

Research Article

Hyperspectral Image Denoising via Nonconvex Logarithmic Penalty

Shuo Wang,¹ Zhibin Zhu^{ID, 2}, Ruwen Zhao,^{1,2} and Benxin Zhang¹

¹School of Electronic Engineering and Automation, Key Laboratory of Automatic Detecting Technology and Instruments, Guilin University of Electronic Technology, Guilin 541004, China

²School of Mathematics and Computing Science, Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, Guilin University of Electronic Technology, Guilin 541004, China

Correspondence should be addressed to Zhibin Zhu; optimization_zhu@163.com

Received 5 January 2021; Revised 22 March 2021; Accepted 8 June 2021; Published 24 June 2021

Academic Editor: Ana C. Teodoro

Copyright © 2021 Shuo Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Hyperspectral images (HSIs) can help deliver more reliable representations of real scenes than traditional images and enhance the performance of many computer vision tasks. However, in real cases, an HSI is often degraded by a mixture of various types of noise, including Gaussian noise and impulse noise. In this paper, we propose a logarithmic nonconvex regularization model for HSI mixed noise removal. The logarithmic penalty function can approximate the tensor fibered rank more accurately and treats singular values differently. An alternating direction method of multipliers (ADMM) is also presented to solve the optimization problem, and each subproblem within ADMM is proven to have a closed-form solution. The experimental results demonstrate the effectiveness of the proposed method.

1. Introduction

A hyperspectral image (HSI) consists of multiple discrete bands at specific frequencies. It can deliver additional information that the human eye fails to capture for real scenes and has been attracting much interest from researchers in a wide range of application fields, such as land use analysis, environmental monitoring, and field surveillance [1–3]. However, HSIs always suffer from various degradations, such as Gaussian noise, impulse noise, and random noise, which can affect the subsequent image processing, such as unmixing, classification, and target detection [4, 5]. Improving the HSI quality merely through a hardware scheme is unsustainable and impractical. Therefore, it is natural to introduce image processing-based approaches to obtain a high-quality HSI before subsequent applications.

The numerical optimization algorithm plays an important role in HSI denoising, such as Liu et al. [6] proposed a two-step wavelet-domain estimation method

to extract the noise map, and Lu et al. [7] presented some representative high-order variational models in the context of image denoising. From the perspective of the prior data format, we classify the existing HSI restoration methods into three categories: (1) 1D vector-based sparse representation methods [8–13]; (2) 2D matrix-based low-rank matrix recovery (LRMR) methods [14–21]; and, (3) 3D tensor-based approximation methods [22–33]. Although the existing works have made significant progress in HSI restoration, there are still drawbacks that need to be improved, such as when the multidimensional HSI data is transformed into a vector or matrix, it usually breaks the spectral-spatial structural correlation. The tensor low-rankness characterization of an HSI is expected to explore the global correlation and preserve the intrinsic structural information. The tensors' recovery under a limited number of measurements is an important problem that arises in a variety of applications, such as computer vision [34–36]. Based on low tubal-rank tensor recovery, Zhang et al. [37] proposed an HSI mixed noise removal model.

However, the framework of the tensor singular value decomposition (t -SVD) lacks flexibility for handling different correlations along the different modes of HSIs, leading to suboptimal denoising performance. Then, Zheng et al. [38] proposed an HSI mixed noise removal model with tensor fibered rank, which is based on the mode- k t -SVD. Moreover, Zheng et al. [38] also proposed a three-directional tensor nuclear norm (3DTNN) as its convex relaxation to provide an efficient numerical solution and a three-directional log-based tensor nuclear norm (3DLogTNN) as its nonconvex relaxation to promote the low rankness of the solution. Compared to 3DTNN, 3DLogTNN has two advantages. First, it is a closer approximation to the fibered rank than 3DTNN. Second, by using the sum of the log function of singular values, 3DLogTNN can better approximate to the fibered rank than 3DTNN.

It is well known that suitable nonconvex penalty functions induce sparsity among the singular values more effectively. However, the use of nonconvex penalty functions will lead to nonconvex optimization problems. Then, it suffers from numerous issues such as spurious local minima in the subproblem, for example, 3DLogTNN in [38]. To avoid the intrinsic difficulties, we introduce a new nonconvex logarithmic regularization model, which allows the use of nonconvex penalty function while maintaining convexity of the subproblem within ADMM. Also, the new model can provide a good approximation for the fibered rank and preserve the major information well.

The rest of the paper is structured as follows. Section 2 presents some notations, explains t -SVD and defines mode- k t -SVD. Section 3 introduces the proposed ADMM based on the parametric penalty function for HSI denoising. The experimental results and analysis are reported in Section 4. Finally, the conclusion is given in Section 5.

2. Brief Overview of Tensor Singular Value Decomposition

In this section, we first describe the notations used throughout the paper and then introduce the tensor decomposition proposed in [39–41] and mode- k t -SVD proposed in [38].

2.1. Notation and Indexing. For a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, denote $\mathcal{A}(k, :, :)$, $\mathcal{A}(:, k, :)$, and $\mathcal{A}(:, :, k)$ as the k th horizontal, lateral, and frontal slices, respectively. $\mathcal{A}(:, i, j)$, $\mathcal{A}(i, :, j)$, and $\mathcal{A}(i, j, :)$ denote the (i, j) th mode-1, mode-2, and mode-3 fibers, respectively; Figures 1 and 2 show the horizontal, lateral, and frontal slides, denoted by $\mathcal{A}(i, :, :)$, $\mathcal{A}(:, j, :)$, and $\mathcal{A}(:, :, k)$, respectively, of a third-order tensor \mathcal{A} . Suppose $\mathcal{A} = (a_{ijk}) \in \mathbb{R}^{n_1 \times n_2 \times n_3}$; we adopt the definition of the Frobenius norm of a tensor $\|\mathcal{A}\|_F = \sqrt{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} a_{ijk}^2}$ and the definition of the l_1 norm of a tensor, $\|\mathcal{A}\|_1 = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} |a_{ijk}|$.

Let A_i denote $\mathcal{A}(:, :, i)$, that is, the i th frontal slice of A . Then,

$$\text{unfold}(\mathcal{A}) = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_{n_3} \end{bmatrix} \in \mathbb{R}^{n_1 n_3 \times n_2}. \quad (1)$$

Similarly, $A_i \in \mathbb{R}^{n_1 n_3 \times n_2}$, and

$$\text{fold} \left(\begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_{n_3} \end{bmatrix} \right) = \mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}. \quad (2)$$

It can be seen in Figure 3, for $\mathcal{A} \in \mathbb{R}^{n \times n \times 3}$, let $A_i = \mathcal{A}(:, :, i)$; then, \mathcal{D}_i are the frontal slices of tensor \mathcal{D} , where \mathcal{D} is computed by applying the Fast Fourier Transform (FFT) along each tube of \mathcal{A} , i.e., $\mathcal{D}_i = \text{fft}(\mathcal{A}, [], i)$ and $\mathcal{A} = \text{ifft}(\mathcal{D}_i, [], i)$.

2.2. t -SVD. In this section, we exploit the proposed t -SVD. A t -SVD interprets third-order tensors as linear operators on the space of oriented matrices [39]. Based on a t -SVD, O. Semerci exploited the decomposition, completion, and recovery of multilinear data [42], and Zhang et al. [34] proposed novel methods for completion and denoising of multilinear data and, as an application, considered 3D and 4D (color) video data completion and denoising.

Definition 1 (t -product). For $\mathcal{G} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{K} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, the t -product $\mathcal{D} = \mathcal{G} * \mathcal{K}$ is a tensor of size $n_1 \times n_4 \times n_3$. For $i = 1, 2, \dots, n_1$ and $j = 1, 2, \dots, n_4$,

$$\mathcal{D}(i, j, :) = \sum_{k=1}^{n_2} \mathcal{G}(i, k, :) * \mathcal{K}(k, j, :). \quad (3)$$

The t -product is analogous to matrix multiplication except that circular convolution replaces the multiplication operations between the elements, which are represented by tubes.

Theorem 1 (t -SVD). For $\mathcal{D} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the t -SVD of \mathcal{D} is given by

$$\mathcal{D} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T, \quad (4)$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors, $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a rectangular diagonal tensor, and $*$ denotes the t -product.

Figure 4 illustrates the decomposition for the 3D case. Additionally, one can obtain this decomposition by computing matrix SVDs in the Fourier domain, see Algorithm 1.

However, when handling different correlations along different modes of an HSI, the t -SVD and the induced tubal rank lack flexibility. This inflexible HSI characterization usually does not have good denoising effects. Zheng et al. [38] proposed a novel tensor decomposition by generalizing the t -SVD to the mode- k t -SVD, which achieves a more flexible and accurate HSI low-rankness characterization.

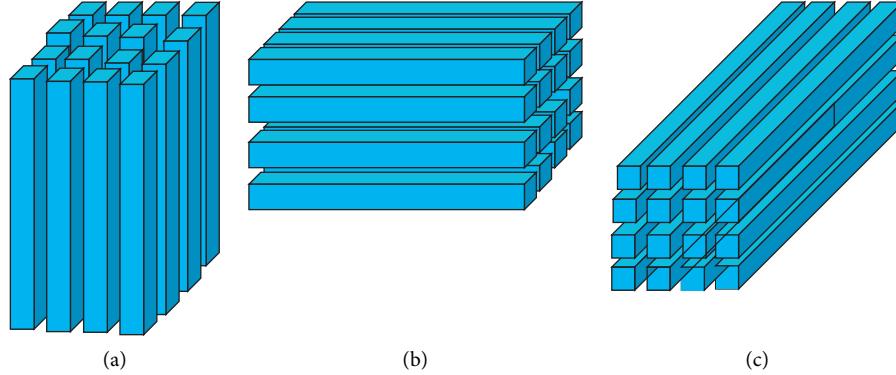


FIGURE 1: Fibers of a 3rd-order tensor. (a) Mode-1 (column) fiber: $x_{:jk}$. (b) Mode-2 (row) fiber: $x_{i:k}$. (c) Mode-3 (tube) fiber: $x_{ij:}$.

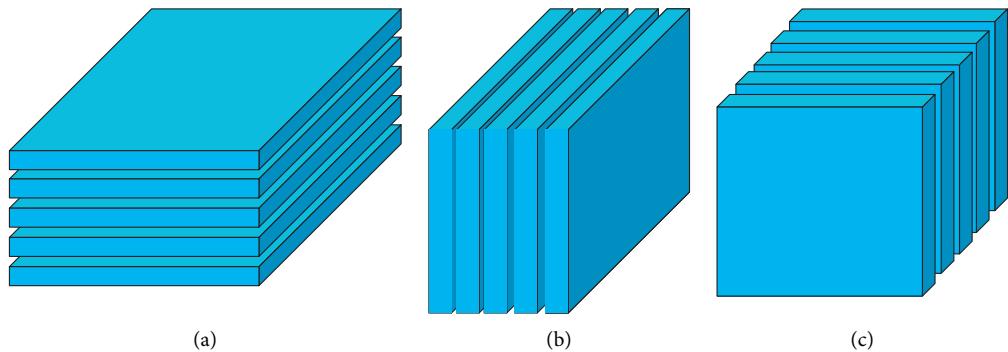


FIGURE 2: Slices of a 3rd-order tensor. (a) Horizontal slices: $x_{i::}$. (b) Lateral slices: $x_{::j}$. (c) Frontal slices: $x_{::k}$.

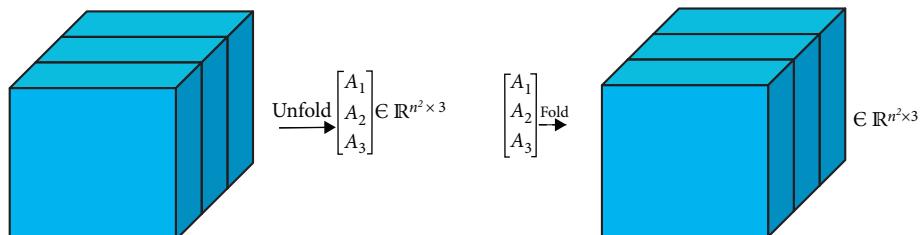


FIGURE 3: Examples of the unfold and fold operators.

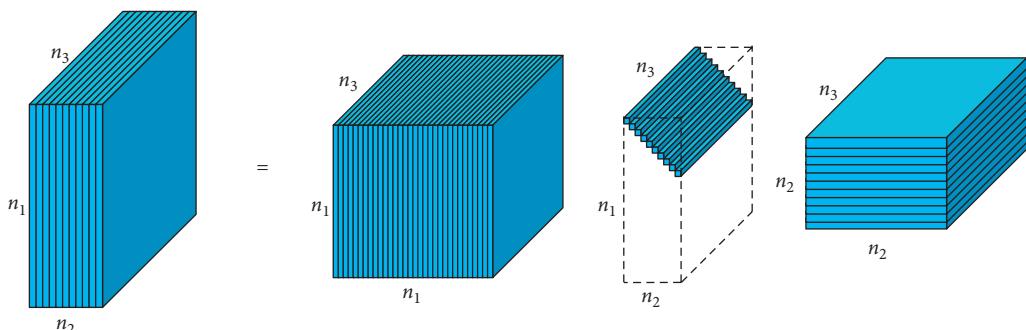


FIGURE 4: The t-SVD of a tensor.

```

Input:  $\mathcal{D} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_N}$ ,  $\bar{w} = n_3 n_4, \dots, n_N$ .
for  $i = 3$  to  $N$  do
  fft( $\mathcal{D}, i$ )  $\longrightarrow \mathcal{D}$ 
end for
for  $i = 1$  to  $\bar{w}$  do
   $[U, S, V] = \text{SVD}(\overline{\mathcal{D}}(:, :, i))$ 
   $\overline{\mathcal{U}}(:, :, i) = U; \overline{\mathcal{S}}(:, :, i) = S; \overline{\mathcal{V}}(:, :, i) = V;$ 
end for
for  $i = 3$  to  $N$  do
  ifft( $\overline{\mathcal{U}}, i$ )  $\longrightarrow \mathcal{U}$ ; ifft( $\overline{\mathcal{S}}, i$ )  $\longrightarrow \mathcal{S}$ ; ifft( $\overline{\mathcal{V}}, i$ )  $\longrightarrow \mathcal{V}$ ;
end for

```

ALGORITHM 1: t -SVD.

2.3. Mode- k t -SVD. In this section, we introduce the mode- k t -SVD and the related definitions.

For a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the mode- k block circulation operation is denoted as

$$\text{bcirc}(\mathcal{A}, k) := \begin{pmatrix} A_k^{(1)} & A_k^{(n_k)} & \dots & A_k^{(2)} \\ A_k^{(2)} & A_k^{(1)} & \dots & A_k^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ A_k^{(n_k)} & A_k^{(n_k-1)} & \dots & A_k^{(1)} \end{pmatrix}, \quad (5)$$

where $A_k^{(i)}$ is the i th mode- k slice of \mathcal{A} .

The mode- k block diagonalization operation and its inverse operation are defined as

$$\text{bdiag}(\mathcal{A}, k) := \begin{pmatrix} A_k^{(1)} \\ A_k^{(2)} \\ \ddots \\ A_k^{(n_k)} \end{pmatrix}, \quad (6)$$

$$\text{bdfold}(\text{bdiag}(\mathcal{A}, k), k) := \mathcal{A}.$$

The mode- k block vectorization operation and its inverse operation are defined as

$$\text{bvec}(\mathcal{A}, k) := \begin{pmatrix} A_k^{(1)} \\ A_k^{(2)} \\ \vdots \\ A_k^{(n_k)} \end{pmatrix}, \quad (7)$$

$$\text{bvfold}(\text{bvec}(\mathcal{A}, k), k) := \mathcal{A}.$$

Definition 2 (Mode- k t -product). For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_1 \times n_3 \times n_4}$, the mode-1 t -product is a tensor of size $n_1 \times n_2 \times n_4$:

$$\begin{aligned} \mathcal{C} &= \mathcal{A} *_1 \mathcal{B} = \text{bvfold}(\text{bcirc}(\mathcal{A}, 1)\text{bvec}(\mathcal{B}, 1), 1) \iff \mathcal{C}(:, j, t) \\ &= \sum_{s=1}^{n_3} \mathcal{A}(:, j, s) * \mathcal{B}(:, s, t). \end{aligned} \quad (8)$$

For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_4 \times n_2 \times n_1}$, the mode-2 t -product is a tensor of size $n_4 \times n_2 \times n_3$:

$$\begin{aligned} \mathcal{C} &= \mathcal{A} *_2 \mathcal{B} = \text{bvfold}(\text{bcirc}(\mathcal{A}, 2)\text{bvec}(\mathcal{B}, 2), 2) \iff \mathcal{C}(i, :, t) \\ &= \sum_{s=1}^{n_1} \mathcal{A}(s, :, t) * \mathcal{B}(i, :, s). \end{aligned} \quad (9)$$

For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, the mode-3 t -product is a tensor of size $n_1 \times n_4 \times n_3$:

$$\begin{aligned} \mathcal{C} &= \mathcal{A} *_3 \mathcal{B} = \text{bvfold}(\text{bcirc}(\mathcal{A}, 3)\text{bvec}(\mathcal{B}, 3), 3) \iff \mathcal{C}(i, j, :) \\ &= \sum_{s=1}^{n_1} \mathcal{A}(i, s, :) * \mathcal{B}(s, j, :). \end{aligned} \quad (10)$$

Definition 3 (Mode- k identity tensor). $\mathcal{I}_k \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the mode- k identity tensor, whose first mode- k slice is an identity matrix and other mode- k slices are all zeros.

Definition 4 (Mode- k conjugate transpose). For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{A}^{T_k} \in \mathbb{R}^{n_2 \times n_1 \times n_3}$ is the mode- k conjugate transpose of \mathcal{A} , which is obtained by transposing each of the mode- k slices and then reversing the order of transposed mode- k slices 2 through n_k .

Definition 5 (Mode- k diagonal tensor). For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, each of its mode- k slices is a diagonal matrix, and then, \mathcal{A} is a mode- k diagonal.

Definition 6 (Mode- k orthogonal tensor). If

$$\mathcal{A}^{T_k} *_k \mathcal{A} = \mathcal{A} *_k \mathcal{A}^{T_k} = \mathcal{I}_k, \quad (11)$$

where the tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is mode- k orthogonal.

Definition 7 (Tensor mode- k permutation). For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\overline{\mathcal{A}}^k := \text{permute}(\mathcal{A}, k)$ is the mode- k permutation of \mathcal{A} , whose i_{th} mode-3 slice is the i_{th} mode- k slice of \mathcal{A} , and its inverse operation is $\mathcal{A} := \text{ipermute}(\overline{\mathcal{A}}^k, k)$.

Theorem 2 (Mode- k t-SVD). The factorization of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is

$$\mathcal{A} = \mathcal{U}_k *_k \mathcal{S}_k *_k \mathcal{V}_k^{T_k}, \quad k = 1, 2, \text{ and } 3, \quad (12)$$

where \mathcal{U}_k and \mathcal{V}_k are the mode- k orthogonal tensors and $\mathcal{S}_k \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the mode- k diagonal tensor.

Theorem 2 is proven in [38] (Th. 2).

Definition 8 (Tensor fibered rank). For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\text{rank}_f(\mathcal{A})$ is the fibered rank of \mathcal{A} , whose k_{th} element is the mode- k tensor fibered rank $\text{rank}_{f_k}(\mathcal{A})$. $\text{rank}_{f_k}(\mathcal{A}) = \max(\text{rank}_{m_k}(\mathcal{A}))$, where the i_{th} element of $\text{rank}_{m_k}(\mathcal{A})$ is the rank of the i_{th} mode- k slice of $\overline{\mathcal{A}}_k = \text{fft}(\mathcal{A}, k)$.

The mode- k t-SVD can be efficiently obtained by computing a series of matrix SVDs in the Fourier domain and achieves more flexible and accurate HSI low-rankness characterization (see Algorithm 2).

3. HSI Denoising Model and Its ADMM

In this section, we show our new denoising model and ADMM for solving the proposed model is also derived.

3.1. The Logarithmic Penalty Function. This article mainly proposes logarithmic penalty function [43] that serves as the model for the penalty function developed in the HSI denoising model below and is designed to have less bias. The logarithmic penalty is given by

$$\phi(x; a) = \frac{1}{a} \log(1 + a|x|), \quad (13)$$

where $a > 0$ controls the degree of nonconvexity of the penalty function. This function satisfies the following conditions:

- (A1) $\phi(x; a) \in \mathbb{C}^2(\mathbb{R}_+^*), \phi(x; a) \in C^0(\mathbb{R}_+), \mathbb{R}_+ := \{t \in \mathbb{R}: t \geq 0\}, \text{ and } \mathbb{R}_+^* := \{t \in \mathbb{R}: t > 0\}$
- (A2) $\phi'(x; a) > 0$ and $\forall t \in \mathbb{R}_+^*$
- (A3) $\phi''(x; a) \leq 0$ and $\forall t \in \mathbb{R}_+^*$
- (A4) $\sup_{t \in \mathbb{R}_+^*} \phi'(x; a) = 1$ and $\inf_{t \in \mathbb{R}_+^*} \phi''(x; a) = \phi''(0^+; a) = -a$

The proximity operator $\Theta: \mathbb{R} \rightarrow \mathbb{R}$ associated with the nonconvex function $\phi(x; a)$ is

$$\Theta(y; \bar{\lambda}, a) = \text{prox}_\phi(y; \bar{\lambda}, a)$$

$$= \arg \min_{x \in \mathbb{R}} \left\{ f(x) = \frac{1}{2}(y - x)^2 + \bar{\lambda}\phi(x; a) \right\}. \quad (14)$$

In [44–46], the authors prove that, for $0 < a \leq (1/\bar{\lambda})$, the function f in equation (14) is convex. Therefore, the proximity operator finds an optimal solution for convex minimization problem (14). The proximity operator associated with logarithmic penalty equation (13) is a continuous nonlinear threshold function with $\bar{\lambda}$ as the threshold value [44], namely,

$$\text{prox}_{\bar{\lambda}\phi}(y; \bar{\lambda}, a) = 0, \forall |y| < \bar{\lambda}, \quad (15)$$

and is given by

$$\text{prox}_{\bar{\lambda}\phi}(y; \bar{\lambda}, a) = \begin{cases} \left[\frac{|y|}{2} - \frac{1}{2a} + \sqrt{\left(\frac{|y|}{2} + \frac{1}{2a} \right)^2 - \frac{\bar{\lambda}}{a}} \right], & |y| > \bar{\lambda}, \\ 0, & |y| \leq \bar{\lambda}. \end{cases} \quad (16)$$

3.2. HSI Denoising Model. Zhang et al. proposed an HSI mixed noise removal model based on low tubal-rank tensor recovery (LRTR) [37]. It can address the mixed noise in HSIs and decompose a noisy HSI into three parts, i.e., a low-tubal-rank part (the clean HSI), a Gaussian noise part, and a sparse noise part. Zheng [38] proposed an HSI denoising model based on a low-fibered-rank prior, and it is formulated as

$$\begin{aligned} & \min_{\mathcal{A}, \mathcal{N}, \mathcal{S}} \text{rank}_f(\mathcal{A}) + \lambda_1 \|\mathcal{N}\|_F^2 + \lambda_2 \|\mathcal{S}\|_1 \\ & \text{s.t. } \mathcal{Y} = \mathcal{A} + \mathcal{N} + \mathcal{S}, \end{aligned} \quad (17)$$

where \mathcal{Y} is the observed noisy HSI, \mathcal{N} is Gaussian noise, \mathcal{S} is sparse noise, $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the clean hyperspectral signal, which has the low-fibered-rank tensor property, and λ_1 and λ_2 are regularization parameters.

The Gaussian noise model (large degree of freedom) corresponds to a dense noise type [47, 48]. \mathcal{N} should not be low in rank. Otherwise, \mathcal{A} cannot be recovered from random noise. Due to the stochastic nature of Gaussian noise, it is assumed that there is no correlation (or a weak correlation) among noise components. Thus, the rank of \mathcal{N} is normally full and much larger than the rank of \mathcal{A} . Therefore, the low-fibered-rank component \mathcal{A} can be recovered from equation (17) by properly choosing the tuning parameters λ_1 and λ_2 .

Directly minimizing the tensor fibered rank is NP-hard. Based on logarithmic penalty equation (13), we propose that the following 3DNLogATNN HSI denoising model is commonly formulated:

```

Input:  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ ,  $k$ 
 $\text{fft}(\mathcal{A}, , k) \longrightarrow \overline{\mathcal{A}}_k$ 
for  $i = 1$  to  $n_k$  do
     $[U, S, V] = \text{SVD}((\overline{\mathcal{A}}_k)_k^{(i)})$ ;  $U \longrightarrow (\overline{\mathcal{U}}_k)_k^{(i)}$ ;  $S \longrightarrow (\overline{\mathcal{S}}_k)_k^{(i)}$ ;  $V \longrightarrow (\overline{\mathcal{V}}_k)_k^{(i)}$ 
end for
 $\text{ifft}(\overline{\mathcal{U}}_k, , k) \longrightarrow \mathcal{U}_k$ ;  $\text{ifft}(\overline{\mathcal{S}}_k, , k) \longrightarrow \mathcal{S}_k$ ;  $\text{ifft}(\overline{\mathcal{V}}_k, , k) \longrightarrow \mathcal{V}_k$ 
output:  $\mathcal{U}_k, \mathcal{S}_k, \mathcal{V}_k$ 

```

ALGORITHM 2: Mode- k t -SVD for three-way tensors.

$$\begin{aligned} & \min_{\mathcal{A}, \mathcal{N}, \mathcal{S}} \sum_{k=1}^3 \tau_k \mathcal{F}_k(\mathcal{A}) + \lambda_1 \|\mathcal{N}\|_F^2 + \lambda_2 \|\mathcal{S}\|_1 \\ & \text{s.t. } \mathcal{Y} = \mathcal{A} + \mathcal{N} + \mathcal{S}, \end{aligned} \quad (18)$$

where $\tau_k \geq 0$ ($k = 1, 2$, and 3) and $\sum_{k=1}^3 \tau_k = 1$ is the weight of the fibered rank. $\mathcal{F}_k(\mathcal{A})$ are set as LogATNN $_k(\mathcal{A})$ in the HIS denoising model:

$$\begin{aligned} \text{LogATNN}_k(\mathcal{A}) &= \sum_{i=1}^{n_k} \sum_{j=1}^m \frac{1}{a} \log \left(1 + a \sigma_j \left((\overline{\mathcal{A}}_k)_k^{(i)} \right) \right), \\ n_k &= n_1, n_2, \text{ and } n_3, \end{aligned} \quad (19)$$

where m is the number of singular values of $\overline{\mathcal{A}}$, $\sigma_j((\overline{\mathcal{A}}_k)_k^{(i)})$ is the j th singular values of $(\overline{\mathcal{A}}_k)_k^{(i)}$, and $(\overline{\mathcal{A}}_k)_k^{(i)}$ is the i th mode- k slice of $\overline{\mathcal{A}} = \text{fft}(\mathcal{A}, , k)$.

3.3. ADMM for Solving Model (18). We use the ADMM to solve equation (18). By casting the three auxiliary variables \mathcal{X}_k ($k = 1, 2$, and 3), equation (18) can be rewritten as follows:

$$\begin{aligned} & \min_{\mathcal{A}, \mathcal{N}, \mathcal{S}} \sum_{k=1}^3 \tau_k \mathcal{F}_k(\mathcal{X}_k) + \lambda_1 \|\mathcal{N}\|_F^2 + \lambda_2 \|\mathcal{S}\|_1 \\ & \text{s.t. } \begin{cases} \mathcal{Y} - (\mathcal{A} + \mathcal{N} + \mathcal{S}) = 0 \\ \mathcal{A} - \mathcal{X}_k = 0 \\ k = 1, 2, \text{ and } 3. \end{cases} \end{aligned} \quad (20)$$

The augmented Lagrangian function of equation (18) is

$$\begin{aligned} L_{\zeta_k, \rho}(\mathcal{X}_k, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{W}, \mathcal{Q}) &= \sum_{k=1}^3 \left\{ \tau_k \mathcal{F}_k(\mathcal{X}_k) + \langle \mathcal{A} - \mathcal{X}_k, \mathcal{W}_k \rangle + \frac{\zeta_k}{2} \|\mathcal{A} - \mathcal{X}_k\|_F^2 \right\} + \lambda_1 \|\mathcal{N}\|_F^2 + \lambda_2 \|\mathcal{S}\|_1 \\ &+ \langle \mathcal{Y} - (\mathcal{A} + \mathcal{N} + \mathcal{S}), \mathcal{Q} \rangle + \frac{\rho}{2} \|\mathcal{Y} - (\mathcal{A} + \mathcal{N} + \mathcal{S})\|_F^2, \end{aligned} \quad (21)$$

where \mathcal{W}_k and \mathcal{Q} are the Lagrange multipliers and ζ_k ($k = 1, 2, 3$) and ρ are positive scalars. Now we can solve the problem within the ADMM framework.

With the other parameters fixed, \mathcal{X} can be updated by solving

$$\min_{\mathcal{X}} L_{\zeta_k, \rho}(\mathcal{X}_k, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{W}, \mathcal{Q}) = \min_{\mathcal{X}} \sum_{k=1}^3 \left\{ \tau_k \mathcal{F}_k(\mathcal{X}_k) + \langle \mathcal{A} - \mathcal{X}_k, \mathcal{W}_k \rangle + \frac{\zeta_k}{2} \|\mathcal{A} - \mathcal{X}_k\|_F^2 \right\}, \quad (22)$$

which is equivalent to the following subproblem for $\mathcal{X}_k \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ ($k = 1, 2$, and 3):

$$\mathcal{X}_k^{l+1} = \arg \min_{\mathcal{X}_k} \tau_k \mathcal{F}_k(\mathcal{X}_k) + \frac{\zeta_k}{2} \left\| \mathcal{A} - \mathcal{X}_k + \frac{\mathcal{W}_k^l}{\zeta_k} \right\|_F^2. \quad (23)$$

To solve equation (23), we can rewrite it as

$$\arg \min_{\mathcal{X}} \bar{\lambda} \log \text{ATNN}_k(\mathcal{X}) + \frac{1}{2} \|\mathcal{X} - \mathcal{M}\|_F^2, \quad (24)$$

where $\mathcal{M} = \mathcal{A} + (\mathcal{W}_k / \zeta_k) = \mathcal{U} * \mathcal{S} * \mathcal{V}^{T_k}$, $\overline{\mathcal{S}}_k = \text{fft}(\mathcal{S}, , k)$, and $\bar{\lambda} = \tau_k / \zeta_k$. From equation (16),

$$\widehat{S}_k = \theta(\overline{\mathcal{S}}_k(i, j, r)) = \begin{cases} \left[\frac{|\overline{\mathcal{S}}_k(i, j, r)|}{2} - \frac{1}{2a} + \sqrt{\left(\frac{|\overline{\mathcal{S}}_k(i, j, r)|}{2} + \frac{1}{2a} \right)^2 - \frac{\bar{\lambda}}{a}} \right] \text{sign}(\overline{\mathcal{S}}_k(i, j, r)), & |\overline{\mathcal{S}}_k(i, j, r)| \geq \bar{\lambda}, \\ 0, & |\overline{\mathcal{S}}_k(i, j, r)| < \bar{\lambda}, \end{cases} \quad (25)$$

where $\tilde{\mathcal{S}}_k = \text{ifft}(\widehat{S}_k, k)$ and $\mathcal{X}^l = \mathcal{D}(\mathcal{M}, k) = \mathcal{U} *_{\mathcal{k}} \tilde{\mathcal{S}} *_{\mathcal{k}}$. $\mathcal{V}^{\top k}$, $\bar{\lambda} > 0$ and $0 < a \leq (1/\bar{\lambda})$. Therefore,

$$\mathcal{X}_k^{l+1} = \mathcal{D}\left(\mathcal{A}^l + \frac{\mathcal{W}_k^l}{\zeta_k}, k\right). \quad (26)$$

Though model (20) is based on a nonconvex penalty function, the parameters of which are constrained to ensure the convexity of the subproblem \mathcal{X}_k .

With other parameters fixed, \mathcal{A} can be updated by solving

$$\arg \min_{\mathcal{Y}} L_{\zeta_k, \rho}(\mathcal{X}_k, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{W}, \mathcal{Q}) = \sum_{i=1}^3 \left\{ \langle \mathcal{A} - \mathcal{X}_k, \mathcal{W}_k \rangle + \frac{\zeta_k}{2} \|\mathcal{A} - \mathcal{X}_k\|_F^2 \right\} + \langle \mathcal{Y} - (\mathcal{A} + \mathcal{N} + \mathcal{S}), \mathcal{Q} \rangle + \frac{\rho}{2} \|\mathcal{Y} - (\mathcal{A} + \mathcal{N} + \mathcal{S})\|_F^2, \quad (27)$$

which is equivalent to the following subproblem:

$$\mathcal{A}^{l+1} = \arg \min_{\mathcal{Y}} \sum_{k=1}^3 \frac{\zeta_k}{2} \left\| \mathcal{A} - \mathcal{X}_k^{l+1} + \frac{\mathcal{W}_k^l}{\zeta_k} \right\|_F^2 + \frac{\rho}{2} \left\| \mathcal{Y} - (\mathcal{A} + \mathcal{N} + \mathcal{S}) + \frac{\mathcal{Q}^l}{\rho} \right\|_F^2. \quad (28)$$

It has the following closed-form solution:

$$\mathcal{A}^{l+1} = \frac{\sum_{k=1}^3 \zeta_k \left(\mathcal{X}_k^{l+1} - \left(\mathcal{W}_k^l / \zeta_k \right) \right) + \rho \left(\mathcal{Y} - \mathcal{N}^l - \mathcal{S}^l + \left(\mathcal{Q}^l / \rho \right) \right)}{\sum_{k=1}^3 \zeta_k + \rho}. \quad (29)$$

With the other parameters fixed, \mathcal{N} can be updated by solving

$$\begin{aligned} \arg \min_{\mathcal{N}} L_{\zeta_k, \rho}(\mathcal{X}_k, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{W}, \mathcal{Q}) &= \lambda_1 \|\mathcal{N}\|_F^2 + \langle \mathcal{Y} - (\mathcal{A} + \mathcal{N} + \mathcal{S}), \mathcal{Q} \rangle \\ &\quad + \frac{\rho}{2} \|\mathcal{Y} - (\mathcal{A} + \mathcal{N} + \mathcal{S})\|_F^2, \end{aligned} \quad (30)$$

which is equivalent to the following subproblem:

$$\mathcal{N}^{l+1} = \arg \min_{\mathcal{N}} \lambda_1 \|\mathcal{N}\|_F^2 + \frac{\rho}{2} \left\| \mathcal{Y} - \left(\mathcal{A}^{l+1} + \mathcal{N}^l + \mathcal{S}^l \right) + \frac{\mathcal{Q}^l}{\rho} \right\|_F^2, \quad (31)$$

and it has the following closed-form solution:

$$\mathcal{N}^{l+1} = \frac{\rho \left(\mathcal{Y} - \mathcal{A}^{l+1} - \mathcal{S}^l + \left(\mathcal{Q}^l / \rho \right) \right)}{2\lambda_1 + \rho}. \quad (32)$$

With the other parameters fixed, \mathcal{S} can be updated by solving

$$\begin{aligned} \arg \min_{\mathcal{S}} L_{\zeta_k, \rho}(\mathcal{X}_k, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{M}, \mathcal{Q}) &= \lambda_2 \|\mathcal{S}\|_1 + \langle \mathcal{Y} - (\mathcal{A} + \mathcal{N} + \mathcal{S}), \mathcal{Q} \rangle + \frac{\rho}{2} \|\mathcal{Y} - (\mathcal{A} + \mathcal{N} + \mathcal{S})\|_F^2, \end{aligned} \quad (33)$$

which is equivalent to the following subproblem:

$$\mathcal{S}^{l+1} = \arg \min_{\mathcal{S}} \lambda_2 \|\mathcal{S}\|_1 + \frac{\rho}{2} \left\| \mathcal{Y} - \left(\mathcal{A}^{l+1} + \mathcal{N}^{l+1} + \mathcal{S} \right) + \frac{\mathcal{Q}^l}{\rho} \right\|_F^2. \quad (34)$$

Using the tensor soft thresholding operator, the following solution can be obtained [38]:

$$S^{l+1} = \text{sign} \left(\mathcal{Y} - \mathcal{A}^{l+1} - \mathcal{N}^{l+1} + \frac{\mathcal{Q}^l}{\rho} \right) \max \left(\left| \mathcal{Y} - \mathcal{A}^{l+1} - \mathcal{N}^{l+1} + \frac{\mathcal{Q}^l}{\rho} \right| - \frac{\lambda_2}{\rho}, 0 \right). \quad (35)$$

The Lagrange multipliers \mathcal{W}_k and \mathcal{Q} can be updated by solving

$$\mathcal{W}_k^{l+1} = \mathcal{W}_k^l + \zeta_k \left(\mathcal{A}^{l+1} - \mathcal{X}_k^{l+1} \right), \quad (36)$$

$$\mathcal{Q}^{l+1} = \mathcal{Q}^l + \rho \left(\mathcal{Y} - \mathcal{A}^{l+1} - \mathcal{N}^{l+1} - \mathcal{S}^{l+1} \right). \quad (37)$$

Hence, the proposed algorithm for HSI denoising is summarized in Algorithm 3.

4. Experiment Results

To validate the effectiveness of the proposed method for HSI denoising, we perform experiments on simulated data and compare the experimental results both quantitatively and visually. The Washington DC Mall data, Pavia City Center data, and the Indian Pines data are used. In our experiments, the Washington DC Mall data uses only a subimage (191 bands and size of each band is 256×256). The Pavia City Center data uses only a subimage (80 bands and size of each band is 200×200). And, the synthetic data by Zhang et al. [37] was generated using the ground truth of the Indian Pines dataset; the size of the synthetic HSI was $145 \times 145 \times 224$. In addition, to facilitate the numerical calculation and visualization, all the bands of the HSI are normalized into $[0, 1]$, and they will be stretched to the original level after restoration.

The three evaluation measures are the mean peak signal-to-noise ratio (MPSNR), mean structure similarity (MSSIM), and spectral angle mapping (SAM). The three metrics are defined as follows to measure the quality of the denoised result:

$$\begin{aligned} \text{PSNR} &= 10 * \log_{10} \frac{L^2 MN}{\sum_{x=1}^M \sum_{y=1}^N [\hat{I}(x, y) - I(x, y)]^2}, \\ \text{MPSNR} &= \frac{1}{B} \sum_{i=1}^B \text{PSNR}_i, \end{aligned} \quad (38)$$

where $M \times N$ represents the image size in the space, $\hat{I}(x, y)$ represents the original image, $I(x, y)$ represents the distortion image, L is a pixel that can achieve the maximum value, and B is the number of PSNR:

$$\text{SSIM} = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \quad (39)$$

$$\text{MSSIM}(X, Y) = \frac{1}{Q} \sum_{j=1}^Q \text{SSIM}(x_j, y_j),$$

where C_1 is a constant for $\mu_x^2 + \mu_y^2$, C_2 is the same as C_1 , σ_x and σ_y represent the x and y standard deviations, respectively, X and Y represent the original image and the distorted image, respectively, x_j and y_j represent the j th local window contents, and Q is the number of local windows:

$$\text{SAM}(\nu, \nu') = \cos^{-1} \left(\frac{\sum_{i=1}^Z \nu_i \nu'_i}{\sqrt{\sum_{i=1}^Z \nu_i^2} \sqrt{\sum_{i=1}^Z \nu'^2_i}} \right), \quad (40)$$

where Z is the number of wavelengths, ν and ν' represent the spectrum vectors, and

$$\begin{aligned} \nu &= \{\nu_1, \nu_2, \dots, \nu_n\}, \\ \nu' &= \{\nu'_1, \nu'_2, \dots, \nu'_n\}. \end{aligned} \quad (41)$$

The PSNR and structural similarity index measure (SSIM) are two conventional perceptual quality indexes (PQIs) in image processing and computer vision. They evaluate the similarities between the target image and the reference image based on the mean square error (MSE) and structural consistency. The larger these two measures are, the closer the target HSI is to the reference HSI. The SAM is a physically based spectral classification method that uses an N -dimensional angle to match pixels to reference spectra. Different from the former two measures, the smaller the SAM is, the more similar the target HSI is to the reference HSI.

Real HSIs usually include several different types of noise. To simulate real-noise scenarios, we consider the variance of the Gaussian noise β and the variance of the impulse noise δ . We use statistical structures to simulate different types of noise, including independent and identically distributed (i.i.d.) and non-i.i.d. noise, which are elaborated as follows:

- (1) Case 1 (non-i.i.d. Gaussian noise): all the bands of the test dataset are contaminated by zero-mean Gaussian noise with different intensities. The variance in the Gaussian noise β is randomly changed from $U(0.1, 0.2)$ and $U(0.55, 0.65)$.
- (2) Case 2 (non-i.i.d. impulse noise): in this case, all bands are contaminated by impulse noise with different ratios. The ratios of impulse noise δ are randomly changed from $U(0.35, 0.45)$, $U(0.45, 0.55)$, and $U(0.55, 0.65)$.
- (3) Case 3 (i.i.d. Gaussian + i.i.d. impulse noise): in this case, all bands are corrupted by zero-mean Gaussian noise and impulse noise. The variance in the Gaussian noise β is 0.3, and the ratio of the impulse noise δ is 0.1; the variance in the Gaussian noise β is 0.1, and the ratio of the impulse noise δ is 0.4; the variance in the Gaussian noise β is 0.3, and the ratio of the impulse noise δ is 0.5.
- (4) Case 4 (non-i.i.d. Gaussian + i.i.d. impulse noise): in this case, all bands are corrupted by zero-mean Gaussian noise and impulse noise with different intensities. The variance in the Gaussian noise β is randomly changed from $U(0.3, 0.4)$, and the ratio of the impulse noise δ is 0.2; the variance in the Gaussian noise β is randomly changed from $U(0.2, 0.3)$, and the ratio of impulse noise δ is 0.3; the variance in the Gaussian noise β is randomly changed from $U(0.4, 0.5)$, and the ratio of impulse noise δ is 0.1.
- (5) Case 5 (i.i.d. Gaussian + non-i.i.d. impulse noise): in this case, all bands are corrupted by zero-mean Gaussian noise and impulse noise with different intensities. The variance in the Gaussian noise β is 0.1, and the ratios of impulse noise δ are randomly

```

Input: The noisy HSI  $\mathcal{Y}$ , parameters  $\tau_k, \zeta_k$  ( $k = 1, 2$ , and  $3$ ),  $\lambda_1, \lambda_2, \rho, t = 1.2, l = 0$ .  

 $\mathcal{X}^0 = 0, \mathcal{N}^0 = 0, \mathcal{S}^0 = 0, \mathcal{X}_k^0 = 0, \mathcal{W}_k^0 = 0$ , and  $\mathcal{Q}^0 = 0$ .  

while not converged do  

    Update  $\mathcal{X}_k^{l+1}$  with equation (26),  $k = 1, 2$ , and  $3$ .  

    Update  $\mathcal{A}_k^{l+1}$  with equation (29).  

    Update  $\mathcal{N}^{l+1}$  with equation (32).  

    Update  $\mathcal{S}^{l+1}$  with equation (35).  

    Update  $\mathcal{W}_k^{l+1}$  with equation (36),  $k = 1, 2$ , and  $3$ .  

    Update  $\mathcal{Q}^{l+1}$  with equation (37).  

    Let  $\zeta_k = t\zeta_k, k = 1, 2$ , and  $3$ ;  $\rho = t\rho$ ; and  $l = l + 1$ .  

    Check the convergence condition:  

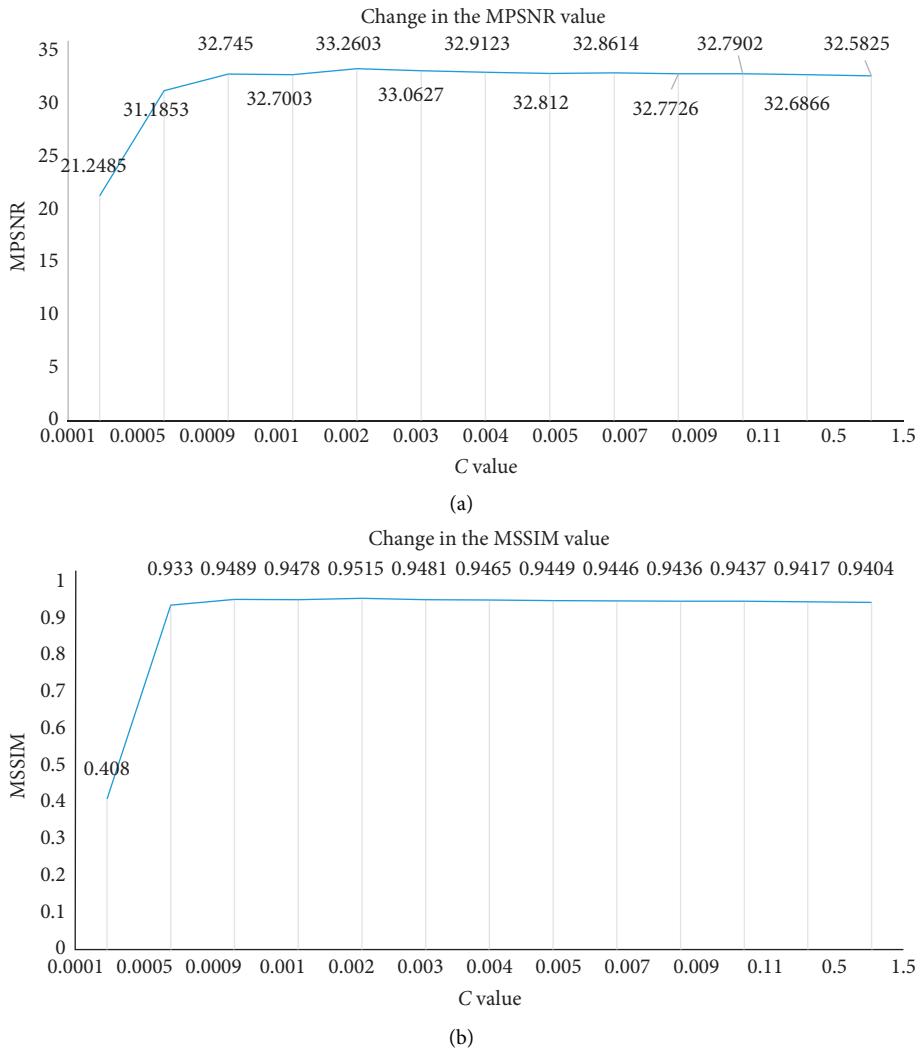
     $(\|\mathcal{A}^{l+1} - \mathcal{A}^l\|_F / \|\mathcal{A}^l\|_F) < 10^{-4}$ .  

end while  

Output: Denoised HIS  $\mathcal{X}$ .

```

ALGORITHM 3: Algorithm for HSI denoising with the ADMM framework.

FIGURE 5: Sensitivity analysis of parameter λ_1 for 3DLogATNN. (a) Change in the MPSNR value. (b) Change in the MSSIM value.

varied from $U(0.5, 0.6)$; the variance in the Gaussian noise β is 0.3, and the ratios of impulse noise δ are randomly varied from $U(0.3, 0.4)$; the variance in

the Gaussian noise β is 0.2, and the ratios of impulse noise δ are randomly varied from $U(0.4, 0.5)$.

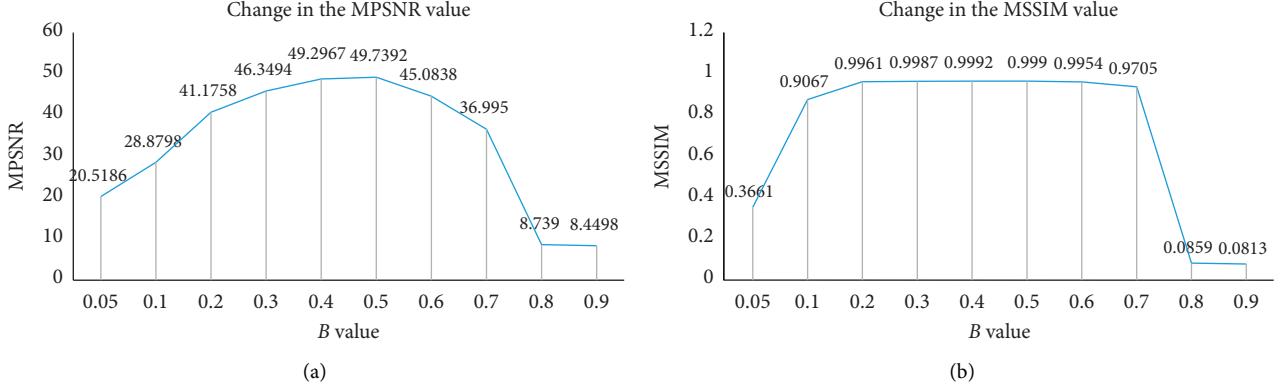


FIGURE 6: Sensitivity analysis of parameter λ_2 for 3DLogATNN. (a) Change in the MPSNR value. (b) Change in the MSSIM value.

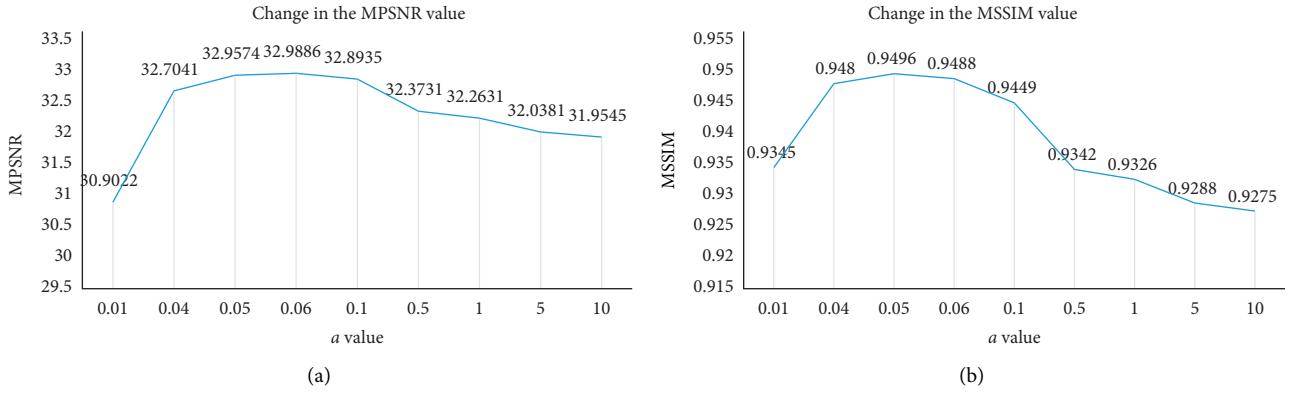


FIGURE 7: Sensitivity analysis of parameter a for 3DLogATNN. (a) Change in the MPSNR value. (b) Change in the MSSIM value.

TABLE 1: Parameter settings in the HSI denoising methods for simulated data.

Method	Data	τ_k	ξ	λ_1	λ_2	ε	a
3DTNN	Washington DC Mall						
	Pavia City Center	(1, 1, 0.001)/2.001	$(10^2, 10^2, 10^2)$	$(0.002/\beta)$	λ	—	—
	Indian Pines						
3DLogTNN	Washington DC Mall					50	
	Pavia City Center	(1, 1, 0.001)/2.001	$(10^5, 10^5, 10^5)$	$(0.005/\beta)$	0.011λ	80	—
	Indian Pines					50	
3DLogATNN	Washington DC Mall						
	Pavia City Center	(1, 1, 0.001)/2.001	$(10^3, 10^3, 10^3)$	$(0.002/\beta)$	0.5λ	—	0.05
	Indian Pines						

- (6) Case 6 (non-i.i.d. Gaussian + non-i.i.d. impulse noise): in this case, all bands are corrupted by zero-mean Gaussian noise and impulse noise with different intensities. The variance in the Gaussian noise β is randomly changed from $U(0.2, 0.3)$, $U(0.1, 0.2)$, and $U(0.4, 0.5)$, and the ratios of impulse noise δ are randomly varied from $U(0.2, 0.3)$, $U(0.3, 0.4)$, $U(0.1, 0.2)$, and $U(0.4, 0.5)$.

Parameter setting: we analyze the parameters involved in the proposed method on HSIs' Washington DC Mall, Pavia City Center, and Indian Pines, i.e., the weight τ_k , the regularization parameters λ_1 and λ_2 , the threshold parameter $\varsigma_k = (\tau_k/\xi)$, the penalty parameter

$\rho = (1/\text{mean}(\xi))$, and a constant a in 3DLogATNN. In all the following experiments, the parameters in these compared methods were manually adjusted according to their default strategies.

The regularization parameter λ_1 for 3DLogATNN: it is easy to see that λ_1 is the parameter used to restrict the sparsity of the Gaussian noise. We set $\lambda_1 = (C/\beta)$, where β is the standard deviation of Gaussian noise and C is a tuning parameter. The results were based on the simulated data experiment in case 1-1. Figure 5 shows the restoration results as C varied in the set $\{0.0001, 0.0005, 0.0009, 0.001, 0.002, 0.003, 0.004, 0.005, 0.007, 0.009, 0.11, 0.5, 0.15\}$. It can be clearly observed from this figure that the results of the 3DLogATNN solver are relatively stable in terms of both MPSNR and

TABLE 2: Performance comparison of competing methods. The best result in each case is highlighted in bold.

Case	Data	Gaussian noise	Impulse noise	Indicators	Noise	3DTNN	3DLogTNN	3DLogATNN
Case 1-1	Washington DC Mall	$\beta \in U(0.1, 0.2)$	—	MPSNR	16.5472	30.7752	32.1420	32.5883
				MSSIM	0.3633	0.9387	0.9449	0.9508
				SAM	36.3941	5.5380	5.8969	6.7688
				Time (s)	119.688	310.708	207.411	
Case 1-2	Washington DC Mall	$\beta \in U(0.55, 0.65)$	—	MPSNR	4.3820	22.9905	18.1073	24.9584
				MSSIM	0.0399	0.7128	0.3943	0.7436
				SAM	67.7679	9.8912	31.6428	11.0287
				Time (s)	241.556	371.838	202.24	
Cases 1-3	Pavia City Center	$\beta \in U(0.1, 0.2)$	—	MPSNR	16.7095	30.3380	31.5492	32.9023
				MSSIM	0.2742	0.9037	0.8933	0.9288
				SAM	34.6329	3.9214	5.7779	3.6840
				Time (s)	41.121	117.2938	83.271	
Cases 1-4	Pavia City Center	$\beta \in U(0.55, 0.65)$	—	MPSNR	4.4469	20.8191	10.4980	24.9550
				MSSIM	0.0219	0.3613	0.0785	0.6598
				SAM	68.2502	19.9347	52.9761	6.1381
				Time (s)	112.223	120.206	71.473	
Cases 1-5	Indian Pines	$\beta \in U(0.1, 0.2)$	—	MPSNR	16.6719	29.2903	30.9744	31.7479
				MSSIM	0.2705	0.9046	0.8793	0.8438
				SAM	16.9301	3.0329	2.5550	2.4113
				Time (s)	90.68	226.048	163.207	
Cases 1-6	Indian Pines	$\beta \in U(0.55, 0.65)$	—	MPSNR	4.4358	19.7342	7.7503	23.2606
				MSSIM	0.0427	0.7354	0.0734	0.6672
				SAM	49.1909	6.6535	38.7423	5.9968
				Time (s)	120.883	232.497	137.487	
Case 2-1	Washington DC Mall	—	$\delta \in U(0.35, 0.45)$	MPSNR	8.4747	33.5183	42.1350	49.3435
				MSSIM	0.0822	0.9474	0.9965	0.9992
				SAM	48.8256	7.4283	1.3193	0.7387
				Time (s)	235.137	332.130	227.661	
Case 2-2	Washington DC Mall	—	$\delta \in U(0.45, 0.55)$	MPSNR	7.5305	19.5300	33.5233	45.1362
				MSSIM	0.0576	0.4601	0.8926	0.9981
				SAM	50.0452	25.8593	14.2450	1.0228
				Time (s)	263.654	376.93	292.934	
Case 2-3	Washington DC Mall	—	$\delta \in U(0.55, 0.65)$	MPSNR	7.4583	18.8927	24.8326	42.2501
				MSSIM	0.0565	0.4286	0.6793	0.9964
				SAM	50.1369	27.0410	22.7390	1.3989
				Time (s)	239.041	337.120	224.460	
Cases 2-4	Pavia City Center	—	$\delta \in U(0.35, 0.45)$	MPSNR	8.8542	41.2600	45.9213	53.6297
				MSSIM	0.0498	0.9956	0.9980	0.9995
				SAM	45.6300	0.3693	0.3204	0.1519
				Time (s)	91.438	140.218	112.862	
Cases 2-5	Pavia City Center	—	$\delta \in U(0.55, 0.65)$	MPSNR	7.0649	18.4363	7.2921	39.0625
				MSSIM	0.0232	0.2896	0.0251	0.9888
				SAM	46.8397	26.4785	46.6534	1.1001
				Time (s)	111.606	76.873	112.652	
Cases 2-6	Indian Pines	—	$\delta \in U(0.35, 0.45)$	MPSNR	9.2778	39.0637	37.3473	38.3288
				MSSIM	0.0774	0.9730	0.9763	0.9784
				SAM	34.4106	1.0303	0.8421	0.7242
				Time (s)	149.143	213.263	150.303	
Cases 2-7	Indian Pines	—	$\delta \in U(0.55, 0.65)$	MPSNR	7.4881	14.1189	13.5063	36.9272
				MSSIM	0.0402	0.1636	0.1542	0.9704
				SAM	40.3902	21.6086	22.9239	0.8861
				Time (s)	150.468	229.334	165.124	
Cases 3-1	Washington DC Mall	$\beta = 0.3$	$\delta = 0.1$	MPSNR	9.0000	25.5926	27.8435	27.3793
				MSSIM	0.1087	0.8129	0.8624	0.8470
				SAM	53.8905	8.5845	7.8496	8.5841
				Time (s)	137.651	317.362	198.887	

TABLE 2: Continued.

Case	Data	Gaussian noise	Impulse noise	Indicators	Noise	3DTNN	3DLogTNN	3DLogATNN
Case 3-2	Washington DC Mall	$\beta = 0.1$	$\delta = 0.4$	MPSNR	8.1591	24.5273	30.4000	30.5960
				MSSIM	0.0751	0.7226	0.9213	0.9226
				SAM	49.8593	12.8686	6.3862	6.4664
				Time (s)		202.923	332.368	162.639
Case 3-3	Washington DC Mall	$\beta = 0.3$	$\delta = 0.5$	MPSNR	5.7009	16.3370	14.9055	20.9115
				MSSIM	0.0359	0.2983	0.2406	0.6499
				SAM	55.9567	21.9255	28.7635	13.2614
				Time (s)		220.647	367.041	200.073
Case 3-4	Pavia City Center	$\beta = 0.3$	$\delta = 0.1$	MPSNR	9.0974	25.7997	13.1319	26.8088
				MSSIM	0.0642	0.7448	0.1327	0.8029
				SAM	53.2858	5.4009	42.7153	5.6316
				Time (s)		51.955	93.249	65.679
Cases 3-5	Pavia City Center	$\beta = 0.1$	$\delta = 0.4$	MPSNR	8.4711	28.4963	30.2146	31.0972
				MSSIM	0.0443	0.8615	0.8613	0.9098
				SAM	46.8083	4.2606	6.4614	3.9908
				Time (s)		63.319	116.127	63.407
Cases 3-6	Indian Pines	$\beta = 0.1$	$\delta = 0.4$	MPSNR	8.9083	25.9148	30.1284	30.7026
				MSSIM	0.0719	0.6817	0.8644	0.8164
				SAM	35.5125	5.2059	2.8185	2.7058
				Time (s)		134.676	234.822	165.194
Cases 3-7	Indian Pines	$\beta = 0.3$	$\delta = 0.1$	MPSNR	9.2176	23.6223	26.8626	27.1730
				MSSIM	0.0971	0.8105	0.7807	0.7415
				SAM	34.2324	5.5598	4.0714	3.9805
				Time (s)		102.124	224.456	150.277
Cases 4-1	Washington DC Mall	$\beta \in U(0.3, 0.4)$	$\delta = 0.2$	MPSNR	7.1533	22.7893	24.8984	25.2959
				MSSIM	0.0702	0.7124	0.8004	0.8008
				SAM	57.1506	11.2378	9.4788	9.3893
				Time (s)		220.148	370.379	226.554
Cases 4-2	Washington DC Mall	$\beta \in U(0.2, 0.3)$	$\delta = 0.3$	MPSNR	7.6829	22.1301	25.1477	25.7826
				MSSIM	0.0737	0.6854	0.8146	0.8178
				SAM	53.4819	12.5629	9.5805	9.3802
				Time (s)		200.652	338.836	165.677
Case 4-3	Washington DC Mall	$\beta \in U(0.4, 0.5)$	$\delta = 0.1$	MPSNR	6.2377	23.7276	9.6080	25.6329
				MSSIM	0.0602	0.7283	0.1128	0.8029
				SAM	61.5814	10.0163	53.2827	9.1467
				Time (s)		190.534	323.047	172.353
Case 4-4	Pavia City Center	$\beta \in U(0.2, 0.3)$	$\delta = 0.3$	MPSNR	7.8602	23.5873	24.6273	25.6363
				MSSIM	0.0421	0.5911	0.6943	0.7801
				SAM	51.6427	9.9142	6.0669	5.9457
				Time (s)		113.882	150.160	91.212
Case 4-5	Pavia City Center	$\beta \in U(0.3, 0.4)$	$\delta = 0.2$	MPSNR	7.2535	23.5193	24.3368	25.1019
				MSSIM	0.0395	0.5704	0.6672	0.7372
				SAM	56.2301	10.1307	6.0174	5.9217
				Time (s)		92.428	139.006	87.948
Cases 4-6	Indian Pines	$\beta \in U(0.2, 0.3)$	$\delta = 0.3$	MPSNR	8.1970	22.1915	26.0036	26.5240
				MSSIM	0.0698	0.7802	0.7771	0.7409
				SAM	37.6925	7.0587	4.5808	4.3126
				Time (s)		117.283	230.989	152.424
Cases 4-7	Indian Pines	$\beta \in U(0.4, 0.5)$	$\delta = 0.1$	MPSNR	6.3572	20.8380	15.0486	23.8612
				MSSIM	0.0578	0.7511	0.2008	0.6976
				SAM	43.2485	7.0333	19.4183	5.5472
				Time (s)		106.580	226.007	133.670
Cases 5-1	Washington DC Mall	$\beta = 0.1$	$\delta \in U(0.5, 0.6)$	MPSNR	6.8766	16.5893	10.6705	28.4585
				MSSIM	0.0447	0.2855	0.1130	0.8819
				SAM	51.2579	27.3488	43.5216	7.8981
				Time (s)		213.342	317.644	172.836

TABLE 2: Continued.

Case	Data	Gaussian noise	Impulse noise	Indicators	Noise	3DTNN	3DLogTNN	3DLogATNN
Cases 5-2	Washington DC Mall	$\beta = 0.3$	$\delta \in U(0.3, 0.4)$	MPSNR	6.6987	19.9605	12.1413	23.0269
				MSSIM	0.0557	0.5540	0.1581	0.7539
				SAM	55.5938	15.5002	41.9549	11.3455
				Time (s)		200.605	323.034	166.227
Cases 5-3	Pavia City Center	$\beta = 0.1$	$\delta \in U(0.5, 0.6)$	MPSNR	7.2143	20.5012	26.6374	29.0107
				MSSIM	0.0262	0.3658	0.7900	0.8651
				SAM	47.5691	19.0650	5.4284	4.5068
				Time (s)		105.725	137.456	103.408
Case 5-4	Pavia City Center	$\beta = 0.3$	$\delta \in U(0.3, 0.4)$	MPSNR	6.9026	21.0954	22.9767	23.7700
				MSSIM	0.0317	0.4285	0.6131	0.6875
				SAM	53.5636	12.6393	6.3647	6.2442
				Time (s)		88.2	136.355	86.321
Case 5-5	Indian Pines	$\beta = 0.1$	$\delta \in U(0.5, 0.6)$	MPSNR	7.6214	17.6787	27.6585	28.9263
				MSSIM	0.0454	0.2527	0.7442	0.7765
				SAM	39.8736	14.3420	4.1642	3.2634
				Time (s)		143.077	243.403	159.432
Case 5-6	Indian Pines	$\beta = 0.2$	$\delta \in U(0.4, 0.5)$	MPSNR	7.6177	20.6385	26.2863	26.3867
				MSSIM	0.0528	0.5139	0.7733	0.7446
				SAM	39.7418	9.1172	4.4909	4.3622
				Time (s)		106.580	226.007	133.670
Cases 6-1	Washington DC Mall	$\beta \in U(0.2, 0.3)$	$\delta \in U(0.2, 0.3)$	MPSNR	8.1127	23.3022	26.3351	26.2174
				MSSIM	0.0845	0.7464	0.8378	0.8400
				SAM	53.2040	11.0123	8.8103	8.8638
				Time (s)		196.620	340.366	167.766
Cases 6-2	Washington DC Mall	$\beta \in U(0.1, 0.2)$	$\delta \in U(0.3, 0.4)$	MPSNR	8.2418	23.9231	27.9402	28.4975
				MSSIM	0.0806	0.7256	0.8802	0.8831
				SAM	50.5831	12.2193	7.8257	7.9288
				Time (s)		199.073	341.498	167.085
Cases 6-3	Washington DC Mall	$\beta \in U(0.4, 0.5)$	$\delta \in U(0.1, 0.2)$	MPSNR	5.9576	22.7822	9.1294	24.5838
				MSSIM	0.0544	0.6896	0.0988	0.7766
				SAM	61.4170	11.3025	53.7084	9.8922
				Time (s)		212.886	326.244	173.614
Cases 6-4	Pavia City Center	$\beta \in U(0.1, 0.2)$	$\delta \in U(0.4, 0.5)$	MPSNR	7.6849	23.1404	25.8334	27.7378
				MSSIM	0.0339	0.5315	0.7519	0.8332
				SAM	58.4695	12.9726	5.9927	5.2728
				Time (s)		90.948	136.677	89.459
Case 6-5	Pavia City Center	$\beta \in U(0.4, 0.5)$	$\delta \in U(0.1, 0.2)$	MPSNR	5.9498	22.3013	24.0139	24.3767
				MSSIM	0.0301	0.4513	0.6279	0.6621
				SAM	61.3247	13.4368	6.2548	6.2623
				Time (s)		94.386	139.482	85.760
Case 6-6	Indian Pines	$\beta \in U(0.1, 0.2)$	$\delta \in U(0.4, 0.5)$	MPSNR	8.0591	21.6393	27.9846	28.3006
				MSSIM	0.0573	0.4807	0.8167	0.7662
				SAM	38.3217	8.5698	3.6572	3.5710
				Time (s)		139.625	232.800	155.648
Case 6-7	Indian Pines	$\beta \in U(0.4, 0.5)$	$\delta \in U(0.1, 0.2)$	MPSNR	6.0436	20.2932	13.1701	23.4520
				MSSIM	0.0526	0.7367	0.1551	0.6912
				SAM	44.3514	7.7401	23.6309	5.9105
				Time (s)		113.345	236.529	137.036

MSSIM values, with the value of C changing from 0.002 to 0.003. Therefore, we suggest the use of $C = 0.002$ in all the simulated data experiments.

The regularization parameter λ_2 for 3DLogATNN: it is easy to see that λ_2 is the parameter used to restrict the sparsity of the impulse noise. We set $\lambda_2 = B\lambda$, where

$$\lambda = \frac{\tau_1}{\sqrt{\max(n_2, n_3)n_1}} + \frac{\tau_2}{\sqrt{\max(n_3, n_1)n_2}} + \frac{\tau_3}{\sqrt{\max(n_1, n_2)n_3}}, \quad (42)$$

and B is a tuning parameter. The results were based on the simulated data experiment in case 2-1. Figure 6 shows the

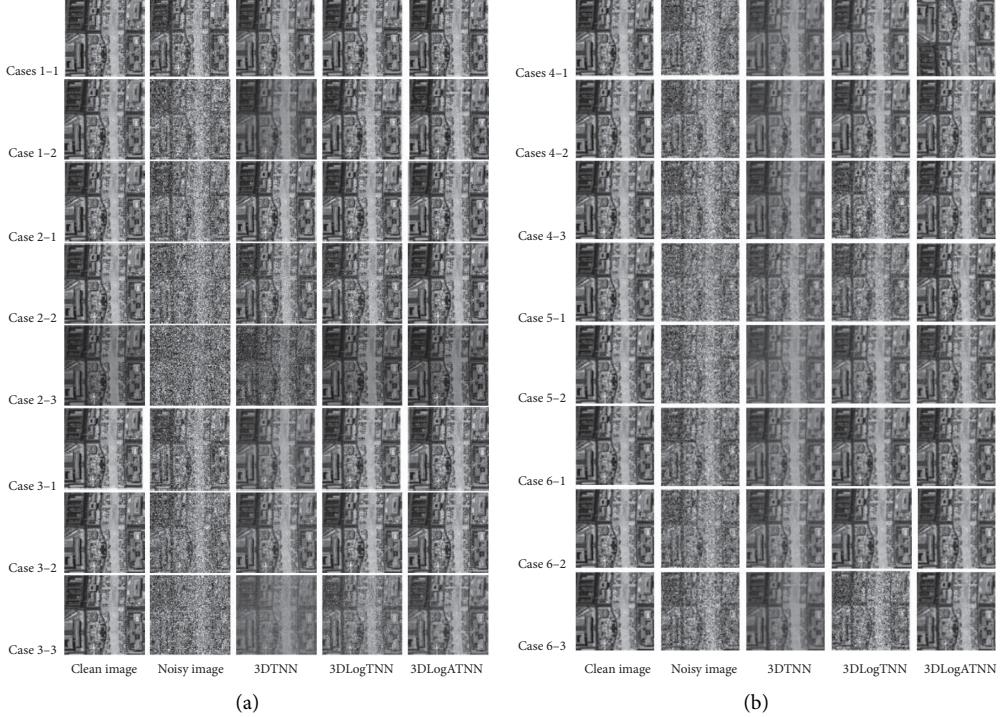


FIGURE 8: Denoising results of the Washington DC Mall dataset with mixture noise.

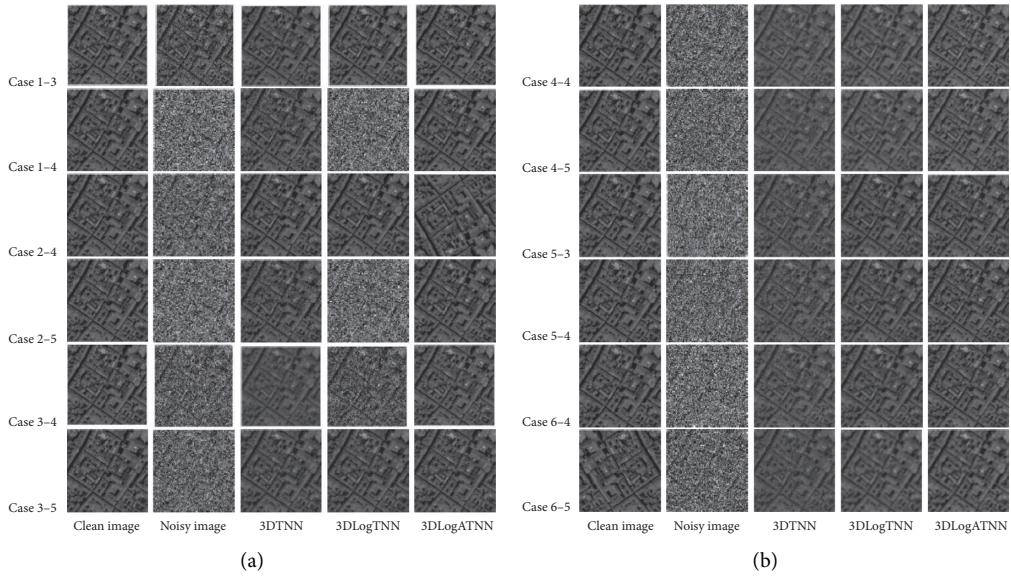


FIGURE 9: Denoising results of the Pavia City Center dataset with mixture noise.

restoration results as B varied in the set $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. It can be clearly observed from this figure that the results of the 3DLogATNN solver are relatively stable in terms of both MPSNR and MSSIM values, with the value of B changing from 0.4 to 0.5. Therefore, we suggest the use of $B = 0.5$ in all the simulated data experiments.

The constant a for 3DLogATNN: the parameter $a > 0$ controls the degree of nonconvexity of the penalty function. The results were based on the simulated data experiment in case 2-1. Figure 7 shows the restoration results as a varied in the set $\{0.01, 0.04, 0.05, 0.06, 0.1, 0.5, 1, 5, 10\}$. It can be clearly observed from this figure that the results of the 3DLogATNN solver are relatively stable in terms of both

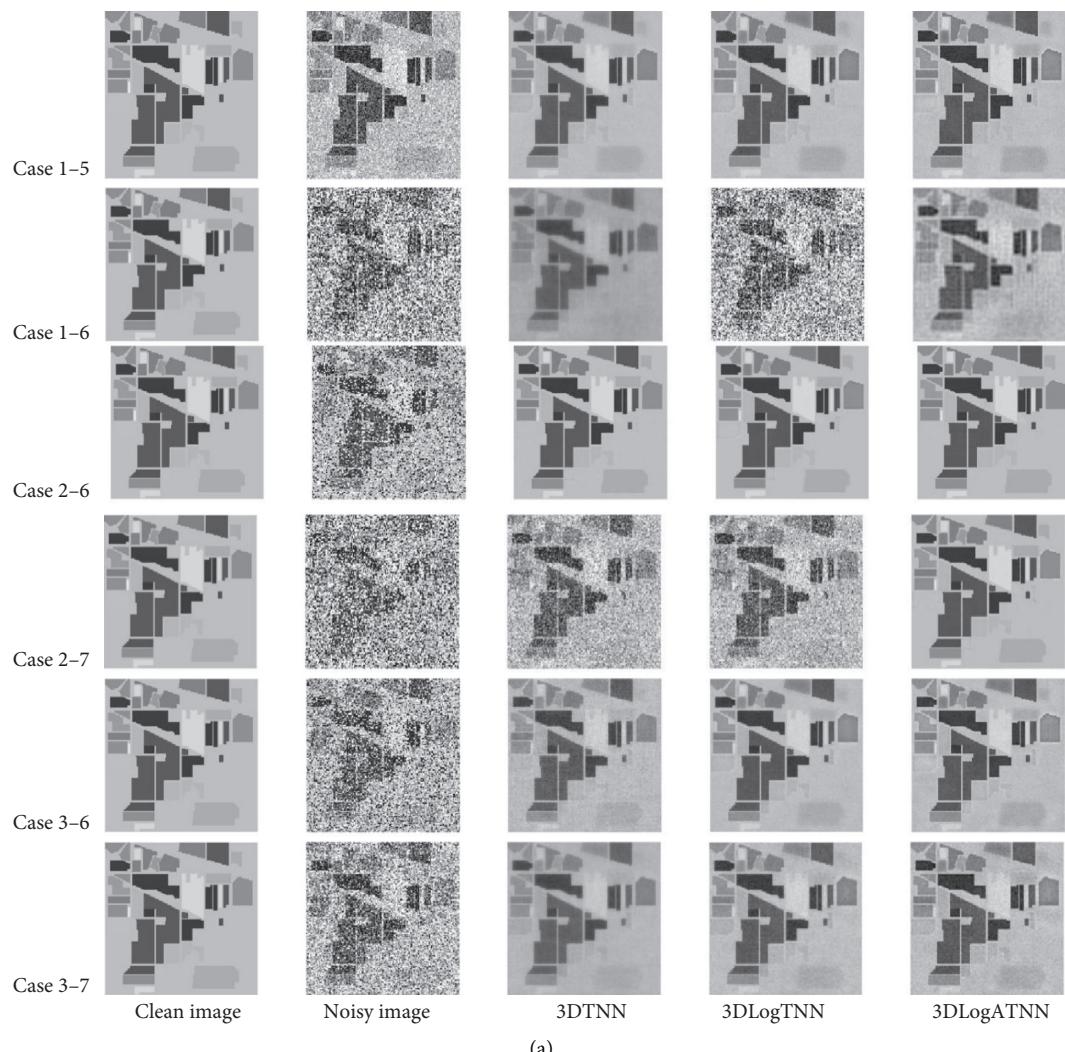


FIGURE 10: Continued.

(a)

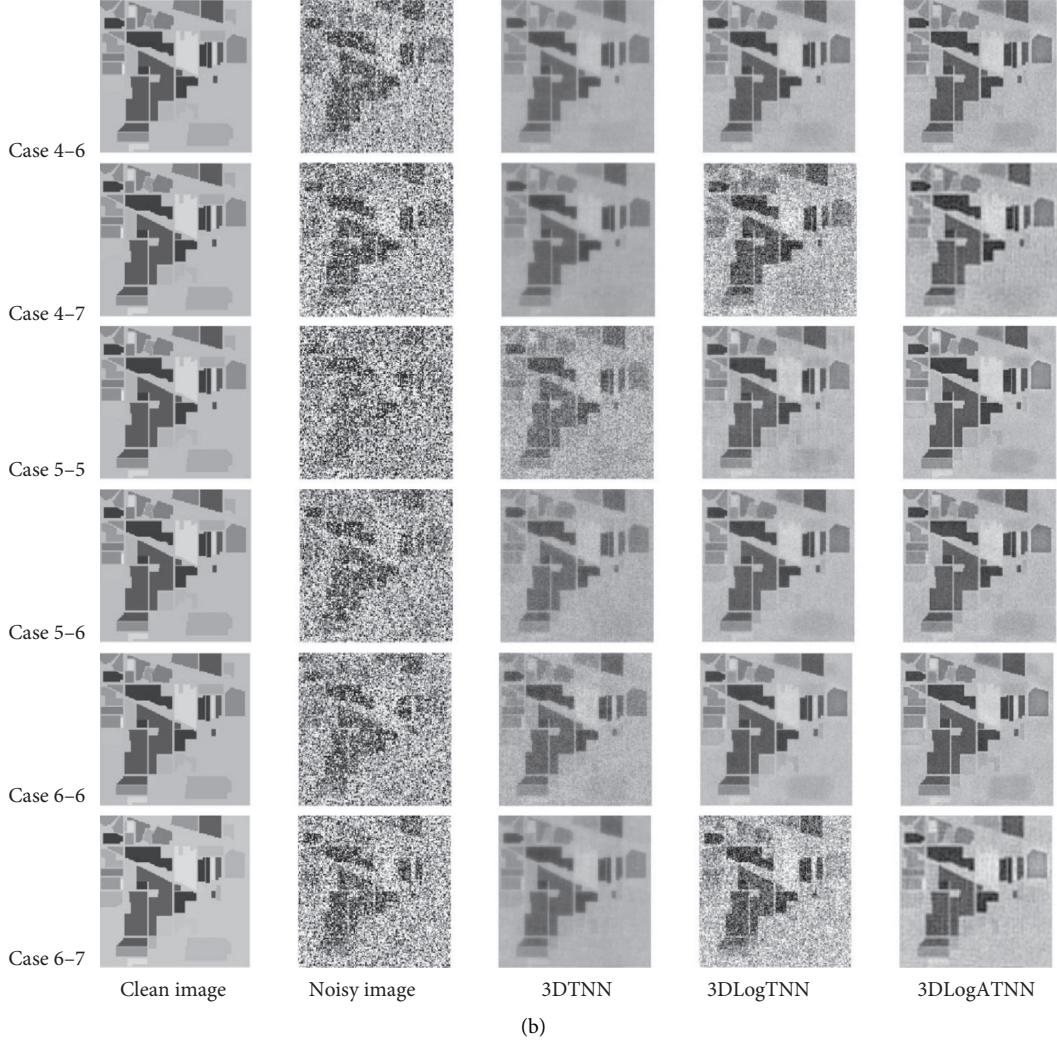


FIGURE 10: Denoising results of the Indian Pines dataset with mixture noise.

MPSNR and MSSIM values, with the value of α changing from 0.05 to 0.06. Therefore, we suggest the use of $\alpha = 0.05$ in all the simulated data experiments.

We adjust the parameters to achieve the best visual result, and the parameter setting is presented in Table 1.

Compared with the state-of-the-art methods, including TRPCA + BM4D [36, 49], LRMRR [37], LRTR [37], LRTDTV [50], and NMoG [51], on low-rank matrix/tensor approximation and noise modeling, the extensive experimental results demonstrate that the 3DTNN and 3DLogTNN [38] methods are better at removing the mixed noise. Therefore, the denoising results of the proposed method are quantitatively and visually compared with two state-of-the-art HSI denoising methods, i.e., 3DTNN and 3DLogTNN. The denoising results of all the methods in six cases are shown in Table 2. Three typical bands of the denoised HSIs in the mixture noise case obtained with different methods are shown in Figures 8–10. Figure 8 shows the denoising results at band 71 of the Washington DC Mall HSI, Figure 9 shows the denoising results at band 52 of the Pavia City Center HSI, and Figure 10 shows the denoising results at band 28 of the

Indian Pines HSI. It can be seen that the proposed 3DLogATNN can effectively remove the mixed noise and preserve the detailed information of the original image. The proposed method obtains the best visual quality by removing all the mixture noise and preserving the details well. Table 2 shows that the 3DLogATNN method converges faster than the 3DLogTNN-based method on all the Washington DC Mall, Pavia City Center data, and Indian Pines data, and our method outperforms the compared ones for the Pavia City Center data. Besides, 3DLogATNN is more stable than the other two algorithms, because it can get the best results in most cases.

5. Conclusion

In this paper, we present a new 3DLogATNN method for HSI denoising by mode- k t -SVD. The logarithmic penalty function is introduced in 3DLogATNN, which enables it to extract low-rank and sparse components more accurately from a degraded HSI contaminated by several types of noise. In addition, the ADMM-based algorithm is applied to effectively solve the proposed HSI denoising model. The

experiments have substantiated the superiority of the proposed method over state-of-the-art methods.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61967004, 11901137, 11961011, and 72061007), Guangxi Natural Science Foundation (2018GXNSFBA281023), China Postdoctoral Science Foundation (2020M682959), Guangxi Key Laboratory of Cryptography and Information Security (GCIS201621 and GCIS201927), Guangxi Key Laboratory of Automatic Detecting Technology and Instruments (YQ20113 and YQ20114), and Promotion Project of Basic Ability of Young and Middle-Aged Teachers in Universities of Guangxi under Grant (2019KY0253).

References

- [1] Y. Yuan, J. Lin, and Q. Wang, "Hyperspectral image classification via multitask joint sparse representation and stepwise MRF optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2966–2977, 2017.
- [2] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 3, pp. 809–823, 2012.
- [3] Y. Yuan, D. Ma, and Q. Wang, "Hyperspectral anomaly detection by graph pixel selection," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 3123–3134, 2016.
- [4] C. Chunzhi Li, X. Xiaohua Chen, and Y. Yunliang Jiang, "On diverse noises in hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 10, pp. 5388–5402, 2015.
- [5] R. Ji, Y. Gao, R. Hong, Q. Liu, D. Tao, and X. Li, "Spectral-spatial constraint hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 3, pp. 1811–1824, 2014.
- [6] R. W. Liu, L. Shi, W. Huang, J. Xu, S. C. H. Yu, and D. Wang, "Generalized total variation-based MRI rician denoising model with spatially adaptive regularization parameters," *Magnetic Resonance Imaging*, vol. 32, no. 6, pp. 702–720, 2014.
- [7] W. Lu, J. Duan, Z. Qiu, Z. Pan, R. W. Liu, and L. Bai, "Implementation of high-order variational models made easy for image processing," *Mathematical Methods in the Applied Sciences*, vol. 39, no. 14, pp. 4208–4233, 2016.
- [8] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [9] H. Othman and S. E. Shen-En Qian, "Noise reduction of hyperspectral imagery using hybrid spatial-spectral derivative-domain wavelet shrinkage," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 2, pp. 397–408, 2006.
- [10] W. Dong, F. Fu, G. Shi et al., "Hyperspectral image super-resolution via non-negative structured sparse representation," *IEEE Transactions on Image Processing*, vol. 25, no. 5, pp. 2337–2352, 2016.
- [11] Q. Wei, J. Bioucas-Dias, N. Dobigeon, and J.-Y. Tourneret, "Hyperspectral and multispectral image fusion based on a sparse representation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 7, pp. 3658–3668, 2015.
- [12] N. Akhtar, F. Shafait, and A. Mian, "Sparse spatio-spectral representation for hyperspectral image super-resolution," in *Proceedings of the 2014 European Conference on Computer Vision*, pp. 63–78, Zurich, Switzerland, September 2014.
- [13] Q. Yuan, L. Zhang, and H. Shen, "Hyperspectral image denoising employing a spectral-spatial adaptive total variation model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 10, pp. 3660–3677, 2012.
- [14] J. Wright, A. Ganesh, S. Rao et al., "Robust principal component analysis: exact recovery of corrupted low-rank matrices," in *Proceedings of the 2009 Neural Information Processing Systems*, pp. 2080–2088, Vancouver, British Columbia, Canada, December 2009.
- [15] Y. Q. Zhao and J. Yang, "Hyperspectral image denoising via sparse representation and low-rank constraint," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 1, pp. 296–308, 2015.
- [16] Y. Xie, Y. Qu, D. Tao, W. Wu, Q. Yuan, and W. Zhang, "Hyperspectral image restoration via iteratively regularized weighted schatten $\$p\$$ -norm minimization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4642–4659, 2016.
- [17] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, and L. Zhang, "Weighted schatten $\$p\$$ -norm minimization for image denoising and background subtraction," *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4842–4857, 2016.
- [18] W. He, H. Zhang, L. Zhang, and H. F. Shen, "Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 1, pp. 178–188, 2015.
- [19] M. A. Veganzones, M. Simoes, G. Licciardi, N. Yokoya, J. M. Bioucas-Dias, and J. Chanussot, "Hyperspectral super-resolution of locally low rank images from complementary multisource data," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 274–288, 2016.
- [20] Y. Chang, L. Yan, T. Wu, and S. Zhong, "Remote sensing image stripe noise removal: from image decomposition perspective," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7018–7031, 2016.
- [21] L. Zhuang and J. M. Bioucas-Dias, "Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 99, pp. 730–742, 2018.
- [22] D. L. Lieven, D. M. Bart, and V. Joos, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [23] D. Letexier and S. Bourennane, "Noise removal from hyperspectral images by multidimensional filtering," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 7, pp. 2061–2069, 2008.
- [24] X. Liu, S. Bourennane, and C. Fossati, "Denoising of hyperspectral images using the PARAFAC model and statistical performance analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 2, pp. 397–408, 2006.

- Geoscience and Remote Sensing*, vol. 50, no. 10, pp. 3717–3724, 2012.
- [25] X. Guo, X. Huang, L. Zhang, and L. Zhang, “Hyperspectral image noise reduction based on rank-1 tensor decomposition,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 83, no. 9, pp. 50–63, 2013.
- [26] Y. Peng, D. Meng, Z. Xu, C. Q. Gao, Y. Yang, and B. Zhang, “Decomposable nonlocal tensor dictionary learning for multispectral image denoising,” in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2949–2956, Columbus, OH, USA, June 2014.
- [27] W. Dong, G. Li, G. Shi, X. Li, and Y. Ma, “Low-rank tensor approximation with laplacian scale mixture modeling for multiframe image denoising,” in *Proceedings of the 2015 IEEE International Conference on Computer Vision*, pp. 442–449, Santiago, Chile, December 2015.
- [28] Q. Xie, Q. Zhao, D. Meng et al., “Multispectral images denoising by intrinsic tensor sparsity regularization,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1692–1700, Las Vegas, NV, USA, June 2016.
- [29] Y. Chang, L. Yan, and S. Zhong, “Hyper-laplacian regularized unidirectional low-rank tensor recovery for multispectral image denoising,” in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4260–4268, Honolulu, HI, USA, July 2017.
- [30] X. Zhang, X. Yuan, and L. Carin, “Nonlocal low-rank tensor factor Analysis for image restoration,” in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8232–8241, Salt Lake City, UT, USA, June 2018.
- [31] T. Yokota, B. Erem, S. Guiker, S. K. Warfield, and H. Hontani, “Missing slice recovery for tensors using a low-rank model in embedded space,” in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8251–8259, Salt Lake City, UT, USA, June 2018.
- [32] H. He, Q. Yao, C. Li, N. Yokoya, and Q. Zhao, “Non-local meets global: an integrated paradigm for hyperspectral denoising,” in *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6868–6877, Long Beach, CA, USA, June 2019.
- [33] Q. Xie, M. Zhou, Q. Zhao, D. Meng, W. M. Zuo, and Z. B. Xu, “Multispectral and hyperspectral image fusion by MS/HS fusion net,” in *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1585–1594, Long Beach, CA, USA, June 2019.
- [34] Z. Zhang, G. Ely, S. Aeron et al., “Novel methods for multilinear aata completion and de-noising based on tensor-SVD,” in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3842–3849, Columbus, OH, USA, June 2014.
- [35] Z. Zhang and S. Aeron, “Exact tensor completion using t-SVD,” *IEEE Transactions on Signal Processing*, vol. 65, no. 6, pp. 1511–1526, 2015.
- [36] C. Lu, J. Feng, Y. Chen et al., “Tensor robust principal component analysis: exact recovery of corrupted low-rank tensors via convex optimization,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5249–5257, Las Vegas, NV, USA, June 2016.
- [37] H.-Y. Zhang, W. He, L.-P. Zhang, H.-F. Shen, and Q. Q. Yuan, “Hyperspectral image restoration using low-rank tensor recovery,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 52, no. 8, pp. 4589–4604, 2017.
- [38] Y. B. Zheng, T. Z. Huang, X. L. Zhao et al., “Mixed noise removal in hyperspectral image via low-fibered-rank regularization,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 1, pp. 734–749, 2019.
- [39] K. Braman, “Third-order tensors as linear operators on a space of matrices,” *Linear Algebra and Its Applications*, vol. 433, no. 7, pp. 1241–1253, 2010.
- [40] M. E. Kilmer and C. D. Martin, “Factorization strategies for third-order tensors,” *Linear Algebra and Its Applications*, vol. 435, no. 3, pp. 641–658, 2011.
- [41] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, “Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging,” *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 148–172, 2013.
- [42] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller, “Tensor-based formulation and nuclear norm regularization for multienergy computed tomography,” *IEEE Transactions on Image Processing*, vol. 23, no. 4, pp. 1678–1693, 2014.
- [43] D. Lazzaro and S. Morigi, “Matrix completion for matrices with low-rank displacement,” *ETNA-Electronic Transactions on Numerical Analysis*, vol. 53, pp. 481–499, 2020.
- [44] I. W. Selesnick and I. Bayram, “Sparse signal estimation by maximally sparse convex optimization,” *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1078–1092, 2014.
- [45] I. W. Selesnick, A. Parekh, and I. Bayram, “Convex 1-D total variation denoising with non-convex regularization,” *IEEE Signal Processing Letters*, vol. 22, no. 2, pp. 141–144, 2014.
- [46] A. Parekh and I. W. Selesnick, “Convex denoising using non-convex tight frame regularization,” *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1786–1790, 2015.
- [47] V. Ollier, R. Boyer, M. N. Korso, and P. Larzabal, “Bayesian lower bounds for dense or sparse (outlier) noise in the RMT framework,” in *Proceedings of the 2016 IEEE Sensor Array Multichannel Signal Process*, pp. 1–5, Rio de Janeiro, Brazil, July 2016.
- [48] S. Sundin and M. Jansson, “Combined modeling of sparse and dense noise improves bayesian RVM,” in *Proceedings of the 2014 22nd European Signal Processing Conference (EUSIPCO)*, pp. 1841–1845, Lisbon, Portugal, September 2014.
- [49] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, “Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms,” *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 3952–3966, 2012.
- [50] Y. Wang, J. Peng, Q. Zhao, Y. Leung, X.-L. Zhao, and D. Meng, “Hyperspectral image restoration via total variation regularized low-rank tensor decomposition,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 4, pp. 1227–1243, 2018.
- [51] Y. Chen, X. Cao, Q. Zhao et al., “Denoising hyperspectral image with non-i.i.d. noise structure,” *IEEE Transactions on Cybernetics*, vol. 48, no. 3, pp. 1054–1066, 2017.