

Hyperspectral Image Denoising via Convex Low-Fibered-Rank Regularization

Yu-Bang Zheng

Ting-Zhu Huang, Xi-Le Zhao, Tai-Xiang Jiang, and Jie Huang

University of Electronic Science and Technology of China (UESTC)

30 July 2019



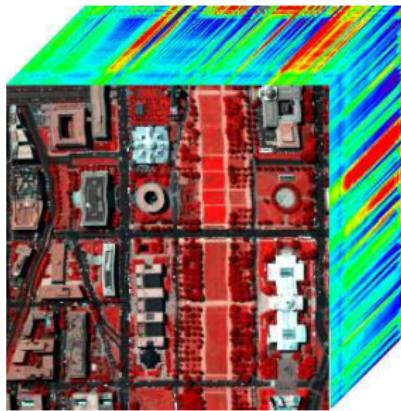
Outline

- 1 Introduction
- 2 The Proposed Model and Algorithm
- 3 Numerical Experiments



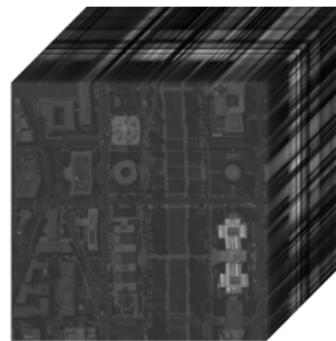
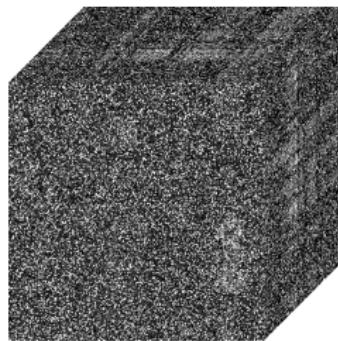
Hyperspectral Image (HSI)

HSIs contain wealthy spatial-spectral knowledge and have been widely used in many applications, such as material identification, mineral detection, and forest inspection.



Why Study HSI Denoising?

HSIs in real applications always suffer from various noises, such as Gaussian noise, sparse noise, and stripes.



Conclusive Issue for HSI Denoising

Exploring accurate spatial-spectral prior knowledge of HSIs:

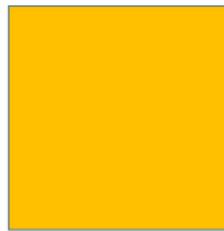
- piecewise smoothness;
- nonlocal self-similarity;
- **low rankness**;
- ...



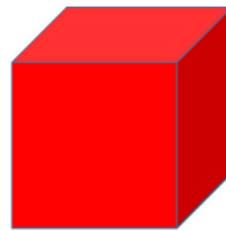
Tensor



vector



matrix



tensor



Tensor Basics (Fibers and Slices)

A *fiber* of a tensor \mathcal{X} is a vector generated by fixing every index but one.

A slice of a tensor \mathcal{X} is a matrix generated by fixing every index but two.

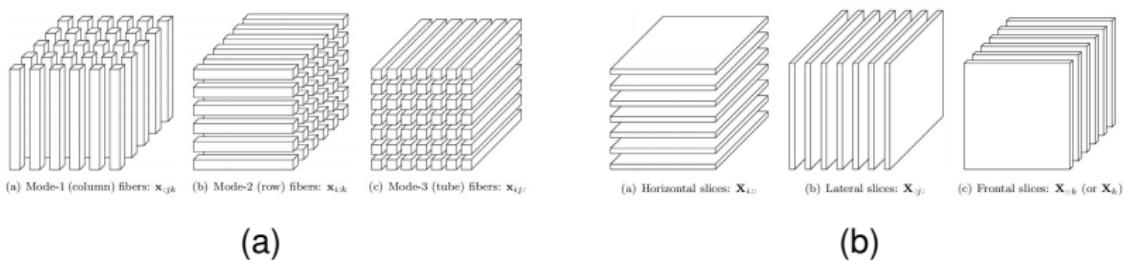


Figure 1: Fibers and slices of three-way tensors.



T-Product, T-SVD, and Tubal Rank

M-product: $F = X \cdot Y \Leftrightarrow F(i, j) = \sum_{t=1}^{n_2} X(i, t)Y(t, j).$

T-product: $\mathcal{F} = \mathcal{X} * \mathcal{Y} \Leftrightarrow \mathcal{F}(i, j, :) = \sum_{t=1}^{n_2} \mathcal{X}(i, t, :) * \mathcal{Y}(t, j, :)$,
where $*$ denotes the circular convolution between two tubes.



T-Product, T-SVD, and Tubal Rank

M-product: $F = X \cdot Y \Leftrightarrow F(i, j) = \sum_{t=1}^{n_2} X(i, t)Y(t, j)$.

T-product: $\mathcal{F} = \mathcal{X} * \mathcal{Y} \Leftrightarrow \mathcal{F}(i, j, :) = \sum_{t=1}^{n_2} \mathcal{X}(i, t, :) * \mathcal{Y}(t, j, :)$,
 where $*$ denotes the circular convolution between two tubes.

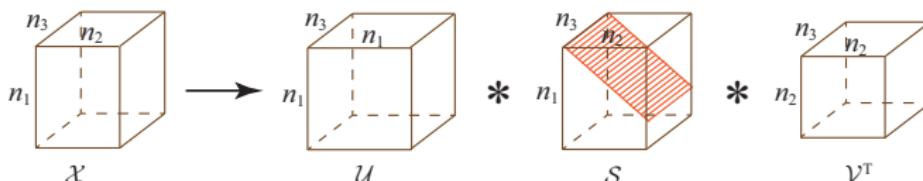


Figure 2: The t-SVD for three-way tensors.



T-Product, T-SVD, and Tubal Rank

M-product: $F = X \cdot Y \Leftrightarrow F(i, j) = \sum_{t=1}^{n_2} X(i, t) Y(t, j)$.

T-product: $\mathcal{F} = \mathcal{X} * \mathcal{Y} \Leftrightarrow \mathcal{F}(i, j, :) = \sum_{t=1}^{n_2} \mathcal{X}(i, t, :) * \mathcal{Y}(t, j, :)$,
 where $*$ denotes the circular convolution between two **tubes**.

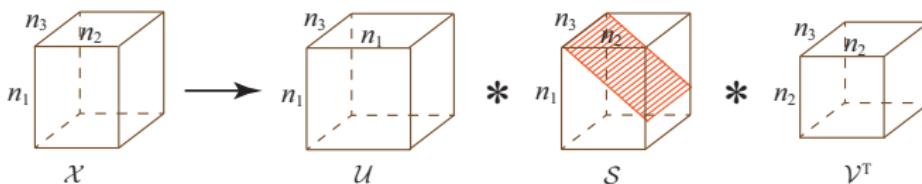


Figure 2: The t-SVD for three-way tensors.

The **tubal rank** of \mathcal{X} is defined as the number of non-zero **tubes** of \mathcal{S} , i.e., $\text{rank}_t(\mathcal{X}) := \#\{i : \mathcal{S}(i, i, :) \neq 0\}$.



Motivation

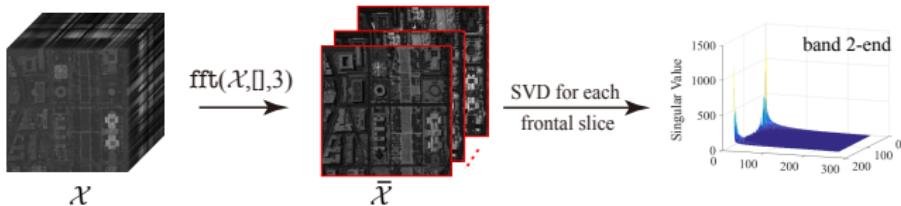


Figure 3: The t-SVD for an HSI.

When setting the band of an HSI to be the frontal slice of a three-way tensor, the t-SVD characterizes its **spatial correlations** via **SVDs**, while describes its **spectral correlation** by the embedded **circular convolution** or DFT.



The Proposed Mode- k T-Product

Mode- k t-product ($*_k$):

$$\mathcal{F} = \mathcal{X} *_1 \mathcal{Y} \Leftrightarrow \mathcal{F}(:, j, s) = \sum_{t=1}^{n_3} \mathcal{X}(:, j, t) \star \mathcal{Y}(:, t, s),$$

$$\mathcal{F} = \mathcal{X} *_2 \mathcal{Y} \Leftrightarrow \mathcal{F}(i, :, s) = \sum_{t=1}^{n_1} \mathcal{X}(t, :, s) \star \mathcal{Y}(i, :, t),$$

$$\mathcal{F} = \mathcal{X} *_3 \mathcal{Y} \Leftrightarrow \mathcal{F}(i, j, :) = \sum_{t=1}^{n_2} \mathcal{X}(i, t, :) \star \mathcal{Y}(t, j, :).$$



The Proposed Mode- k T-SVD and Fibered Rank

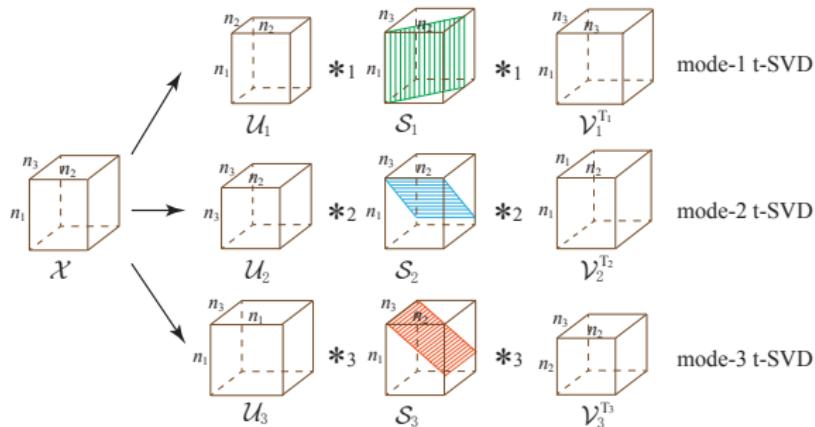


Figure 4: The mode- k t-SVD for three-way tensors ($k=1,2,3$).



The Proposed Mode- k T-SVD and Fibered Rank

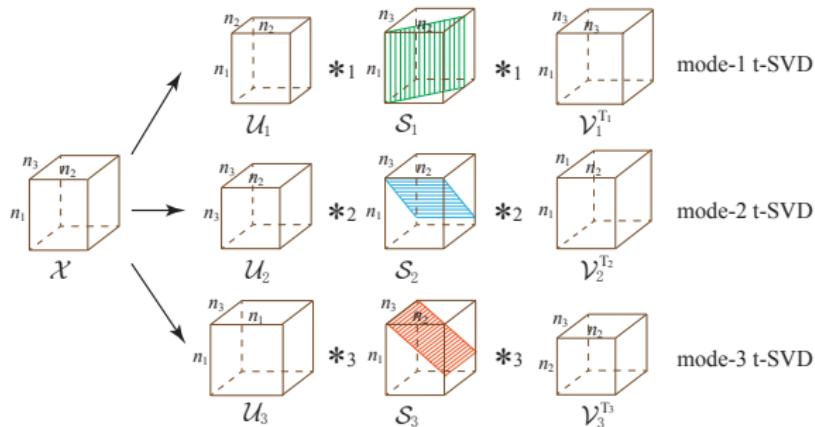


Figure 4: The mode- k t-SVD for three-way tensors ($k=1,2,3$).

The **mode- k fibered rank**: $\text{rank}_{f_k}(\mathcal{X})$ is defined as the number of non-zero mode- k fibers of \mathcal{S}_k .



The Proposed Mode- k T-SVD and Fibered Rank

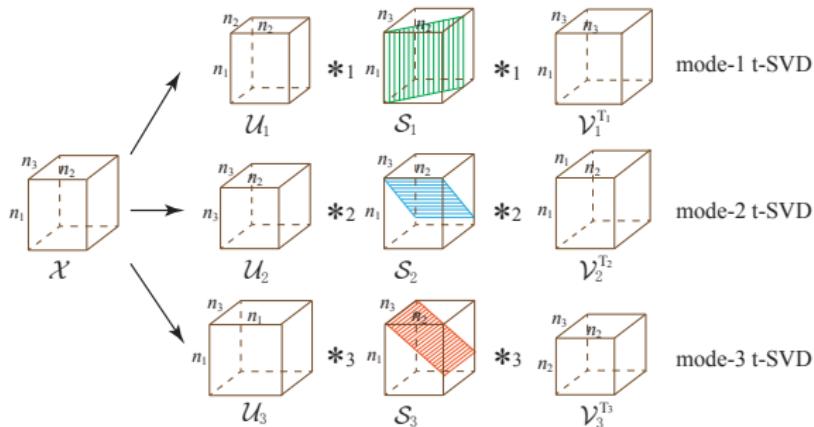


Figure 4: The mode- k t-SVD for three-way tensors ($k=1,2,3$).

The **mode- k fibered rank**: $\text{rank}_{f_k}(\mathcal{X})$ is defined as the number of non-zero mode- k fibers of \mathcal{S}_k .

The **fibered rank**: $\text{rank}_f(\mathcal{X}) = (\text{rank}_{f_1}(\mathcal{X}), \text{rank}_{f_2}(\mathcal{X}), \text{rank}_{f_3}(\mathcal{X}))$.



Low-Fibered-Rank Prior for An HSI

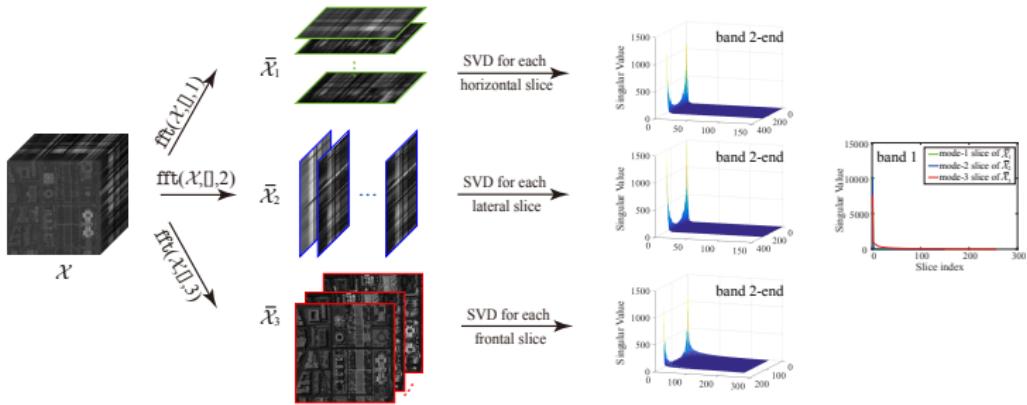


Figure 5: The mode- k t-SVD for an HSI.



Low-Fibered-Rank Prior for An HSI

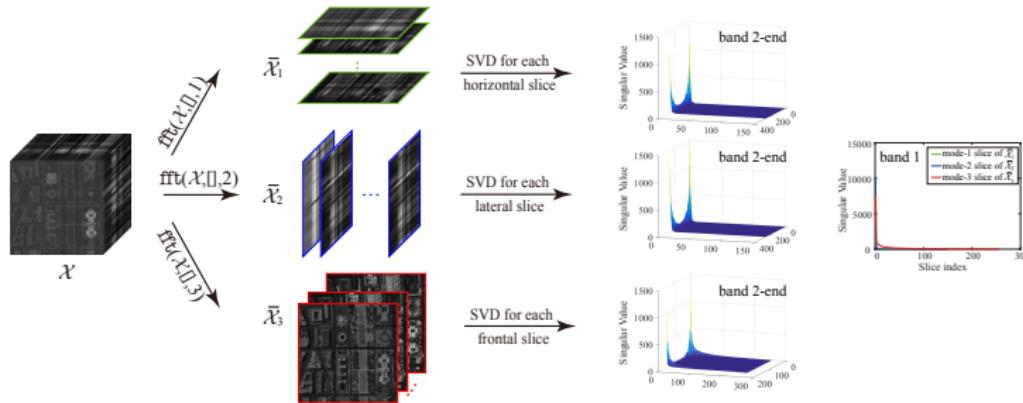


Figure 5: The mode- k t-SVD for an HSI.

Table 1: The rank estimation of an HSI.

Data	Size	Tucker rank	Tubal rank	Fibered rank
Washington DC Mall	$256 \times 256 \times 150$	(107,110,6)	182	(8,8,182)

Convex Relaxation: Three-Directional Tensor Nuclear Norm (3DTNN)

Mode- k TNN: $\|\mathcal{X}\|_{\text{TNN}_k}$ is defined as the sum of singular values of all the mode- k slices of $\bar{\mathcal{X}}_k$, i.e.,

$$\|\mathcal{X}\|_{\text{TNN}_k} := \sum_{i=1}^{n_k} \|(\bar{\mathcal{X}}_k)_k^{(i)}\|_*,$$

where $(\bar{\mathcal{X}}_k)_k^{(i)}$ is the i -th mode- k slice of $\bar{\mathcal{X}}_k$ with $\bar{\mathcal{X}}_k = \text{fft}(\mathcal{X}, [], k)$.



Convex Relaxation: Three-Directional Tensor Nuclear Norm (3DTNN)

Mode- k TNN: $\|\mathcal{X}\|_{\text{TNN}_k}$ is defined as the sum of singular values of all the mode- k slices of $\bar{\mathcal{X}}_k$, i.e.,

$$\|\mathcal{X}\|_{\text{TNN}_k} := \sum_{i=1}^{n_k} \|(\bar{\mathcal{X}}_k)_k^{(i)}\|_*,$$

where $(\bar{\mathcal{X}}_k)_k^{(i)}$ is the i -th mode- k slice of $\bar{\mathcal{X}}_k$ with $\bar{\mathcal{X}}_k = \text{fft}(\mathcal{X}, [], k)$.

3DTNN: $\|\mathcal{X}\|_{\text{3DTNN}}$ is defined as

$$\|\mathcal{X}\|_{\text{3DTNN}} := \sum_{k=1}^3 \alpha_k \|\mathcal{X}\|_{\text{TNN}_k},$$

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



Outline

1 Introduction

2 The Proposed Model and Algorithm

3 Numerical Experiments



3DTNN-Based HSI Denoising Model

Considering a target HSI $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the proposed 3DTNN-based HSI denoising model is formulated as

$$\begin{aligned} & \min_{\mathcal{X}, \mathcal{N}, \mathcal{S}} \|\mathcal{X}\|_{3\text{DTNN}} + \lambda_1 \|\mathcal{N}\|_F^2 + \lambda_2 \|\mathcal{S}\|_1, \\ & \text{s.t. } \mathcal{Y} = \mathcal{X} + \mathcal{N} + \mathcal{S}, \end{aligned} \tag{1}$$

where \mathcal{Y} is the observed HSI, \mathcal{N} is Gaussian noise, and \mathcal{S} is sparse noise.



3DTNN-Based HSI Denoising Model

Considering a target HSI $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the proposed 3DTNN-based HSI denoising model is formulated as

$$\begin{aligned} & \min_{\mathcal{X}, \mathcal{N}, \mathcal{S}} \|\mathcal{X}\|_{3\text{DTNN}} + \lambda_1 \|\mathcal{N}\|_F^2 + \lambda_2 \|\mathcal{S}\|_1, \\ & \text{s.t. } \mathcal{Y} = \mathcal{X} + \mathcal{N} + \mathcal{S}, \end{aligned} \quad (1)$$

where \mathcal{Y} is the observed HSI, \mathcal{N} is Gaussian noise, and \mathcal{S} is sparse noise. The problem (1) can be rewritten as

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



ADMM-Based Algorithm

We use the ADMM to solve (2). We introduce three auxiliary tensors \mathcal{Z}_k ($k = 1, 2, 3$) and reformulate (2) as



ADMM-Based Algorithm

We use the ADMM to solve (2). We introduce three auxiliary tensors \mathcal{Z}_k ($k = 1, 2, 3$) and reformulate (2) as

$$\begin{aligned} & \min_{\mathcal{X}, \mathcal{N}, \mathcal{S}, \mathcal{Z}_k} \sum_{k=1}^3 \alpha_k \|\mathcal{Z}_k\|_{\text{TNN}_k} + \lambda_1 \|\mathcal{N}\|_F^2 + \lambda_2 \|\mathcal{S}\|_1, \\ & \text{s.t. } \mathcal{Y} - (\mathcal{X} + \mathcal{N} + \mathcal{S}) = 0, \quad \mathcal{X} - \mathcal{Z}_k = 0, \quad k = 1, 2, 3. \end{aligned} \tag{3}$$



ADMM-Based Algorithm

We use the ADMM to solve (2). We introduce three auxiliary tensors \mathcal{Z}_k ($k = 1, 2, 3$) and reformulate (2) as

$$\begin{aligned} & \min_{\mathcal{X}, \mathcal{N}, \mathcal{S}, \mathcal{Z}_k} \sum_{k=1}^3 \alpha_k \|\mathcal{Z}_k\|_{\text{TNN}_k} + \lambda_1 \|\mathcal{N}\|_F^2 + \lambda_2 \|\mathcal{S}\|_1, \\ & \text{s.t. } \mathcal{Y} - (\mathcal{X} + \mathcal{N} + \mathcal{S}) = 0, \quad \mathcal{X} - \mathcal{Z}_k = 0, \quad k = 1, 2, 3. \end{aligned} \quad (3)$$

The augmented Lagrangian function of (3) is

$$\begin{aligned} L_{\mu_k, \beta}(\mathcal{Z}_k, \mathcal{X}, \mathcal{N}, \mathcal{S}, \mathcal{M}_k, \mathcal{P}) = & \sum_{k=1}^3 \left\{ \alpha_k \|\mathcal{Z}_k\|_{\text{TNN}_k} \right. \\ & + \langle \mathcal{X} - \mathcal{Z}_k, \mathcal{M}_k \rangle + \mu_k / 2 \|\mathcal{X} - \mathcal{Z}_k\|_F^2 \Big\} + \lambda_1 \|\mathcal{N}\|_F^2 + \lambda_2 \|\mathcal{S}\|_1 \\ & + \langle \mathcal{Y} - (\mathcal{X} + \mathcal{N} + \mathcal{S}), \mathcal{P} \rangle + \beta / 2 \|\mathcal{Y} - (\mathcal{X} + \mathcal{N} + \mathcal{S})\|_F^2. \end{aligned}$$



ADMM-Based Algorithm

Algorithm 1 ADMM-based optimization algorithm for the 3DTNN-based HSI denosing model.

Input: The noisy HSI \mathcal{Y} , parameters $\alpha = (\alpha_1, \alpha_2, \alpha_3)$, $\mu = (\mu_1, \mu_2, \mu_3)$, $\lambda_1, \lambda_2, \beta$ and $\rho = 1.2$.

Initialization: $p = 0$, $\mathcal{X}^0 = 0$, $\mathcal{N}^0 = 0$, $\mathcal{S}^0 = 0$, $\mathcal{Z}_k^0 = 0$, $\mathcal{M}_k^0 = 0$, and $\mathcal{P}^0 = 0$.

while not converged **do**

Update $\mathcal{Z}_k^{p+1} = \mathcal{D}_{\alpha_k/\mu_k}(\mathcal{X}^p + \mathcal{M}_k^p/\mu_k, k)$, $k = 1, 2, 3$.

Update $\mathcal{X}^{p+1} = (\sum_{k=1}^3 (\mu_k \mathcal{Z}_k^{p+1} - \mathcal{M}_k^p) + (\beta \mathcal{Y} - \beta \mathcal{N}^p - \beta \mathcal{S}^p + \mathcal{P}^p)) / (\sum_{k=1}^3 \mu_k + \beta)$.

Update $\mathcal{N}^{p+1} = (\beta \mathcal{Y} - \beta \mathcal{X}^{p+1} - \beta \mathcal{S}^p + \mathcal{P}^p) / (2\lambda_1 + \beta)$.

Update $\mathcal{S}^{p+1} = \text{shrink}(\mathcal{Y} - \mathcal{X}^{p+1} - \mathcal{N}^{p+1} + \frac{\mathcal{P}^p}{\beta}, \frac{\lambda_2}{\beta})$.

Update $\mathcal{M}_k^{p+1} = \mathcal{M}_k^p + \mu_k(\mathcal{X}^{p+1} - \mathcal{Z}_k^{p+1})$, $k = 1, 2, 3$; $\mathcal{P}^{p+1} = \mathcal{P}^p + \beta(\mathcal{Y} - (\mathcal{X}_k^{p+1} + \mathcal{N}_k^{p+1} + \mathcal{S}_k^{p+1}))$.

Let $\mu = \rho\mu$; $\beta = \rho\beta$; $p = p + 1$.

Check the convergence condition $\|\mathcal{X}^{(p+1)} - \mathcal{X}^{(p)}\|_F / \|\mathcal{X}^{(p)}\|_F < 10^{-4}$.

end while

Output: The restored HSI \mathcal{X} .



Computational Cost and Convergence

Computational cost:

$$\mathcal{O}\left(n_1 n_2 n_3 \left(\log(n_1 n_2 n_3) + \sum_{i=1}^3 \min(n_i, n_{i+1}) \right)\right), \text{ where } n_4 = n_1.$$



Computational Cost and Convergence

Computational cost:

$$\mathcal{O}\left(n_1 n_2 n_3 \left(\log(n_1 n_2 n_3) + \sum_{i=1}^3 \min(n_i, n_{i+1}) \right)\right), \text{ where } n_4 = n_1.$$

Convergence:

guaranteed theoretically \Leftarrow convex optimization problem



Outline

1 Introduction

2 The Proposed Model and Algorithm

3 Numerical Experiments



Compared Methods

Compared Methods:

- BM4D+TRPCA [*Maggioni et al., IEEE TIP 2012; Lu et al., CVPR 2016*];
- SSTV [*Aggarwal and Majumdar, IEEE GRSL 2016*];
- LRMR [*Zhang et al., IEEE TGRS 2016*];
- LRTR [*Fan et al., IEEE JSTARS 2017*].



Case 1: different Gaussian noise, fixed impulse noise, and fixed stripe noise.

Table 2: The performance comparison of five competing methods with respect to different Gaussian noise levels.

Case	Case 1														
	$\sigma = 0.02$			$\sigma = 0.06$			$\sigma = 0.10$								
Gaussian noise															
Impulse noise	proportion $v = 0.2$														
Stripes	added to 10 bands and proportion $p = 10\%$.														
Method	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM						
Noise	11.373	0.1212	47.389	11.188	0.1137	48.025	10.839	0.1023	49.172						
TRPCA+BM4D	38.798	0.9790	3.7193	33.657	0.9342	5.8150	30.991	0.8821	7.3463						
SSTV	39.043	0.9754	4.3674	34.377	0.9326	6.6053	31.251	0.8734	8.8027						
LRMR	35.196	0.9488	5.6839	33.653	0.9301	6.8313	31.516	0.8952	8.6890						
LRTR	36.479	0.9629	5.1349	33.928	0.9331	6.2357	30.968	0.8923	8.4193						
3DTNN	41.658	0.9920	1.8010	35.554	0.9655	3.9101	32.398	0.9317	5.5411						



Case 2: **fixed Gaussian noise, different impulse noise, and fixed stripe noise.**

Table 3: The performance comparison of five competing methods with respect to **different impulse noise levels**.

Case	Case 2								
Gaussian noise	$\sigma = 0.02$								
Impulse noise	$v = 0.1$			$v = 0.3$			$v = 0.4$		
Stripes	added to 10 bands and proportion $p = 10\%$.								
Method	PSNR	SSIM	SAM	PSNR	SSIM	SAM	PSNR	SSIM	SAM
Noise	14.357	0.2531	41.766	9.6182	0.0718	49.704	8.3756	0.0470	50.771
TRPCA+BM4D	39.900	<u>0.9832</u>	<u>3.2894</u>	37.273	<u>0.9708</u>	<u>4.4344</u>	33.336	0.9240	7.0538
SSTV	40.239	0.9804	4.0178	37.839	0.9682	4.8038	36.336	<u>0.9562</u>	<u>5.4216</u>
LRMR	38.597	0.9730	3.8940	32.704	0.9189	7.4550	30.588	0.8819	9.2499
LRTR	38.663	0.9741	3.7062	34.617	0.9428	6.2333	31.113	0.8717	9.2404
3DTNN	42.794	0.9937	1.6046	40.345	0.9897	2.0145	38.629	0.9856	2.3506



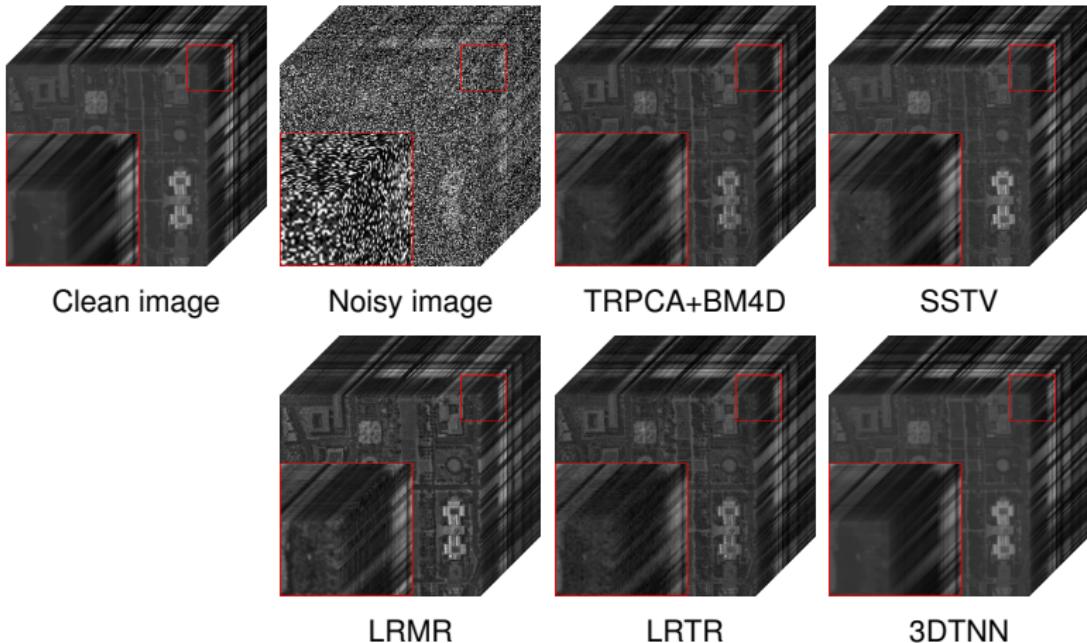


Figure 6: The three dimensional visualization of the denoising results for Gaussian noise with $\sigma = 0.02$ and impulse noise with $v = 0.4$.



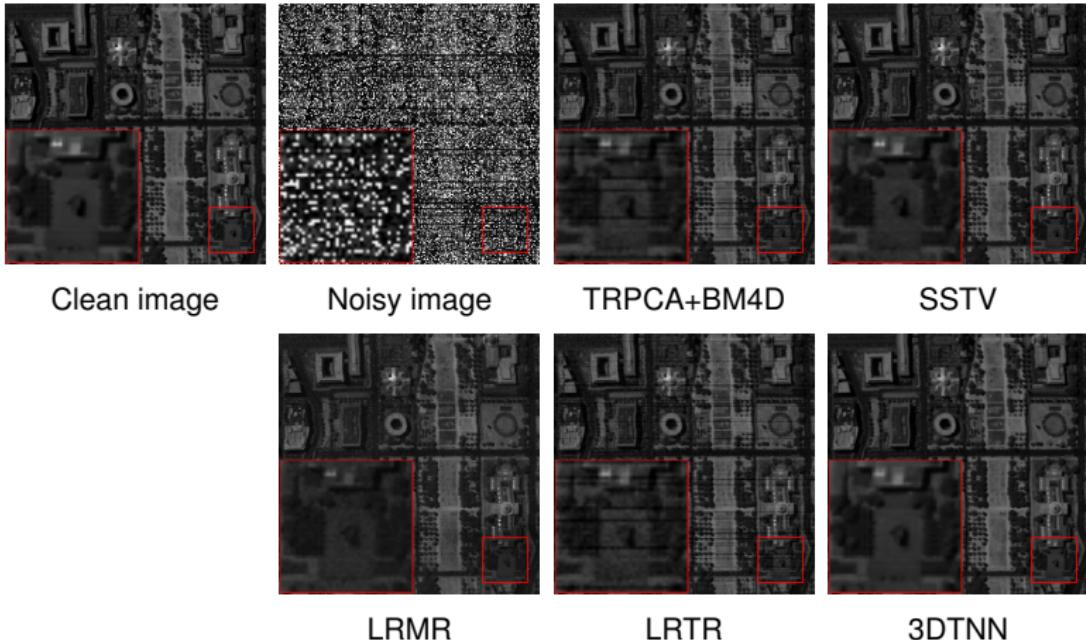


Figure 7: The denoising results at band 131 for Gaussian noise with $\sigma = 0.02$ and impulse noise with $v = 0.4$.



Thank you very much for listening.



Homepage



Wechat

