\mathcal{V}}, [], 3). \tag{10}$$

The resulting t-SVD provides a means for optimally approximating the tensor as a sum of outer products of matrices.

Definition 1: (Tensor multi-rank and tubal rank)[41] The tensor multirank of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a vector $\mathbf{r} \in \mathbb{R}^{n_3}$ with the ith entry being defined as the rank of $\hat{\mathbf{A}}^{(i)}$. The tensor tubal rank $\mathtt{rank}_t(\mathcal{A})$ is defined as the number of nonzero singular tubes of $\mathcal{S}$ obtained from the t-SVD of tensor $\mathcal{A}$ in (5), i.e.,

$$\mathtt{rank}_t(\mathcal{A}) = \sharp\{i : \mathcal{S}(i,i,:) \neq 0\} = \mathtt{max}_i r_i. \tag{11}$$

Definition 2: (Tensor nuclear norm)[42] The nuclear norm of tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, denoted as $\|\mathcal{A}\|_*$, is defined as the average of the nuclear norm of all the frontal slices of $\hat{\mathcal{A}}$, i.e.,

$$\|\mathcal{A}\|_* = \frac{1}{n_3} \sum_{i=1}^{n_3} \hat{\mathbf{A}}^{(i)} = \frac{1}{n_3} \sum_{i,j} \hat{\mathcal{S}}(i,i,j). \tag{12}$$

The above-mentioned tensor nuclear norm is defined in the Fourier domain. It is closely related to the nuclear norm of the block circulant matrix in the original domain. That is

$$\begin{aligned} \|\mathcal{A}\|_* &= \tfrac{1}{n_3} \textstyle\sum_{i=1}^{n_3} \hat{\mathbf{A}}^{(i)} = \tfrac{1}{n_3} \|\bar{\mathbf{A}}\|_* \\ &= \tfrac{1}{n_3} \|(\mathbf{F}_{n_3} \otimes \mathbf{I}_{n_1}) \cdot \mathrm{bcirc}(\mathcal{A}) \cdot (\mathbf{F}_{n_3}^{-1} \otimes \mathbf{I}_{n_2})\|_* \\ &= \tfrac{1}{n_3} \|\mathrm{bcirc}(\mathcal{A})\|_*. \end{aligned} \tag{13}$$

The above-mentioned relationship gives an equivalent definition of the tensor nuclear norm in the original domain. So the tensor nuclear norm is the nuclear norm [with a factor of a new matricization (block circulant matrix)] of a tensor. Compared to previous matricizations along a particular dimension (for HSIs denoising, HSI data are often matricized along the spectral dimension), the block circulant matricization may preserve more spacial relationship among entries.

Definition 3: (Tensor inner product) The inner product of $\mathcal{A}$ and $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_m}$ is denoted as

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1, i_2, \ldots, i_m} \mathcal{A}_{i_1 i_2 \cdots i_m} \cdot \mathcal{B}_{i_1 i_2 \cdots i_m}. \tag{14}$$

III. Proposed Method

A. Problem Formulation

HSI can be represented as a three-order tensor, where the spatial information lies in the first two dimensions and the spectral information lies in the third dimension. The hyperspectral signal for each pixel can be modeled as the sum of two components: the signal and the noise term. The noises can be divided into two classes according to the density of their distributions: sparse noise and dense noise, in which the sparse noise mainly contains impulse noise, dead lines, dead pixels, and stripes. The dense noise is assumed to be Gaussian distributed. In this paper, the observation $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ ($n_1 \times n_2$ is the dimensionality in spatial domain and n_3 is the number of spectral bands) is expressed as the sum of three parts, that is

$$\mathcal{Y} = \mathcal{F} + \mathcal{N} + \mathcal{S} \tag{15}$$

where $\mathcal{Y}$ is the observed noisy HSI, $\mathcal{F}$ is the clean hyperspectral signal, $\mathcal{N}$ is the Gaussian noise, and $\mathcal{S}$ is the sparse noise.

B. Recovery of Low-Rank Tensor From Corrupted HSI Data

In this paper, we study the model defined in (15) to recover the low-rank component $\mathcal{F}$ from the observations $\mathcal{Y} = \mathcal{F} + \mathcal{N} + \mathcal{S}$. Without considering the Gaussian noise $\mathcal{N}$, Lu *et al.* [41] used the Tensor Robust Principal Component Analysis (TRPCA) to recover a low-rank tensor via convex optimization, i.e.,

$$\begin{aligned} \min_{\mathcal{F},\mathcal{S}} \quad & \|\mathcal{F}\|_* + \lambda_2 \|\mathcal{S}\|_1 \\ \text{s.t.} \quad & \tilde{\mathcal{Y}} = \mathcal{F} + \mathcal{S}. \end{aligned} \tag{16}$$

In (16), $\lambda_2 = 1/\sqrt{\max(n_1, n_2)n_3}$ guarantees accurate recovery when the low-rank tensor $\mathcal{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ obeys the incoherence conditions, provided that the tubal rank of $\mathcal{F}$ is not too large, and that $\mathcal{S}$ is reasonably sparse [43]. Moreover, (16) can be solved by polynomial-time algorithms, e.g., the standard ADMM [41], [43].

Then, we follow this convex optimization scheme to recover the low-tubal-rank tensor $\mathcal{F}$ from the observations $\mathcal{Y}$ degraded by both outlying observations $\mathcal{S}$ and noise $\mathcal{N}$. In this paper, the low-rank approximation denoising is formulated as

$$\begin{aligned} \min_{\mathcal{F},\mathcal{N},\mathcal{S}} \quad & \|\mathcal{F}\|_* + \lambda_1 \|\mathcal{N}_{SI}\|_F^2 + \lambda_2 \|\mathcal{S}\|_1 \\ \text{s.t.} \quad & \mathcal{Y} = \mathcal{F} + \mathcal{N} + \mathcal{S}. \end{aligned} \tag{17}$$

As stated in [28], the clean HSI data $\mathcal{F}$ has the low-rank tensor property. Apart from t-SVD, there are mainly two types of LRTR methods in the literature, i.e., the methods that are based on CANDECOMP/PARAFAC(CP) format, and the methods based on Tucker decomposition. In fact, if a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a low-rank tensor (in either CP or Tucker sense), it is also a low tubal rank tensor. Specifically, if a third-order tensor $\mathcal{A}$ is of CP rank r, then its tubal rank is up to r [43]. In other words, the low-rank tensor $\mathcal{F}$ can also be recovered by low tubal rank constraint, rather than CP or Tucker decomposition. A tensor of size $\mathbb{R}^{n_1 \times n_2 \times n_3}$ with rank r under t-SVD (referred to as the tubal rank) can be recovered by solving a convex optimization problem [43]. On one hand, tensor tubal rank has similar optimal properties as matrix rank. On the other hand, using tensor tubal rank allows processing HSI data as a third-order tensor, which can preserve the global structure of HSI data, instead of adapting data to classical matrix-based techniques.

As observed in [41] and [43], low-tubal-rank tensor $\mathcal{F}$ should not be sparse. If recovery is possible, the following incoherence conditions should be satisfied.

Definition 9: (Tensor Incoherence Conditions) For $\mathcal{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, suppose its tubal rank $\texttt{rank}_t(\mathcal{F}) = r$. Its t-SVD is $\mathcal{F} = \mathcal{Q} * \mathcal{S} * \mathcal{V}^*$, where $\mathcal{Q} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times r \times n_3}$ are orthogonal. $\mathcal{S} \in \mathbb{R}^{r \times r \times n_3}$ is an *f*-diagonal tensor. Then, $\mathcal{F}$ should satisfy the following incoherence conditions:

$$\max_{i=1,\ldots,n_1} \|\mathcal{Q}^* * \mathring{\mathfrak{e}}_i\|_F \leq \sqrt{\frac{\mu_0 r}{n_1 n_3}} \tag{18}$$

$$\max_{j=1,\ldots,n_2} \|\mathcal{V}^* * \mathring{\mathfrak{e}}_j\|_F \leq \sqrt{\frac{\mu_0 r}{n_2 n_3}} \tag{19}$$

where $\mathring{\mathfrak{e}}_i$ is the row basis of size $n_1 \times 1 \times n_3$ with its $(i, 1, 1)$th entry $\mathring{\mathfrak{e}}_{i11} = 1$ and other entries being zero. $\mathring{\mathfrak{e}}_j$ is the column basis of size $1 \times n_2 \times n_3$ with its $(1, j, 1)$th entry $\mathring{\mathfrak{e}}_{1j1} = 1$ and other entries being zero. Unlike matrix PCA, for Tensor PCA, the joint incoherence condition is also necessary

$$\|\mathcal{Q} * \mathcal{V}^*\|_\infty \leq \sqrt{\frac{\mu_0 r}{n_1 n_2 n_3^2}}. \tag{20}$$

These incoherence conditions assert that for small values of μ_0, the singular vectors of $\mathcal{F}$ are reasonably spread out, in other words, they are not sparse. Another identifiability issue arises if the rank of sparse tensor $\mathcal{S}$ is low. This will occur, if the nonzero entries of $\mathcal{S}$ occur in a column or in a few columns. To avoid trivial situations, we assume that the sparsity pattern of the sparse component $\mathcal{S}$ is selected uniformly at random [37].

Apart from the above-mentioned incoherence conditions, the rank of the low-rank component $\mathcal{F}$ should not be too large, and the sparse component $\mathcal{S}$ should be reasonably sparse, if we want to perfectly recover $\mathcal{F}$ and $\mathcal{S}$. We denote that $n_{(1)} = \max\{n_1, n_2\}$ and $n_{(2)} = \min\{n_1, n_2\}$.

Theorem 2: Assume that the low-rank tensor $\mathcal{F}$ obeys the incoherence conditions (18)–(20). We first fix any $n_1 \times n_2 \times n_3$ tensor $\mathcal{R}$ of signs. Suppose that the support set Ω of $\mathcal{S}$ is uniformly distributed among all sets of cardinality m, and $\text{sign}(\mathcal{S}_{ijk}) = \mathcal{R}_{ijk}$ for all $(i, j, k) \in \Omega$. For perfect recover, the rank of $\mathcal{F}$ and the number of nonzero entries of $\mathcal{S}$ should satisfy that

$$\texttt{rank}_t(\mathcal{F}) \leq \frac{\rho_r n_{(2)}}{\mu_0 (\log(n_{(1)} n_3))^2} \quad \text{and} \quad m \leq \rho_s n_1 n_2 n_3^2$$

where ρ_r and ρ_s are positive constants. In fact, the above-mentioned conditions indicate that this tensor recovery algorithm works for large values of the rank, that is, on the order of $\frac{n_{(2)}}{(log(n_{(1)} n_3))^2}$ when μ_0 is not too large. For sparse component $\mathcal{S}$, we only make assumption on the locations of nonzero entries of $\mathcal{S}$, but no assumption is given on the magnitudes or signs of the nonzero entries [37], [41].

Theorem 3: λ_2 is free from being a tuning parameter. Under the incoherence conditions defined in (18)–(20) and Theorem 2, the parameter λ_2 used in (16) and (17) can be determined as a constant $\lambda_2 = 1/\sqrt{n_{(1)} n_3}$, which is independent of the values of $\mathcal{F}$ and $\mathcal{S}$ [37] [41].

It is now clear that $\mathcal{F}$ and $\mathcal{S}$ should be satisfied for tensor recovery via TRPCA; another question arises: what conditions $\mathcal{N}$ has to meet in order to perfectly recover $\mathcal{F}$ from (17). It is expected that $\mathcal{N}$ should not either be sparse or low in rank. If $\mathcal{N}$ is sparse, it is unable to identify $\mathcal{N}$ from the sparse component $\mathcal{S}$. In other words, $\mathcal{N}$ should be dense enough. As described in Section III-A, the elementwise model for noise component n_{i_1, i_2, i_3} is an ergodic Gaussian noise. It is well known that a Gaussian noise model (large degree of freedom) corresponds to a dense noise type [44], [45]. On the other hand, $\mathcal{N}$ should not be low in rank. Otherwise, $\mathcal{F}$ cannot be recovered from random noise. Due to the stochastic nature of Gaussian noise, it is assumed that there is no correlation (or weak correlation) between noise components. Thus, the rank of $\mathcal{N}$ is normally full and much larger than the rank of $\mathcal{F}$, that is, $\texttt{rank}_t(\mathcal{N}) \gg \texttt{rank}_t(\mathcal{F})$. Therefore, the low-rank component $\mathcal{F}$ can be recovered from (17) by properly choosing the tuning parameter λ_1 and λ_2.

IV. ADMM for LRTR

The constrained problem defined in (17) can be addressed by a quadratic penalty approach, i.e., by solving

$$\begin{aligned} \min_{\mathcal{F},\mathcal{N},\mathcal{S}} & \|\mathcal{F}\|_* + \lambda_1 \|\mathcal{N}\|_F^2 + \lambda_2 \|\mathcal{S}\|_1 \\ & + \frac{\beta}{2} \|\mathcal{Y} - (\mathcal{F} + \mathcal{N} + \mathcal{S})\|_F^2. \end{aligned} \tag{21}$$

The solution of (21) can be approached to that of (17) by alternating this minimization with respect to variables $\mathcal{F}, \mathcal{N}$, and $\mathcal{S}$. However, the intermediate minimization becomes increasingly ill-posed when β becomes large [46]–[47]. The augmented Lagrangian method (ALM) provides another term to mimic Lagrange multiplier and to overcome the ill-posed problem caused by large value of β. The augmented Lagrangian function for problem defined in (17) is

$$\begin{aligned}&\mathcal{L}\left(\mathcal{F},\mathcal{N},\mathcal{S},\Lambda_1;\lambda_1,\lambda_2,\beta\right)\\&=\|\mathcal{F}\|_* + \lambda_1\|\mathcal{N}\|_F^2 + \lambda_2\|\mathcal{S}\|_1\\&\quad + < \Lambda_1, \mathcal{Y} - (\mathcal{F}+\mathcal{N}+\mathcal{S}) >\\&\quad + \frac{\beta}{2}\|\mathcal{Y} - (\mathcal{F}+\mathcal{N}+\mathcal{S})\|_F^2 \end{aligned} \tag{22}$$

where Λ_1 is the Lagrangian multipliers. ALM is used to minimize $\mathcal{L}(\cdot)$ with respect to variables $\mathcal{F}, \mathcal{N}$, and $\mathcal{S}$ while keeping Λ_1 fixed. It then updates Λ_1 according to the following rule:

$$\Lambda_1^{k+1} \leftarrow \Lambda_1^k + \beta(\mathcal{Y} - (\mathcal{F}^{k+1} + \mathcal{N}^{k+1} + \mathcal{S}^{k+1})). \tag{23}$$

However, minimizing $\mathcal{L}(\cdot)$ in (22) with respect to all variables $\mathcal{F}, \mathcal{N}$, and $\mathcal{S}$ can be a nontrivial problem. Instead, we use the ADMM method, which is a variant of the ALM. ADMM uses partial updates by keeping other variables fixed each time.

Using the ADMM framework for (22), we can update the variables $\mathcal{F}, \mathcal{N}$, and $\mathcal{S}$ in the $(k+1)$th iteration, by alternately minimizing the following function while keeping the value of Λ_1 fixed at Λ_1^k

$$\begin{aligned}&\tilde{\mathcal{L}}\left(\mathcal{F},\mathcal{N},\mathcal{S},\mathcal{A},\Lambda_1^k;\lambda_1,\beta,\right)\\&=\|\mathcal{F}\|_* + \lambda_1\|\mathcal{N}\|_F^2 + \lambda_2\|\mathcal{S}\|_1\\&\quad + \frac{\beta}{2}\|\mathcal{Y} - (\mathcal{F}+\mathcal{N}+\mathcal{S}) + \frac{\Lambda_1^k}{\beta}\|_F^2. \end{aligned} \tag{24}$$

In the $(k+1)$th iteration, $\mathcal{F}^{k+1}$, $\mathcal{N}^{k+1}$, $\mathcal{S}^{k+1}$, and Λ_1^{k+1} are updated as follows:

For the updating of $\mathcal{F}^{k+1}$, we have

$$\begin{aligned}\mathcal{F}^{k+1} &:= \operatorname*{argmin}_{\mathcal{F}} \bar{\mathcal{L}}\left(\mathcal{F},\mathcal{N}^k,\mathcal{S}^k,\Lambda_1^k\right)\\&= \operatorname*{argmin}_{\mathcal{F}} \left(\|\mathcal{F}\|_* + \frac{\beta}{2}\|\mathcal{Y}\right.\\&\qquad \left. - (\mathcal{F}+\mathcal{N}^k+\mathcal{S}^k) + \frac{\Lambda_1^k}{\beta}\|_F^2\right). \end{aligned} \tag{25}$$

The updating of $\mathcal{F}^{k+1}$ has a closed-form solution [38]. For the sake of simplicity, we denote the iteration of $\mathcal{F}^{k+1}$ in (25) as

$$\mathcal{F}^{k+1} := \operatorname*{argmin}_{\mathcal{F}} \left(\|\mathcal{F}\|_* + \frac{\beta}{2}\|\mathcal{F} - \mathcal{X}^k\|_F^2\right) \tag{26}$$

where $\mathcal{X}^k = \mathcal{Y} - (\mathcal{N}^k + \mathcal{S}^k)$. Solving this optimization problem is equivalent to solving the following tensor recovery problem in frequency domain

$$\hat{\mathcal{F}}^{k+1} := \operatorname*{argmin}_{\hat{\mathcal{F}}} \left(\|\bar{\mathbf{F}}\|_* + \frac{\beta}{2}\|\hat{\mathcal{F}} - \hat{\mathcal{X}}^k\|_F^2\right) \tag{27}$$

where $\hat{\mathcal{F}}$ is the DFT along the third dimension of $\mathcal{F}$ represented by $\hat{\mathcal{F}} = \texttt{fft}(\mathcal{F}, [], 3)$. Similarly, $\mathcal{F}$ can be calculated from $\hat{\mathcal{F}}$ via $\texttt{ifft}(\hat{\mathcal{F}}, [], 3)$. $\hat{\mathcal{X}}^k$ is the DFT along the third dimension of $\mathcal{X}^k$. $\bar{\mathbf{F}} \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$ represents the block diagonal matrix with each block on diagonal as the frontal slice $\hat{\mathbf{F}}^{(i)}$ of $\hat{\mathcal{F}}$. For more details about these notations, one can refer to Section II.

For problem defined in (27), we can break it up to n_3 independent minimization problems. Let $\hat{\mathbf{F}}^{k+1,(i)}$ denote the ith frontal slice of $\hat{\mathbf{F}}^{k+1}$. Similarly, we can define $\hat{\mathbf{F}}^{(i)}$ and $\hat{\mathbf{F}}^{k,(i)}$.

$$\hat{\mathbf{F}}^{k+1,(i)} := \operatorname*{argmin}_{\hat{\mathbf{F}}^{(i)}} \left(\|\hat{\mathbf{F}}^{(i)}\|_* + \frac{\beta}{2}\|\hat{\mathbf{F}}^{(i)} - \hat{\mathbf{F}}^{k,(i)}\|_F^2\right). \tag{28}$$

In order to solve (28), we give the following lemma.

Lemma 1: Consider the SVD of a matrix $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$ of rank r.

$$\mathbf{A} = \mathbf{Q} * \mathbf{S} * \mathbf{V}, \quad \mathbf{S} = \text{diag}(\{\sigma_i\}_{1 \le i \le r}) \tag{29}$$

where $\mathbf{Q} \in \mathbb{R}^{n_1 \times r}$ and $\mathbf{V} \in \mathbb{R}^{n_2 \times r}$ are orthogonal, and the singular values σ_i are real and positive. Then, for all $\tau > 0$, we define the soft-thresholding operator $\mathcal{D}$

$$\mathcal{D}_\tau(\mathbf{A}) := \mathbf{Q} * \mathcal{D}_\tau(\mathbf{S}) * \mathbf{V}, \mathcal{D}_\tau(\mathbf{S}) = \text{diag}(\{(\sigma_i - \tau)_+\}_{1 \le i \le r}) \tag{30}$$

where x_+ is an operator $x_+ = \max(0, x)$. Then, for each $\tau > 0$ and $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2}$, the singular value shrinkage operator defined in (30) obeys

$$\mathcal{D}_\tau(\mathbf{A}) = \operatorname*{argmin}_{\mathbf{B}} \left\{\frac{1}{2}\|\mathbf{B} - \mathbf{A}\|_F^2 + \tau\|\mathbf{B}\|_*\right\}. \tag{31}$$

Using (29)–(31) in Lemma 1, we have

$$\hat{\mathbf{F}}^{k+1,(i)} := \mathcal{D}_{\frac{1}{\beta}}(\hat{\mathbf{X}}^{k,(i)}) \tag{32}$$

for $i = 1, \ldots, n_3$ and $\mathcal{X}^k = \mathcal{Y} - (\mathcal{N}^k + \mathcal{S}^k)$. The $(k+1)$th updating of $\mathcal{F}^{k+1}$ can be obtained via inverse Fourier transform

$$\mathcal{F}^{k+1} := \texttt{ifft}(\hat{\mathcal{F}}^{k+1}, [], 3). \tag{33}$$

For the updating of $\mathcal{S}^{k+1}$, we have

$$\begin{aligned}\mathcal{S}^{k+1} &:= \operatorname*{argmin}_{\mathcal{S}} \bar{\mathcal{L}}\left(\mathcal{F}^{k+1},\mathcal{N}^k,\mathcal{S},\Lambda_1^k\right)\\&= \operatorname*{argmin}_{\mathcal{S}} \left(\lambda_2\|\mathcal{S}\|_1 + \frac{\beta}{2}\|\mathcal{Y}\right.\\&\qquad \left. - (\mathcal{F}^{k+1}+\mathcal{N}^k+\mathcal{S}) + \frac{\Lambda_1^k}{\beta}\|_F^2\right). \end{aligned} \tag{34}$$

Equation (34) can be solved by a soft-shrinkage operator

$$\mathcal{S}^{k+1} := \text{shrink}\left(\mathcal{Y} - (\mathcal{F}^{k+1} + \mathcal{N}^k) + \frac{\Lambda_1^k}{\beta}, \frac{\lambda_2}{\beta}\right) \tag{35}$$

where $\text{shrink}(\cdot,\cdot)$ is an elementwise soft-shrinkage operator. For each element a of tensor $\mathcal{Y} - (\mathcal{F}^{k+1} + \mathcal{N}^k) + \frac{\Lambda_1^k}{\beta}$, we have

$$\text{shrink}(a, \tfrac{\lambda_2}{\beta}) \begin{cases} a - \frac{\lambda_2}{\beta}, & a > \frac{\lambda_2}{\beta} \\ 0, & |a| \le \frac{\lambda_2}{\beta} \\ a + \frac{\lambda_2}{\beta}, & a < -\frac{\lambda_2}{\beta}. \end{cases} \tag{36}$$

Algorithm 1: The Optimization Framework for LRTR

```
Input: Observation $\mathcal{Y}$, parameters $\lambda_1$, $\lambda_2$
Output: Useful signal $\mathcal{F}$
Initialization: Choose parameters $\eta > 1$ and $\beta_{\min}, \beta_{\max}$
with $0 < \beta_{\min} < \beta_{\max} < +\infty$, initialize an iteration
number $k \leftarrow 0$ and a bounded starting point;
repeat
    Initialize $\beta^k \in [\beta_{\min}, \beta_{\max}]$;
    repeat
        Update $\mathcal{F}^{k+1}$ by solving (32) with $\beta$ being
        replaced by $\beta^k$,
        Update $\mathcal{S}$ by solving (35) with $\beta$ being replaced
        by $\beta^k$,
        Update $\mathcal{N}^{k+1}$ by solving (38) with $\beta$ being
        replaced by $\beta^k$,
        if $\beta \in (0, \frac{6\mu_2}{13})$ (the definition of $\mu_2$ can be found in
        Assumption 1) then
            Break;
        else
            $\beta^{k+1} \leftarrow \min(\eta\beta^k, \beta_{\max})$;
        end if
    until the maximum number of inner loop is reached;
    Update $\Lambda_1^{k+1}$ using (39),
    if Stopping criterion is satisfied; then
        Break;
    else
        $k \leftarrow k+1$;
    end if
until the maximum number of outer loop is reached.
```

The soft-shrinkage operator $\operatorname{shrink}(\cdot,\cdot)$ is the proximity operator of the l_1-norm [48] [49]. For $\mathcal{N}^{k+1}$, we have

$$\begin{aligned}\mathcal{N}^{k+1} &:= \underset{\mathcal{N}}{\operatorname{argmin}}\, \bar{\mathcal{L}}\left(\mathcal{F}^{k+1}, \mathcal{N}, \mathcal{S}^{k+1}, \Lambda_1^k\right)\\ &= \underset{\mathcal{N}}{\operatorname{argmin}} \left(\lambda_1\|\mathcal{N}\|_F^2 + \frac{\beta}{2}\|\mathcal{Y} - (\mathcal{F}^{k+1}\right.\\ &\qquad \left. + \mathcal{N} + \mathcal{S}^{k+1}) + \frac{\Lambda_1^k}{\beta}\|_F^2\right). \qquad (37)\end{aligned}$$

Similarly, we can obtain the closed-form solution for the updating of $\mathcal{N}^{k+1}$

$$\mathcal{N}^{k+1} := \frac{\beta}{(2\lambda_1 + \beta)}\left(\mathcal{Y} + \frac{\Lambda_1^k}{\beta} - (\mathcal{F}^{k+1} + \mathcal{S}^{k+1})\right). \qquad (38)$$

The update of Λ_1^{k+1} can be formulated as

$$\Lambda_1^{k+1} := \Lambda_1^k + \beta[\mathcal{Y} - (\mathcal{F}^{k+1} + \mathcal{N}^{k+1} + \mathcal{S}^{k+1})] \qquad (39)$$

A. Initialization and Penalty Parameter Updating

It is known that a good initialization of the step size for outer iterations can significantly reduce the line search cost and hence speed up the overall convergence of an algorithm. In this paper, we adopt the so-called last rule to initialize the penalty parameter. Specifically, β^{k+1} is initialized using the finally accepted value of β at the last iteration.

With a fixed β, ADMM converges slowly [50]. Consequently, an adaptive updating strategy for the penalty parameter is adopted

$$\beta_{k+1} = \min(\beta_{\max}, \eta\beta_k) \qquad (40)$$

where $\beta_{\max}$ is the upper bound for β. In our experiments, we empirically set the lower bound $\beta_{\min}$ as 10^{-4}, and the upper bound $\beta_{\max}$ as 10^4. By combining the updating rule for step size β proposed in [50] and our problem, we obtain a proximal value assignment of η

$$\eta = \begin{cases}\eta_0, & \text{if} \quad \beta_k\gamma_k/\|\mathcal{Y}\| < \epsilon_2\\ 1, & \text{otherwise}\end{cases} \qquad (41)$$

where $\gamma_k = \max(\|\mathcal{F}^{k+1} - \mathcal{F}^k\|, \|\mathcal{N}^{k+1} - \mathcal{N}^k\|, \|\mathcal{S}^{k+1} - \mathcal{S}^k\|)$, $\beta_0 = \alpha\epsilon_2$, and α depends on the size of $\mathcal{Y}$.

B. Global Analysis

In this section, we mainly analyze the convergence of ADMM for solving problem defined in (17). We observe that $f_1(\cdot) = \|\cdot\|_*$, $f_2(\cdot) = \|\cdot\|_F^2$, and $f_3(\cdot) = \|\cdot\|_1$. $f_2(\cdot)$ are strongly convex, while $f_1(\cdot)$ and $f_3(\cdot)$ are convex terms, but may not be strongly convex. The problem defined in (17) can be reformulated as

$$\begin{aligned}\min_{\mathcal{F},\mathcal{N},\mathcal{S}} \quad & f_1(\mathcal{F}) + \lambda_1 f_2(\mathcal{N}) + \lambda_2 f_3(\mathcal{S})\\ \text{s.t.} \quad & \mathcal{Y} = \mathcal{F} + \mathcal{N} + \mathcal{S}.\end{aligned} \qquad (42)$$

Definition 4: (Convex and Strongly Convex) Let $f : \mathbb{R}^n \rightarrow [-\infty, +\infty]$, if the domain of f denoted by $\operatorname{dom} f := \{\mathbf{x} \in \mathbb{R}^n, f(\mathbf{x}) < +\infty\}$ is not empty, f is considered to be proper. If for any $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$, we always have $f(t\mathbf{x} + (1-t)\mathbf{y}) \leq tf(\mathbf{x}) + (1-t)f(\mathbf{y}), \forall t \in [0,1]$, then it is considered that f is convex. Furthermore, f is considered to be strongly convex with the modulus $\mu > 0$, if

$$\begin{aligned}f(t\mathbf{x} + (1-t)\mathbf{y}) &\leq tf(\mathbf{x}) + (1-t)f(\mathbf{y})\\ &- \frac{1}{2}\mu t(1-t)\|\mathbf{x} - \mathbf{y}\|^2, \forall t \in [0,1].\end{aligned} \qquad (43)$$

Cai *et al.* [51] and Li *et al.* [52] have proved the convergence of Extended ADMM (e-ADMM) with only one strongly convex function for the case $m = 3$.

Assumption 1: In (42), f_1 and f_3 are convex, and f_2 is strongly convex with modulus $\mu_2 > 0$.

Assumption 2: The optimal solution set for the problem defined in (17) is nonempty, i.e., there exist $(\mathcal{F}^*, \mathcal{N}^*, \mathcal{S}^*, \Lambda^*) \in \Omega^*$ such that the following requirements can be satisfied:

$$\mathbf{0} = \nabla f_1(\mathcal{F}^*) - \Lambda^* \qquad (44)$$

$$\mathbf{0} = \lambda_1\nabla f_2(\mathcal{N}^*) - \Lambda^* \qquad (45)$$

$$\mathbf{0} = \lambda_2\nabla f_3(\mathcal{S}^*) - \Lambda^* \qquad (46)$$

$$\mathbf{0} = \mathcal{F}^* + \mathcal{N}^* + \mathcal{S}^* - \mathcal{Y}. \qquad (47)$$

Theorem 4: Assume that Assumptions 1 and 2 hold. Let $(\mathcal{F}^k, \mathcal{N}^k, \mathcal{S}^k, \Lambda^k)$ be the sequence generated by Algorithm 1 for solving the problem defined in (17). If $\beta \in (0, \frac{6\mu_2}{13})$, the limit point of $(\mathcal{F}^k, \mathcal{N}^k, \mathcal{S}^k, \Lambda^k)$ is an optimal solution to (17). Moreover, the objective function converges to the optimal value and the constraint violation converges to zero, i.e.,

$$\lim_{k\to\infty} \|f_1(\mathcal{F}^*) + \lambda_1 f_2(\mathcal{N}^*) + \lambda_2 f_3(\mathcal{S}^*) - f^*\| = 0 \tag{48}$$

and

$$\lim_{k\to\infty} \|\mathcal{Y} - (\mathcal{F} + \mathcal{N} + \mathcal{S})\| = 0 \tag{49}$$

where f^* denotes the optimal objective value for the problem defined in (17). In our specific application, $\beta \in (0, \frac{6*2\lambda_1}{13})$ can sufficiently ensure the convergence [51].

C. Stopping Criterion

Main Result 1: It can be easily verified that the iterations generated by the proposed ADMM algorithm can be characterized by

$$\begin{cases} \mathbf{0} \in \partial\|\mathcal{F}^{k+1}\|_* - [\Lambda_1^k - \beta(\mathcal{F}^{k+1} + \mathcal{S}^k + \mathcal{N}^k - \mathcal{Y})] \\ \mathbf{0} \in \partial(\lambda_2\|\mathcal{S}^{k+1}\|_1) - [\Lambda_1^k - \beta(\mathcal{F}^{k+1} + \mathcal{S}^{k+1} + \mathcal{N}^k - \mathcal{Y})] \\ \mathbf{0} \in \partial(\lambda_1\|\mathcal{N}^{k+1}\|_F^2) - [\Lambda_1^k - \beta(\mathcal{F}^{k+1} + \mathcal{S}^{k+1} \\ \quad + \mathcal{N}^{k+1} - \mathcal{Y})] \\ \Lambda_1^{k+1} = \Lambda_1^k - \beta[(\mathcal{F}^{k+1} + \mathcal{N}^{k+1} + \mathcal{S}^{k+1}) - \mathcal{Y}] \end{cases} \tag{50}$$

which is equivalent to

$$\begin{cases} \mathbf{0} \in \partial\|\mathcal{F}^{k+1}\|_* - \Lambda_1^{k+1} + \beta(\mathcal{S}^k - \mathcal{S}^{k+1}) + \beta(\mathcal{N}^k - \mathcal{N}^{k+1}) \\ \mathbf{0} \in \partial(\lambda_2\|\mathcal{S}^{k+1}\|_1) - \Lambda_1^{k+1} + \beta(\mathcal{N}^k - \mathcal{N}^{k+1}) \\ \mathbf{0} \in \partial(\lambda_1\|\mathcal{N}^{k+1}\|_F^2) - \Lambda_1^{k+1} \\ \Lambda_1^{k+1} = \Lambda_1^k - \beta[(\mathcal{F}^{k+1} + \mathcal{N}^{k+1} + \mathcal{S}^{k+1}) - \mathcal{Y}]. \end{cases} \tag{51}$$

Equation (51) shows that the distance of the iterations $(\mathcal{F}^{k+1}, \mathcal{N}^{k+1}, \mathcal{S}^{k+1})$ to the solution $(\mathcal{F}^*, \mathcal{N}^*, \mathcal{S}^*, \Lambda^*)$ can be characterized by $\beta(\|\mathcal{S}^k - \mathcal{S}^{k+1}\| + \|\mathcal{N}^k - \mathcal{N}^{k+1}\|)$ and $\frac{1}{\beta}\|\Lambda_1^k - \Lambda_1^{k+1}\|$. Thus, a straightforward stopping criterion for Algorithm 1 is

$$\min\{\beta(\|\mathcal{S}^k - \mathcal{S}^{k+1}\| + \|\mathcal{N}^k - \mathcal{N}^{k+1}\|), \frac{1}{\beta}\|\Lambda_1^k - \Lambda_1^{k+1}\|\} \leq \epsilon. \tag{52}$$

Here, ϵ is an infinitesimal number, e.g., 10^{-6}.

D. Choice of Parameters

In our optimization framework, given in (24), there are three parameters β, λ_1, and λ_2. As previously mentioned in Section III-B, λ_2 does not need any tuning and can be set to $1/\sqrt{n_{(1)}n_3}$, where $n_{(1)} = \max\{n_1, n_2\}$. Besides, the value of β is limited to the range of $(0, \frac{6*2\lambda_1}{13})$ to ensure the convergence of our algorithm (based on the analysis in Theorem 4). Thus, the value of λ_1 is important for the performance of our algorithm.

Theorem 5: Supposing that the Gaussian noise term $\mathbf{N} \in \mathbb{R}^{n\times n}$, and each entry $n_{i,j}$ is i.i.d. normally distributed, we can have that for $\mathcal{N}(0, \sigma^2)$, $\|\mathbf{N}\|_F^2 \leq (n + \sqrt{8n})\sigma^2$ with a high probability [53].

Theorem 5 can be extended to the tensor case. For a Gaussian noise term $\mathcal{N} \in \mathbb{R}^{n_1\times n_2\times n_3}$, and each entry $n_{i,j,k}$ is i.i.d. normally distributed, we can have that for $\mathcal{N}(0, \sigma^2)$, $\|\mathcal{N}\|_F^2 \leq (n_{(1)}n_3 + \sqrt{8n_{(1)}n_3})\sigma^2$ with a high probability. We define $\delta = \sqrt{(n_{(1)}n_3 + \sqrt{8n_{(1)}n_3})}\sigma$. Let the value of $1/2\lambda_1$ be changed within the range of $(\frac{\delta}{10}, 10\delta)$ to find the optimal value of λ_1 to achieve satisfying restoration results.

E. Analysis of Computational Complexity

The updating of $\mathcal{F}^{k+1}$, $\mathcal{S}^{k+1}$, and $\mathcal{F}^{k+1}$ have closed-form solutions, as shown in Algorithm 1. It is easy to see that the major cost in each iteration lies in the updating of $\mathcal{F}^{k+1}$, which requires the calculation of FFT and n_3 SVDs of $n_1 \times n_2$ matrices. Thus, the complexity for each iteration is $O(n_1n_2n_3\log(n_3) + n_{(1)}n_{(2)}n_3)$. We denote that $n_{(1)} = \max\{n_1, n_2\}$ and $n_{(2)} = \min\{n_1, n_2\}$.

V. Experiments

In this section, we conduct experiments on simulated and real image datasets to test the HSI denoising performance of the proposed LRTR algorithm. To thoroughly evaluate the proposed algorithm, we select four methods for comparison, i.e., the LRMR method [3], the LRTA method [28], the PARAFAC method [31], and the VBM3D method [33]. LRMR is a well-established low-rank matrix recovery method for HSI denoising. LRTA and PARAFAC are tensor decomposition based approaches for HSI denoising. VBM3D is an extension of the block-matching and 3-D filtering (BM3D) for single-image denoising. As the VBM3D method does not remove impulse noise. We first use robust PCA (RPCA) to filter the impulse noise, and then VBM3D is applied to the low-rank part of the RPCA model. We denote this restoration method as RPCA+VBM3D. In the experiments, we tune the parameters for the proposed method and the other four compared methods. LRTA is a parameter-free method. For the PARAFAC method, we set the thresholds for constancy of diagonal elements and the energy finiteness of off-diagonal elements in residual covariance matrix to be 10^{-7} and 10^{-6}, respectively. For the LRMR method, the upper bound rank for LRMR is manually adjusted to achieve its best performance. The noise variance for VBM3D and the proposed LRTR method is estimated using the multiple regression theory-based approach [54]. In addition, the gray values in each HSI band are normalized to [0,1] before denoising.

A. Mixed-Noise Removal on Simulated Data

The synthetic data were generated using two public HSI datasets, including the Washington DC Mall dataset and the Pavia city center dataset. The Washington DC Mall dataset was collected by the hyperspectral digital imagery collection experiment, and the whole image contains 1208×307 pixels and 191 spectral bands. In our experiments, only a subimage of size $256 \times 256 \times 191$ was used, as shown in Fig. 1(a). The Pavia city center dataset was collected by the reflective optics system imaging spectrometer (ROSIS-03). The first 22 bands (containing all the noisy bands) of this data were removed and the size of the subimage was set to $200 \times 200 \times 80$, as shown in Fig. 1(b). To test different noise reduction methods, the Peak SNR (PSNR) and structural similarity index measurement (SSIM) [55] were used. For HSI, we computed the PSNR and SSIM values

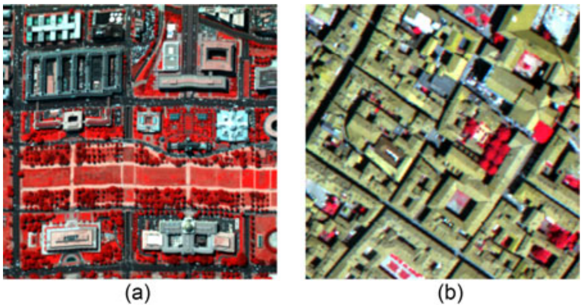


Fig. 1. (a) Pseudocolor image of the Washington DC Mall dataset (R: 60, G: 27, and B: 17). (b) Pseudocolor image of the Pavia City Center dataset (R: 80, G: 34, and B: 9).

between each noise-free band and the denoised band, and then averaged them. These metrics are denoted as mean PSNR (MPSNR) and mean SSIM (MSSIM).

In experiments, noisy hyperspectral data were generated by adding sparse noise and Gaussian noise to the two reference HSI datasets, as the following three cases:

Case One: To test the sparse noise removal performance of different algorithms, we added sparse noise S with different sparsity and Gaussian noise with a fixed variance to all bands of the Washington DC Mall data and the Pavia City Center data. The mean of Gaussian noise was set to zero and its variance was set to 0.02. Then, different percentages of sparse noise were considered. The percentages were set to 0.05, 0.10, 0.15, and 0.20, respectively.

Case Two: To test the Gaussian noise removal performance of different algorithms, we added Gaussian noise with different variance and sparse noise $\mathcal{S}$ with a fixed sparsity. Gaussian noise was added to the Washington DC Mall data and the Pavia City Center data. The mean of Gaussian noise was set to zero while different variance were considered, including 0.02 and 0.04. Sparse noise with a percentage of 0.10 was added to the data.

Case Three: Zero-mean Gaussian noise with different variances was added to each band of the two HSI datasets. The variance values were randomly selected from 0.02 to 0.04. Impulse noise was added to all bands of the Washington DC Mall dataset and the Pavia City Center dataset. The percentage of impulse noise was set to 0.10. For the Washington DC Mall data, dead lines were added to 11 selected bands from band 80 to 90. The width of the dead lines ranged from one line to three lines. Stripes were added to four bands from band 91 to 94. For the Pavia City Center data, dead lines were added to 11 bands from band 30 to 40. The width of the dead lines ranged from one line to three lines. Stripes were added to 4 bands from band 41 to 44.

These types of sparse noise are simulated in different ways.

1) Speckle noise: the speckle noise is generated by adding salt and pepper noise to the HSIs.
2) Dead pixels or lines, and stripes: the model for stripes and dead pixels can be described as

$$g_{x,y} = A_{x,y} z_{x,y} + B_{x,y} \tag{53}$$

where (x, y) denotes pixel coordinate, $z_{x,y}$ is the useful signal for pixel (x, y) and $A_{x,y}$ and $B_{x,y}$ are the relative gain and offset parameter. For healthy pixels, the gain and bias should be one and zero, respectively. For dead pixels, the gain is set to zero and the bias is set to be the pixel value. For stripes, the pixel values in a whole row or column are the same [56].

The denoising results achieved by different algorithms on the Washington DC Mall and the Pavia City Center data in Case One are presented in Tables I and II, respectively. The best result for each metric is labeled in bold. From Tables I and II, it can be seen that the proposed method provides the highest values in both MPSNR and MSSIM, which clearly demonstrates the advantage of the proposed method over the other methods for impulse noise removal. The denoising results achieved by different algorithms on the two HSI datasets in Case Two are shown in Tables III and IV. In Tables III and IV, the MPSNR and MSSIM values achieved by LRMR are comparable to those achieved by LRTR. They are significantly superior to RPCA+VBM3D, LRTA, and PARAFAC. The results indicate that these low-rank-regularized methods can better remove Gaussian noise with the existence of impulse noise.

For Case Three, we select six typical bands of the two HSI datasets for visual inspection. Figs. 2 and 5 show denoising performance with different methods on band 6 of the Washington DC Mall data and band 1 of the Pavia City Center data, which are only contaminated by impulse noise and zero-mean Gaussian noise. Figs. 3 and 6, respectively, show denoising results on band 88 of the Washington DC Mall data and band 40 of the Pavia City Center data, which are corrupted by dead lines, apart from Gaussian noise and impulse noise. Figs. 4 and 7, respectively, show denoising results on band 94 of the Washington DC Mall data and band 43 of the Pavia City Center data, which are contaminated by stripes, Gaussian noise and impulse noise. From Figs. 2 and 5, it can be observed that the proposed method and the LRMR method can remove Gaussian noise efficiently and preserve the edge and detailed information simultaneously, as compared to the other three methods. Some detailed information is lost for RPCA+VBM3D. There is still some Gaussian noise remaining in images denoised by LRTA and PARAFAC. Meanwhile, the proposed LRTR algorithm outperforms LRMR and can more effectively suppress Gaussian noise. From Figs. 3 and 6,it can be observed that the proposed algorithm performs best and can effectively remove dead lines. RPCA+VBM3D, LRTA, and PARAFAC fail to remove either Gaussian noise, impulse noise, or dead lines. LRMR can effectively remove Gaussian noise and impulse noise. But, the dead lines remaining in the image restored by LRMR are more obvious than the image restored by LRTR. From Figs. 4 and 7, it can be seen that the LRTR method performs best in stripe removal and preserves the edge and detailed information. The effectiveness of the proposed algorithm can also be illustrated by MPSNR and MSSIM shown in Tables V and VI. The MPSNR and MSSIM values achieved by the proposed method are higher than the other four compared methods, which are consistent with the visual results shown in Figs. 2–7.

B. *Experiments on Real HSI Data Corrupted by Sparse and Gaussian Noise*

In this part, we consider the real noisy Indian Pines dataset. This dataset was acquired by the NASA Airborne Visible/ Infrared Imaging Spectrometer (AVIRIS) instrument over the

TABLE I
QUANTITATIVE EVALUATION OF DIFFERENT DENOISING ALGORITHMS ON THE WASHINGTON DC MALL DATASET IN CASE ONE

Sparse percentage	Metrics	RPCA+VBM3D	LRTA	PARAFAC	LRMR	LRTR
0.05	MPSNR (dB)	29.1156	24.3114	24.8592	32.6698	**33.7431**
	MSSIM	0.5580	0.4491	0.4834	0.7521	**0.7949**
0.10	MPSNR (dB)	28.4188	20.6820	20.5284	32.1473	**32.9455**
	MSSIM	0.5361	0.3899	0.3452	0.7306	**0.7612**
0.15	MPSNR (dB)	27.2548	17.0897	16.9002	31.6098	**32.1761**
	MSSIM	0.4999	0.2987	0.2358	0.7075	**0.7281**
0.20	MPSNR (dB)	25.4779	14.2354	14.0786	31.0435	**31.4059**
	MSSIM	0.4455	0.2304	0.1561	0.6819	**0.6943**

TABLE II
QUANTITATIVE EVALUATION OF DIFFERENT DENOISING ALGORITHMS ON THE PAVIA CITY CENTER DATASET IN CASE ONE

Sparse percentage	Metrics	RPCA+VBM3D	LRTA	PARAFAC	LRMR	LRTR
0.05	MPSNR (dB)	28.2734	26.4367	26.7632	29.8486	**30.0233**
	MSSIM	0.8024	0.7871	0.8126	0.8633	**0.9266**
0.10	MPSNR (dB)	27.6808	22.2802	22.7387	29.2546	**30.0122**
	MSSIM	0.7791	0.6392	0.6996	0.8479	**0.9108**
0.15	MPSNR (dB)	26.5413	18.7495	19.0216	28.7257	**29.7580**
	MSSIM	0.7309	0.4729	0.5378	0.8322	**0.8715**
0.20	MPSNR (dB)	24.6956	15.8978	16.1163	28.1046	**28.2005**
	MSSIM	0.6443	0.3172	0.3814	0.5092	**0.5596**

TABLE III
QUANTITATIVE EVALUATION OF DIFFERENT DENOISING ALGORITHMS ON THE WASHINGTON DC MALL DATASET IN CASE TWO

Noise variance	Metrics	RPCA+VBM3D	LRTA	PARAFAC	LRMR	LRTR
0.02	MPSNR (dB)	28.7544	22.4198	20.6719	32.1473	**32.9455**
	MSSIM	0.5464	0.7055	0.6417	0.7306	**0.7612**
0.04	MPSNR (dB)	26.7479	22.2973	20.5941	30.7509	**31.6802**
	MSSIM	0.4840	0.4030	0.3544	0.6727	**0.7108**

TABLE IV
QUANTITATIVE EVALUATION OF DIFFERENT DENOISING ALGORITHMS ON THE PAVIA CITY CENTER DATASET IN CASE TWO

Noise variance	Metrics	RPCA+VBM3D	LRTA	PARAFAC	LRMR	LRTR
0.02	MPSNR (dB)	27.9607	23.9926	24.4596	29.2546	**30.1751**
	MSSIM	0.7905	0.7055	0.7533	0.8479	**0.8809**
0.04	MPSNR (dB)	26.3082	22.4087	22.8522	27.4661	**27.9068**
	MSSIM	0.7190	0.6417	0.6960	0.7927	**0.8233**

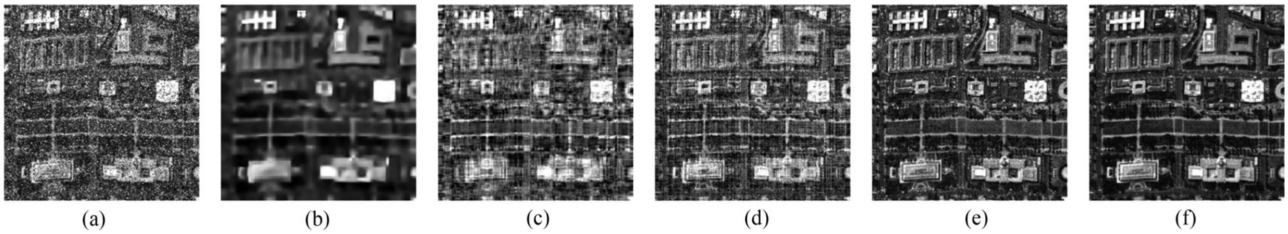


Fig. 2. Denoising results on the Washington DC Mall dataset in Case Three. (a) Noisy band 6, (b) RPCA+VBM3D, (c) LRTA, (d) PARAFAC, (e) LRMR, and (f) LRTR.

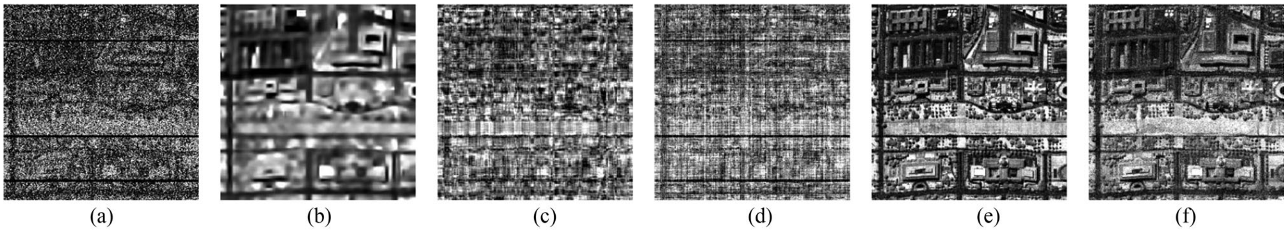


Fig. 3. Denoising results on the Washington DC Mall dataset in Case Three. (a) Noisy band 88, (b) RPCA+VBM3D, (c) LRTA, (d) PARAFAC, (e) LRMR, and (f) LRTR.

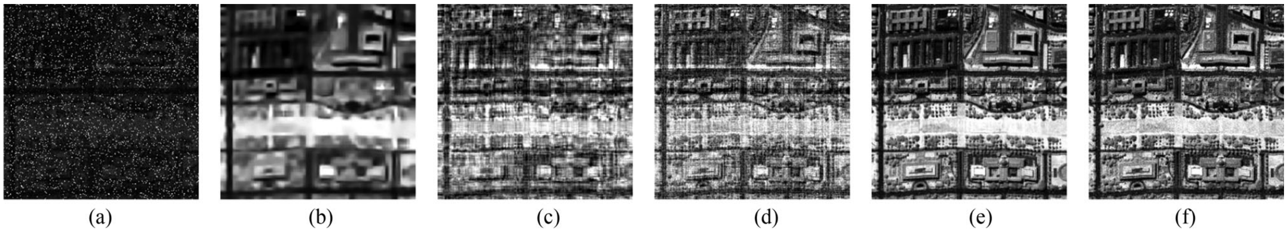


Fig. 4. Denoising results on the Washington DC Mall dataset in Case Three. (a) Noisy band 94, (b) RPCA+VBM3D, (c) LRTA, (d) PARAFAC, (e) LRMR, and (f) LRTR.

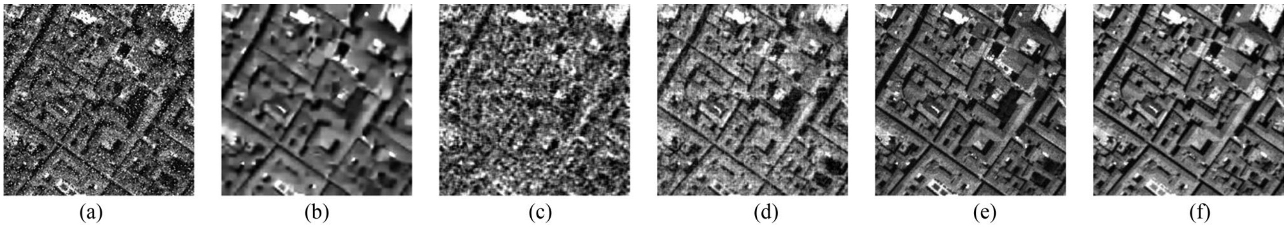


Fig. 5. Denoising results on the Pavia City Center dataset in Case Three. (a) Noisy band 1, (b) RPCA+VBM3D, (c) LRTA, (d) PARAFAC, (e) LRMR, and (f) LRTR.

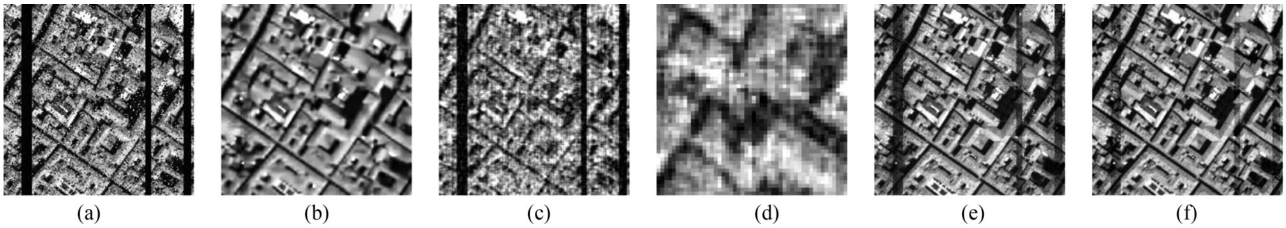


Fig. 6. Denoising results on the Pavia City Center dataset in Case Three. (a) Noisy band 40, (b) RPCA+VBM3D, (c) LRTA, (d) PARAFAC, (e) LRMR, and (f) LRTR.

Indian Pines test site in Northwestern Indiana in 1992. The Indian Pines dataset has a size of $145 \times 145 \times 220$, covering the wavelength range of 0.4–2.5 μm. The number of bands is reduced to 200 by removing the bands covering the region of water absorption: 104–108, 150–163, and 220. The pseudocolor images composed of bands 3, 147, and 219, before and after denoising, are shown in Fig. 8. It can be observed that LRTR obtains the best performance. The proposed LRTR method effectively suppresses Gaussian noise and sparse noise, while preserving local details and structural information of the image. LRMR can obtain a comparable result to LRTR, but some dense noise can still be found in the restored image. For RPCA+VBM3D, the noise is well suppressed, but the result is oversmoothed and some details are lost. The PARAFAC introduces some distortions. LRTA can remove noise and preserve details, but not completely.

Since the original images in bands 1 and 219 are seriously corrupted by noise, there is almost no useful information without denoising. Figs. 9 and 11 show the denoising results achieved by the proposed method and four compared methods in bands 1 and 219, respectively. In Figs. 9 and 11, it can be seen that the proposed LRTR method performs best. The edge

TABLE V
QUANTITATIVE EVALUATION OF DIFFERENT DENOISING ALGORITHMS ON THE WASHINGTON DC MALL DATASET IN CASE THREE

Metrics	RPCA+VBM3D	LRTA	PARAFAC	LRMR	LRTR
MPSNR (dB)	36.2625	24.8013	25.4436	35.9243	**40.6451**
MSSIM	0.8003	0.4470	0.4853	0.7755	**0.8950**

TABLE VI
QUANTITATIVE EVALUATION OF DIFFERENT DENOISING ALGORITHMS ON THE PAVIA CITY CENTER DATASET IN CASE THREE

Metrics	RPCA+VBM3D	LRTA	PARAFAC	LRMR	LRTR
MPSNR (dB)	29.0943	24.1558	26.1668	34.7077	**37.0560**
MSSIM	0.8232	0.6715	0.8051	0.9498	**0.9665**

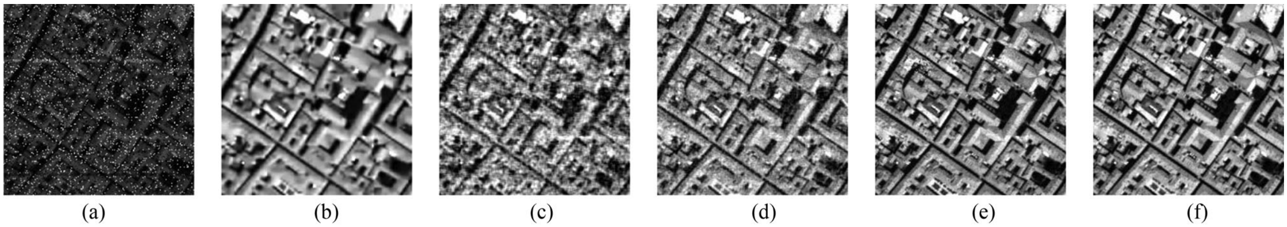


Fig. 7. Denoising results on the Pavia City Center dataset in Case Three. (a) Noisy band 43, (b) RPCA+VBM3D, (c) LRTA, (d) PARAFAC, (e) LRMR, and (f) LRTR.

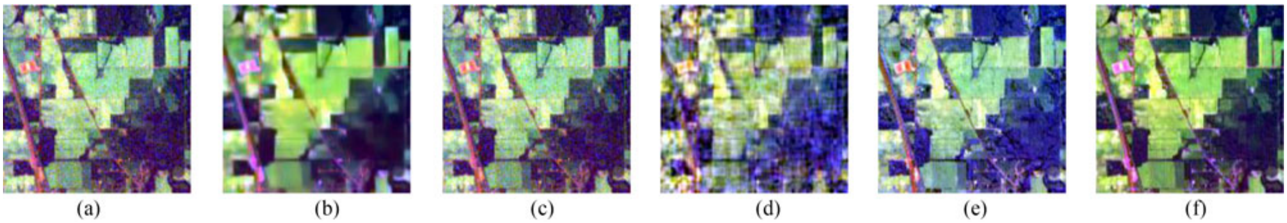


Fig. 8. Denoising results on the AVIRIS Indian Pines dataset (R: 3, G: 147, and B: 219). (a) Original, (b) RPCA+VBM3D, (c) LRTA, (d) PARAFAC, (e) LRMR, and (f) LRTR.

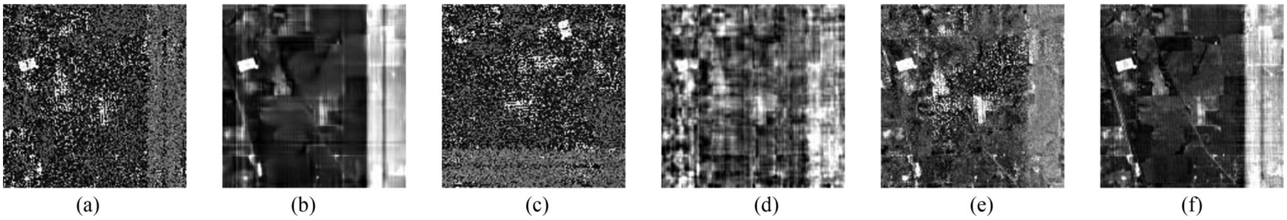


Fig. 9. Denoising results on band 1 of the Indian Pines dataset. (a) Original image, (b) RPCA+VBM3D, (c) LRTA, (d) PARAFAC, (e) LRMR, and (f) LRTR.

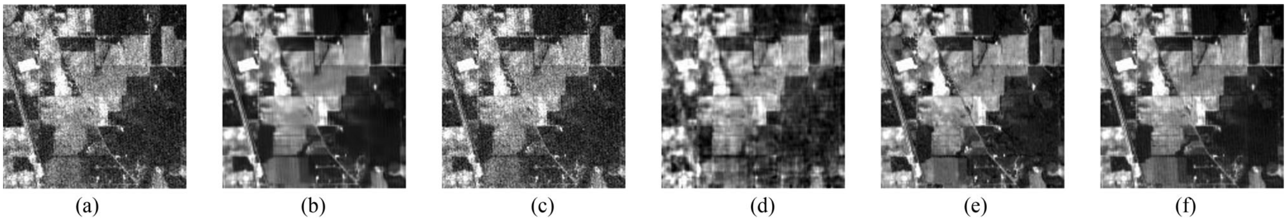


Fig. 10. Denoising results on band 3 of the Indian Pines dataset. (a) Original image, (b) RPCA+VBM3D, (c) LRTA, (d) PARAFAC, (e) LRMR, and (f) LRTR.

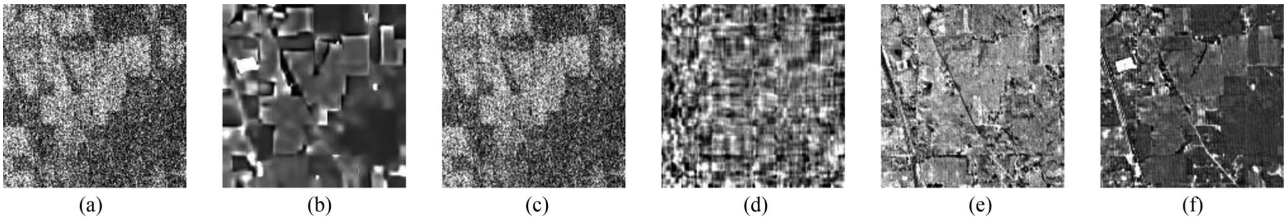


Fig. 11. Denoising results on band 219 of the Indian Pines dataset. (a) Original image, (b) RPCA+VBM3D, (c) LRTA, (d) PARAFAC, (e) LRMR, and (f) LRTR.

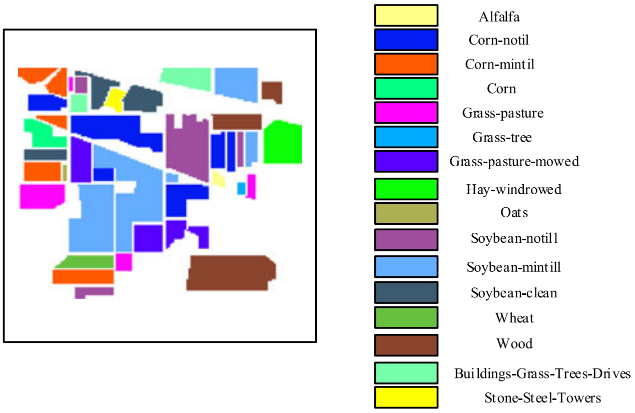


Fig. 12. Ground truth of class labels of the 16 land-cover classes in Indian Pines dataset

information can be better preserved by LRTR than the other four methods. The LRTA and PARAFAC methods perform poorly and fail to remove noise as compared to the original noisy images. Quite a few distortions are introduced in the PARAFAC method. RPCA+BM3D can well suppress heavy noise, but a large amount of local details and structural information in the recovered image are lost.The image restored by LRMR still has heavy noise. The restoration results on band 3 are given in Fig. 10. In Fig. 10, it can be observed that the result achieved by RPCA+VBM3D is oversmoothed. It indicates that VBM3D cannot perform well in heavy noise, that is, because the similarity among grouped blocks is dependent on noise level. There is still a lot of stripe noise remaining in the image recovered by the LRTA method. The image restored by PARAFAC has some distortions. The performance of the LRMR method is good except in the marginal area since LRMR is a patch-based denoising method, which may lead to a loss of interdimensional information and details. It can be noticed that noise can be removed by LRTR and the edge and texture information are well preserved in the image.

HSI classification is an important subsequent application for an HSI denoising method. Thus, the classification accuracy can be adopted to evaluate the denoising performance. Here, a support vector machine (SVM) is utilized to conduct supervised classification for HSI data [57]. The main idea of SVM is to project the data to a higher dimensional space and to use a hyperplane to obtain a better separation. The effectiveness of these methods is evaluated in terms of classification accuracy: the overall accuracy (OA) and class-specific accuracy (CA). A total of 10 249 samples are divided into 16 classes and are not mutually exclusive, as shown in Fig. 12. A total of 1045 samples (about 10%) are selected as training samples. The number of training samples for each class is shown in Table VII. The left samples are used for testing. To make the classification performance achieved by different methods more reliable, the training samples are randomly selected for 100 times. The performances of all algorithms are compared using the mean and standard deviation of the overall classification accuracy.

Table VII shows the OA and CA values achieved by the proposed method and other four compared methods. It can be seen that the OA and CA values are improved when noise is reduced. It indicates that the denoising step can improve the classification performance. Among all the classification results achieved by the five denoising methods, the proposed LRTR method achieves the highest OA and CA values, indicating the best performance in noise removal.

C. Discussion

1) Convergence of LRTR: To verify the convergence of LRTR, we first define variables Error, chg$\mathcal{Y}$, chg$\mathcal{F}$, chg$\mathcal{S}$, and chg for each iteration k, as shown in the following equations:

$$\text{Error} := \|\mathcal{Y} - \mathcal{F}^k - \mathcal{S}^k - \mathcal{N}^k\|_F \tag{54}$$

$$\begin{aligned}\text{chg}\mathcal{Y} &:= \|\mathcal{Y} - \mathcal{F}^k - \mathcal{S}^k - \mathcal{N}^k\|_\infty \\ &= \max_{i_1,i_2,i_3} |\mathcal{Y}_{i_1,i_2,i_3} - \mathcal{F}^k_{i_1,i_2,i_3} - \mathcal{S}^k_{i_1,i_2,i_3} - \mathcal{N}^k_{i_1,i_2,i_3}|\end{aligned} \tag{55}$$

$$\text{chg}\mathcal{F} := \|\mathcal{F}^k - \mathcal{F}^{k-1}\|_\infty = \max_{i_1,i_2,i_3} |\mathcal{F}^k_{i_1,i_2,i_3} - \mathcal{F}^{k-1}_{i_1,i_2,i_3}| \tag{56}$$

$$\text{chg}\mathcal{S} := \|\mathcal{S}^k - \mathcal{S}^{k-1}\|_\infty = \max_{i_1,i_2,i_3} |\mathcal{S}^k_{i_1,i_2,i_3} - \mathcal{S}^{k-1}_{i_1,i_2,i_3}|. \tag{57}$$

We can define chg for each iteration k as

$$\text{chg} := \max\{\text{chg}\mathcal{F}, \text{chg}\mathcal{S}, \text{chg}\mathcal{Y}\}. \tag{58}$$

Then, we show the values of Error, chg$\mathcal{F}$, chg$\mathcal{S}$, and chg achieved at each iteration in the simulated experiment in Case Three on the Washington DC Mall data and the Pavia City Center data in Figs. 13 and 14, respectively. From Fig. 13, it can be observed that the values of Error, chg$\mathcal{F}$, chg$\mathcal{S}$, and chg decrease monotonically. The maximum number of iterations is 60. Similarly, values of these variables decrease monotonically on the Pavia City Center data, as shown in Fig. 14. When the stopping criterion is satisfied, the maximum number of iterations on the Pavia City Center data is 70.

2) Denoising Performance on Gaussian Noise Only: As illustrated in the previous experiments, the proposed LRTR method performs better than the other four compared method

TABLE VII
CLASSIFICATION ACCURACY (%) ACHIEVED ON THE INDIAN PINES DATASET USING SVM CLASSIFIER AND DIFFERENT DENOISING METHODS

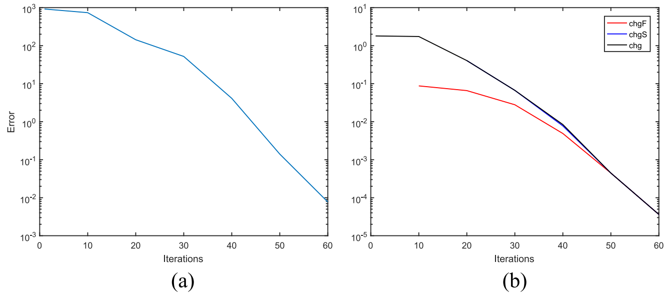
	♯ Samples	Methods					
Class	Training/Total	SVM	RPCA+VBM3D	LRTA	PARAFAC	LRMR	LRTR
Alfalfa	15/46	88.24	92.60	87.95	86.32	93.65	**93.97**
Corn-notill	100/1428	73.98	80.66	75.20	76.53	77.01	**81.28**
Corn-mintill	100/830	71.43	77.82	72.68	72.82	79.50	**81.50**
Corn	50/237	86.92	91.80	88.42	87.59	92.32	**92.72**
Grass-pasture	50/483	93.58	91.18	90.98	90.15	92.60	**93.62**
Grass-tree	100/730	97.20	99.06	98.26	98.42	98.88	**99.52**
Grass-pasture-mowed	15/28	92.38	98.46	93.90	90.05	96.24	**99.68**
Hay-windrowed	50/478	97.82	98.32	98.10	98.34	99.23	**99.75**
Oats	15/20	96.23	99.68	97.18	96.56	98.68	**99.92**
Soybeans-notill	100/972	78.78	85.20	81.35	81.48	84.54	**88.60**
Soybeans-mintill	150/2455	76.28	83.22	77.80	81.58	**84.69**	84.53
Soybeans-cleantill	50/593	74.85	81.68	76.26	77.15	84.80	**85.48**
Wheat	50/205	**99.08**	97.78	97.85	95.56	98.20	98.90
Woods	100/1265	94.75	95.83	95.85	96.10	98.16	**97.90**
Buildings-Grass-Trees-Drives	50/386	59.78	72.46	65.46	71.18	76.26	**87.60**
Stone-Steel-Towers	50/93	98.22	98.92	98.52	98.20	97.87	**99.96**
OA		81.34 ± 0.99	85.86 ± 0.58	82.60 ± 1.00	83.98 ± 0.72	86.36 ± 0.80	**88.26 ± 0.50**

TABLE VIII
QUANTITATIVE EVALUATION OF DIFFERENT DENOISING ALGORITHMS ON THE WASHINGTON DC MALL DATASET WITH GAUSSIAN NOISE ONLY

Noise variance	Metrics	VBM3D	LRTA	PARAFAC	LRMR	LRTR
0.02	MPSNR (dB)	37.0733	**41.0602**	38.4658	33.8050	39.5427
	MSSIM	0.8435	0.9107	0.9007	0.7983	**0.9175**
0.04	MPSNR (dB)	28.2912	**39.6259**	38.7180	31.6576	37.5267
	MSSIM	0.5182	0.8746	**0.8673**	0.7147	0.8742

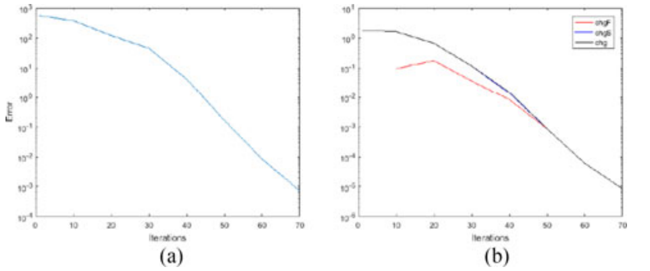
TABLE IX
QUANTITATIVE EVALUATION OF DIFFERENT DENOISING ALGORITHMS ON THE PAVIA CITY CENTER DATASET WITH GAUSSIAN NOISE ONLY

Noise Variance	Metrics	VBM3D	LRTA	PARAFAC	LRMR	LRTR
0.02	MPSNR (dB)	29.7310	31.2154	29.5260	30.6308	**32.8627**
	MSSIM	0.8453	0.7055	0.8629	0.8844	**0.9246**
0.04	MPSNR (dB)	23.1374	29.0799	29.4932	28.3887	**30.4867**
	MSSIM	0.5536	0.6417	0.8566	0.8239	**0.8811**



(a) (b)

Fig. 13. Divergence results of LRTR in the simulated data experiment in Case Three on the Washington DC Mall data. (a) Value of error at each step. (b) Value of chg$\mathcal{F}$, chg$\mathcal{S}$, and chg at each step.



(a) (b)

Fig. 14. Divergence results of LRTR in the simulated data experiment in Case Three on the Pavia City Center data. (a) Value of error at each step. (b) Value of chg$\mathcal{F}$, chg$\mathcal{S}$, and chg at each step.

when HSI data is corrupted by a mixture of Gaussian noise and sparse noise. Then, we compare these algorithms on data with Gaussian noise only. Gaussian noise was added to the Washington DC Mall data and the Pavia City Center data. The mean of Gaussian noise was set to 0 while different variances were considered, including 0.02 and 0.04. The denoising results achieved on the two HSI datasets are shown in Tables VIII and IX. In Tables VIII and IX, it can be seen that the LRTR method still achieves better results in terms of MPSNR and MSSIM.

TABLE X
RUNNING TIME COSTED BY DIFFERENT METHODS ON THE INDIAN PINES DATASET

Method	RPCA+VBM3D	LRTA	PARAFAC	LRMR	LRTR
Time(s)	25.00	1.62	150.26	41.30	30.76

It indicates that the proposed algorithm can achieve promising performance for the removal of Gaussian noise only.

3) Computational Time Comparison: In Section IV-E, a computational complexity analysis of the proposed algorithm is presented. Here, we compare the running time of the proposed method with other methods. The experiments were performed on a laptop with 3.6-GHz Intel Core CPU and 8-GB memory using MATLAB. The time costs on the Indian Pines dataset in Section V-B are reported in Table X. It can be seen that the time cost of our denoising method ranks in the middle. To make our LRTR more efficient, we can seek faster optimization scheme (such as the accelerated proximal gradient method [58]) to solve the proposed denoising model and find possible methods to accelerate the SVD when solving the rank minimization subproblem.

VI. CONCLUSION

In this paper, we have proposed an LRTR method to remove mixed noise from HSI data. In the LRTR model, the NP-hard task of tensor recovery from Gaussian and sparse noise can be well accomplished by integrating the tensor nuclear norm, l_1-norm and least square term in a unified convex relaxation framework. The convergence of the proposed algorithm is discussed. Experiments on both simulated and real HSI datasets have been conducted to demonstrate the effectiveness of the proposed denoising method.

There is space for further improvement of the proposed noise removal method. The low-rank constraint and the TV regularization can be integrated into a unified framework to complement each other rather than simply using the low-rank constraint. The low-rank and sparse tensor decomposition can provide sparse noise components for the TV regularization denoising. With the sparse noise information confirmed, the TV regularization can provide an enhanced clean image, and in return help the separation of low-rank and sparse components.

REFERENCES

[1] A. F. H. Goetz, "Three decades of hyperspectral remote sensing of the earth: A personal view," *Remote Sens. Environ.*, vol. 113, no. 9–, 2009, Art. no. S5–S16.

[2] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, and P. Scheunders, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.

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

[4] C. Li, Y. Ma, J. Huang, X. Mei, and J. Ma, "Hyperspectral image denoising using the robust low-rank tensor recovery," *J. Opt. Soc. Amer A Opt. Image Sci. Vis.*, vol. 32, no. 9, pp. 1604–1612, 2015.

[5] 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.

[6] M. J. Montag and H. Stephani, "Hyperspectral unmixing from incomplete and noisy data," *J. Imag.*, vol. 2, no. 1, p. 7, 2016.

[7] Z. Zhang, E. Pasolli, M. M. Crawford, and J. C. Tilton, "An active learning framework for hyperspectral image classification using hierarchical segmentation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 640–654, Feb. 2016.

[8] J. Ma, H. Zhou, J. Zhao, Y. Gao, J. Jiang, and J. Tian, "Robust feature matching for remote sensing image registration via locally linear transforming," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6469–6481, Dec. 2015.

[9] H. Zhang, J. Li, Y. Huang, and L. Zhang, "A nonlocal weighted joint sparse representation classification method for hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2056–2065, Jun. 2014.

[10] H. Zhang, H. Zhai, L. Zhang, and P. Li, "Spectral–spatial sparse subspace clustering for hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3672–3684, Jun. 2016.

[11] J. Yang, Y. Zhao, C. Yi, and J. C.-W. Chan, "No-reference hyperspectral image quality assessment via quality-sensitive features learning," *Remote Sens.*, vol. 9, no. 4, p. 305, 2017.

[12] 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.

[13] A. Buades, B. Coll, and J. M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, vol. 2, 2005, pp. 60–65.

[14] 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.

[15] G. Chen and S. E. Qian, "Simultaneous dimensionality reduction and denoising of hyperspectral imagery using bivariate wavelet shrinking and principal component analysis," *Can. J. Remote Sens.*, vol. 82, no. 34, pp. 447–454, 2008.

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

[17] 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.

[18] 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.

[19] 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.

[20] 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.*, vol. 397, pp. 48–68, 2017.

[21] Y. Qian and M. Ye, "Hyperspectral imagery restoration using nonlocal spectral-spatial structured sparse representation with noise estimation," *IEEE J. Selected Topics Appl. Earth Observations Remote Sensing*, vol. 6, no. 2, pp. 499–515, Apr. 2013.

[22] 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.

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

[24] M. Ye, Y. Qian, and J. Zhou, "Multitask sparse nonnegative matrix factorization for joint spectral–spatial hyperspectral imagery denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2621–2639, May 2015.

[25] 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.

[26] S. Tariyal, H. K. Aggarwal, and A. Majumdar, "Removing sparse noise from hyperspectral images with sparse and low-rank penalties," *J. Electron. Imag.*, vol. 25, no. 2, 2016, Art. no. 020501.

[27] J. Ma, J. Zhao, J. Tian, X. Bai, and Z. Tu, "Regularized vector field learning with sparse approximation for mismatch removal," *Pattern Recog.*, vol. 46, no. 12, pp. 3519–3532, 2013.

[28] N. Renard, S. Bourennane, and J. Blanc-Talon, "Denoising and dimensionality reduction using multilinear tools for hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 2, pp. 138–142, Apr. 2008.

[29] A. Karami, M. Yazdi, and A. Z. Asli, "Noise reduction of hyperspectral images using kernel non-negative tucker decomposition," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 3, pp. 487–493, Jun. 2011.

[30] D. Letexier and S. Bourennane, "Noise removal from hyperspectral images by multidimensional filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 7, pp. 2061–2069, Jul. 2008.

[31] X. Liu, S. Bourennane, and C. Fossati, "Denoising of hyperspectral images using the parafac model and statistical performance analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 10, pp. 3717–3724, Oct. 2012.

[32] X. G. Guo, X. Huang, L. Zhang, and L. Zhang, "Hyperspectral image noise reduction based on rank-1 tensor decomposition," *ISPRS J. Photogrammetry Remote Sens.*, vol. 83, pp. 50–63, 2013.

[33] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3d transform-domain collaborative filtering," in *Proc. Eur. Signal Process. Conf.*, 2007, pp. 2080–2095.

[34] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 119–33, Jan. 2013.

[35] D. Wu, Y. Zhang, and Y. Chen, "3d sparse coding based denoising of hyperspectral images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 3115–3118.

[36] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *Siam J. Matrix Anal. Appl.*, vol. 35, no. 1, pp. 225–253, 2013.

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

[38] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-SVD," *Comput. Sci.*, vol. 44, no. 9, pp. 3842–3849, 2014.

[39] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebr. Appl.*, vol. 435, no. 3, pp. 641–658, 2011.

[40] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller, "Tensor-based formulation and nuclear norm regularization for multienergy computed tomography," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1678–1693, Apr. 2014.

[41] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5249–5257.

[42] Z. Zhang, D. Liu, S. Aeron, and A. Vetro, "An online tensor robust PCA algorithm for sequential 2D data," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 2434–2438.

[43] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *Comput. Sci.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2015.

[44] V. Ollier, R. Boyer, M. N. El Korso, and P. Larzabal, "Bayesian lower bounds for dense or sparse (outlier) noise in the RMT framework," in *Proc. 2016 IEEE Sensor Array Multichannel Signal Process. Workshop*, May 2016, pp. 1–5.

[45] S. C. M. Sundin and M. Jansson, "Combined modeling of sparse and dense noise improves Bayesian RVM," in *Proc. 22nd Eur. Signal Process. Conf., Lisbon, Portugal*, 2014, pp. 1841–1845.

[46] J. M. Bioucas-Dias and M. A. Figueiredo, "Multiplicative noise removal using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1720–1730, Jul. 2010.

[47] J. Nocedal and S. J. Wright, *Numerical optimization*. Berlin, Germany: Springer, 2006.

[48] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for hierarchical sparse coding," *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 2297–2334, 2011.

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

[50] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," *Adv. Neural Inf. Process. Syst.*, pp. 612–620, 2011.

[51] X. Cai, D. Han, and X. Yuan, "The direct extension of ADMM for three-block separable convex minimization models is convergent when one function is strongly convex," *Optim. Online*, 2014, 26 pages.

[52] M. Li, D. Sun, and K. Toh, "A convergent 3-block semi-proximal admm for convex minimization problems with one strongly convex block," *Asia Pacific J. Oper. Res.*, vol. 32, no. 4, 2014, Art. no. 1550024.

[53] M. Tao and X. Yuan, "Recovering low-rank and sparse components of matrices from incomplete and noisy observations," *Siam J. Optim.*, vol. 21, no. 1, pp. 57–81, 2011.

[54] L. Gao, Q. Du, B. Zhang, W. Yang, and Y. Wu, "A comparative study on linear regression-based noise estimation for hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 488–498, Apr. 2013.

[55] 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.

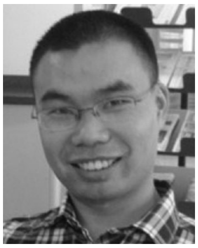
[56] H. Shen and L. Zhang, "A map-based algorithm for destriping and inpainting of remotely sensed images," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 5, pp. 1492–1502, May 2009.

[57] P. K. Gotsis, C. C. Chamis, and L. Minnetyan, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[58] H. Ji, S. Huang, Z. Shen, and Y. Xu, "Robust video restoration by joint sparse and low rank matrix approximation," *Siam J. Imag. Sci.*, vol. 4, no. 4, pp. 1122–1142, 2011.

Haiyan Fan received the B.S. degree in information engineering in 2011 and the Master's degree in information and communication engineering in January 2013 from the National University of Defense Technology, Changsha, China, where since February 2013, she has been working toward the Ph.D. degree in graph and image processing.

Her research interests include statistical signal processing, tensor analysis, and machine learning.



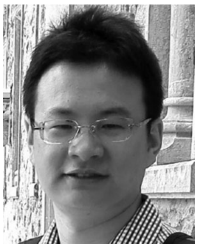
Yunjin Chen received the B.Sc. degree in applied physics from Nanjing University of Aeronautics and Astronautics, Nanjing, China, the M.Sc. degree in optical engineering from the National University of Defense Technology, Changsha, China, and the Ph.D. degree in computer science from Graz University of Technology, Graz, Austria, in 2007, 2010, and 2015, respectively.

From July 2015 to April 2017, he was a Scientific Researcher in the Government of China. He has published more than 20 papers in top-tier academic journals and conferences. He was a Tutorial Organizer in European Conference on Computer Vision 2016. More information can be found in his homepage http://www.escience.cn/people/chenyunjin/index.html. His research interests include learning image prior models for low-level computer vision problems and convex optimization.

Yulan Guo received the B.S. and Ph.D. degrees in information and communication engineering from National University of Defense Technology (NUDT), Changsha, China, in 2008 and 2015, respectively. From November 2011 to November 2014, he was a Visiting Ph.D. Student at the University of Western Australia, Perth, Western Australia.

He is currently an Assistant Professor in the College of Electronic Science and Engineering, NUDT. He has authored more than 40 peer reviewed articles in journals and conferences, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCEand *International Journal of Computer Vision*. His current research focuses on computer vision and pattern recognition, particularly on three-dimensional (3-D) feature learning, 3-D modeling, 3-D object recognition, and 3-D face recognition.

Dr. Guo was the Reviewer for more than 30 international journals and conferences. He received the NUDT Distinguished Ph.D. Thesis Award in 2015 and the CAAI Distinguished Ph. D. Thesis Award in 2016.



Hongyan Zhang (M'13–SM'16) received the B.S. degree in geographic information system and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2005 and 2010, respectively.

He is currently a Professor in the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuha University. He has authored/coauthored more than 60 research papers His research interests include image reconstruction for quality improvement, hyperspectral image processing, sparse representation, and low-rank methods for sensing image imagery.

Dr. Zhang is the reviewer of more than 20 international academic journals, including the IEEE TRANSACTION ON GEOSCIENCE AND REMOTE SENSING, THE IEEE TRANSACTION ON IMAGE PROCESSING, THE IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, and the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS.

Gangyao Kuang (M'11–SM'16) received B.S. and M.S. degrees in geophysics from the Central South University of Technology, Changsha, China, in 1998 and 1991, and the Ph.D. degree in communication and information from the National University of Defense Technology, Changsha, China, in 1995. He is currently a Professor and Director of the Remote Sensing Information Processing Laboratory, School of Electronic Science and Engineering, National University of Defense Technology. His research interests include remote sensing, SAR image processing, change detection, SAR ground moving target indication, and classification with polarimetric SAR images.