Robust Low-Rank Tensor Completion via Transformed Tensor Nuclear Norm with Total Variation Regularization

Duo Qiu[*], Minru Bai[†], Michael K. Ng[‡], and Xiongjun Zhang[§]

February 18, 2021

Abstract

Robust low-rank tensor completion plays an important role in multidimensional data analysis against different degradations, such as Gaussian noise, sparse noise, and missing entries, and has a variety of applications in image processing and computer vision. In this paper, we investigate the problem of low-rank tensor completion with different degradations for third-order tensors, and propose a transformed tensor nuclear norm method combined the tensor ℓ_1 norm with total variational (TV) regularization. Our model is based on a recently proposed algebraic framework in which the transformed tensor nuclear norm is introduced to capture lower transformed multi-rank by using suitable unitary transformations. We adopt the tensor ℓ_1 norm to detect the sparse noise, and the TV regularization to preserve the piecewise smooth structure along the spatial and tubal dimensions. Moreover, a symmetric Gauss-Seidel based alternating direction method of multipliers is developed to solve the resulting model and its global convergence is established under very mild conditions. Extensive numerical examples on both hyperspectral images and video datasets are carried out to demonstrate the superiority of the proposed model compared with several existing state-of-the-art methods.

Key Words: Low-rank tensor completion, transformed tensor nuclear norm, mixed noise, total variation regularization

Mathematics Subject Classification 2010: 15A69, 68U10, 90C25

1 Introduction

Tensors, which are higher-order generalizations of vectors and matrices, are a useful tool for data analysis and dimensionality reduction [20]. Owing to the low dimensional structure of the underlying tensor data, the low-rank tensor is used to investigate the high-order data, and has received much attention in

[*]School of Mathematics, Hunan University, Changsha 410082, China (e-mail: qiuduoduo13@hnu.edu.cn).

[†]Corresponding author. School of Mathematics, Hunan University, Changsha 410082, China (e-mail: minrubai@163.com). Research supported in part by the National Natural Science Foundation of China under grant 11971159.

[‡]Department of Mathematics, The University of Hong Kong, Pokfulam, Hong Kong (e-mail: mng@maths.hku.hk). Research supported in part by the HKRGC GRF 12306616, 12200317, 12300218, 12300519 and 17201020, and HKU Grant 104005583.

[§]School of Mathematics and Statistics and Hubei Key Laboratory of Mathematical Sciences, Central China Normal University, Wuhan 430079, China (e-mail: xjzhang@mail.ccnu.edu.cn). Research supported in part by the National Natural Science Foundation of China under grants 11801206, 11871025, Hubei Provincial Natural Science Foundation of China under grant 2018CFB105, and Fundamental Research Funds for the Central Universities under grant CCNU19ZN017.

various application fields such as hyperspectral image processing [43, 45, 46, 53], machine learning [39], and computer vision [23, 26, 48]. Among these applications, the underlying low-rank tensor may be corrupted by different noises [32, 43, 46] and only partially observed entries are available. In this paper, we mainly study the problem of recovering an unknown low-rank third-order tensor from partial observations with noises (named as robust low-rank tensor completion), where the underlying low-rank tensor is degraded by Gaussian noise and sparse noise simultaneously.

The robust tensor completion problem reduces to robust matrix completion for second-order tensors, which has been intensively studied in the past decades, see [6–8, 19, 54] and references therein. The problem of robust matrix completion is to recover a low-rank matrix when part of its entries are not observed and some of the observed entries are corrupted. In particular, Candès et al. [7] studied the matrix robust principal component analysis and showed that one can recover a low-rank matrix and a sparse matrix exactly with overwhelming probability under incoherence conditions, where the observed data matrix is the superposition of a low-rank component and a sparse component. Zhou et al. [54] studied the problem of recovering a low-rank matrix from a high-dimensional data matrix despite both small entry-wise noise and gross sparse errors, and proved that, by solving a convex programm, one can recover a low-rank matrix with an error bound proportional to the noise level under certain incoherence conditions even though a positive fraction of its entries are arbitrarily corrupted. Moreover, Klopp et al. [19] investigated robust matrix completion from partial and corrupted observations in the presence of Gaussian and sparse noises, and analyzed the error bounds of the solutions of a constrained convex optimization problem consisted of the nuclear norm penalty for the underlying matrix and a convex relaxation penalty for the sparse constraint.

Compared with robust matrix completion, robust tensor completion is more difficult due to the complexity of the rank of a tensor. The widely used ranks of a tensor are CANDECOMP/PARAFAC (CP) rank [9], Tucker rank [41], tensor train (TT) rank [33], and tubal rank and multi-rank [18]. In general, computing the CP rank of a tensor is NP-hard [13]. Also the convex relaxation of CP rank is intractable, which makes the low CP rank tensor recovery challenging. There have been great efforts for robust tensor completion via Tucker rank minimization. Liu et al. [25] first proposed the sum of nuclear norms of unfolding matrices (SNN) of a tensor to approximate the Tucker rank minimization. The SNN can explore the intrinsic structure of the tensor data compared with the matrix based methods. Furthermore, Goldfarb et al. [12] proposed a model combined the SNN with the ℓ_1 norm for robust tensor recovery, where missing entries and gross corruptions were considered. Although the SNN is easy to compute, Romera-Paredes et al. [35] showed that it is not the convex envelope of the sum of entries of the Tucker rank of a tensor. And the SNN is to minimize the sum of nuclear norms of the unbalanced matrices, which will influence its recovery performance. Moreover, Mu et al. [29] also showed that the SNN is just suboptimal for tensor completion. For TT rank minimization, Bengua et al. [4] proposed a TT method by using a well-balanced matricization scheme, which was able to capture hidden information from tensors. While the TT method needs to reshape the original tensor into a higher-order tensor and is just effective for some special sizes of a tensor, which may destroy the intrinsic structure of the original tensor and limit the applications of the tensor with general sizes.

Recently, Kilmer et al. [18] proposed a new algebra operator for third-order tensors and then Martin et al. [28] generalized it to higher-order tensors. Moreover, the tensor singular value decomposition (SVD) and the tubal rank and multi-rank of a tensor were proposed based on the tensor-tensor product in the Fourier domain [17]. In order to maintain the intrinsic structure of the tensor data, Semerci et al. [38] proposed a tensor nuclear norm method to approximate the tensor tubal rank minimization problem, where Zhang et al. [52] showed that the tensor nuclear norm is the convex envelope of the sum of entries of its multi-rank. However, the approach in [38] was not able to address the observations with mixed noise and missing entries. Based on the tubal rank and multi-rank, many researchers studied the tensor completion problem and tensor robust principal component analysis, see [14, 15, 24, 27, 38,

50, 51] and references therein. For example, Jiang et al. [15] proposed a unified framework for tensor completion [51] and tensor robust principal component analysis [27], i.e., robust tensor completion, and showed that one can recover a low tubal rank tensor exactly from partially noisy observations with overwhelming probability. However, the previous methods may suffer from disadvantage due to the limitation of Fourier transform and cannot address Gaussian noise and sparse noise simultaneously, where the tensor-tensor product and tensor SVD are performed by Fourier transform. Recently, Song et al. [40] proposed a unitary transform method for robust tensor completion by using transformed tensor nuclear norm (TTNN) and transformed tensor SVD, and also analyzed its exact recovery under some mild assumptions. Compared with the tensor nuclear norm in [38], the TTNN may get a lower tubal rank and multi-rank tensor with suitable unitary transformations [40]. However, many real-word tensors may be corrupted by mixed noise including Gaussian noise and sparse noise, and the underlying tensors possess local smoothness in spatial and tubal dimensions, such as videos and hyperspectral images. Moreover, the method by using transformed tensor SVD in [40] was just used to remove sparse noise from partial observations, and did not consider the prior information of imaging data.

On the other hand, there exist some methods to remove mixed noise for hyperspectral image restoration with full observations, see [1, 11, 43, 46, 47, 53] and references therein. For example, by vectorizing the spatial dimension of hyperspectral images, Yuan et al. [46] proposed a spectral-spatial adaptive total variation (TV) model, and Aggarwal et al. [1] exploited the inherent structure of hyperspectral images by utilizing two dimensional TV along the spatial dimension and one dimensional TV along the spectral dimension. Although the two methods utilized the TV regularization to explore the local smoothness of the underlying hyperspectral images, the unfolding methods will destroy the spatial information of hyperspectral images, and may be difficult to preserve the intrinsic structure of the underlying tensor data. Recently, Wang et al. [43] proposed to exploit the low-rankness of the hyperspectral images via Tucker decomposition, which can describe the global correlation among all bands of the hyperspectral images. While this method is nonconvex, and is difficult to get its global solution. More importantly, the above mentioned methods cannot deal with the problem when only partial entries are observed.

In this paper, we propose a novel method for robust tensor completion, which aims to recover a third-order tensor when some of its entries are not observed and all of its observed entries are corrupted by mixed noise including Gaussian noise and sparse noise. The proposed model consists of the data-fitting term combined the TTNN, the tensor ℓ_1 norm with TV regularization (TNTV for short). The TV regularization is used to investigate the local smoothness of the underlying image data along the spatial and tubal dimensions, which was first proposed by Rudin et al. [37] for image restoration with Gaussian noise removal, and then applied to other kinds of noises, see [3, 49] and references therein. In image processing, the TV regularization has been validated to keep more edge details of imaging data. Instead of using the traditional tensor nuclear norm in [38], the TTNN is used to explore the global low-rankness of a tensor, and may get a lower transformed multi-rank tensor by using suitable unitary transformations than the tensor nuclear norm method in the Fourier domain [40]. Besides, the tensor ℓ_1 norm is further used to enforce the sparsity of sparse noise. Moreover, we develop a symmetric Gauss-Seidel based alternating direction method of multipliers (sGS-ADMM) [10, 22] to solve the resulting TNTV model and establish its global convergence under very mild conditions. Numerical examples on hyperspectral images and video datasets are presented to verify the superiority of the proposed approach.

The contributions of this paper are summarized as follows. (i) We propose a TNTV model for robust tensor completion by incorporating the TTNN and TV regularization into the model, where the TTNN is used to explore the low-rankness, and the TV regularization is utilized to depict the local smoothness of the underlying tensor data along the spatial and tubal dimensions. (ii) An sGS-ADMM with convergence guarantee is developed to solve the TNTV model. (iii) Extensive numerical

experiments on hyperspectral images and video datasets are conducted to demonstrate the superior performance of the TNTV model compared with several existing methods.

The remaining parts of this paper are organized as follows. In the next section, some notation and notions about tensors are introduced, and the transformed tensor SVD is reviewed. In Section 3, we propose a TNTV model consisting of the TTNN, the tensor ℓ_1 norm, and TV regularization for robust tensor completion with different degradations. Then the sGS-ADMM is developed to solve the resulting model and its global convergence is established in Section 4. Extensive numerical experiments are shown to demonstrate the effectiveness of the proposed TNTV method for hyperspectral images and video datasets in Section 5. Finally, we conclude this paper with a brief discussion in Section 6.

2 Preliminaries

In this section, we give some notation and notions about tensors, and also review the transformed tensor SVD in [40]. Let $\mathbb{R}$ and $\mathbb{C}$ denote the sets of real and complex numbers, respectively. Scalars are denoted by lowercase letters, e.g., a. Vectors and matrices are denoted by lowercase boldface letters and capitalized boldface letters, respectively, e.g., $\mathbf{a}$ and $\mathbf{A}$. Tensors are denoted by capitalized calligraphic letters, e.g., $\mathcal{A}$. The order of a tensor is the number of its dimension, also known as ways or modes [20]. For a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its (i,j,k)th entry is denoted by $\mathcal{A}_{ijk}$. A tube of a third-order tensor $\mathcal{A}$, denoted by $\mathcal{A}(i,j,:)$, is a vector by fixing the first two indices and varying the third one. $\mathcal{A}(i,:,:), \mathcal{A}(:,i,:), \mathcal{A}(:,:,i)$ are used to denote the ith lateral, horizontal, and frontal slices of $\mathcal{A}$, respectively. For the sake of simplicity, the ith frontal slice of $\mathcal{A}$ is also denoted by $\mathbf{A}^{(i)}$.

For any matrix $\mathbf{A}$, $\|\mathbf{A}\|_*$ denotes the nuclear norm of $\mathbf{A}$, which is the sum of all singular values. The spectral norm of $\mathbf{A}$, denoted by $\|\mathbf{A}\|$, is its maximum singular value. The Frobenius norm of a matrix $\mathbf{A}$ is denoted by $\|\mathbf{A}\|_F$. $\mathbf{A}^H$ denotes the conjugate transpose of $\mathbf{A}$. The inner product between two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ is defined as

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i=1}^{n_3} \langle \mathbf{A}^{(i)}, \mathbf{B}^{(i)} \rangle,$$

where $\langle \mathbf{A}^{(i)}, \mathbf{B}^{(i)} \rangle$ is the standard inner product of two matrices. The tensor Frobenius norm of $\mathcal{A}$, denoted by $\|\mathcal{A}\|_F$, is defined as $\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. The tensor ℓ_1 norm of $\mathcal{A}$ is defined as $\|\mathcal{A}\|_1 = \sum_{i,j,k} |\mathcal{A}_{ijk}|$.

Let $g : \mathfrak{U} \to (-\infty, +\infty]$ be a proper and lower semicontinuous function, where $\mathfrak{U}$ is an arbitrary finite dimensional Euclidean space. Given any $y \in \mathfrak{U}$, the proximal mapping of g at y is defined by

$$\mathrm{Prox}_g(y) := \arg\min_{x \in \mathfrak{U}} \left\{ g(x) + \frac{1}{2}\|y - x\|^2 \right\}.$$

The indicator function of a set $\mathfrak{E}$ is defined by

$$\delta_{\mathfrak{E}}(\mathcal{X}) = \begin{cases} 0, & \text{if } \mathcal{X} \in \mathfrak{E}, \\ +\infty, & \text{otherwise.} \end{cases}$$

2.1 Transformed Tensor Singular Value Decomposition

In this subsection, we briefly review the transformed tensor SVD of a third-order tensor, which can be found in [40] for more details.

Denote $\mathbf{U} \in \mathbb{C}^{n_3 \times n_3}$ to be a unitary matrix. For any tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\widehat{\mathcal{A}}$ is denoted to be the tensor via multiplying by $\mathbf{U}$ to all tubes of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, i.e., $\widehat{\mathcal{A}}(i,j,:) = \mathbf{U}(\mathcal{A}(i,j,:)), i =$

$1, \ldots, n_1, j = 1, \ldots, n_2$. In order to describe the notation intuitively, $\widehat{\mathcal{A}}$ is also denoted by $\mathbf{U}[\mathcal{A}]$. Conversely, we could get $\mathcal{A}$ from $\widehat{\mathcal{A}}$ by $\mathcal{A} = \mathbf{U}^H[\widehat{\mathcal{A}}]$.

Next, we define the tensor-tensor product with respect to $\mathbf{U}$.

Definition 2.1 *[40, Definition 1] For any two tensors $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ and $\mathcal{Y} \in \mathbb{C}^{n_2 \times n_4 \times n_3}$, the $\mathbf{U}$-product, denoted by $\mathcal{X} \diamond_{\mathbf{U}} \mathcal{Y}$, is a tensor $\mathcal{Z} \in \mathbb{C}^{n_1 \times n_4 \times n_3}$ given by*

$$\mathcal{Z} = \mathcal{X} \diamond_{\mathbf{U}} \mathcal{Y} = \mathbf{U}^H \left[\text{Fold} \left(\text{Block}(\widehat{\mathcal{X}}) \cdot \text{Block}(\widehat{\mathcal{Y}}) \right) \right],$$

where

$$\text{Fold}(\text{Block}(\widehat{\mathcal{X}})) = \widehat{\mathcal{X}} \ \textit{and} \ \text{Block}(\widehat{\mathcal{X}}) = \begin{pmatrix} \widehat{\mathbf{X}}^{(1)} & & & \\ & \widehat{\mathbf{X}}^{(2)} & & \\ & & \ddots & \\ & & & \widehat{\mathbf{X}}^{(n_3)} \end{pmatrix}.$$

Before giving the definition of transformed tensor SVD, the definitions of the identity tensor, unitary tensor, diagonal tensor with respect to any unitary matrix $\mathbf{U}$ are listed as follows:

- The identity tensor with respect to $\mathbf{U}$, denoted by $\mathcal{I}_{\mathbf{U}} \in \mathbb{C}^{n \times n \times n_3}$, is defined to be a tensor such that $\mathcal{I}_{\mathbf{U}} = \mathbf{U}^H[\mathcal{I}]$, where each frontal slice $\mathbf{I}^{(i)}$ of the tensor $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$, $i = 1, \ldots, n_3$, is the $n \times n$ identity matrix [16, Proposition 4.1].
- The conjugate transpose of any tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ with respect to a unitary matrix $\mathbf{U} \in \mathbb{C}^{n_3 \times n_3}$, denoted by $\mathcal{A}^H \in \mathbb{C}^{n_2 \times n_1 \times n_3}$, is defined as $\mathcal{A}^H = \mathbf{U}^H[\text{Fold}(\text{Block}(\widehat{\mathcal{A}})^H)]$ [40, Definition 2].
- The unitary tensor with respect to any unitary matrix $\mathbf{U}$ is defined as $\mathcal{U}^H \diamond_{\mathbf{U}} \mathcal{U} = \mathcal{U} \diamond_{\mathbf{U}} \mathcal{U}^H = \mathcal{I}_{\mathbf{U}}$ [16, Definition 5.1].
- A tensor is called to be diagonal if each frontal slice is a diagonal matrix [18].

By the $\mathbf{U}$-product of two tensors, the transformed tensor SVD of a tensor can be stated in the following theorem.

Theorem 2.1 *[16, Theorem 5.1] For any $\mathcal{Z} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, the transformed tensor SVD is given by*

$$\mathcal{Z} = \mathcal{U} \diamond_{\mathbf{U}} \mathcal{S} \diamond_{\mathbf{U}} \mathcal{V}^H, \tag{2.1}$$

where $\mathcal{U} \in \mathbb{C}^{n_1 \times n_1 \times n_3}$, $\mathcal{V} \in \mathbb{C}^{n_2 \times n_2 \times n_3}$ are unitary tensors with respect to $\mathbf{U}$-product, and $\mathcal{S} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ is diagonal.

The transformed tensor SVD could be implemented efficiently by the SVDs of the frontal slices in the transformed domain. We also refer the reader to [40, Algorithm 1] for more details about the computation of transformed tensor SVD.

Definition 2.2 *[40, Definition 6] The transformed multi-rank of a tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ is a vector $\mathbf{r} := (r_1, \ldots, r_{n_3})^T \in \mathbb{R}^{n_3}$, where $r_i = \text{rank}(\widehat{\mathbf{A}}^{(i)})$, $i = 1, \ldots, n_3$.*

Definition 2.3 *[40, Definition 7] The transformed tensor nuclear norm of a tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, denoted by $\|\mathcal{A}\|_{\text{TTNN}}$, is defined as the sum of nuclear norms of all frontal slices $\widehat{\mathbf{A}}^{(i)}$ of $\widehat{\mathcal{A}}$, i.e.,*

$$\|\mathcal{A}\|_{\text{TTNN}} = \sum_{i=1}^{n_3} \|\widehat{\mathbf{A}}^{(i)}\|_*.$$

Recently, Song et al. showed that the transformed tensor nuclear norm (TTNN) of a tensor is the convex envelope of the sum of entries of the transformed multi-rank over the unit ball of the tensor spectral norm [40, Lemma 1], where the tensor spectral norm of $\mathcal{A}$ is defined as the spectral norm of the block diagonal matrix Block($\widehat{\mathcal{A}}$). In this case, the TTNN could be used to approximate the sum of entries of the transformed multi-rank of a tensor.

3 Robust Tensor Completion by TTNN with TV Regularization

In this section, we propose a TNTV model for robust tensor completion from partial observations with Gaussian noise and sparse noise, which combined the TTNN, tensor ℓ_1 norm, and TV regularization. First, the problem of the observed model is formulated.

3.1 Problem Formulation

For a real-world tensor in image processing, it may be corrupted by sparse noise, such as impulse noise, and also corrupted by Gaussian noise [46, 53], which implies that the underlying low-rank tensor may be corrupted by sparse noise and Gaussian noise simultaneously. Given the noisy tensor $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, only partial entries of $\mathcal{Y}$ are observed. Then the resulting degradation model can be described as

$$\mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{P}_\Omega(\mathcal{L} + \mathcal{S} + \mathcal{N}), \tag{3.2}$$

where $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the underlying low-rank tensor, $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{N} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ are the sparse noise and Gaussian noise, respectively, $\Omega \subset \{1, 2, \ldots, n_1\} \times \{1, 2, \ldots, n_2\} \times \{1, 2, \ldots, n_3\}$ is the index set, and $\mathcal{P}_\Omega$ is the projection operator onto Ω such that

$$(\mathcal{P}_\Omega(\mathcal{X}))_{ijk} = \begin{cases} \mathcal{X}_{ijk}, & \text{if } (i,j,k) \in \Omega, \\ 0, & \text{otherwise.} \end{cases}$$

Our aim is to recover the underlying low-rank tensor $\mathcal{L}$ from $\mathcal{P}_\Omega(\mathcal{Y})$.

3.2 The TNTV Model with TV Regularization

In order to stabilize the recovery of $\mathcal{L}$, some prior information on the underlying tensor should be utilized. Since the TTNN has the ability to exploit the global low-rankness for a third-order tensor [40], we use the TTNN to get a low transformed multi-rank tensor. Compared with tensor nuclear norm in [38, 52], the TTNN could generate a lower transformed multi-rank tensor with suitable unitary transformations. Moreover, to explore the local smoothness of the underlying low-rank imaging data, the TV is enforced to preserve sharp edges, which was first proposed by Rudin et al. [37] for grey image denoising, and then generalized to image deconvolution [36]. Being different from the traditional TV, we use the finite differences of the imaging data along the spatial and tubal dimensions. For the sparse noise $\mathcal{S}$, the tensor ℓ_1 norm regularization is enforced to exploit a sparse solution and remove sparse noise, such as impulse noise. Furthermore, the tensor Frobenius norm of the difference between $\mathcal{P}_\Omega(\mathcal{Y})$ and $\mathcal{P}_\Omega(\mathcal{L} + \mathcal{S})$ is used to remove Gaussian noise in the observation model. Therefore, we propose to incorporate the TV regularization into robust tensor completion by combining the TTNN with the tensor ℓ_1 norm (TNTV for short) as follows:

$$\min_{\mathcal{L},\mathcal{S}} \|\mathcal{L}\|_{\text{TTNN}} + \gamma\|\mathcal{L}\|_{\text{TV}} + \lambda\|\mathcal{S}\|_1 + \frac{\rho}{2}\|\mathcal{P}_\Omega(\mathcal{Y} - \mathcal{L} - \mathcal{S})\|_F^2, \tag{3.3}$$

where $\gamma, \lambda, \rho > 0$ are regularization parameters. Here the anisotropic TV for a third-order tensor is defined as

$$\|\mathcal{L}\|_{\text{TV}} = \|D\mathcal{L}\|_1 = \|D_h\mathcal{L}\|_1 + \|D_v\mathcal{L}\|_1 + \|D_t\mathcal{L}\|_1,$$

where D_h, D_v, D_t denote the first-order forward finite-difference operators along the horizontal, vertical, and tubal directions, respectively. Here, for the (i, j, k)th entry, each forward finite-difference operator is defined by

$$\begin{cases} D_h\mathcal{X} = \mathcal{X}(i, j+1, k) - \mathcal{X}(i, j, k), \\ D_v\mathcal{X} = \mathcal{X}(i+1, j, k) - \mathcal{X}(i, j, k), \\ D_t\mathcal{X} = \mathcal{X}(i, j, k+1) - \mathcal{X}(i, j, k), \end{cases}$$

where the period boundary conditions are used for the finite-difference operators, i.e., $\mathcal{X}(i, 1, k) - \mathcal{X}(i, n_2, k), \mathcal{X}(1, j, k) - \mathcal{X}(n_1, j, k), \mathcal{X}(i, j, 1) - \mathcal{X}(i, j, n_3)$ are used for the differences on the boundary.

The TV regularization along three different directions is used to investigate the local smoothness along the spatial and tubal dimensions in robust tensor completion. For example, the TV could account for both the spatial and spectral correlations for hyperspectral images [43]. For video data, the TV could not only explore the spatial smoothness, but also preserve locally smooth structure of videos along the time dimension.

Compared with the tensor nuclear norm based on Fourier transform in [15, 38, 52], the TTNN can exploit the low-rankness of a tensor better [40], which employs unitary transform matrices instead of discrete Fourier transform matrix in the tensor-tensor product and tensor SVD. A lower transformed multi-rank tensor may be obtained by using other unitary transform matrices than that by using discrete Fourier transform matrix. Therefore, the TTNN may obtain a lower transformed multi-rank solution than the tensor nuclear norm in the Fourier domain. We also refer the reader to [40] for a more detailed discussion about the superiority of the TTNN compared with the tensor nuclear norm based on Fourier transform.

Remark 3.1 *The TNTV is fundamentally different from the existing methods for robust tensor completion in the literature [12, 15]. In order to address sparse noise and missing entries, Goldfarb et al. [12] proposed an SNN based method for robust tensor completion, and Jiang et al. [15] proposed a model consisted of the tensor nuclear norm for the underlying tensor and tensor ℓ_1 norm for the sparse errors. However, our TNTV model can address Gaussian noise, sparse noise, and missing entries simultaneously. Moreover, we use the TTNN to explore the low-rankness of the underlying tensor, while the method in [12] used the SNN to depict its low-rankness. Compared with the tensor nuclear norm in [15], the TTNN can get a lower transformed multi-rank tensor by using suitable unitary transformations [40]. Besides, the TV regularization is enforced in the TNTV model along the spatial and tubal dimensions, where the TV can enhance the piecewise smoothness and preserve sharp edges for imaging data. However, the methods in [12, 15] do not consider this prior information of imaging data.*

Remark 3.2 *By vectorizing each frontal slice of the hyperspectral images into a vector along the spatial dimension and stacking these vectors into a matrix, Aggarwal et al. [1] and Yuan et al. [46] proposed TV regularization based models combined the nuclear norm of the underlying matrix with the ℓ_1 norm for the sparse errors. However, the two methods cannot deal with missing entries. Moreover, the matricization methods in [1, 46] may destroy the internal structure of the tensor data. Our TNTV method can address the tensor data directly, which can explore the correlation of the original data better. Besides, the TNTV can deal with missing entries.*

Remark 3.3 *Based on the tensor structure, Wang et al. [43] proposed a method by combining Tucker decomposition with the TV regularization, which can remove Gaussian noise and sparse noise simultaneously. But the method in [43] cannot address missing entries. Moreover, the model in [43] is*

nonconvex and it is difficult to get the global optimal solution, where the convergence of the algorithm for solving the nonconvex model is not given. However, we use the TTNN to explore the low-rankness of the underlying tensor, which maintains its internal structure. Also we show the globally convergence of the proposed algorithm under very mild conditions.

4 A Symmetric Gauss-Seidel Based ADMM

In this section, we develop an sGS-ADMM [22] to solve problem (3.3). The sGS-ADMM has been validated efficiently in various fields, e.g., semidefinite programming [10, 22], support vector machine [21, 42], and tensor completion [2, 40, 50].

Let $\mathcal{Z} = \mathcal{P}_\Omega(\mathcal{Y}) - \mathcal{L} - \mathcal{S}$, $\mathcal{F} = D\mathcal{M}$, $\mathcal{L} = \mathcal{M}$, then problem (3.3) can be reformulated as

$$\begin{aligned} \min_{\mathcal{L},\mathcal{F},\mathcal{S},\mathcal{M},\mathcal{Z}} \ & \|\mathcal{L}\|_{\text{TTNN}} + \gamma\|\mathcal{F}\|_1 + \lambda\|\mathcal{S}\|_1 + \frac{\rho}{2}\|\mathcal{P}_\Omega(\mathcal{Z})\|_F^2 \\ \text{s.t. } & \mathcal{L} + \mathcal{S} + \mathcal{Z} = \mathcal{P}_\Omega(\mathcal{Y}),\ \mathcal{F} = D\mathcal{M},\ \mathcal{L} = \mathcal{M}, \end{aligned} \tag{4.4}$$

where $D = [D_h, D_v, D_t]$ is the three-dimensional difference operator. Note that problem (4.4) is equivalent to

$$\begin{aligned} \min_{\mathcal{L},\mathcal{F},\mathcal{S},\mathcal{Z},\mathcal{M}} \ & (\|\mathcal{L}\|_{\text{TTNN}} + \gamma\|\mathcal{F}\|_1) + \lambda\|\mathcal{S}\|_1 + \frac{\rho}{2}\|\mathcal{P}_\Omega(\mathcal{Z})\|_F^2 \\ \text{s.t. } & \begin{pmatrix} \mathcal{I} & \mathbf{0} \\ \mathbf{0} & \mathcal{I} \\ \mathcal{I} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathcal{L} \\ \mathcal{F} \end{pmatrix} + \begin{pmatrix} \mathcal{I} & \mathbf{0} \\ \mathbf{0} & -D \\ \mathbf{0} & -\mathcal{I} \end{pmatrix} \begin{pmatrix} \mathcal{S} \\ \mathcal{M} \end{pmatrix} + \begin{pmatrix} \mathcal{I} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \mathcal{Z} = \begin{pmatrix} \mathcal{P}_\Omega(\mathcal{Y}) \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \end{aligned} \tag{4.5}$$

where $\mathcal{I}$ denotes the identity operator. Since there are three separable blocks in problem (4.5), the direct extended ADMM is not applicable. Fortunately, the function corresponding to block $\mathcal{Z}$ in the objective function of (4.5) is quadratic. And the functions corresponding to blocks $(\mathcal{L}, \mathcal{F})$ and $(\mathcal{S}, \mathcal{M})$ are nonsmooth. Therefore, we can develop an sGS-ADMM to solve problem (4.5).

For a given parameter $\beta > 0$, the augmented Lagrangian function associated with problem (4.5) is given by

$$\begin{aligned} & L_A(\mathcal{L}, \mathcal{F}, \mathcal{S}, \mathcal{M}, \mathcal{Z}, \mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3) \\ = & \|\mathcal{L}\|_{\text{TTNN}} + \gamma\|\mathcal{F}\|_1 + \lambda\|\mathcal{S}\|_1 + \frac{\rho}{2}\|\mathcal{P}_\Omega(\mathcal{Z})\|_F^2 + \langle \mathcal{T}_1, \mathcal{L} + \mathcal{S} + \mathcal{Z} - \mathcal{P}_\Omega(\mathcal{Y})\rangle + \langle \mathcal{T}_2, \mathcal{F} - D\mathcal{M}\rangle \\ & + \langle \mathcal{T}_3, \mathcal{L} - \mathcal{M}\rangle + \frac{\beta}{2}\left(\|\mathcal{L} + \mathcal{S} + \mathcal{Z} - \mathcal{P}_\Omega(\mathcal{Y})\|_F^2 + \|\mathcal{F} - D\mathcal{M}\|_F^2 + \|\mathcal{L} - \mathcal{M}\|_F^2\right), \end{aligned}$$

where $\mathcal{T}_i$ are the Lagrangian multipliers, $i = 1, 2, 3$. Now we give the framework of the sGS-ADMM algorithm for solving (4.5) based on the algorithm solving multi-block convex conic programming in [22]. Problem (4.5) can be viewed as a linearly constrained nonsmooth convex programming problem with three blocks of variables grouped as $(\mathcal{L}, \mathcal{F})$, $(\mathcal{S}, \mathcal{M})$, $\mathcal{Z}$. The iteration of the sGS-ADMM is given

as follows:

$$\mathcal{Z}^{k+\frac{1}{2}} = \arg\min_{\mathcal{Z}} \left\{ L_A(\mathcal{L}^k, \mathcal{F}^k, \mathcal{S}^k, \mathcal{M}^k, \mathcal{Z}, \mathcal{T}_1^k, \mathcal{T}_2^k, \mathcal{T}_3^k) \right\}, \tag{4.6}$$

$$(\mathcal{L}^{k+1}, \mathcal{F}^{k+1}) = \arg\min_{\mathcal{L},\mathcal{F}} \left\{ L_A(\mathcal{L}, \mathcal{F}, \mathcal{S}^k, \mathcal{M}^k, \mathcal{Z}^{k+\frac{1}{2}}, \mathcal{T}_1^k, \mathcal{T}_2^k, \mathcal{T}_3^k) \right\}, \tag{4.7}$$

$$\mathcal{Z}^{k+1} = \arg\min_{\mathcal{Z}} \left\{ L_A(\mathcal{L}^{k+1}, \mathcal{F}^{k+1}, \mathcal{S}^k, \mathcal{M}^k, \mathcal{Z}, \mathcal{T}_1^k, \mathcal{T}_2^k, \mathcal{T}_3^k) \right\}, \tag{4.8}$$

$$(\mathcal{S}^{k+1}, \mathcal{M}^{k+1}) = \arg\min_{\mathcal{S},\mathcal{M}} \left\{ L_A(\mathcal{L}^{k+1}, \mathcal{F}^{k+1}, \mathcal{S}, \mathcal{M}, \mathcal{Z}^{k+1}, \mathcal{T}_1^k, \mathcal{T}_2^k, \mathcal{T}_3^k) \right\}, \tag{4.9}$$

$$\mathcal{T}_1^{k+1} = \mathcal{T}_1^k + \tau\beta(\mathcal{L}^{k+1} + \mathcal{S}^{k+1} + \mathcal{Z}^{k+1} - \mathcal{P}_\Omega(\mathcal{Y})), \tag{4.10}$$

$$\mathcal{T}_2^{k+1} = \mathcal{T}_2^k + \tau\beta(\mathcal{F}^{k+1} - D\mathcal{M}^{k+1}), \tag{4.11}$$

$$\mathcal{T}_3^{k+1} = \mathcal{T}_3^k + \tau\beta(\mathcal{L}^{k+1} - \mathcal{M}^{k+1}), \tag{4.12}$$

where $\tau \in (0, \frac{1+\sqrt{5}}{2})$ is the dual step-length. Next we solve the subproblems with respect to the three blocks $(\mathcal{L}, \mathcal{F})$, $(\mathcal{S}, \mathcal{M})$, $\mathcal{Z}$ in detail.

i) Since $\mathcal{L}, \mathcal{F}$ are separable, the optimal solution of L_A with respect to $(\mathcal{L}, \mathcal{F})$ in (4.7) can be reformulated as

$$\mathcal{L}^{k+1} = \arg\min_{\mathcal{L}} \left\{ \|\mathcal{L}\|_{\text{TTNN}} + \beta \left\| \mathcal{L} - \frac{1}{2}\left(\mathcal{P}_\Omega(\mathcal{Y}) + \mathcal{M}^k - \mathcal{S}^k - \mathcal{Z}^{k+\frac{1}{2}} - \frac{1}{\beta}\left(\mathcal{T}_1^k + \mathcal{T}_3^k \right) \right) \right\|_F^2 \right\} \tag{4.13}$$

and

$$\mathcal{F}^{k+1} = \arg\min_{\mathcal{F}} \left\{ \gamma\|\mathcal{F}\|_1 + \frac{\beta}{2} \left\| \mathcal{F} - \left(D\mathcal{M}^k - \frac{1}{\beta}\mathcal{T}_2^k \right) \right\|_F^2 \right\}. \tag{4.14}$$

By [40, Theorem 3], the optimal solution of (4.13) can be given as follows:

$$\mathcal{L}^{k+1} = \mathcal{U} \diamond_{\mathbf{U}} \mathcal{S}_{\frac{1}{2\beta}} \diamond_{\mathbf{U}} \mathcal{V}^H, \tag{4.15}$$

where $\mathcal{U}, \mathcal{V}$ can be obtained by the following transformed tensor SVD

$$\frac{1}{2}\left(\mathcal{P}_\Omega(\mathcal{Y}) + \mathcal{M}^k - \mathcal{S}^k - \mathcal{Z}^{k+\frac{1}{2}} - \frac{1}{\beta}\left(\mathcal{T}_1^k + \mathcal{T}_3^k \right) \right) = \mathcal{U} \diamond_{\mathbf{U}} \mathcal{S} \diamond_{\mathbf{U}} \mathcal{V}^H,$$

and $\mathcal{S}_{\frac{1}{2\beta}} = \mathbf{U}^H[\widehat{\mathcal{S}}_{\frac{1}{2\beta}}]$ with $\widehat{\mathcal{S}}_{\frac{1}{2\beta}} = \max\{\widehat{\mathcal{S}} - \frac{1}{2\beta}, 0\}$.

Let $\mathcal{A} := D\mathcal{M}^k - \frac{1}{\beta}\mathcal{T}_2^k$. Then by the well-known soft-shrinkage operator, (4.14) can be written as

$$\mathcal{F}^{k+1} = \text{sgn}(\mathcal{A}) \circ \max\left\{ |\mathcal{A}| - \frac{\gamma}{\beta}, 0 \right\}, \tag{4.16}$$

where $\text{sgn}(x)$ denotes the sign of $x \neq 0$ and $\text{sgn}(0) = 0$, and $\circ$ denotes point-wise product. Here sgn and $|\cdot|$ are the point-wise operators.

ii) Note that $\mathcal{S}, \mathcal{M}$ are separable in L_A. Then the optimal solution of L_A with respect to $\mathcal{S}$ in (4.9) is given by

$$\mathcal{S}^{k+1} = \arg\min_{\mathcal{S}} \left\{ \lambda\|\mathcal{S}\|_1 + \frac{\beta}{2} \left\| \mathcal{S} - \left(\mathcal{P}_\Omega(\mathcal{Y}) - \mathcal{L}^{k+1} - \mathcal{Z}^{k+1} - \frac{1}{\beta}\mathcal{T}_1^k \right) \right\|_F^2 \right\}. \tag{4.17}$$

Let $\mathcal{Q} := \mathcal{P}_\Omega(\mathcal{Y}) - \mathcal{L}^{k+1} - \mathcal{Z}^{k+1} - \frac{1}{\beta}\mathcal{T}_1^k$. Then

$$\mathcal{S}^{k+1} = \mathrm{sgn}(\mathcal{Q}) \circ \max\left\{|\mathcal{Q}| - \frac{\lambda}{\beta}, 0\right\}. \tag{4.18}$$

For the solution with respect to $\mathcal{M}$, it needs to solve the following equation

$$\beta(\mathcal{I} + D^*D)\mathcal{M} = D^*(\mathcal{T}_2^k + \beta\mathcal{F}^{k+1}) + \mathcal{T}_3^k + \beta\mathcal{L}^{k+1}, \tag{4.19}$$

where D^* indicates the adjoint operator of D. By the block circulant structure of the matrix corresponding to the operator D^*D [31], it can be diagonalized by the three-dimensional fast Fourier transform (3DFFT) matrix. Then (4.19) can be solved explicitly by

$$\mathcal{M}^{k+1} = \mathrm{iFFT3}\left(\frac{\mathrm{FFT3}(D^*(\mathcal{T}_2^k + \beta\mathcal{F}^{k+1}) + \mathcal{T}_3^k + \beta\mathcal{L}^{k+1})}{\beta\mathbb{I} + \beta|\mathrm{FFT3}(D_h)|^2 + \beta|\mathrm{FFT3}(D_v)|^2 + \beta|\mathrm{FFT3}(D_t)|^2}\right), \tag{4.20}$$

where FFT3 and iFFT3 denote the 3DFFT and its inverse transform, respectively, $\mathbb{I}$ denotes a tensor with all entries being 1, and the division and square are performed in a point-wise manner.

iii) The optimal solution of L_A with respect to $\mathcal{Z}$ is given by

$$\mathcal{Z} = \frac{1}{\beta+\rho}\mathcal{P}_\Omega\Big(\beta\Big(\mathcal{P}_\Omega(\mathcal{Y}) - \mathcal{L} - \mathcal{S}\Big) - \mathcal{T}_1\Big) + \mathcal{P}_{\overline{\Omega}}\left(-\mathcal{L} - \mathcal{S} - \frac{1}{\beta}\mathcal{T}_1\right),$$

where $\overline{\Omega}$ is the complementary set of Ω.

Now we are ready to state the sGS-ADMM algorithm for solving problem (4.5) in Algorithm 1.

Algorithm 1: A Symmetric Gauss-Seidel Based Multi-Block ADMM for Solving (4.5).
Step 0. Let $\tau \in (0, (1+\sqrt{5})/2), \beta > 0$ be given constants. Given $\mathcal{L}^0, \mathcal{S}^0, \mathcal{M}^0, \mathcal{T}_i^0, i = 1, 2, 3$. Set $k := 0$.
Step 1. Compute $\mathcal{Z}^{k+\frac{1}{2}}$ via

$$\mathcal{Z}^{k+\frac{1}{2}} = \frac{1}{\beta+\rho}\mathcal{P}_\Omega\Big(\beta\Big(\mathcal{P}_\Omega(\mathcal{Y}) - \mathcal{L}^k - \mathcal{S}^k\Big) - \mathcal{T}_1^k\Big) + \mathcal{P}_{\overline{\Omega}}\left(-\mathcal{L}^k - \mathcal{S}^k - \frac{1}{\beta}\mathcal{T}_1^k\right).$$

Step 2. Compute $\mathcal{L}^{k+1}$ and $\mathcal{F}^{k+1}$ via (4.15) and (4.16), respectively.
Step 3. Compute $\mathcal{Z}^{k+1}$ by

$$\mathcal{Z}^{k+1} = \frac{1}{\beta+\rho}\mathcal{P}_\Omega\Big(\beta\Big(\mathcal{P}_\Omega(\mathcal{Y}) - \mathcal{L}^{k+1} - \mathcal{S}^k\Big) - \mathcal{T}_1^k\Big) + \mathcal{P}_{\overline{\Omega}}\left(-\mathcal{L}^{k+1} - \mathcal{S}^k - \frac{1}{\beta}\mathcal{T}_1^k\right).$$

Step 4. Compute $\mathcal{S}^{k+1}$ and $\mathcal{M}^{k+1}$ by (4.18) and (4.20), respectively.
Step 5. Update $\mathcal{T}_1^{k+1}$, $\mathcal{T}_2^{k+1}$, $\mathcal{T}_3^{k+1}$ via (4.10), (4.11), and (4.12), respectively.
Step 6. If a termination criterion is not satisfied, set $k := k+1$ and go to Step 1.

Notice that the two blocks with respect to $(\mathcal{L}, \mathcal{F})$ and $(\mathcal{S}, \mathcal{M})$ are nonsmooth and the block with respect to $\mathcal{Z}$ is quadratic in the objective function of (4.5). By [22, Theorem 3], we can show the convergence of sGS-ADMM easily, which is summarized in the following theorem.

Theorem 4.1 *Suppose that $\tau \in (0, (1+\sqrt{5})/2)$ and $\beta > 0$. Let the sequence $\{(\mathcal{L}^k, \mathcal{F}^k, \mathcal{S}^k, \mathcal{M}^k, \mathcal{Z}^k, \mathcal{T}_1^k, \mathcal{T}_2^k, \mathcal{T}_3^k)\}$ be generated by Algorithm 1. Then the sequence $\{(\mathcal{L}^k, \mathcal{F}^k, \mathcal{S}^k, \mathcal{M}^k, \mathcal{Z}^k)\}$ converges to an optimal solution to (4.5) and $\{(\mathcal{T}_1^k, \mathcal{T}_2^k, \mathcal{T}_3^k)\}$ converges to an optimal solution to the dual problem of (4.5).*

Proof. Denote

$$f(\mathcal{L},\mathcal{F},\mathcal{Z},\mathcal{S},\mathcal{M}) := \|\mathcal{L}\|_{\text{TTNN}} + \gamma\|\mathcal{F}\|_1 + \lambda\|\mathcal{S}\|_1 + \frac{\rho}{2}\|\mathcal{P}_\Omega(\mathcal{Z})\|_F^2 + \delta_{\mathfrak{E}}(\mathcal{L},\mathcal{F},\mathcal{S},\mathcal{M},\mathcal{Z}),$$

where $\mathfrak{E} := \{(\mathcal{L},\mathcal{F},\mathcal{S},\mathcal{M},\mathcal{Z}) : \mathcal{L}+\mathcal{S}+\mathcal{Z} = \mathcal{P}_\Omega(\mathcal{Y}),\ \mathcal{F} = D\mathcal{M},\ \mathcal{L} = \mathcal{M}\}$. It can be easily obtain that f is level bounded, i.e., $f(\mathcal{L},\mathcal{F},\mathcal{Z},\mathcal{S},\mathcal{M}) \to +\infty$ as $\|(\mathcal{L},\mathcal{F},\mathcal{Z},\mathcal{S},\mathcal{M})\|_F \to +\infty$. It follows from [34, Theorem 1.9] that the optimal solution set of (4.5) is nonempty and compact. Note that

$$\begin{pmatrix} \mathcal{I} & \mathbf{0} \\ \mathbf{0} & \mathcal{I} \\ \mathcal{I} & \mathbf{0} \end{pmatrix}^* \begin{pmatrix} \mathcal{I} & \mathbf{0} \\ \mathbf{0} & \mathcal{I} \\ \mathcal{I} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathcal{I} & \mathbf{0} \\ \mathbf{0} & \mathcal{I} \end{pmatrix}$$

and

$$\begin{pmatrix} \mathcal{I} & \mathbf{0} \\ \mathbf{0} & -D \\ \mathbf{0} & -\mathcal{I} \end{pmatrix}^* \begin{pmatrix} \mathcal{I} & \mathbf{0} \\ \mathbf{0} & -D \\ \mathbf{0} & -\mathcal{I} \end{pmatrix} = \begin{pmatrix} \mathcal{I} & \mathbf{0} \\ \mathbf{0} & D^*D+\mathcal{I} \end{pmatrix}$$

are positive definite operators. By [22, Theorem 3], we can easily obtain the conclusion of this theorem. □

5 Numerical Results

In this section, numerical experiments are presented to validate the effectiveness of the proposed TNTV method. We compare the TNTV with the singleton high-order robust principal component analysis (SHRPCA)[1] [12], robust tensor completion by tensor nuclear norm (TNN) [15], the spatio-spectral total variation regularization (SSTV)[2] [1], and the TV-regularized low-rank tensor decomposition (LRTDTV)[3] [43]. Since the SSTV and LRTDTV methods do not consider missing entries and are not effective for the case of missing entries, especially for low sampling ratios, we first employ the TNN method to fill the missing pixels and then apply the SSTV and LRTDTV to the intermediate result. For simplicity, they are also called SSVT and LRTDTV, respectively. All experiments are performed under a desktop computer running on 64-bit Windows Operating System (4 cores, Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz, 20 RAM).

For the TNTV approach, the transformations will influence its performance. In the experiments, we use three different transformations in the transformed tensor SVD. The first two are fast Fourier transform (FFT) and discrete cosine transform (DCT), respectively. The corresponding methods are called TNTV (FFT) and TNTV (DCT) for short. The third one is based on the given data [40], which is named as TNTV (Data). The detailed construction is given as follows: First, an initial estimator $\mathcal{L}_1$ of the underlying tensor $\mathcal{L}$ is generated, which can be obtained by TNTV (DCT). Then we unfold $\mathcal{L}_1$ into a matrix along the third dimension (denoted by $\mathbf{L}_{(3)}$). Let the SVD of $\mathbf{L}_{(3)}$ be $\mathbf{L}_{(3)} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$. We use $\mathbf{U}^H$ as the unitary matrix in the transformed tensor SVD. A lower transformed multi-rank tensor may be obtained by using this transformation. More details about the choice based on data for transformed tensor SVD can be referred to [40].

For the sampling ratio (SR) of observations, it is defined as SR $= \frac{|\Omega|}{n_1 n_2 n_3}$, where $|\Omega|$ denotes the cardinality of Ω and Ω is generated uniformly at random. The observed tensor is generated as follows: First, we add salt-and-pepper impulse noise with ratio κ, and then add zero-mean Gaussian noise with variance σ. Finally, we generate Ω with a given SR to get the observed tensor.

[1] https://sites.google.com/site/tonyqin/research

[2] https://sites.google.com/view/hkaggarwal/publications

[3] http://gr.xjtu.edu.cn/web/dymeng/3;jsessionid=A74F12A4182A91A7FEDF43CB18FEE27F

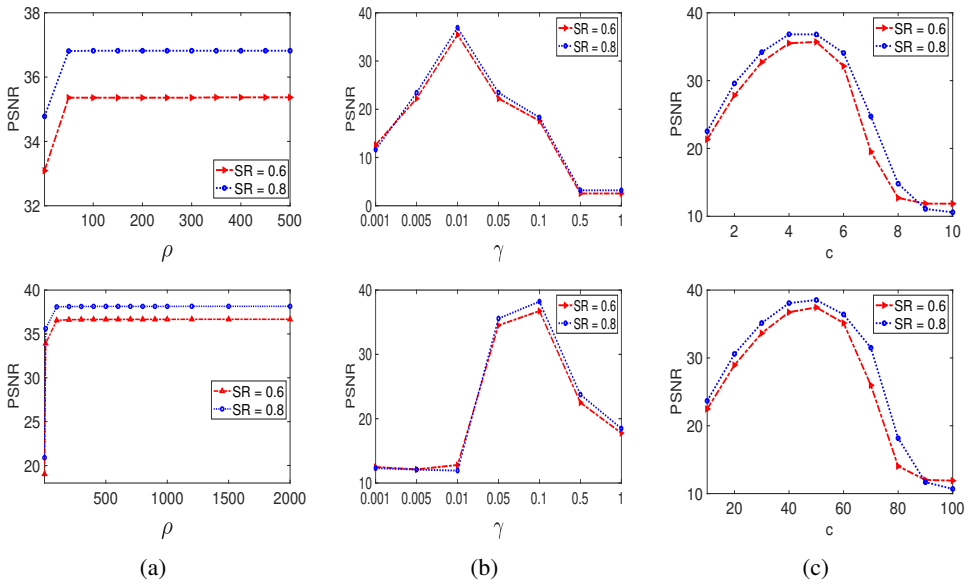


Figure 5.1: The performance of different parameters in TNTV (FFT) and TNTV (DCT) for the Samson dataset, where $\sigma = 0.01$ and $\kappa = 0.3$. First row: TNTV (FFT). Second row: TNTV (DCT).

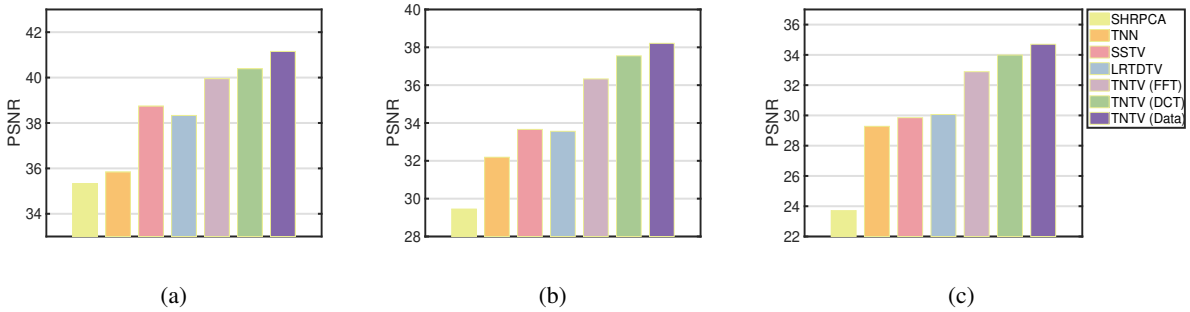


Figure 5.2: PSNR values of different methods for the Samson dataset with different impulse noise ratios, where SR $= 0.7$ and $\sigma = 0.01$. (a) $\kappa = 0.1$. (b) $\kappa = 0.3$. (c) $\kappa = 0.5$.

In order to evaluate the performance of different methods, the peak signal-to-noise ratio (PSNR) is used to measure the recovered tensor, and is defined as

$$\text{PSNR} = 10\log_{10}\frac{n_1 n_2 n_3(\max_{i,j,k}\mathcal{L}_{ijk} - \min_{i,j,k}\mathcal{L}_{ijk})^2}{\|\widetilde{\mathcal{L}} - \mathcal{L}\|_F^2},$$

where $\widetilde{\mathcal{L}}$ and $\mathcal{L} \in \mathbb{R}^{n_1\times n_2\times n_3}$ denote the recovered tensor and the ground-truth tensor, respectively. The structure similarity (SSIM) index [44] of the tensor denotes the mean of SSIM over all images.

The Karush-Kuhn-Tucker (KKT) conditions of problem (4.5) is given by

$$\begin{cases} 0 \in \partial\|\mathcal{L}\|_{\text{TTNN}} + \mathcal{T}_1 + \mathcal{T}_3,\ 0 \in \partial(\gamma\|\mathcal{F}\|_1) + \mathcal{T}_2,\ 0 \in \partial\lambda\|\mathcal{S}\|_1 + \mathcal{T}_1, \\ \rho\mathcal{P}_\Omega(\mathcal{Z}) + \mathcal{T}_1 = 0,\ D^*\mathcal{T}_2 + \mathcal{T}_3 = 0, \\ \mathcal{L} + \mathcal{S} + \mathcal{Z} - \mathcal{P}_\Omega(\mathcal{Y}) = 0,\ \mathcal{F} - D\mathcal{M} = 0,\ \mathcal{L} - \mathcal{M} = 0, \end{cases} \tag{5.21}$$

where $\partial f(x)$ denotes the subdifferential of f at x. Based on the KKT conditions in (5.21), we use the

following relative residuals to measure the accuracy of an approximate optimal solution:

$$\eta = \max\{\eta_l, \eta_f, \eta_s, \eta_z, \eta_m, \eta_{t_1}, \eta_{t_2}, \eta_{t_3}\},$$

where

$$\eta_l = \frac{\|\mathcal{L} - \text{Prox}_{\|\cdot\|_{\text{TTNN}}}(\mathcal{L} - \mathcal{T}_1 - \mathcal{T}_3)\|_F}{1 + \|\mathcal{L}\|_F + \|\mathcal{T}_1\|_F + \|\mathcal{T}_3\|_F}, \ \eta_f = \frac{\|\mathcal{F} - \text{Prox}_{\gamma\|\cdot\|_1}(\mathcal{F} - \mathcal{T}_2)\|_F}{1 + \|\mathcal{F}\|_F + \|\mathcal{T}_2\|_F},$$
$$\eta_s = \frac{\|\mathcal{S} - \text{Prox}_{\lambda\|\cdot\|_1}(\mathcal{S} - \mathcal{T}_1)\|_F}{1 + \|\mathcal{S}\|_F + \|\mathcal{T}_1\|_F}, \ \eta_z = \frac{\|\rho\mathcal{P}_\Omega(\mathcal{Z}) + \mathcal{T}_1\|_F}{1 + \|\rho\mathcal{P}_\Omega(\mathcal{Z})\|_F + \|\mathcal{T}_1\|_F}, \ \eta_m = \frac{\|D^*\mathcal{T}_2 + \mathcal{T}_3\|_F}{1 + \|D^*\mathcal{T}_2\|_F + \|\mathcal{T}_3\|_F},$$
$$\eta_{t_1} = \frac{\|\mathcal{L} + \mathcal{S} + \mathcal{Z} - \mathcal{P}_\Omega(\mathcal{Y})\|_F}{1 + \|\mathcal{P}_\Omega(\mathcal{Y})\|_F}, \ \eta_{t_2} = \frac{\|\mathcal{F} - D\mathcal{M}\|_F}{1 + \|\mathcal{F}\|_F + \|D\mathcal{M}\|_F}, \ \eta_{t_3} = \frac{\|\mathcal{L} - \mathcal{M}\|_F}{1 + \|\mathcal{L}\|_F + \|\mathcal{M}\|_F}.$$

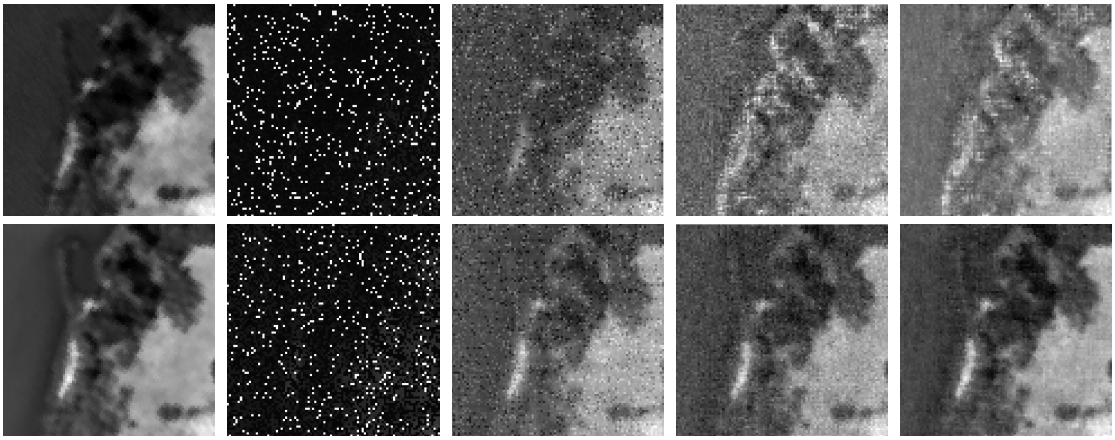
We terminate Algorithm 1 once $\eta \leq 5 \times 10^{-3}$ or the number of iterations reaches the maximum of 500.

5.1 The Setting of Parameters

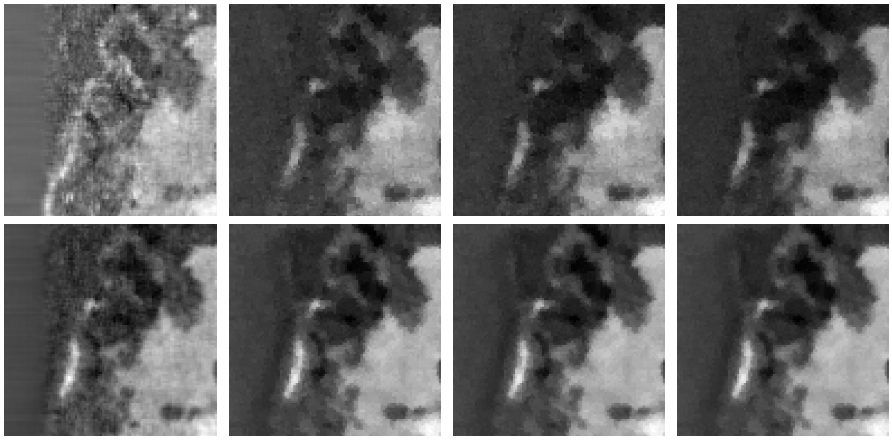
Let $\eta_p = \max\{\eta_l, \eta_f, \eta_s, \eta_z, \eta_m\}$ and $\eta_d = \max\{\eta_{t_1}, \eta_{t_2}, \eta_{t_3}\}$. In order to improve the convergence speed of Algorithm 1, we adapt the following strategy to update β in practice, where similar strategies are also employed in [21] and [50]. Set $\vartheta = \frac{\eta_p}{\eta_d}$. If $\vartheta > \theta$, set $\beta^{k+1} = \xi\beta^k$; elseif $\frac{1}{\vartheta} > \theta$, set $\beta^{k+1} = \frac{1}{\xi}\beta^k$; otherwise, $\beta^{k+1} = \beta^k$. In our experiments, the initial β^0 is set to be 0.1 for TNTV (FFT) and 1 for TNTV (DCT) and TNTV (Data). θ and ξ are set to be 50 and 1.2, respectively. τ is set to be 1.618 for the convergence of the sGS-ADMM.

Now we analyze the parameters γ, λ, ρ in the model (4.5). All parameters of TNTV (DCT) and TNTV (Data) are set the same in the experiments. Therefore, we just analyze the robust of different parameters in TNTV (FFT) and TNTV (DCT). The parameter ρ controls the Gaussian noise term and we test different ρ for the Samson dataset (see in the next subsection) with SR $= 0.6$ and 0.8, respectively, where the testing image is corrupted by impulse noise with $\kappa = 0.3$ and Gaussian noise with variance $\sigma = 0.01$. We vary ρ in the set $\{1, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500\}$ in TNTV (FFT) and $\{1, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1200, 1500, 2000\}$ in TNTV (DCT), respectively. In Figure 5.1(a), we show the PSNR values versus different ρ in the testing experiments. It can be seen that both TNTV (FFT) and TNTV (DCT) are robust when ρ is slightly larger. Therefore, we set ρ to be 400 for TNTV (FFT) and 1000 for TNTV (DCT) in all experiments, respectively.

The parameter γ is used to balance the TV regularization, which dominates the local smoothness of the underlying tensor. In Figure 5.1(b), we plot the PSNR values of TNTV (FFT) and TNTV (DCT) versus γ within the range of $[0.001, 1]$. It can be observed that the γ of TNTV (FFT) is smaller than that of TNTV (DCT) for stable recovery performance. In all experiments, we set γ to be 0.01 for TNTV (FFT) and 0.1 for TNTV (DCT), respectively. λ restrains the sparsity of the sparse noise. As suggested in [40] for robust tensor completion, we set $\lambda = \frac{c}{\sqrt{n_3 \max\{n_1, n_2\}\text{SR}}}$ for an $n_1 \times n_2 \times n_3$ tensor and adjust c slightly. In Figure 5.1(c), we test different c for TNTV (FFT) and TNTV (DCT) to see the sensitivity of c for the recovery performance, where c is changed from 1 to 10 with a step size of 1 for TNTV (FFT), and from 10 to 100 with a step size of 10 for TNTV (DCT). It can be observed that the PSNR increases gradually and then declines fast when c is larger for different SR. The PSNR values research the peaks when c reaches 4 for TNTV (FFT) and 50 for TNTV (DCT), respectively. This demonstrates that the parameter c is relatively robust within a certain range. We will adjust c around these ranges for TNTV (FFT) and for TNTV (DCT). For c in TNTV (FFT), we choose it from $\{4, 5, 6, 7, 12\}$ in all experiments to get the highest PSNR values. For c in TNTV (DCT), we choose it from $\{35, 40, 45, 85, 120, 125\}$ for hyperspectral image datasets, and from $\{60, 65, 70, 75, 80\}$ for video datasets.

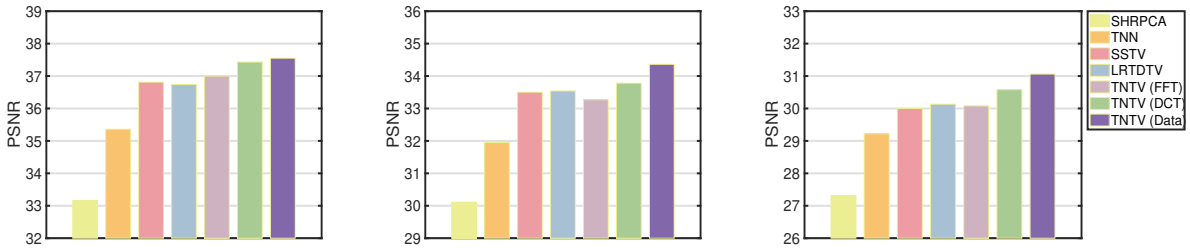


(a) Original (b) Observation (c) SHRPCA (d) TNN (e) SSTV



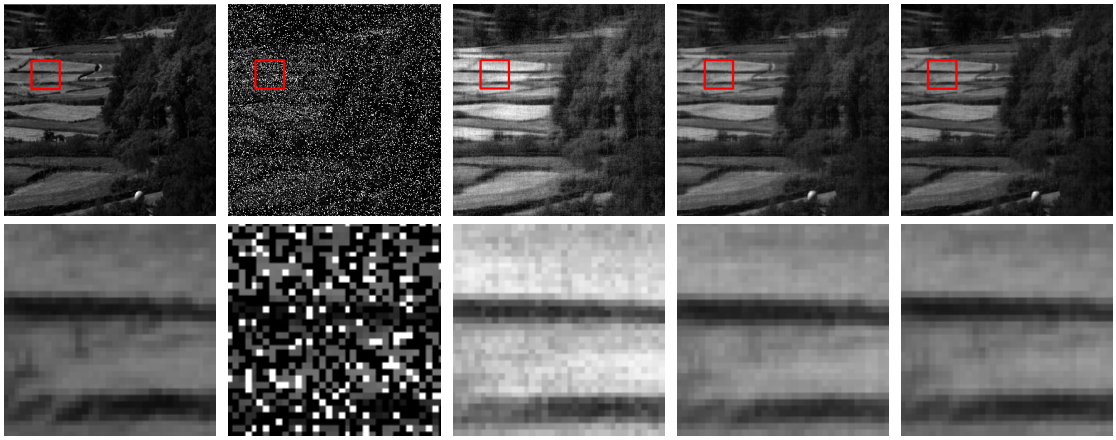
(f) LRTDTV (g) TNTV (FFT) (h) TNTV (DCT) (i) TNTV (Data)

Figure 5.3: Recovered images of different methods for the 5th and 40th bands of the Samson dataset, where SR $= 0.6, \sigma = 0.01$, and $\kappa = 0.2$.

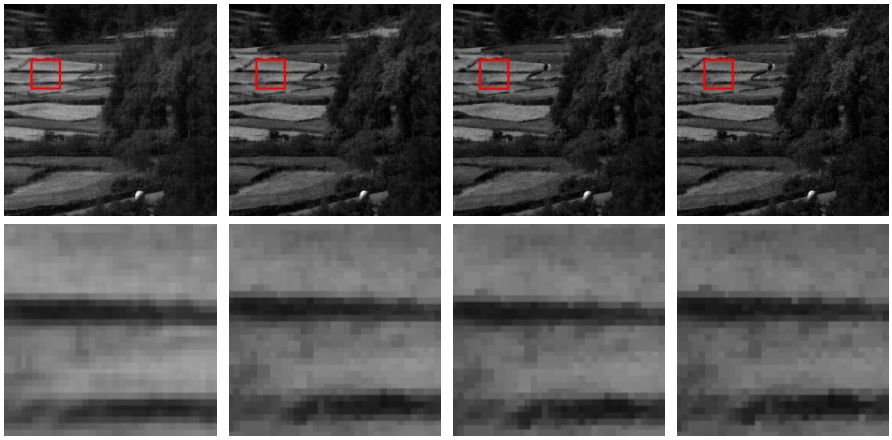


(a) (b) (c)

Figure 5.4: PSNR values of different methods for the Scene 5 dataset with different impulse noise ratios, where SR $= 0.7$ and $\sigma = 0.01$. (a) $\kappa = 0.1$. (b) $\kappa = 0.3$. (c) $\kappa = 0.5$.

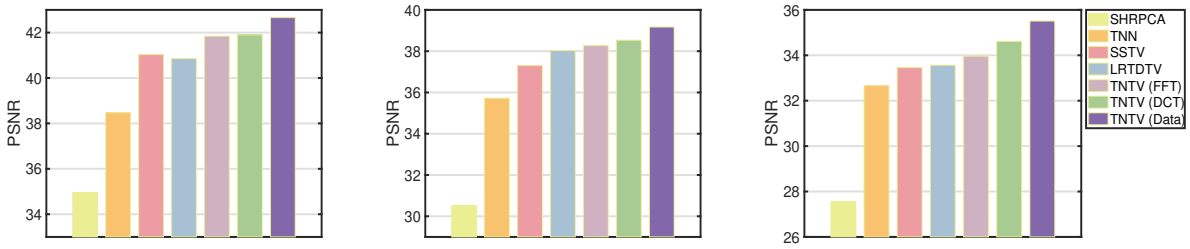


(a) Original (b) Observation (c) SHRPCA (d) TNN (e) SSTV



(f) LRTDTV (g) TNTV (FFT) (h) TNTV (DCT) (i) TNTV (Data)

Figure 5.5: Recovered images and zoomed regions of different methods for The 30th band of the Scene 5 dataset, where SR $= 0.6, \sigma = 0.001$, and $\kappa = 0.3$.



(a) (b) (c)

Figure 5.6: PSNR values of different methods for the Salinas dataset with different impulse noise ratios, where SR $= 0.7$ and $\sigma = 0.01$. (a) $\kappa = 0.1$. (b) $\kappa = 0.3$. (c) $\kappa = 0.5$.

5.2 Hyperspectral Images

In this subsection, we test the hyperspectral image datasets (length $\times$ width $\times$ bands) to validate the effectiveness of the proposed TNTV method. The first hyperspectral image dataset is the Samson

Table 5.1: PSNR and SSIM values of different methods for the Samson dataset. The boldface number is the best and the underline number is the second best.

		SR	σ	κ	SHRPCA	TNN	SSTV	LRTDTV	TNTV (FFT)	TNTV (DCT)	TNTV (Data)
Samson	PSNR	0.6	0.01	0.1	34.69	35.29	38.31	37.32	39.17	39.43	**40.27**
				0.2	31.28	33.29	34.95	34.89	36.74	38.09	**38.82**
				0.3	28.49	31.86	33.11	33.08	35.55	36.61	**37.30**
				0.4	25.80	30.42	30.90	30.86	33.46	35.64	**36.04**
				0.5	22.81	28.62	28.91	28.96	32.06	33.89	**34.29**
			0.001	0.1	37.06	39.11	40.21	39.33	41.18	42.71	**43.80**
				0.2	32.81	35.38	36.07	35.43	38.82	40.50	**41.21**
				0.3	29.27	33.08	34.21	33.72	36.56	38.76	**39.08**
				0.4	25.99	31.54	31.72	31.62	34.25	36.48	**36.80**
				0.5	22.99	29.43	29.86	29.75	32.52	34.63	**34.93**
		0.8	0.01	0.1	35.82	36.20	39.09	38.63	40.58	41.05	**41.92**
				0.2	32.73	34.28	36.63	36.52	38.76	39.62	**40.45**
				0.3	30.02	32.86	35.27	34.36	36.92	38.24	**38.88**
				0.4	27.31	31.49	32.16	32.13	34.74	36.50	**37.16**
				0.5	24.32	29.84	30.57	30.73	32.46	34.77	**35.42**
			0.001	0.1	39.79	40.79	41.87	41.06	43.75	45.85	**47.07**
				0.2	34.77	36.97	39.27	38.09	41.02	42.90	**43.81**
				0.3	31.17	34.67	36.84	34.89	38.56	40.69	**41.10**
				0.4	27.94	32.81	33.00	32.90	35.91	38.37	**38.57**
				0.5	24.53	30.72	31.17	31.08	33.86	35.93	**36.10**
	SSIM	0.6	0.01	0.1	0.8982	0.9007	0.9696	0.9694	0.9693	0.9785	**0.9799**
				0.2	0.8285	0.8979	0.9505	0.9574	0.9650	0.9680	**0.9751**
				0.3	0.7655	0.8830	0.9612	0.9496	0.9401	0.9665	**0.9686**
				0.4	0.7232	0.8920	0.9420	0.9429	0.9331	0.9534	**0.9564**
				0.5	0.4948	0.8738	0.9213	0.9238	0.9307	0.9371	**0.9415**
			0.001	0.1	0.9698	0.9724	0.9785	0.9824	0.9888	0.9925	**0.9931**
				0.2	0.9101	0.9685	0.9729	0.9773	0.9802	0.9876	**0.9884**
				0.3	0.8439	0.9498	0.9706	0.9646	0.9661	0.9809	**0.9818**
				0.4	0.7917	0.9476	0.9643	0.9609	0.9493	0.9701	**0.9716**
				0.5	0.5274	0.8973	0.9399	0.9378	0.9452	0.9551	**0.9577**
		0.8	0.01	0.1	0.8934	0.9263	0.9782	0.9756	0.9739	0.9826	**0.9837**
				0.2	0.8721	0.9115	0.9675	0.9660	0.9648	0.9785	**0.9800**
				0.3	0.8329	0.9112	0.9616	0.9551	0.9514	0.9730	**0.9747**
				0.4	0.6820	0.8978	0.9534	0.9533	0.9252	0.9647	**0.9670**
				0.5	0.6355	0.8872	0.9379	0.9225	0.8919	0.9535	**0.9565**
			0.001	0.1	0.9799	0.9818	0.9919	0.9836	0.9932	0.9958	**0.9963**
				0.2	0.9215	0.9808	0.9856	0.9781	0.9866	0.9924	**0.9929**
				0.3	0.8420	0.9645	0.9725	0.9765	0.9762	0.9867	**0.9874**
				0.4	0.7631	0.9583	0.9734	0.9700	0.9596	0.9788	**0.9795**
				0.5	0.7030	0.9166	0.9550	0.9535	0.9572	0.9655	**0.9667**

dataset ($95 \times 95 \times 156$) [55]. The second one is the Hyperspectral Images of Natural Scenes 2004 (Scene 5)[4] [30]. Each frontal slice of the Scene 5 dataset is resized to 300×300 in order to improve the computational time. Then the size of the resulting tensor is $300 \times 300 \times 32$. The third one is the Salinas scene[5] ($512 \times 217 \times 224$), which is collected by the 224-band AVIRIS sensor over Salinas Valley, California, and is larger than the previous two datasets. The three testing hypspectral image datasets are normalized into [0,1].

Figure 5.2 shows the PSNR values of different methods for the Samson dataset with different impulse noise ratios, where SR $= 0.7$ and $\sigma = 0.01$. It can be seen that the three TNTV methods outperform SHRPCA, TNN, SSTV, and LRTDTV in terms of PSNR values, where the impulse noise ratios $\kappa = 0.1, 0.3, 0.5$. Moreover, the SSTV and LRTDTV perform almost the same for the testing cases, which outperform SHRPCA and TNN. In Figure 5.3, we show the images of the 5th and 40th

[4] https://personalpages.manchester.ac.uk/staff/d.h.foster/default.html

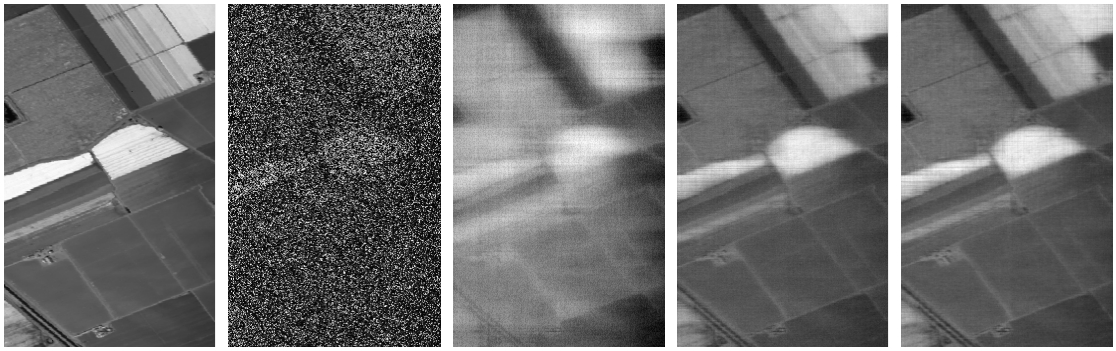
[5] http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes#Salinas

Table 5.2: PSNR and SSIM values of different methods for the Scene 5 dataset. The boldface number is the best and the underline number is the second best.

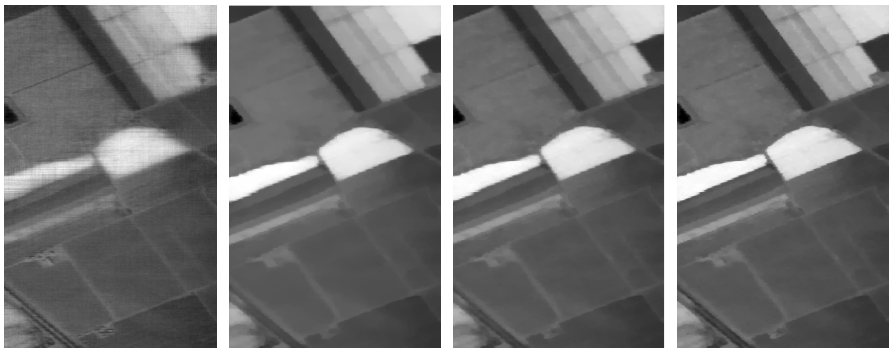
		SR	σ	κ	SHRPCA	TNN	SSTV	LRTDTV	TNTV (FFT)	TNTV (DCT)	TNTV (Data)
Scene 5	PSNR	0.6	0.01	0.1	32.55	34.83	35.80	36.36	36.33	36.68	**37.17**
				0.2	30.89	33.29	34.13	33.99	33.54	34.63	**34.93**
				0.3	29.52	31.87	32.94	32.98	32.51	33.12	**33.78**
				0.4	28.26	30.34	30.95	30.94	31.03	31.46	**31.78**
				0.5	26.72	28.37	29.27	29.48	29.36	30.01	**30.39**
			0.001	0.1	33.78	37.02	37.48	37.60	37.44	37.77	**38.39**
				0.2	31.62	34.75	35.24	35.15	35.17	35.73	**36.02**
				0.3	30.05	32.45	33.48	33.28	32.95	33.63	**34.33**
				0.4	28.62	31.07	31.29	31.21	31.45	32.06	**32.62**
				0.5	27.01	29.04	29.76	29.71	29.07	29.72	**30.27**
		0.8	0.01	0.1	33.20	35.88	37.60	37.76	37.72	38.39	**38.82**
				0.2	31.54	34.29	36.08	36.06	35.56	36.42	**36.80**
				0.3	30.20	32.85	33.99	34.02	33.73	34.58	**34.92**
				0.4	28.92	31.46	32.32	32.26	32.28	32.71	**33.18**
				0.5	27.76	29.31	30.56	30.88	30.56	31.01	**31.58**
			0.001	0.1	35.22	38.48	40.01	39.73	39.66	40.29	**41.05**
				0.2	32.83	35.99	37.24	37.08	37.10	37.81	**38.14**
				0.3	31.08	34.09	35.18	35.17	34.34	35.49	**35.51**
				0.4	29.63	32.34	32.63	32.62	32.84	33.41	**34.02**
				0.5	28.04	30.26	31.12	31.02	30.88	31.33	**31.76**
	SSIM	0.6	0.01	0.1	0.8557	0.9013	0.9457	0.9461	0.9478	0.9488	**0.9492**
				0.2	0.8429	0.8960	0.9210	0.9196	0.9208	0.9259	**0.9282**
				0.3	0.7602	0.8354	0.9005	0.9003	0.8997	0.9024	**0.9041**
				0.4	0.7494	0.8340	0.8668	0.8666	0.8711	0.8744	**0.8773**
				0.5	0.6372	0.7187	0.8215	0.8218	0.8291	0.8364	**0.8403**
			0.001	0.1	0.9159	0.9548	0.9588	0.9565	0.9605	0.9623	**0.9640**
				0.2	0.8528	0.9410	0.9458	0.9496	0.9437	0.9493	**0.9508**
				0.3	0.8129	0.9211	0.9249	0.9252	0.9248	0.9256	**0.9268**
				0.4	0.7942	0.9003	0.8938	0.8907	0.8969	0.8970	**0.8988**
				0.5	0.6746	0.7878	0.8706	0.8755	0.8545	0.8583	**0.8613**
		0.8	0.01	0.1	0.8825	0.9138	0.9465	0.9465	0.9398	0.9493	**0.9517**
				0.2	0.8481	0.8871	0.9311	0.9306	0.9348	0.9337	**0.9360**
				0.3	0.8150	0.8777	0.9062	0.9060	0.9058	0.9099	**0.9118**
				0.4	0.7926	0.8502	0.8882	0.8871	0.8910	0.8939	**0.8956**
				0.5	0.7230	0.7324	0.8537	0.8523	0.8542	0.8560	**0.8617**
			0.001	0.1	0.9452	0.9566	0.9750	0.9784	0.9754	0.9789	**0.9796**
				0.2	0.8860	0.9359	0.9545	0.9530	0.9488	0.9589	**0.9601**
				0.3	0.8374	0.9214	0.9260	0.9257	0.9252	0.9307	**0.9317**
				0.4	0.8085	0.9119	0.9114	0.9188	0.9192	0.9182	**0.9190**
				0.5	0.7897	0.8142	0.8739	0.8799	0.8826	0.8838	**0.8843**

bands recovered by SHRPCA, TNN, SSTV, LRTDTV, TNTV (FFT), TNTV (DCT), and TNTV (Data) for the Samson dataset, where SR $= 0.6, \sigma = 0.01$, and $\kappa = 0.2$. It can be observed that the images recovered by the three TNTV methods are more clear than those recovered by SHRPCA, TNN, SSTV, and LRTDTV. Furthermore, the TNTV (Data) performs the best in the three TNTV methods in terms of visual quality, where TNTV (Data) keeps more details than TNTV (FFT) and TNTV (DCT) for the recovered images.

Figure 5.4 dispalys the PSNR values of different methods for the Scene 5 dataset with different impulse noise ratios, where SR $= 0.7$ and $\sigma = 0.01$. It can been seen that the TNTV (Data) and TNTV (DCT) perform better than SHRPCA, TNN, SSTV, LRTDTV, and TNTV (FFT) in terms of PSNR values. The performance of SSTV and LRTDTV are almost the same for different impulse noise ratios, which are slightly better than TNTV (FFT) for $\kappa = 0.3$. Figure 5.5 shows the visual comparisons and corresponding zoomed regions of different methods for the 30th band of the Scene 5 dataset, where SR $= 0.6, \sigma = 0.001$, and $\kappa = 0.3$. From the zoomed regions, we can see that the images recovered by



(a) Original (b) Observation (c) SHRPCA (d) TNN (e) SSTV



(f) LRTDTV (g) TNTV (FFT) (h) TNTV (DCT) (i) TNTV (Data)

Figure 5.7: Recovered images of different methods for the 50th band of the Salinas dataset, where SR $= 0.6, \sigma = 0.01$, and $\kappa = 0.5$.

TNTV (Data) are more clear and keep more textures than those recovered by other methods. Besides, the SSTV and LRTDTV outperform SHRPCA and TNN in terms of visual quality.

In Tables 5.1 and 5.2, we display the PSNR and SSIM values of different methods for the Samson and Scene 5 datasets, where we test the cases with SR $= 0.6, 0.8, \sigma = 0.01, 0.001$, and $\kappa = 0.1, 0.2, 0.3, 0.4, 0.5$. It is shown that the PSNR and SSIM values obtained by TNTV (Data) and TNTV (DCT) are higher than those obtained by SHRPCA, TNN, SSTV, LRTDTV, and TNTV (FFT), and the PSNR values obtained by TNTV (Data) are higher than those obtained by TNTV (DCT). For the Samson dataset, TNTV (FFT) performs better than SHRPCA, TNN, SSVT, and LRTDTV in terms of PSNR values. For the Scene 5 dataset, the SSTV and LRTDTV are slightly better than TNTV (FFT) for most cases, which perform better than SHRPCA and TNN.

In addition, we test a larger hyperspectral datatset, i.e., Salinas. Figure 5.6 shows the PSNR values versus impulse noise ratio of different methods for the Salinas dataset, where SR $= 0.7$ and $\sigma = 0.01$. It can be observed from this figure that the TNTV methods achieve higher PSNR values than SHRPCA, TNN, SSTV, and LRTDTV, which implies that the TNTV performs better. Again the TNTV (Data) outperforms TNTV (FFT) and TNTV (DCT) in terms of PSNR values. In Figure 5.7, we show the visual quality of different methods for the 50th band of the Salinas dataset, where SR $= 0.6, \kappa = 0.5$, and $\sigma = 0.01$. We can see that the images recovered by the three TNTV methods are more clear than those recovered by SHRPCA, TNN, SSTV, and LRTDTV, especially for the white regions of the recovered images. And the TNTV (Data) performs slightly better than TNTV (FFT) and TNTV (DCT) in terms of visual quality. In Table 5.3, the PSNR and SSIM values of different methods are

Table 5.3: PSNR and SSIM values of different methods for the Salinas dataset. The boldface number is the best and the underline number is the second best.

		SR	σ	κ	SHRPCA	TNN	SSTV	LRTDTV	TNTV (FFT)	TNTV (DCT)	TNTV (Data)
Salinas	PSNR	0.6	0.01	0.1	33.81	38.65	40.86	40.74	40.97	41.09	**41.89**
				0.3	29.61	35.21	36.96	37.45	37.48	37.81	**38.51**
				0.5	26.90	32.00	32.85	33.28	33.56	34.29	**35.13**
			0.001	0.1	34.96	42.73	44.03	44.14	43.96	44.15	**45.09**
				0.3	29.93	37.72	38.45	38.85	38.83	39.13	**39.90**
				0.5	27.14	33.50	33.91	33.93	34.10	34.73	**35.70**
		0.8	0.01	0.1	35.24	38.85	41.79	41.49	42.37	42.46	**43.14**
				0.3	31.10	36.29	38.27	38.31	38.83	39.17	**39.64**
				0.5	27.83	32.90	34.31	34.59	34.45	35.08	**35.85**
			0.001	0.1	37.42	44.71	46.05	46.50	46.60	46.76	**48.05**
				0.3	32.25	39.32	40.29	40.67	40.72	41.47	**41.77**
				0.5	28.62	34.92	35.48	35.52	34.81	35.58	**36.47**
	SSIM	0.6	0.01	0.1	0.8903	0.9226	0.9419	0.9501	0.9608	0.9669	**0.9690**
				0.3	0.8747	0.9026	0.9115	0.9058	0.9251	0.9405	**0.9412**
				0.5	0.8670	0.8171	0.8540	0.8892	0.9157	0.9230	**0.9272**
			0.001	0.1	0.9745	0.9720	0.9779	0.9797	0.9890	0.9905	**0.9907**
				0.3	0.9441	0.9482	0.9532	0.9649	0.9646	0.9709	**0.9723**
				0.5	0.9229	0.9201	0.9513	0.9536	0.9545	0.9591	**0.9633**
		0.8	0.01	0.1	0.8734	0.9299	0.9452	0.9590	0.9675	0.9709	**0.9725**
				0.3	0.7918	0.9051	0.9362	0.9072	0.9385	0.9477	**0.9482**
				0.5	0.6860	0.8916	0.9008	0.9185	0.9226	0.9298	**0.9339**
			0.001	0.1	0.9821	0.9814	0.9832	0.9857	0.9932	0.9941	**0.9943**
				0.3	0.9426	0.9637	0.9689	0.9764	0.9739	0.9795	**0.9796**
				0.5	0.8673	0.9588	0.9617	0.9640	0.9601	0.9657	**0.9700**

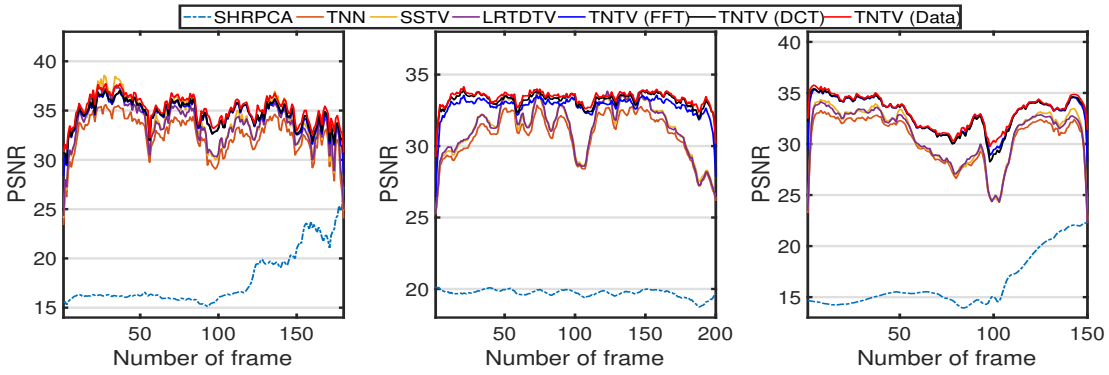


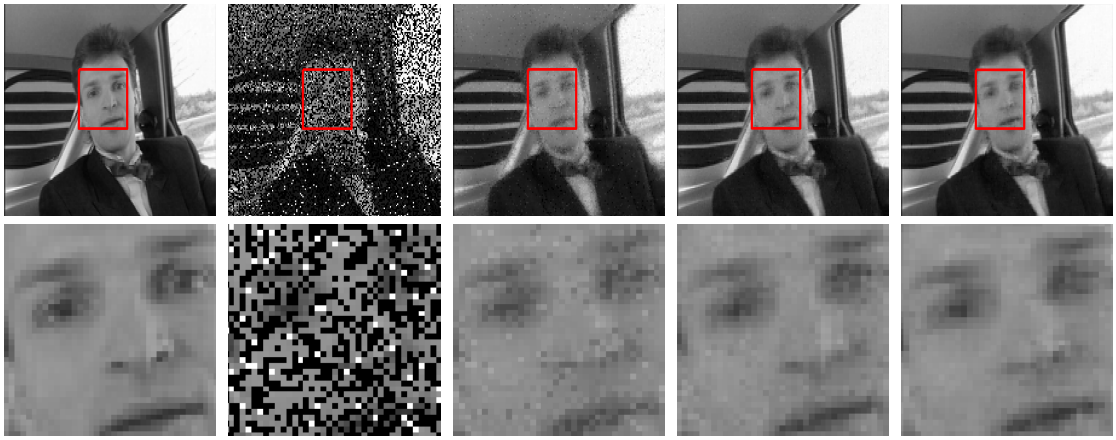
Figure 5.8: PSNR values versus number of frame for different video datasets. First column: Carphone. Second column: Claire. Third column: Suzie.

displayed for the Salinas dataset with different sampling ratios, κ, and σ. It can be seen that the TNTV (Data) acquires highest PSNR and SSIM values for these testing cases compared with other methods. Moreover, the TNTV (DCT) outperforms the SHRPCA, TNN, SSTV, LRTDTV, and TNTV (FFT) for both PSNR and SSIM values.

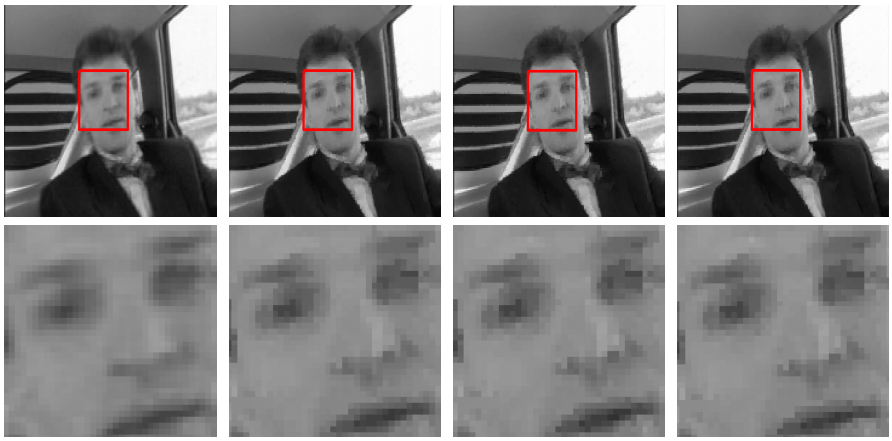
5.3 Video Images

In this subsection, we test three video datasets (length $\times$ width $\times$ frames) including Carphone ($144 \times 176 \times 180$), Claire ($144 \times 176 \times 200$), and Suzie ($144 \times 176 \times 150$)[6]. We just choose the first channel of

[6]http://trace.eas.asu.edu/yuv/



(a) Original (b) Observation (c) SHRPCA (d) TNN (e) SSTV



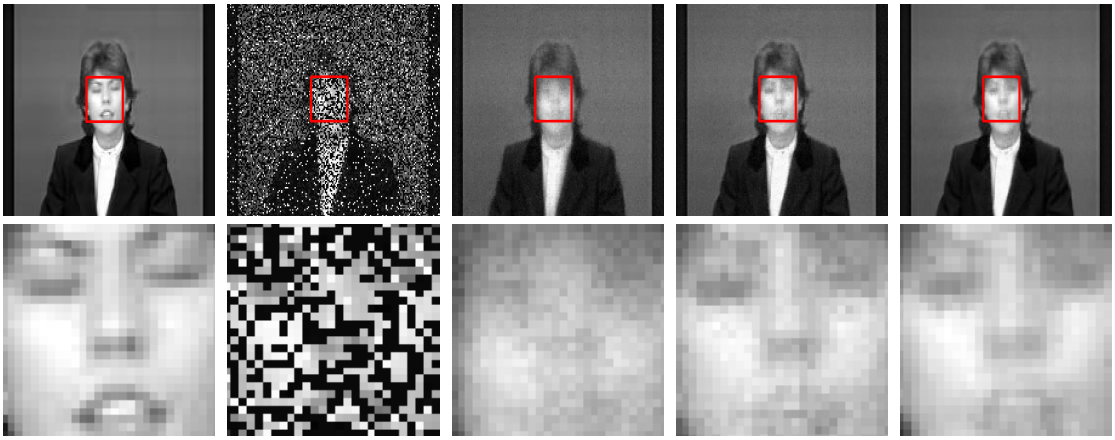
(f) LRTDTV (g) TNTV (FFT) (h) TNTV (DCT) (i) TNTV (Data)

Figure 5.9: Recovered images and zoomed regions of different methods for the 5th frame of the Carphone dataset, where SR $= 0.6, \sigma = 0.001$, and $\kappa = 0.1$.

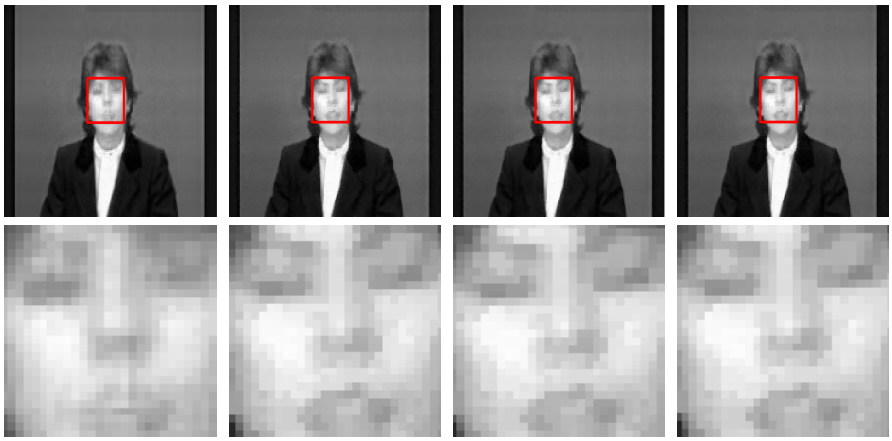
the color images in the videos and then formulate them as third-order tensors. The three testing videos are normalized into $[0, 1]$.

In Figure 5.8, we show the PSNR values versus number of frames of SHRPCA, TNN, SSTV, LRTDTV, TNTV (FFT), TNTV (DCT), and TNTV (Data) for the three video datasets, where SR $= 0.8, \kappa = 0.1, \sigma = 0.01$ for the Carphone dataset, SR $= 0.6, \kappa = 0.5, \sigma = 0.01$ for the Claire dataset, and SR $= 0.8, \kappa = 0.3, \sigma = 0.001$ for the Suzie dataset. We can observe that the three TNTV methods perform better than SHRPCA, TNN, SSTV, and LRTDTV in terms of PSNR values in most frames for the Carphone dataset. For the Claire and Suzie datasets, the TNTV (Data) outperforms other methods in all frames. Moreover, the performance of TNTV (FFT), TNTV (DCT), and TNTV (Data) are almost the same for the Carphone and Suzie dataset in most frames, and the PSNR values obtained by TNTV (Data) are slightly higher than those obtained by TNTV (FFT) and TNTV (DCT) for the Claire datasets in most frames. The SSTV and LRTDTV perform almost the same for different frames, which outperform SHRPCA and TNN.

Figure 5.9 shows the visual comparisons of different methods for the 5th frame of the Carphone dataset, where SR $= 0.6, \sigma = 0.001$, and $\kappa = 0.1$. It can be seen that the visual quality of the three



(a) Original (b) Observation (c) SHRPCA (d) TNN (e) SSTV

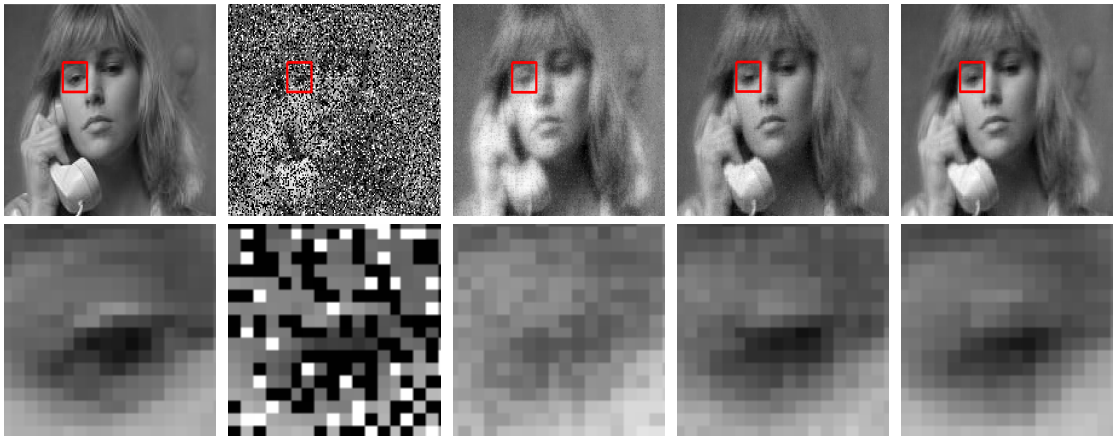


(f) LRTDTV (g) TNTV (FFT) (h) TNTV (DCT) (i) TNTV (Data)

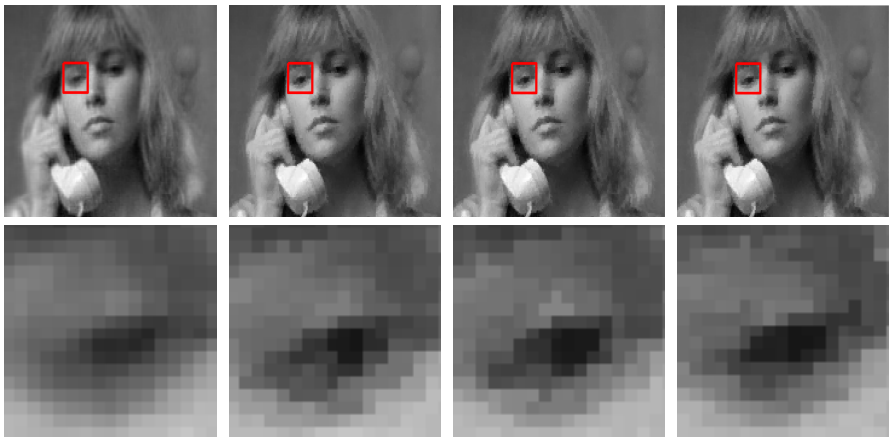
Figure 5.10: Recovered images and zoomed regions of different methods for the 10th frame of the Claire dataset, where SR $= 0.6, \sigma = 0.01$, and $\kappa = 0.2$.

TNTV methods are better than those of SHRPCA, TNN, SSTV, and LRTDTV. In Figure 5.10, we show the visual quality and corresponding zoomed regions of different methods for the 10th frame of the Claire dataset, where SR $= 0.6, \sigma = 0.01$, and $\kappa = 0.2$. Compared with SHRPCA, TNN, SSTV, and LRTDTV, the images recovered by the three TNTV methods are better in terms of visual quality. Figure 5.11 shows the recovered images and zoomed regions by different methods for the 30th frame of the Suzie dataset, where SR $= 0.8, \sigma = 0.001$, and $\kappa = 0.3$. Again it can be observed from this figure that the images recovered by TNTV (FFT), TNTV (DCT), and TNTV (Data) are more clear than those recovered by SHRPCA, TNN, SSTV, and LRTDTV. Moreover, the performance of TNTV (Data) is slightly better than those of TNTV (FFT) and TNTV (DCT) in terms of visual quality.

Tables 6.4 and 6.5 present the PSNR and SSIM values of different methods for the three video datasets, respectively, where SR $= 0.6, 0.8, \sigma = 0.01, 0.001$, and $\kappa = 0.1, 0.2, 0.3, 0.4, 0.5$. We can see that the PSNR and SSIM values obtained by the three TNTV methods are higher than those obtained by SHRPCA, TNN, SSTV, and LRTDTV. Moreover, the TNTV (Data) is slightly better than TNTV (FFT) and TNTV (DCT) in PSNR and SSIM values for most cases, and the TNTV (DCT) outperforms TNTV (FFT) slightly for most cases. The performance of SSTV and LRTDTV is almost the same,



(a) Original (b) Observation (c) SHRPCA (d) TNN (e) SSTV



(f) LRTDTV (g) TNTV (FFT) (h) TNTV (DCT) (i) TNTV (Data)

Figure 5.11: Recovered images and zoomed regions of different methods for the 30th frame of the Suzie dataset, where SR $= 0.8, \sigma = 0.001$, and $\kappa = 0.3$.

which is better than that of SHRPCA and TNN.

6 Concluding Remarks

In this paper, we have proposed a TNTV model for robust low-rank tensor completion with different degradations from partial observations, including Gaussian noise and sparse noise. The TNTV model consists of the TTNN, TV regularization for the underlying low-rank tensor combined the tensor ℓ_1 norm with the tensor Frobenius norm. The TTNN has the capacity to exploit the global low-rankness of the underlying tensor and get a lower transformed multi-rank tensor by using suitable unitary transformations [40], and the TV regularization is utilized to enhance the piecewise smooth structure along the spatial and tubal dimensions for imaging data. Besides, the tensor ℓ_1 norm is enforced to detect the sparse noise. Then three different transformations are utilized in transformed tensor SVD for the TNTV model. Furthermore, an sGS-ADMM is developed to solve the resulting model with three blocks of variables and its global convergence is established under very mild conditions. Numerical experiments on hyperspectral image and video datasets are presented to illustrate that the TNTV method can

Table 6.4: PSNR values of different methods for the video datasets. The boldface number is the best and the underline number is the second best.

		SR	σ	κ	SHRPCA	TNN	SSTV	LRTDTV	TNTV (FFT)	TNTV (DCT)	TNTV (Data)
PSNR	Carphone	0.6	0.01	0.1	27.23	30.40	31.10	31.08	32.65	32.89	**33.32**
				0.2	24.73	28.86	29.70	29.90	30.80	31.13	**31.42**
				0.3	22.69	27.57	28.34	28.56	28.69	29.67	**29.90**
				0.4	20.78	26.56	27.01	26.86	27.95	28.31	**28.53**
				0.5	19.00	25.30	25.76	25.80	26.34	26.90	**27.04**
			0.001	0.1	27.60	31.06	31.82	31.80	32.92	33.21	**33.73**
				0.2	24.92	29.30	29.94	29.98	31.00	31.35	**31.69**
				0.3	22.78	27.96	28.56	28.68	28.79	29.77	**30.02**
				0.4	20.85	26.85	27.12	26.88	28.03	28.39	**28.59**
				0.5	19.03	25.63	25.84	25.83	26.46	26.99	**27.16**
		0.8	0.01	0.1	28.90	31.67	32.95	32.85	34.29	34.52	**35.03**
				0.2	26.19	29.83	30.80	30.95	32.26	32.55	**32.93**
				0.3	24.05	28.52	29.21	29.39	30.16	30.90	**31.19**
				0.4	22.04	27.32	27.94	28.07	28.99	29.35	**29.57**
				0.5	20.11	26.14	26.61	26.69	27.26	27.76	**27.94**
			0.001	0.1	29.49	32.67	33.74	33.58	34.81	35.12	**35.76**
				0.2	26.44	30.59	31.34	31.69	32.51	32.85	**33.31**
				0.3	24.27	29.06	29.32	29.36	30.44	31.15	**31.49**
				0.4	22.11	27.74	28.03	28.42	29.20	29.60	**29.85**
				0.5	20.19	26.45	26.70	26.74	27.41	27.88	**28.06**
	Claire	0.6	0.01	0.1	31.49	35.62	37.43	37.59	38.14	38.53	**38.74**
				0.2	28.70	34.14	35.71	36.13	36.62	37.05	**37.26**
				0.3	26.60	32.78	33.98	34.33	35.33	35.83	**36.05**
				0.4	24.09	31.47	31.83	31.85	34.11	34.54	**34.76**
				0.5	21.17	30.20	30.82	30.77	32.76	33.26	**33.45**
			0.001	0.1	32.47	38.23	38.85	38.85	39.10	39.80	**40.10**
				0.2	29.28	35.95	36.69	36.78	37.40	37.95	**38.22**
				0.3	27.04	34.16	34.46	34.47	35.72	36.37	**36.60**
				0.4	24.23	32.46	32.94	32.89	34.44	34.92	**35.16**
				0.5	21.54	30.86	31.35	31.33	32.99	33.53	**33.73**
		0.8	0.01	0.1	32.80	36.68	38.91	39.11	39.46	39.87	**40.07**
				0.2	30.05	35.16	36.99	37.26	37.85	38.27	**38.48**
				0.3	27.88	33.83	35.02	35.22	36.38	36.88	**37.07**
				0.4	25.66	32.42	32.87	33.79	35.01	35.53	**35.73**
				0.5	22.61	30.95	31.78	32.10	33.70	34.16	**34.36**
			0.001	0.1	34.65	39.66	40.24	40.31	40.90	41.76	**42.08**
				0.2	31.10	37.40	38.32	38.40	38.86	39.44	**39.71**
				0.3	28.22	35.27	36.21	36.40	36.99	37.69	**37.95**
				0.4	26.04	33.59	34.23	34.11	35.47	36.01	**36.25**
				0.5	22.75	31.83	31.85	32.50	33.96	34.50	**34.72**
	Suzie	0.6	0.01	0.1	28.62	31.93	33.22	33.42	34.32	34.36	**34.64**
				0.2	26.46	30.47	31.39	31.51	32.80	32.87	**33.06**
				0.3	24.60	29.28	29.98	30.27	31.71	31.64	**31.84**
				0.4	22.68	28.19	28.56	28.59	30.50	30.48	**30.67**
				0.5	20.59	27.16	27.41	27.43	29.23	29.24	**29.43**
			0.001	0.1	29.17	33.04	33.83	33.82	35.06	35.08	**35.42**
				0.2	26.77	31.19	31.83	31.91	33.09	33.43	**33.68**
				0.3	24.86	29.77	30.32	30.51	31.94	31.95	**32.24**
				0.4	22.80	28.57	28.74	28.65	30.67	30.66	**30.87**
				0.5	20.67	27.39	27.51	27.50	29.64	29.80	**29.92**
		0.8	0.01	0.1	30.46	33.27	34.88	35.05	35.62	35.64	**35.88**
				0.2	28.04	31.54	32.70	33.13	33.94	34.02	**34.25**
				0.3	26.00	30.24	31.03	31.25	32.65	32.67	**32.89**
				0.4	24.09	29.05	29.67	29.85	31.33	31.35	**31.41**
				0.5	21.95	27.84	28.38	28.50	30.04	30.18	**30.33**
			0.001	0.1	31.16	34.74	35.38	35.36	36.82	36.81	**37.23**
				0.2	28.53	32.52	33.05	33.57	34.49	34.75	**35.07**
				0.3	26.22	30.85	31.34	31.44	33.11	33.16	**33.40**
				0.4	24.28	29.46	30.00	30.05	31.60	31.68	**31.86**
				0.5	22.12	28.14	28.65	28.72	30.24	30.31	**30.50**

Table 6.5: SSIM values of different methods for the video datasets. The boldface number is the best and the underline number is the second best.

		SR	σ	κ	SHRPCA	TNN	SSTV	LRTDTV	TNTV (FFT)	TNTV (DCT)	TNTV (Data)
SSIM	Carphone	0.6	0.01	0.1	0.8064	0.8704	0.9334	0.9325	0.9477	0.9457	**0.9483**
				0.2	0.7311	0.8688	0.9001	0.9135	0.9281	0.9278	**0.9308**
				0.3	0.6185	0.8538	0.8715	0.8922	0.8967	0.9117	**0.9148**
				0.4	0.5278	0.7917	0.8584	0.8636	0.8950	0.8948	**0.8975**
				0.5	0.4726	0.7542	0.8256	0.8408	0.8640	0.8725	**0.8753**
			0.001	0.1	0.8384	0.9260	0.9406	0.9396	0.9606	0.9610	**0.9636**
				0.2	0.7600	0.8684	0.9200	0.9228	0.9423	0.9430	**0.9460**
				0.3	0.6374	0.8896	0.8950	0.9027	0.9118	0.9251	**0.9280**
				0.4	0.5415	0.8356	0.8783	0.8687	0.9058	0.9068	**0.9097**
				0.5	0.4843	0.7995	0.8454	0.8479	0.8746	0.8821	**0.8850**
		0.8	0.01	0.1	0.8267	0.9183	0.9473	0.9468	0.9597	0.9579	**0.9600**
				0.2	0.7904	0.9056	0.9254	0.9313	0.9444	0.9448	**0.9463**
				0.3	0.6844	0.8711	0.9039	0.9127	0.9217	0.9293	**0.9320**
				0.4	0.5792	0.8030	0.8767	0.8868	0.9125	0.9129	**0.9156**
				0.5	0.4931	0.7776	0.8486	0.8606	0.8854	0.8918	**0.8945**
			0.001	0.1	0.8683	0.9335	0.9532	0.9531	0.9735	0.9745	**0.9764**
				0.2	0.7615	0.9116	0.9265	0.9367	0.9589	0.9597	**0.9621**
				0.3	0.7172	0.8886	0.9214	0.9211	0.9382	0.9441	**0.9467**
				0.4	0.6033	0.8544	0.8972	0.8763	0.9237	0.9258	**0.9284**
				0.5	0.5120	0.8252	0.8696	0.8718	0.8971	0.9017	**0.9044**
	Claire	0.6	0.01	0.1	0.8684	0.9197	0.9726	0.9761	0.9703	0.9701	**0.9706**
				0.2	0.8281	0.9178	0.9585	0.9611	0.9646	0.9652	**0.9660**
				0.3	0.7584	0.8958	0.9447	0.9412	0.9609	0.9605	**0.9614**
				0.4	0.6970	0.9099	0.9461	0.9461	0.9523	0.9539	**0.9549**
				0.5	0.6408	0.8601	0.9278	0.9271	0.9403	0.9421	**0.9433**
			0.001	0.1	0.9393	0.9741	0.9850	0.9847	0.9848	0.9863	**0.9868**
				0.2	0.7957	0.9666	0.9735	0.9755	0.9786	0.9801	**0.9807**
				0.3	0.8449	0.9535	0.9695	0.9695	0.9701	0.9730	**0.9737**
				0.4	0.7836	0.9205	0.9566	0.9542	0.9619	0.9643	**0.9653**
				0.5	0.7244	0.8890	0.9397	0.9376	0.9503	0.9534	**0.9546**
		0.8	0.01	0.1	0.8596	0.9368	0.9776	0.9807	**0.9769**	0.9747	0.9751
				0.2	0.7967	0.9318	0.9672	0.9740	0.9707	0.9709	**0.9715**
				0.3	0.7843	0.9215	0.9599	0.9648	0.9669	0.9665	**0.9672**
				0.4	0.7227	0.9159	0.9543	0.9446	0.9590	0.9607	**0.9615**
				0.5	0.6676	0.8594	0.9350	0.9228	0.9485	0.9501	**0.9512**
			0.001	0.1	0.9492	0.9878	0.9893	0.9888	0.9895	0.9909	**0.9912**
				0.2	0.8987	0.9779	0.9825	0.9828	0.9844	0.9856	**0.9860**
				0.3	0.8853	0.9697	0.9687	0.9734	0.9768	0.9793	**0.9800**
				0.4	0.8303	0.9570	0.9636	0.9607	0.9691	0.9715	**0.9723**
				0.5	0.7722	0.9436	0.9543	0.9490	0.9586	0.9616	**0.9626**
	Suzie	0.6	0.01	0.1	0.8083	0.8551	0.9062	0.9153	0.9328	0.9312	**0.9330**
				0.2	0.7521	0.8363	0.8745	0.8709	**0.9205**	0.9117	0.9136
				0.3	0.6675	0.8169	0.8349	0.8629	**0.9005**	0.8957	0.8977
				0.4	0.6106	0.7826	0.8362	0.8405	0.8743	0.8816	**0.8835**
				0.5	0.3533	0.7596	0.8053	0.8111	0.8358	0.8642	**0.8656**
			0.001	0.1	0.8562	0.9164	0.9299	0.9230	0.9540	0.9541	**0.9558**
				0.2	0.7971	0.8936	0.9027	0.9072	0.9329	0.9335	**0.9355**
				0.3	0.7109	0.8677	0.8679	0.8796	0.9167	0.9149	**0.9168**
				0.4	0.6483	0.8322	0.8545	0.8485	0.8915	0.8970	**0.8987**
				0.5	0.3627	0.8008	0.8212	0.8188	0.854 2	0.8679	**0.8694**
		0.8	0.01	0.1	0.8349	0.8835	0.9359	0.9401	0.9471	0.9450	**0.9475**
				0.2	0.7599	0.8749	0.8912	0.9053	**0.9351**	0.9290	0.9308
				0.3	0.7254	0.8398	0.8791	0.8934	**0.9174**	0.9140	0.9157
				0.4	0.6555	0.8075	0.8423	0.8571	0.8943	0.8985	**0.9001**
				0.5	0.4356	0.7855	0.8051	0.8217	0.8627	0.8720	**0.8736**
			0.001	0.1	0.8940	0.9473	0.9545	0.9521	0.9691	0.9692	**0.9705**
				0.2	0.8183	0.8988	0.9046	0.9224	0.9516	0.9518	**0.9533**
				0.3	0.7797	0.8657	0.9058	0.9081	0.9342	0.9338	**0.9354**
				0.4	0.7051	0.8607	0.8733	0.8756	0.9120	0.9147	**0.9163**
				0.5	0.4578	0.7709	0.8352	0.8438	0.8815	0.8885	**0.8900**

give superior performance in terms of the qualitative and quantitative comparisons with other existing approaches.

In our proposed TNTV model, we use the TV regularization for the underlying images to preserve sharp edges. However, the TV regularization will sometimes cause undesired oil painting artifacts. Therefore, a possible extension of TNTV is to replace the TV by the total generalized variation [5] for the underlying tensor. The total generalized variation is more precise in describing intensity variations in smooth regions of images, and thus can reduce oil painting artifacts, while it is still able to preserve sharp edges like TV does. Moreover, in future work, we also plan to study the recovery/error bound results of the combination of TV and low rank in the proposed TNTV model (cf. [19, 54]).

Acknowledgments

We would like to thank the associate editor and anonymous referees for their valuable comments and suggestions which have helped to improve the quality of this paper.

References

[1] H. K. Aggarwal and A. Majumdar. Hyperspectral image denoising using spatio-spectral total variation. *IEEE Geosci. Remote Sens. Lett.*, 13(3):442–446, 2016.

[2] M. Bai, X. Zhang, G. Ni, and C. Cui. An adaptive correction approach for tensor completion. *SIAM J. Imaging Sci.*, 9(3):1298–1323, 2016.

[3] M. Bai, X. Zhang, and Q. Shao. Adaptive correction procedure for TVL1 image deblurring under impulse noise. *Inverse Problems*, 32(8):085004, 2016.

[4] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do. Efficient tensor completion for color image and video recovery: Low-rank tensor train. *IEEE Trans. Image Process.*, 26(5):2466–2479, 2017.

[5] K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM J. Imaging Sci.*, 3(3):492–526, 2010.

[6] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.*, 20(4):1956–1982, 2010.

[7] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3):11, 2011.

[8] E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009.

[9] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

[10] L. Chen, D. Sun, and K.-C. Toh. An efficient inexact symmetric Gauss-Seidel based majorized ADMM for high-dimensional convex composite conic programming. *Math. Program.*, 161(1-2):237–270, 2017.

[11] Y. Chen, S. Wang, and Y. Zhou. Tensor nuclear norm-based low-rank approximation with total variation regularization. *IEEE J. Sel. Topics Signal Process.*, 12(6):1364–1377, 2018.

[12] D. Goldfarb and Z. Qin. Robust low-rank tensor recovery: Models and algorithms. *SIAM J. Matrix Anal. Appl.*, 35(1):225–253, 2014.

[13] C. J. Hillar and L.-H. Lim. Most tensor problems are NP-hard. *J. ACM*, 60(6):45, 2013.

[14] W. Hu, D. Tao, W. Zhang, Y. Xie, and Y. Yang. The twist tensor nuclear norm for video completion. *IEEE Trans. Neural Netw. Learn. Syst.*, 28(12):2961–2973, 2017.

[15] Q. Jiang and M. Ng. Robust low-tubal-rank tensor completion via convex optimization. In *Proc. 28th Int. Joint Conf. Artificial Intell.*, pages 2649–2655, 2019.

[16] E. Kernfeld, M. Kilmer, and S. Aeron. Tensor–tensor products with invertible linear transforms. *Linear Algebra Appl.*, 485:545–570, 2015.

[17] 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 J. Matrix Anal. Appl.*, 34(1):148–172, 2013.

[18] M. E. Kilmer and C. D. Martin. Factorization strategies for third-order tensors. *Linear Algebra Appl.*, 435(3):641–658, 2011.

[19] O. Klopp, K. Lounici, and A. B. Tsybakov. Robust matrix completion. *Probab. Theory Relat. Fields*, 169(1-2):523–564, 2017.

[20] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, 2009.

[21] X. Y. Lam, J. S. Marron, D. Sun, and K.-C. Toh. Fast algorithms for large scale generalized distance weighted discrimination. *J. Comput. Graph. Stat.*, 27(2):368–379, 2018.

[22] X. Li, D. Sun, and K.-C. Toh. A Schur complement based semi-proximal ADMM for convex quadratic conic programming and extensions. *Math. Program.*, 155(1-2):333–373, 2016.

[23] X.-T. Li, X.-L. Zhao, T.-X. Jiang, Y.-B. Zheng, T.-Y. Ji, and T.-Z. Huang. Low-rank tensor completion via combined non-local self-similarity and low-rank regularization. *Neurocomputing*, 367:1–12, 2019.

[24] X.-L. Lin, M. K. Ng, and X.-L. Zhao. Tensor factorization with total variation and Tikhonov regularization for low-rank tensor completion in imaging data. *J. Math. Imaging Vis.*, 62(6-7):900–918, 2020.

[25] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. In *Proc. IEEE 12th Int. Conf. Computer Vision*, pages 2114–2121. IEEE, 2009.

[26] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):208–220, 2013.

[27] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(4):925–938, 2020.

[28] C. D. Martin, R. Shafer, and B. LaRue. An order-p tensor factorization with applications in imaging. *SIAM J. Sci. Comput.*, 35(1):A474–A490, 2013.

[29] C. Mu, B. Huang, J. Wright, and D. Goldfarb. Square deal: Lower bounds and improved relaxations for tensor recovery. In *Proc. Int. Conf. Mach. Learn.*, volume 32, pages 73–81, 2014.

[30] S. M. C. Nascimento, F. P. Ferreira, and D. H. Foster. Statistics of spatial cone-excitation ratios in natural scenes. *J. Opt. Soc. Am. A*, 19(8):1484–1490, 2002.

[31] M. K. Ng. *Iterative Methods for Toeplitz Systems*. Oxford, London: Oxford University Press, 2004.

[32] M. K. Ng, Q. Yuan, L. Yan, and J. Sun. An adaptive weighted tensor completion method for the recovery of remote sensing images with missing data. *IEEE Trans. Geosci. Remote Sens.*, 55(6):3367–3381, 2017.

[33] I. V. Oseledets. Tensor-train decomposition. *SIAM J. Sci. Comput.*, 33(5):2295–2317, 2011.

[34] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Berlin: Springer, 2009.

[35] B. Romera-Paredes and M. Pontil. A new convex relaxation for tensor completion. In *Adv. Neural Inform. Process. Syst.*, pages 2967–2975, 2013.

[36] L. I. Rudin and S. Osher. Total variation based image restoration with free local constraints. In *Proc. 1st Int. Conf. Image Process.*, volume 1, pages 31–35. IEEE, 1994.

[37] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms.

Phys. D, 60(1-4):259–268, 1992.

[38] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller. Tensor-based formulation and nuclear norm regularization for multienergy computed tomography. *IEEE Trans. Image Process.*, 23(4):1678–1693, 2014.

[39] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. A. Suykens. Learning with tensors: a framework based on convex optimization and spectral regularization. *Mach. Learn.*, 94(3):303–351, 2014.

[40] G. Song, M. K. Ng, and X. Zhang. Robust tensor completion using transformed tensor singular value decomposition. *Numer. Linear Algebra Appl.*, 27(3):e2299, 2020.

[41] L. R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[42] B. Wang and H. Zou. Another look at distance-weighted discrimination. *J. Royal Stat. Soc. B*, 80(1):177–198, 2018.

[43] 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 J. Sel. Topics Appl. Earth Obs. Remote Sens.*, 11(4):1227–1243, 2018.

[44] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Pocess.*, 13(4):600–612, 2004.

[45] J.-H. Yang, X.-L. Zhao, T.-H. Ma, Y. Chen, T.-Z. Huang, and M. Ding. Remote sensing image destriping using unidirectional high-order total variation and nonconvex low-rank regularization. *J. Comput. Appl. Math.*, 363:124–144, 2020.

[46] Q. Yuan, L. Zhang, and H. Shen. Hyperspectral image denoising employing a spectral–spatial adaptive total variation model. *IEEE Trans. Geosci. Remote Sens.*, 50(10):3660–3677, 2012.

[47] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan. Hyperspectral image restoration using low-rank matrix recovery. *IEEE Trans. Geosci. Remote Sens.*, 52(8):4729–4743, 2013.

[48] X. Zhang. A nonconvex relaxation approach to low-rank tensor completion. *IEEE Trans. Neural Netw. Learn. Syst.*, 30(6):1659–1671, 2019.

[49] X. Zhang, M. Bai, and M. K. Ng. Nonconvex-TV based image restoration with impulse noise removal. *SIAM J. Imaging Sci.*, 10(3):1627–1667, 2017.

[50] X. Zhang and M. K. Ng. A corrected tensor nuclear norm minimization method for noisy low-rank tensor completion. *SIAM J. Imaging Sci.*, 12(2):1231–1273, 2019.

[51] Z. Zhang and S. Aeron. Exact tensor completion using t-SVD. *IEEE Trans. Signal Process.*, 65(6):1511–1526, 2017.

[52] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-SVD. In *Proc. IEEE Conf. Computer Vision Pattern Recognit.*, pages 3842–3849, 2014.

[53] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-H. Ma, and T.-Y. Ji. Mixed noise removal in hyperspectral image via low-fibered-rank regularization. *IEEE Trans. Geosci. Remote Sens.*, 58(1):734–749, 2020.

[54] Z. Zhou, X. Li, J. Wright, E. Candès, and Y. Ma. Stable principal component pursuit. In *Proc. Int. Symp. Inform. Theory*, pages 1518–1522, 2010.

[55] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan. Spectral unmixing via data-guided sparsity. *IEEE Trans. Image Process.*, 23(12):5412–5427, 2014.