Hyperspectral Image Denoising via Nonconvex Logarithmic Penalty

Shuo Wang,1 Zhibin Zhu,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

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 $A \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the $i$th, horizontal, lateral, and frontal slices, respectively, $A(:, :, i)$ and $A(:, :, k)$ as the $k$th horizontal, lateral, and frontal slices, respectively, $A(:, :, k)$, and $A(:, :, k)$ is the tensor of size $n_1 \times n_2 \times n_3$. For $i = 1, 2, \ldots, n_1$ and $j = 1, 2, \ldots, n_2$, $D(i, j, :) = \sum_{k=1}^{n_3} G(i, k, :) \ast H(k, j, :)$.

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 $D \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the $t$-SVD of $D$ is given by $D = U \ast \mathcal{S} \ast V^T$, where $U \in \mathbb{R}^{n_1 \times n_1}$, $V \in \mathbb{R}^{n_2 \times n_2}$, and $\mathcal{S}$ is a rectangular diagonal tensor, and $\ast$ 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.

\[
\text{unfold}(A) = \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} \in \mathbb{R}^{n_1 \times n_2 \times n_3}. \tag{1}
\]

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

\[
\text{fold} \left( \begin{bmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{bmatrix} \right) = A \in \mathbb{R}^{n_1 \times n_2 \times n_3}. \tag{2}
\]
Figure 1: Fibers of a 3rd-order tensor. (a) Mode-1 (column) fiber: $x_{jk}$.
(b) Mode-2 (row) fiber: $x_{ik}$.
(c) Mode-3 (tube) fiber: $x_{ij}$.

Figure 2: Slices of a 3rd-order tensor. (a) Horizontal slices: $x_{ik}$.
(b) Lateral slices: $x_{jk}$.
(c) Frontal slices: $x_{ik}$.

Figure 3: Examples of the unfold and fold operators.

Figure 4: The $t$-SVD of a tensor.
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

$$
\text{bcirc}(\mathcal{A}, k) := \begin{pmatrix}
    A_k^{(1)} & A_k^{(n_2)} & \cdots & A_k^{(2)} \\
    A_k^{(2)} & A_k^{(1)} & \cdots & A_k^{(3)} \\
    \vdots & \vdots & \ddots & \vdots \\
    A_k^{(n_2)} & A_k^{(n_2-1)} & \cdots & A_k^{(1)}
\end{pmatrix},
$$

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

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

$$
\text{bdiag}(\mathcal{A}, k) := \begin{pmatrix}
    A_k^{(1)} \\
    A_k^{(2)} \\
    \vdots \\
    A_k^{(n_2)}
\end{pmatrix},
$$

$$
\text{bd} \text{fold}(\text{bdiag}(\mathcal{A}, k), k) = \mathcal{A}.
$$

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

$$
\text{bvec}(\mathcal{A}, k) := \begin{pmatrix}
    A_k^{(1)} \\
    A_k^{(2)} \\
    \vdots \\
    A_k^{(n_2)}
\end{pmatrix},
$$

$$
\text{bd} \text{fold}(\text{bvec}(\mathcal{A}, k), k) = \mathcal{A}.
$$

Definition 2 (Mode-$k$ $t$-product). For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_4 \times n_5 \times n_3}$, the mode-$1$ $t$-product is a tensor of size $n_1 \times n_2 \times n_4$:

$$
\mathcal{C} = \mathcal{A} \ast_1 \mathcal{B} = \text{bd} \text{fold}(\text{bcirc}(\mathcal{A}, 1)\text{bvec}(\mathcal{B}, 1), 1) \ast \mathcal{C}(\cdot, j, t) = \sum_{s=1}^{n_4} \mathcal{A}(\cdot, j, s) \ast \mathcal{B}(\cdot, s, t).
$$

(8)

For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_4 \times n_5 \times n_3}$, the mode-$2$ $t$-product is a tensor of size $n_1 \times n_2 \times n_3$:

$$
\mathcal{C} = \mathcal{A} \ast_2 \mathcal{B} = \text{bd} \text{fold}(\text{bcirc}(\mathcal{A}, 2)\text{bvec}(\mathcal{B}, 2), 2) \ast \mathcal{C}(\cdot, \cdot, t) = \sum_{s=1}^{n_4} \mathcal{A}(s, \cdot, t) \ast \mathcal{B}(i, \cdot, s).
$$

(9)

For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_4 \times n_5 \times n_3}$, the mode-$3$ $t$-product is a tensor of size $n_1 \times n_4 \times n_3$:

$$
\mathcal{C} = \mathcal{A} \ast_3 \mathcal{B} = \text{bd} \text{fold}(\text{bcirc}(\mathcal{A}, 3)\text{bvec}(\mathcal{B}, 3), 3) \ast \mathcal{C}(i, \cdot, \cdot) = \sum_{s=1}^{n_4} \mathcal{A}(i, s, \cdot) \ast \mathcal{B}(s, j, \cdot).
$$

(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 $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{A}^T_k \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ 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
\[ \mathcal{A}^T_i \ast_k \mathcal{A} = \mathcal{A} \ast_k \mathcal{A}^T_i = \mathcal{I}_k, \] (11)

where the tensor \( \mathcal{A} \in \mathbb{R}^{n_x \times n_y \times n_z} \) is mode-\( k \) orthogonal.

**Definition 7** (Tensor mode-\( k \) permutation). For \( \mathcal{A} \in \mathbb{R}^{n_x \times n_y \times n_z} \), \( \mathcal{A} \) is the mode-\( k \) permutation of \( \mathcal{A} \), where the \( i \)-th mode-\( k \) slice is the \( i \)-th mode-\( k \) slice of \( \mathcal{A} \), and its inverse operation is \( \mathcal{A} = \text{ipermute}(\mathcal{A}^T, k) \).

**Theorem 2** (Mode-\( k \) t-SVD). The factorization of \( \mathcal{A} \in \mathbb{R}^{n_x \times n_y \times n_z} \) is

\[ \mathcal{A} = \mathcal{U}_k \mathcal{S}_k \mathcal{Y}_k^T, \quad k = 1, 2, \text{ and } 3, \] (12)

where \( \mathcal{U}_k \) and \( \mathcal{Y}_k \) are the mode-\( k \) orthogonal tensors and \( \mathcal{S}_k \in \mathbb{R}^{n_x \times n_y \times n_z} \) is the mode-\( k \) diagonal tensor.

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

**Definition 8** (Tensor fibered rank). For \( \mathcal{A} \in \mathbb{R}^{n_x \times n_y \times n_z} \), \( \text{rank}_k(\mathcal{A}) \) is the fibered rank of \( \mathcal{A} \), whose \( i \)-th element is the mode-\( k \) tensor fibered rank \( \text{rank}_k(\mathcal{A}) \), \( \text{rank}_k(\mathcal{A}) = \max(\text{rank}_{m_i}(\mathcal{A})) \), where the \( i \)-th element of \( \text{rank}_{m_i}(\mathcal{A}) \) is the rank of the \( i \)-th mode-\( k \) slice of \( \mathcal{A} \) and \( m_i \) is the \( i \)-th mode-\( k \) slice of \( \mathcal{A} \).

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-rank 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

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

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

(A1) \( \phi(x; a) \in C^2(\mathbb{R}_+), \phi(x; a) \in C^0(\mathbb{R}_+) \), \( \mathbb{R}_+ = \{ t \in \mathbb{R} : t > 0 \} \) 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 \( \inf_{t \in \mathbb{R}_+} \phi''(x; a) = -a \).

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

\[ \Theta(y; \lambda, a) = \text{prox}_{\phi}(y; \lambda, a) = \arg\min_{x \in \mathbb{R}} \left\{ f(x) = \frac{1}{2} (y - x)^2 + \lambda \phi(x; a) \right\}. \] (14)

In [44–46], the authors prove that, for \( 0 < a \leq (1/\lambda) \), 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 \( \lambda \) as the threshold value [44], namely,

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

and is given by

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

\[ \min_{\mathcal{A}, \mathcal{N}, \mathcal{S}} \text{rank}_f(\mathcal{A}) + \lambda_1 \| \mathcal{N} \|_F^2 + \lambda_2 \| \mathcal{S} \|_1 \] (17)

s.t. \( \mathcal{Y} = \mathcal{A} + \mathcal{N} + \mathcal{S} \),

where \( \mathcal{Y} \) is the observed noisy HSI, \( \mathcal{N} \) is Gaussian noise, \( \mathcal{S} \) is sparse noise, \( \mathcal{A} \in \mathbb{R}^{n_x \times n_y \times n_z} \) is the clean hyperspectral signal, which has the low-fibered-rank tensor property, and \( \lambda_1 \) and \( \lambda_2 \) 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 \( \lambda_1 \) and \( \lambda_2 \).

Directly minimizing the tensor fibered rank is NP-hard. Based on logarithmic penalty equation (13), we propose that the following 3DNLATN HSI denoising model is commonly formulated:
Input: $a \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $k$

$\text{fft}(a, k) \rightarrow \overline{a}_k$

for $i = 1$ to $n_k$ do

$[U, S, V] = \text{SVD}((\overline{a}_k)_{ij}^{(i)}); U \rightarrow (\overline{a}_k)_{ij}^{(i)}; S \rightarrow (\overline{a}_k)_{ij}^{(i)}; V \rightarrow (\overline{a}_k)_{ij}^{(i)}$

end for

$\text{ifft}(\overline{a}_k, k) \rightarrow \mathcal{H}_k^1; \text{ifft}(\overline{a}_k, k) \rightarrow \mathcal{H}_k^2; \text{ifft}(\overline{a}_k, k) \rightarrow \mathcal{H}_k^3$

output: $\mathcal{H}_k, \delta_k, \mathcal{Y}_k$

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

$$\min_{\mathcal{A}, \mathcal{N}, \mathcal{S}} \sum_{k=1}^{n_k} \frac{1}{3} \tau_k \mathcal{F}_k (\mathcal{A}) + \lambda_1 \|\mathcal{N}\|_F^2 + \lambda_2 \|\mathcal{S}\|_1$$

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

$$\text{LogATNN}_k (\mathcal{A}) = \sum_{i=1}^{n_k} \sum_{j=1}^{m_k} \frac{1}{n_k} \log \left(1 + \alpha \sigma_j (\overline{A}_k)_{ij}^{(i)} \right),$$

$$\sigma_j (\overline{A}_k)_{ij}^{(i)} = \|\mathcal{A} - \mathcal{X}_k\|_F^2 + \lambda_1 \|\mathcal{N}\|_F^2 + \lambda_2 \|\mathcal{S}\|_1$$

3.3. ADMM for Solving Model (18). We use the ADMM to solve equation (18). By casting the three auxiliary variables $(\mathcal{X}_k, \mathcal{A}, \mathcal{N}, \mathcal{S}, \mathcal{W}, \mathcal{Q})$ equation (18) can be rewritten as follows:

$$\min_{\mathcal{A}, \mathcal{N}, \mathcal{S}} \sum_{k=1}^{n_k} \frac{1}{3} \tau_k \mathcal{F}_k (\mathcal{X}_k) + \langle \mathcal{A} - \mathcal{X}_k, \mathcal{W}_k \rangle + \frac{\xi_k}{2} \|\mathcal{A} - \mathcal{X}_k\|_F^2$$

$$+ \langle \mathcal{Y} - (\mathcal{A} + \mathcal{N} + \mathcal{S}), \mathcal{Q} \rangle + \rho \|\mathcal{Y} - (\mathcal{A} + \mathcal{N} + \mathcal{S})\|_F^2,$$

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

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

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

$$\mathcal{X}_k^{t+1} = \arg \min \tau_k \mathcal{F}_k (\mathcal{X}_k) + \frac{\xi_k}{2} \|\mathcal{A} - \mathcal{X}_k + \mathcal{W}_k\|_F^2,$$

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

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

$$\mathcal{X}_k^{t+1} = \arg \min \tau_k \mathcal{F}_k (\mathcal{X}_k) + \frac{\xi_k}{2} \|\mathcal{A} - \mathcal{X}_k + \mathcal{W}_k\|_F^2,$$

where $\mathcal{M} = \mathcal{A} + (\mathcal{W}_k \mathcal{X}_k) = \mathcal{A} \mathcal{X}_k \mathcal{S}^{*} \mathcal{E}^{T} x, \overline{\mathcal{X}}_k = \text{fft}(\mathcal{X}_k, k)$, and $\mathcal{L} = \tau_k / \xi_k$. From equation (16),
\[
\hat{S}_k = \theta(\mathcal{X}_k(i, j, r)) = \begin{cases} 
\left[\frac{|\mathcal{X}_k(i, j, r)| - \frac{1}{2a}}{2} + \sqrt{\left(\frac{|\mathcal{X}_k(i, j, r)| + \frac{1}{2a}}{2}\right)^2 - \frac{1}{a}}\right] \sgn(\mathcal{X}_k(i, j, r)), & |\mathcal{X}_k(i, j, r)| \geq \lambda, \\
0, & |\mathcal{X}_k(i, j, r)| < \lambda,
\end{cases}
\]

where \( \hat{S}_k = \text{ifft}(\hat{S}_k, k) \) and \( \mathcal{X}_k = \mathcal{X}(M, k) = \mathcal{U}_k \hat{S}_k * \mathcal{X} \) \( \forall \tau, \lambda > 0 \) and \( 0 < a \leq (1/\lambda) \). Therefore,

\[
\mathcal{X}^{(i+1)} = \mathcal{D}\left(\mathcal{A} + \frac{\mathcal{W}_k(i, j, r)}{\xi_k}\right).
\]

solving which is equivalent to the following subproblem:

\[
\mathcal{A}^{i+1} = \arg\min_{\mathcal{A}} \sum_{i=1}^{3} \left[ \langle \mathcal{A} - \mathcal{X}_k, \mathcal{W}_k \rangle + \frac{\mathcal{Y} - (\mathcal{A} + \mathcal{N} + \delta) + \mathcal{Q} \rho}{\mathcal{F}} \right].
\]

It has the following closed-form solution:

\[
\mathcal{A}^{i+1} = \sum_{k=1}^{3} \hat{\xi}_k (\mathcal{X}^{i+1}_k - (\mathcal{W}_k(i, j, r)) + \rho \mathcal{Y} - (\mathcal{A} + \mathcal{N} + \delta) + \mathcal{Q} \rho \mathcal{F}).
\]

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

\[
\arg\min_{\mathcal{N}} L_{\mathcal{U}, \rho}(\mathcal{X}_k, \mathcal{A}, \mathcal{N}, \delta, \mathcal{W}, \mathcal{Q}) = \lambda_1 \mathcal{N}^2 + \langle \mathcal{Y} - (\mathcal{A} + \mathcal{N} + \delta), \mathcal{Q} \rangle + \frac{\rho}{\mathcal{F}} \mathcal{Y} - (\mathcal{A} + \mathcal{N} + \delta)\mathcal{F}^2.
\]

which is equivalent to the following subproblem:

\[
\mathcal{N}^{i+1} = \arg\min_{\mathcal{N}} \lambda_1 \mathcal{N}^2 + \frac{\rho}{\mathcal{F}} \mathcal{Y} - (\mathcal{A}^{i+1} + \mathcal{N} + \delta)\mathcal{F}^2 + \mathcal{Q}^2 \rho \mathcal{F}^2.
\]

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

\[
\hat{S}^{i+1} = \sgn\left(\mathcal{Y} - \mathcal{A}^{i+1} + \mathcal{N}^{i+1} + \mathcal{Q}^2 \rho \mathcal{F}^2 - \frac{\lambda_2}{\rho} \mathcal{Q}^2 \rho \mathcal{F}^2\right)\max\left(\mathcal{Y} - \mathcal{A}^{i+1} + \mathcal{N}^{i+1} + \mathcal{Q}^2 \rho \mathcal{F}^2 - \frac{\lambda_2}{\rho} \mathcal{Q}^2 \rho \mathcal{F}^2, 0\right).
\]

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

\[
\mathcal{W}_k^{i+1} = \mathcal{W}_k^{i} + \xi_k (\mathcal{A}^{i+1} - \hat{S}_k^{i+1}),
\]

\[
\mathcal{Q}^{i+1} = \mathcal{Q}^{i} + \rho \mathcal{Y} - \mathcal{A}^{i+1} + \mathcal{N}^{i+1} - \mathcal{Q}^{i+1}.
\]
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 × 145 × 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:

\[
\text{PSNR} = 10 \times \log_{10} \left( \frac{L^2 MN}{\sum_{x=1}^{M} \sum_{y=1}^{N} \left[ I(x, y) - \bar{I}(x, y) \right]^2} \right).
\]

\[
\text{MPSNR} = \frac{1}{B} \sum_{x=1}^{B} \text{PSNR}_x,
\]

where \( M \times N \) represents the image size in the space, \( \bar{I}(x, y) \) 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.

\[
\text{SSIM} = \frac{(2\mu_I \mu_{I'} + C_1)(2\sigma_{xy} + C_2)}{(\mu_I^2 + \mu_{I'}^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}
\]

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

where \( C_1 \) is a constant for \( \mu_x^2 + \mu_y^2, C_2 \) is the same as \( C_1, \sigma_x \) and \( \sigma_y \) 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 jth local window contents, and \( Q \) is the number of local windows:

\[
\text{SAM}(v, v') = \cos^{-1} \left( \frac{\sum_{j=1}^{Z} v_j v'_j}{\sum_{j=1}^{Z} v_j^2 \sum_{j=1}^{Z} v'_j^2} \right),
\]

where \( Z \) is the number of wavelengths, \( v \) and \( v' \) represent the spectrum vectors, and

\[
v = [v_1, v_2, \ldots, v_N], \quad v' = [v'_1, v'_2, \ldots, v'_N].
\]

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 \( \beta \) and the variance of the impulse noise \( \delta \). 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 \( \beta \) 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 \( \delta \) 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 \( \beta \) is 0.3, and the ratio of the impulse noise \( \delta \) is 0.1; the variance in the Gaussian noise \( \beta \) is 0.1, and the ratio of the impulse noise \( \delta \) is 0.4; the variance in the Gaussian noise \( \beta \) is 0.3, and the ratio of the impulse noise \( \delta \) 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 \( \beta \) is randomly changed from \( U(0.3, 0.4) \), and the ratio of the impulse noise \( \delta \) is 0.2; the variance in the Gaussian noise \( \beta \) is randomly changed from \( U(0.2, 0.3) \), and the ratio of impulse noise \( \delta \) is 0.3; the variance in the Gaussian noise \( \beta \) is randomly changed from \( U(0.4, 0.5) \), and the ratio of impulse noise \( \delta \) 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 \( \beta \) is 0.1, and the ratios of impulse noise \( \delta \) are randomly...
Input: The noisy HSI $Y$, parameters $\tau_k, \zeta_k$ ($k = 1, 2, \text{and } 3), \lambda_1, \lambda_2, \rho, t = 1, 2, I = 0$.
$\mathcal{X}^0 = 0, \mathcal{X}'^0 = 0, \mathcal{Y}'^0 = 0, \mathcal{X}'^0_k = 0$, and $\mathcal{Y}'^0 = 0$.

while not converged do
  Update $\mathcal{X}'^k$ with equation (26), $k = 1, 2, \text{and } 3$.
  Update $\mathcal{X}'^k$ with equation (29).
  Update $\mathcal{X}'^k$ with equation (32).
  Update $\mathcal{X}'^k$ with equation (35).
  Update $\mathcal{X}'^k$ with equation (36), $k = 1, 2, \text{and } 3$.
  Update $\mathcal{X}'^k$ with equation (37).
  Let $\zeta_k = t\zeta_k, k = 1, 2, \text{and } 3, \rho = t\rho$, and $l = l + 1$.
  Check the convergence condition:
  $(\|\mathcal{X}'^{l+1} - \mathcal{X}'^l\|_F/\|\mathcal{X}'^l\|_F) < 10^{-4}$.
end while

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

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

![Graph](image1)

![Graph](image2)

(a) Change in the MPSNR value
(b) Change in the MSSIM value

Figure 5: Sensitivity analysis of parameter $\lambda_1$ 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 $\beta$ is 0.3, and the ratios of impulse noise $\delta$ are randomly varied from $U(0.3, 0.4)$; the variance in the Gaussian noise $\beta$ is 0.2, and the ratios of impulse noise $\delta$ are randomly varied from $U(0.4, 0.5)$. 

Mathematical Problems in Engineering 9
(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 $\rho$ 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 $\delta$ 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 $r_k$, the regularization parameters $\lambda_1$ and $\lambda_2$, the threshold parameter $\xi_k = (r_k/\xi)$, 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 $\lambda_1$ for 3DLogATNN: it is easy to see that $\lambda_1$ is the parameter used to restrict the sparsity of the Gaussian noise. We set $\lambda_1 = (C/\beta)$, where $\beta$ 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 $[0.0001, 0.0005, 0.0009, 0.001, 0.002, 0.003, 0.004, 0.005, 0.007, 0.009, 0.11, 0.15]$. It can be clearly observed from this figure that the results of the 3DLogATNN solver are relatively stable in terms of both MPSNR and MSSIM.
| Case   | Data                   | Gaussian noise | Impulse noise | Indicators | Noise | 3DTNN | 3DLogTNN | 3DLogATNN |
|--------|------------------------|----------------|--------------|------------|-------|-------|----------|-----------|
| Case 1-1 | Washington DC Mall     | β ∈ 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 5.5300 5.8969 6.7688 |
|         |                        |                |              | Time (s)   | 119.688 310.708 207.411 |
| Case 1-2 | Washington DC Mall     | β ∈ 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     | β ∈ 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     | β ∈ 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          | β ∈ 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          | β ∈ 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     | —                | δ ∈ 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     | —                | δ ∈ 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     | —                | δ ∈ 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     | —                | δ ∈ 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     | —                | δ ∈ 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          | —                | δ ∈ 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          | —                | δ ∈ 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     | β = 0.3        | δ = 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 |
| 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 MSSIM: 0.0751 SAM: 49.8593 | 24.5273 0.7226 12.8686 | 30.4000 0.9213 6.3862 | 30.5960 0.9226 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 MSSIM: 0.0359 SAM: 55.9567 | 16.3370 0.2983 21.9255 | 14.9055 0.2406 28.7635 | 20.9115 0.6499 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 MSSIM: 0.0642 SAM: 53.2858 | 25.7997 0.7448 5.4009 | 13.1319 0.1337 42.7153 | 26.8088 0.8029 5.6316 |
|         |                  |                |               | Time (s): 51.955 | 93.249 65.797 |
| Cases 3–5 | Pavia City Center | $\beta = 0.1$ | $\delta = 0.4$ | MPSNR: 8.4711 MSSIM: 0.0443 SAM: 46.8083 | 28.4963 0.8615 4.2606 | 30.2146 0.8613 6.4614 | 31.0972 0.9098 3.9908 |
|         |                  |                |               | Time (s): 63.319 | 116.127 63.407 |
| Cases 3–6 | Indian Pines    | $\beta = 0.1$ | $\delta = 0.4$ | MPSNR: 8.9083 MSSIM: 0.0719 SAM: 35.5125 | 25.9148 0.6817 5.2059 | 30.1284 0.8644 2.8185 | 30.7026 0.8164 2.7058 |
|         |                  |                |               | Time (s): 134.676 | 234.822 165.194 |
| Cases 3–7 | Indian Pines    | $\beta = 0.3$ | $\delta = 0.1$ | MPSNR: 9.2176 MSSIM: 0.0971 SAM: 34.2324 | 23.6223 0.8105 5.5598 | 26.8662 0.7807 4.0714 | 27.1730 0.7415 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 MSSIM: 0.0702 SAM: 57.1506 | 22.7893 0.7124 11.2378 | 24.8984 0.8004 9.4788 | 25.2959 0.8008 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 MSSIM: 0.0737 SAM: 53.4819 | 22.1301 0.6854 12.5629 | 25.1477 0.8146 9.5805 | 25.7826 0.8178 9.3802 |
|         |                  |                |               | Time (s): 200.652 | 338.836 165.677 |
| Cases 4–3 | Washington DC Mall | $\beta \in U(0.4, 0.5)$ | $\delta = 0.1$ | MPSNR: 6.2377 MSSIM: 0.0602 SAM: 61.5814 | 23.7276 0.7283 10.0163 | 9.6080 0.1128 53.2827 | 25.6329 0.8029 9.1467 |
|         |                  |                |               | Time (s): 190.534 | 323.047 172.353 |
| Cases 4–4 | Pavia City Center | $\beta \in U(0.2, 0.3)$ | $\delta = 0.3$ | MPSNR: 7.8602 MSSIM: 0.0421 SAM: 51.6427 | 23.5873 0.5911 9.9142 | 24.6273 0.6943 6.0669 | 25.6363 0.7801 5.9457 |
|         |                  |                |               | Time (s): 113.882 | 150.160 91.212 |
| Cases 4–5 | Pavia City Center | $\beta \in U(0.3, 0.4)$ | $\delta = 0.2$ | MPSNR: 7.2535 MSSIM: 0.0395 SAM: 56.2301 | 23.5193 0.5704 10.1307 | 24.3368 0.6672 6.0174 | 25.1019 0.7372 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 MSSIM: 0.0698 SAM: 37.6925 | 22.1915 0.7802 7.0587 | 26.0036 0.7771 4.5808 | 26.5240 0.7409 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 MSSIM: 0.0578 SAM: 43.2485 | 20.8380 0.7511 7.0333 | 15.0486 0.2008 19.4183 | 23.8612 0.6976 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 MSSIM: 0.0447 SAM: 51.2579 | 16.5893 0.2855 27.3488 | 10.6705 0.1130 43.5216 | 28.4585 0.8819 7.8981 |
|         |                  |                |               | Time (s): 213.342 | 317.644 172.836 |
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 \( \lambda_2 \) for 3DLogATNN: it is easy to see that \( \lambda_2 \) is the parameter used to restrict the sparsity of the impulse noise. We set \( \lambda_2 = B \lambda \), where

\[
\lambda = \frac{\tau_1}{\sqrt{\max(n_2, n_3)n_1}} + \frac{\tau_2}{\sqrt{\max(n_3, n_1)n_2}} + \frac{\tau_3}{\sqrt{\max(n_1, n_2)n_3}}
\]

and \( B \) is a tuning parameter. The results were based on the simulated data experiment in case 2-1. Figure 6 shows the
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 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.
Figure 10: Continued.
MPSNR and MSSIM values, with the value of $a$ changing from 0.05 to 0.06. Therefore, we suggest the use of $a = 0.05$ 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], LRMR [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

![Figure 10: Denoising results of the Indian Pines dataset with mixture noise.](image)
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 Jang, "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. Aharony, "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. 54, 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
