CHAPTER

2

Transform-based tensor singular value decomposition in multidimensional image recovery☆,☆☆

Tai-Xiang Jiang[a], **Michael K. Ng**[b], **and Xi-Le Zhao**[c]

[a] *School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu, Sichuan, China*

[b] *Department of Mathematics, The University of Hong Kong, Pokfulam, Hong Kong*

[c] *School of Mathematical Sciences/Research Center for Image and Vision Computing, University of Electronic Science and Technology of China, Chengdu, Sichuan, China*

CONTENTS

☆ This work was supported in part by the National Natural Science Foundation of China under Grants 12001446, 61876203, and 61772003; in part by the Key Project of Applied Basic Research in Sichuan Province under Grant 2020YJ0216; in part by the Applied Basic Research Project of Sichuan Province under Grant 2021YJ0107; in part by the National Key Research and Development Program of China under Grant 2020YFA0714001; in part by the Fundamental Research Funds for the Central Universities under Grant JBK2102001; and in part by the HKRGC GRF under Grants 12300218, 12300519, 17201020 and 17300021.

☆☆ Three authors contributed equally.

Tensors for Data Processing. https://doi.org/10.1016/B978-0-12-824447-0.00008-X

2.1 Introduction

The rapid advance in imaging technology has given rise to a wealth of multidimensional images (e.g., color images, video, and multispectral/hyperspectral images). However, due to the limitations of imaging conditions, the captured multidimensional data are always degraded. Multidimensional image recovery aims to estimate the underlying high-quality data from the degraded observation and is a fundamental problem in low-level computer vision. Without loss of generality, the degradation process can be modeled as

$$\mathcal{O} = \mathcal{A}(\mathcal{X}) + \mathcal{N}, \tag{2.1}$$

where $\mathcal{O}$ is the observation, $\mathcal{A}$ is a linear mapping, $\mathcal{X}$ is the target high-quality multidimensional image, and $\mathcal{N}$ is the additive noise.[1] Different combinations of $\mathcal{A}$ and $\mathcal{N}$ are related to a wide variety of real-world applications. In the case that $\mathcal{A}$ refers to the sampling operator and the noise term $\mathcal{N}$ vanishes, estimation of the complete data $\mathcal{X}$ from the partial observation $\mathcal{O}$ is a tensor/matrix completion problem [1,2]. If the noise term is taken into consideration, it becomes the noisy tensor/matrix completion problem [3,4]. When $\mathcal{A}$ is an identical mapping and $\mathcal{N}$ is mixed noise consisting of Gaussian noise, sparse noise, and other noise types, multidimensional image recovery becomes the mixed noise removal problem and is common in remote sensing hyperspectral images (HSIs) [5].

Estimation of $\mathcal{X}$ from $\mathcal{O}$ is an ill-posed inverse problem, and the maximum a posteriori (MAP) estimation is an effective approach to solve this problem. Specifically, we need to maximize the posterior probability $P(\mathcal{X}|\mathcal{O})$ under the Bayes rule [6], i.e.,

$$\begin{aligned} \mathcal{X}^* &= \arg\max_{\mathcal{X}} P(\mathcal{X}|\mathcal{O}) = \arg\max_{\mathcal{X}} \frac{P(\mathcal{O}|\mathcal{X})P(\mathcal{X})}{P(\mathcal{O})} \\ &= \arg\min_{\mathcal{X}} \{-\log P(\mathcal{O}|\mathcal{X}) - \log P(\mathcal{X})\}. \end{aligned} \tag{2.2}$$

In (2.2), $P(\mathcal{O}|\mathcal{X})$ is the likelihood term and is determined by (2.1). Generally, we can write $-\log P(\mathcal{O}|\mathcal{X})$ as $f(\mathcal{A}(\mathcal{X}) - \mathcal{O})$, and its minimization indicates that the recovered results should conform to the degradation process in (2.1). We also call it the data fidelity term. For example, when the noise is independent and identically Gaussian distributed with variance σ^2 and $\mathcal{A}$ is an identical mapping, $f(\mathcal{A}(\mathcal{X}) - \mathcal{O})$ can be formulated as $\frac{1}{2\sigma^2}\|\mathcal{X} - \mathcal{O}\|_F^2$ with its factor absorbed by $-\log P(\mathcal{X})$. For the tensor completion problem, $f(\cdot)$ can be written as the Dirac delta function. As for the prior term $-\log P(\mathcal{X})$ that expresses the prior distribution of the data, it can be written as $\lambda\phi(\mathcal{X})$, where lambda is a nonnegative parameter. Oftentimes, $\phi(\mathcal{X})$ is also referred to as the regularization term. Then, (2.2) can be rewritten as a regularized optimization problem, i.e.,

$$\mathcal{X}^* = \arg\min_{\mathcal{X}} \{f(\mathcal{A}(\mathcal{X}) - \mathcal{O}) + \lambda\phi(\mathcal{X})\}. \tag{2.3}$$

[1] Multiplicative noise can be modeled in a nonlinear mapping $\mathcal{A}$ when needed.

Since f is determined by the degradation process, reasonable analysis and effective exploitation of the underlying data's prior knowledge to build ϕ is highly important in this framework.

Multidimensional images are always inner-structured and globally correlated. For instance, the bands of a hyperspectral image are highly correlated such that its spectral vectors live in a low-dimensional subspace [7]. This low-dimensionality can be mathematically formulated as low-rankness, i.e., representation of the high-dimensional data under learned lower-dimensional bases. For matrices, although the direct minimization of the rank is NP-hard, we can minimize the nuclear norm (i.e., the sum of the singular values) to faithfully enhance the low-rankness. However, if multidimensional images are reordered into matrices and the corresponding recovery problem is solved via matrix-based methods, such matricization will inevitably destroy the intrinsic structure of multidimensional images. As the higher-order extension of the matrix, the tensor can provide a more natural and elegant representation for multidimensional images. Thus, in this chapter, we mainly focus on the design of a tensor low-rank regularizer for multidimensional image recovery.

Unlike for the matrix case, there is no unique rank definition for tensors. Thus, the definition of the tensor rank is a fundamental problem. Many research efforts have been devoted to this hot topic [8–11], such as the CANDECOMP/PARAFAC (CP) rank and the Tucker rank. The CP rank is defined based on canonical polyadic decomposition, where an N-th-order tensor is decomposed as the sum of the rank-1 tensors [8,12–14], i.e., the outer product of N vectors. The CP rank is defined as the minimal number of the rank-1 tensors required to express the data. The Tucker rank is based on the Tucker decomposition that decomposes a tensor into a core tensor multiplied by a matrix along each mode [8,15]. The Tucker rank is defined as the vector consisting of the ranks of unfolding matrices along different modes. The Tucker rank has been considered in the low-rank tensor completion problem by minimizing the convex surrogate of its summation, i.e., the sum of the nuclear norm (SNN) [2], or the nonconvex surrogates [16,17].

The tensor singular value decomposition (t-SVD) [18,19], based on the tensor–tensor product (t-prod), has emerged as a powerful tool for preserving the intrinsic structures of the tensor. The tensor nuclear norm (TNN) [20–24] is suggested as a convex surrogate of the tensor tubal rank that is derived from the t-SVD framework. The TNN-based multidimensional image recovery model is given by

$$\min_{\mathcal{X}} \{ f(\mathcal{A}(\mathcal{X}) - \mathcal{O}) + \lambda \|\mathcal{X}\|_{\text{TNN}} \}, \tag{2.4}$$

where $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\|\mathcal{X}\|_{\text{TNN}} = \sum_{i=1}^{n_3} \|\widehat{\mathcal{X}}^{(i)}\|_*$, $\widehat{\mathcal{X}}^{(i)}$ is the i-th frontal slice of $\widehat{\mathcal{X}}$, and $\widehat{\mathcal{X}}$ is the tensor generated by performing discrete Fourier transformation (DFT) along the mode-3 fibers of $\mathcal{X}$, i.e., $\widehat{\mathcal{X}} = \texttt{fft}(\mathcal{X}, [], 3)$ in MATLAB®.

Compared with the Tucker and CP decomposition schemes, t-SVD provides an algebraic framework that is more analogous to the matrix case [19], and has received much attention in recent years. In this chapter, we review the framework of t-SVD and the establishment of the TNN. Then, focusing on the multidimensional image

recovery problem, we revisit recent advances based on t-SVD. Moreover, we delve into the t-SVD framework, and replace its key module, which is the DFT, with other transforms. Then, the transform-based t-SVD is introduced and we offer a deeper insight into the inner mechanism of the whole t-SVD framework.

The rest of this chapter is organized as follows. In Section 2.2, we revisit the t-SVD framework and the TNN. Specifically, Section 2.2.1 provides the basic tensor notations and the t-SVD framework is given in Section 2.2.2. Sections 2.2.3–2.2.4 concentrate on the TNN minimization model and its extensions. Then, we introduce the recent advance of the transform-based t-SVD in Section 2.3, from the linear invertible transform in Section 2.3.1 to the noninvertible transform and data adaptive transform in Section 2.3.2. Next, some numerical experiments are described in Section 2.4. Finally, Section 2.5 draws conclusions and provides some possible directions for future research.

2.2 Recent advances of the tensor singular value decomposition

In this section, we first give some basic notations and definitions for tensors, and then we introduce the framework of t-SVD. Next, the definition of TNN together with the recovery of multidimensional visual data by TNN minimization will be presented. Subsequently, some nonconvex surrogates and additional regularization terms are discussed. Finally, we give numerical examples to illustrate the performance of the methods within the t-SVD framework for multidimensional visual data recovery.

2.2.1 Preliminaries and basic tensor notations

Throughout this chapter, lowercase letters, e.g., x, boldface lowercase letters, e.g., $\mathbf{x}$, boldface uppercase letters, e.g., $\mathbf{X}$, and boldface calligraphic letters, e.g., $\mathcal{X}$, are used to denote scalars, vectors, matrices, and tensors, respectively. Given a third-order tensor[2] $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we use $\mathcal{X}_{ijk}$ or $\mathcal{X}(i, j, k)$ to denote its (i, j, k)-th element. Its (i, j)-th mode-3 fibers, i.e., the vectors along the third dimension, are denoted as $\mathcal{X}(i, j, :)$. The mode-3 fiber is also referred to as the tube. The k-th frontal slice of $\mathcal{X}$ is denoted as $\mathcal{X}^{(k)}$ (or $\mathcal{X}(:, :, k)$, $\mathbf{X}^k$). The **inner product** of two same-sized third-order tensors $\mathcal{X}$ and $\mathcal{Y}$ is defined as

$$\langle \mathcal{X}, \mathcal{Y} \rangle := \sum_{i_1, i_2, i_3} \mathcal{X}_{i_1 i_2 i_3} \cdot \mathcal{Y}_{i_1 i_2 i_3}. \tag{2.5}$$

[2] In this chapter, we mainly focus on third-order tensors. The notations on this page can be naturally extended to higher-order tensors.

The **Frobenius norm** is then defined as

$$\|\mathcal{X}\|_F := \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle} = \sqrt{\sum_{ijk} \mathcal{X}_{ijk}^2}. \tag{2.6}$$

The **mode-3 unfolding** of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is denoted as a matrix $\mathbf{X}_{(3)} \in \mathbb{R}^{n_3 \times n_1 n_2}$, where the tensor's (i, j, k)-th element maps to the matrix's (k, l)-th element satisfying $l = (j-1)n_1 + i$. The mode-3 unfolding operator and its inverse are respectively denoted as $\texttt{unfold}_3$ and $\texttt{fold}_3$, and they satisfy

$$\mathcal{X} = \texttt{fold}_3(\texttt{unfold}_3(\mathcal{X})) = \texttt{fold}_3(\mathbf{X}_{(3)}). \tag{2.7}$$

The mode-3 tensor–matrix product of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with a matrix $\mathbf{A} \in \mathbb{R}^{m \times n_3}$ is denoted by $\mathcal{X} \times_3 \mathbf{A}$ and is of size $n_1 \times n_2 \times m$. Element-wise, we have

$$(\mathcal{X} \times_3 \mathbf{A})_{ijk} = \sum_{n=1}^{n_3} \mathcal{X}_{ijn} \mathbf{A}_{kn}. \tag{2.8}$$

The mode-3 tensor–matrix product can also be expressed in terms of the mode-3 unfolding

$$\mathcal{Y} = (\mathcal{X} \times_3 \mathbf{A}) \quad \Leftrightarrow \quad \mathbf{Y}_{(3)} = \mathbf{A} \cdot \texttt{unfold}_3(\mathcal{X}), \tag{2.9}$$

where $\cdot$ is the matrix product.

Interested readers can refer to [8] for a more extensive overview.

2.2.2 The t-SVD framework

The basis of t-SVD is the definition of the multiplication operation between two third-order tensors. It is well known that in linear algebra, if $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is a $n \times p$ matrix, the matrix product $\mathbf{C} = \mathbf{AB}$ is defined to be the $m \times p$ matrix with

$$\mathbf{C}_{ij} = \sum_{k=1}^{n} \mathbf{A}_{ik} \mathbf{B}_{kj} \quad \text{for } i = 1, 2, \cdots, m \text{ and } j = 1, 2, \cdots, p. \tag{2.10}$$

Similarly, if we replace the numbers in a matrix with vectors of the same length and use the circular convolution between two vectors instead of the multiplication between two numbers, the tensor–tensor product (t-prod) can be defined.

Definition 2.1 (T-prod [19]). Let $\mathcal{A}$ be a tensor with size $n_1 \times n_2 \times n_3$ and let $\mathcal{B}$ be a tensor with size $n_2 \times n_4 \times n_3$. Then, the tensor–tensor product[3] between $\mathcal{A}$ and $\mathcal{B}$

[3] In this chapter, the symbol "$*$" is used to denote t-prod, similar to the literature. However, please note that this notation may be different in other chapters.

is the $n_1 \times n_4 \times n_3$ tensor $\mathcal{C} = \mathcal{A} * \mathcal{B}$ whose (i, j)-th tube is given by

$$\mathcal{C}(i, j, :) = \sum_{k=1}^{n_2} \mathcal{B}(i, k, :) \circledast \mathcal{C}(k, j, :), \tag{2.11}$$

where $i = 1, 2, \cdots, n_1$ and $j = 1, 2, \cdots, n_4$, and "$\circledast$" represents the circular convolution of two same-size vectors.

Since the circular convolution can be converted into element-wise multiplication in the Fourier domain, we now consider the DFT on a third-order tensor. The 1D DFT on a real-valued vector $\mathbf{x} \in \mathbb{R}^n$, denoted as $\bar{\mathbf{x}}$, is given by $\bar{\mathbf{x}} = \mathbf{F}_n \mathbf{x} \in \mathbb{C}^n$, where $\mathbf{F}_n \in \mathbb{C}^{n \times n}$ is the DFT matrix. The (i, j)-th element of $\mathbf{F}_n$ is equal to $\omega^{(i-1)(j-1)}$, where $\omega = e^{\frac{-2\pi\iota}{n}}$ is a primitive n-th root of unity and $\iota^2 = -1$. The inverse DFT matrix can be obtained as $\mathbf{F}_n^{-1} \in \mathbb{C}^{n \times n}$ and we have $\mathbf{F}_n^{-1} = \frac{1}{n}\mathbf{F}_n^{\mathrm{H}}$, where $\cdot^{\mathrm{H}}$ is the conjugate transpose. In this chapter, we use $\widehat{\mathcal{X}}$ to denote the transformed tensor by performing 1D DFT along the mode-3 fibers (tubes) of $\mathcal{X}$. By using the DFT matrix $\mathbf{F}_{n_3} \in \mathbb{C}^{n_3 \times n_3}$, we have

$$\widehat{\mathcal{X}} = \mathcal{X} \times_3 \mathbf{F}_{n_3}. \tag{2.12}$$

Additionally, $\mathcal{X}$ and $\widehat{\mathcal{X}}$ can be transformed to each other via the fast Fourier transform (FFT) and its inverse with a lower computation burden, for example, using the MATLAB command `fft` $(\mathcal{X}, [], 3)$ and `ifft` $(\mathcal{X}, [], 3)$, respectively. Meanwhile, we have

$$\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle} = \sqrt{\frac{1}{n_3} \langle \widehat{\mathcal{X}}, \widehat{\mathcal{X}} \rangle} = \frac{1}{\sqrt{n_3}} \|\widehat{\mathcal{X}}\|_F. \tag{2.13}$$

Returning to the t-prod, the t-prod between $\mathcal{A}$ and $\mathcal{B}$ can be computed efficiently by multiplying each pair of the frontal slices of the FFT-transformed tensors $\widehat{\mathcal{A}}$ and $\widehat{\mathcal{B}}$ and computing the inverse FFT along the third dimension to obtain the result.

Definition 2.2 (Block-diagonal form [21]). Let $\overline{\mathbf{X}}$ denote the block-diagonal matrix of $\widehat{\mathcal{X}} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, i.e.,

$$\begin{aligned} \overline{\mathbf{X}} &= \texttt{blockdiag}(\widehat{\mathcal{X}}) \\ &\triangleq \begin{bmatrix} \widehat{\mathcal{X}}^{(1)} & & & \\ & \widehat{\mathcal{X}}^{(2)} & & \\ & & \ddots & \\ & & & \widehat{\mathcal{X}}^{(n_3)} \end{bmatrix} \in \mathbb{C}^{n_1 n_3 \times n_2 n_3}. \end{aligned} \tag{2.14}$$

Furthermore, we denote the inverse operation of block-diagonal as $\texttt{blockdiag}^{-1}$. Then, for any two tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, we have

$$\mathcal{C} = \mathcal{A} * \mathcal{B} \Leftrightarrow \overline{\mathbf{C}} = \overline{\mathbf{A}} \cdot \overline{\mathbf{B}}. \tag{2.15}$$

The t-prod between two tensors is equivalent to the matrix multiplication in the Fourier-transformed domain.

Definition 2.3 (Tensor conjugate transpose [19]). The conjugate transpose of a tensor $\mathcal{X} \in \mathbb{R}^{n_2 \times n_1 \times n_3}$ is a tensor $\mathcal{X}^{\mathrm{H}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ obtained by the conjugate transpose of each of the frontal slices and then reversing the order of the transposed frontal slices 2 through n_3:

$$\begin{aligned} \left(\mathcal{X}^{\mathrm{H}}\right)^{(1)} &= \left(\mathcal{X}^{(1)}\right)^{\mathrm{H}} \quad \text{and} \\ \left(\mathcal{X}^{\mathrm{H}}\right)^{(i)} &= \left(\mathcal{X}^{(n_3+2-i)}\right)^{\mathrm{H}}, \quad i = 2, \cdots, n_3. \end{aligned} \tag{2.16}$$

More intuitively, in the Fourier-transformed domain, it satisfies

$$\left(\widehat{\mathcal{X}^{\mathrm{H}}}\right)^{(i)} = \left(\widehat{\mathcal{X}}^{(i)}\right)^{\mathrm{H}}. \tag{2.17}$$

Definition 2.4 (Identity tensor [19]). The identity tensor $\mathcal{I} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is defined as a tensor whose first frontal slice is the $n_1 \times n_1$ identity matrix, and the other frontal slices are zero matrices.

Definition 2.5 (F-diagonal tensor [19]). We call a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ f-diagonal if all of its frontal slices are diagonal matrices.

Definition 2.6 (Orthogonal tensor [19]). A tensor $\mathcal{Q} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is an orthogonal tensor if

$$\mathcal{Q}^{\mathrm{H}} * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^{\mathrm{H}} = \mathcal{I}. \tag{2.18}$$

Definitions 2.3–2.6 are analogous to the basic ingredients in matrix format to formulate the matrix SVD. Equipped with the these basic definitions, the t-SVD is suggested.

Theorem 2.1 (T-SVD [19,24]). *Let $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ be a third-order tensor. Then it can be decomposed as*

$$\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\mathrm{H}}, \tag{2.19}$$

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

Please see the proof of Theorem 2.1 in [24]. The t-SVD of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ can be efficiently obtained by computing a series of matrix SVDs on the frontal slices of $\widehat{\mathcal{X}}$ in the Fourier domain. Nonetheless, because the SVD is not unique, it is suggested to adopt the algorithm in Table 2.1 that can ensure that $\mathcal{U}$, $\mathcal{X}$, and $\mathcal{V}$ are real tensors [24].

Thus, the framework of t-SVD has been established. It originates from [18] and has been updated in [24], extending many familiar tools of linear algebra to third-order tensors. In particular, the t-prod, which is closed on the set of third-order tensors, allows tensor factorizations that are analogs of matrix factorizations such

Table 2.1 The algorithm for computing t-SVD.

Input: $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.

1. $\widehat{\mathcal{X}} \leftarrow \texttt{fft}(\mathcal{X}, [], 3)$.
2. $[\mathbf{U}, \mathbf{S}, \mathbf{V}] = \texttt{svd}(\widehat{\mathcal{X}}^{(1)})$.
3. $\widehat{\mathcal{U}}^{(1)} \leftarrow U$, $\widehat{\mathcal{S}}^{(1)} \leftarrow S$, $\widehat{\mathcal{V}}^{(1)} \leftarrow V$.
4. **for** $i = 2$ to $\lceil \frac{n_3+1}{2} \rceil$ **do**
5. $\quad [\mathbf{U}, \mathbf{S}, \mathbf{V}] = \texttt{svd}(\widehat{\mathcal{X}}^{(i)})$.
6. $\quad \widehat{\mathcal{U}}^{(i)} \leftarrow \mathbf{U}$, $\widehat{\mathcal{S}} \leftarrow \mathbf{S}$, $\widehat{\mathcal{V}}^{(i)} \leftarrow \mathbf{V}$
7. $\quad \widehat{\mathcal{U}}^{(n_3-i+2)} \leftarrow \mathbf{U}^{\mathrm{H}}$, $\widehat{\mathcal{S}}^{(n_3-i+2)} \leftarrow \mathbf{S}^{\mathrm{H}}$, $\widehat{\mathcal{V}}^{(n_3-i+2)} \leftarrow \mathbf{V}^{\mathrm{H}}$
8. **end for**
9. $\mathcal{U} \leftarrow \texttt{ifft}(\widehat{\mathcal{U}}, [], 3)$, $\mathcal{S} \leftarrow \texttt{ifft}(\widehat{\mathcal{S}}, [], 3)$, $\mathcal{V} \leftarrow \texttt{ifft}(\widehat{\mathcal{V}}, [], 3)$.

Output: $\mathcal{U}$, $\mathcal{S}$, and $\mathcal{V}$.

as SVD. Meanwhile, it also allows novel extensions of familiar matrix analysis to the multilinear setting while avoiding the loss of information inherent in *matricization* or *flattening* of the tensor [19].

2.2.3 Tensor nuclear norm and tensor recovery

Based on t-SVD, we can deduce the corresponding definitions of tensor multirank, tubal-rank, and average-rank.

Definition 2.7 (Tensor tubal-rank [19], multirank [21], and average-rank [24]). Let $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ be a third-order tensor. The tensor multirank, denoted as $\mathrm{rank}_{\mathrm{m}}(\mathcal{X}) \in \mathbb{R}^{n_3}$, is a vector whose i-th element is the rank of the i-th frontal slice of $\widehat{\mathcal{X}}$, where $\widehat{\mathcal{X}} = \texttt{fft}(\mathcal{X}, [], 3)$. We can write

$$\mathrm{rank}_{\mathrm{m}}(\mathcal{X}) = \left[\mathrm{rank}(\widehat{\mathcal{X}}^{(1)}), \mathrm{rank}(\widehat{\mathcal{X}}^{(2)}), \cdots, \mathrm{rank}(\widehat{\mathcal{X}}^{(n_3)})\right]. \tag{2.20}$$

The tensor tubal-rank of $\mathcal{X}$, denoted as $\mathrm{rank}_{\mathrm{t}}(\mathcal{X})$, is defined as the number of nonzero tubes of $\mathcal{S}$, where $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\mathrm{H}}$. Specifically, we have

$$\mathrm{rank}_{\mathrm{t}}(\mathcal{X}) = \#\{i, \mathcal{S}(i, i, :) \neq 0\}. \tag{2.21}$$

The tensor average-rank[4] of $\mathcal{X}$, denoted as $\mathrm{rank}_{\mathrm{a}}(\mathcal{X})$, is defined as

$$\mathrm{rank}_{\mathrm{a}}(\mathcal{X}) = \frac{1}{n_3} \sum_{k=1}^{n_3} \mathrm{rank}(\widehat{\mathcal{X}}^{(i)}). \tag{2.22}$$

[4] In [24], the tensor average-rank is defined in the original domain.

According to the definitions of the three tensor ranks, given a third-order tensor $\mathcal{X}$, its three tensor ranks are related by [24]

$$\text{rank}_{\text{a}}(\mathcal{X}) \leq \max(\text{rank}_{\text{m}}(\mathcal{X})) = \text{rank}_{\text{t}}(\mathcal{X}). \tag{2.23}$$

Analogous to the matrix case, optimization problems related to the tensor ranks defined in Definition 2.7 are difficult and adequate convex relaxations are required for practical applications. Next, we give the definition of the TNN.

Definition 2.8 (Tensor nuclear norm [20]). The TNN of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, denoted as $\|\mathcal{X}\|_{\text{TNN}}$, is defined as the sum of singular values of all of the frontal slices of $\bar{\mathcal{X}}$, i.e.,

$$\|\mathcal{X}\|_{\text{TNN}} := \sum_{i=1}^{n_3} \|\widehat{\mathcal{X}}^{(i)}\|_*, \tag{2.24}$$

where $\widehat{\mathcal{X}}^{(i)}$ is the i-th frontal slice of $\widehat{\mathcal{X}}$ and $\widehat{\mathcal{X}} = \texttt{fft}(\mathcal{X}, [], 3)$.

We find that the TNN[5] of a third-order tensor $\mathcal{X}$ is defined in the Fourier domain and is equivalent to $\|\overline{\mathbf{X}}\|_*$ where $\overline{\mathbf{X}} = \texttt{blockdiag}(\widehat{\mathcal{X}})$. Thus, it can be proved that TNN is the tightest convex relaxation to the ℓ_1-norm of the tensor multirank [21], which is indeed the rank of the block-diagonal matrix obtained from the Fourier-transformed tensor.

Before proceeding to solve the multidimensional image recovery model using TNN, one key issue is to compute the proximal operator of TNN, i.e.,

$$\mathcal{X}^* = \arg\min_{\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}} \tau \|\mathcal{X}\|_{\text{TNN}} + \frac{1}{2}\|\mathcal{X} - \mathcal{Y}\|_F^2. \tag{2.25}$$

Recalling that $\|\mathcal{X}\|_F = \frac{1}{\sqrt{n_3}}\|\widehat{\mathcal{X}}\|_F$, the optimization problem in (2.25) can be decoupled into n_3 matrix nuclear norm minimization problems in the Fourier-transformed domain as

$$\widehat{\mathcal{X}^*}^{(i)} = \arg\min_{\mathbf{X}} \tau \|\mathbf{X}\|_* + \frac{1}{2n_3}\|\mathbf{X} - \widehat{\mathcal{Y}}^{(i)}\|_F^2, \tag{2.26}$$

for $i = 1, 2, \cdots, n_3$. For each i, the solution can be obtained by using the singular value thresholding (SVT) [25]. Let $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\text{H}}$ be the t-SVD of $\mathcal{X}$. We can define a tensor singular value thresholding (t-SVT) operator as

$$\mathcal{D}_\tau(\mathcal{X}) = \mathcal{U} * \mathcal{S}_\tau * \mathcal{V}^{\text{H}}, \tag{2.27}$$

where $\mathcal{S}_\tau$ can be computed by

$$\left(\mathcal{S} \times_3 \mathbf{F}_{n_3} - n_3\tau\right)_+ \times_3 \mathbf{F}_{n_3}^{-1} \tag{2.28}$$

[5] In [24], the TNN is defined in the original domain instead of the Fourier domain, while it is numerically equal to $\frac{1}{n_3}\sum_{i=1}^{n_3}\|\widehat{\mathcal{X}}^{(i)}\|_*$ and is a tight convex envelope of the tensor average-rank.

or

$$\texttt{ifft}\left((\texttt{fft}(\mathcal{S},[],3)-n_3\tau)_+,[],3\right), \tag{2.29}$$

where $(\cdot)_+$ indicates retaining the nonnegative part and setting the negative values to 0, and n_3 is the length of the third dimension. Then, the solution of (2.25) is $\mathcal{D}_\tau(\mathcal{Y})$.

Once the TNN is established, we provide a general iterative solving scheme based on the alternating direction method of multipliers (ADMM) [26] to optimize the TNN-based multidimensional image recovery model in (2.4). First, after introducing an auxiliary variable $\mathcal{Y}\in\mathbb{R}^{n_1\times n_2\times n_3}$, we rewrite (2.4) as

$$\min_{\mathcal{X},\mathcal{Y}}\{f(\mathcal{A}(\mathcal{X})-\mathcal{O})+\lambda\|\mathcal{Y}\|_{\text{TNN}}\},\quad \text{s.t. } \mathcal{Y}-\mathcal{X}=0. \tag{2.30}$$

Then, the argument Lagrangian function of (2.30) is given by

$$L_\beta(\mathcal{X},\mathcal{Y},\mathcal{M})=f(\mathcal{A}(\mathcal{X})-\mathcal{O})+\|\mathcal{Y}\|_{\text{TNN}}+\langle\mathcal{Y}-\mathcal{X},\mathcal{M}\rangle+\frac{\beta}{2}\|\mathcal{Y}-\mathcal{X}\|_F^2, \tag{2.31}$$

where $\mathcal{M}\in\mathbb{R}^{n_1\times n_2\times n_3}$ is the Lagrangian multiplier and β is a nonnegative parameter.

Next, at the k-th iteration, the ADMM updates each variable and the multiplier according to

$$\begin{cases}\mathcal{Y}^{k+1}=\arg\min\limits_{\mathcal{Y}}\ \|\mathcal{Y}\|_{\text{TNN}}+\dfrac{\beta}{2}\|\mathcal{Y}-\mathcal{X}^k+\dfrac{\mathcal{M}^k}{\beta}\|_F^2\\ \qquad\quad=\mathcal{D}_{\frac{1}{\beta}}\left(\mathcal{X}^k-\dfrac{\mathcal{M}^k}{\beta}\right),\\ \mathcal{X}^{k+1}=\arg\min\limits_{\mathcal{X}}\ f(\mathcal{A}(\mathcal{X})-\mathcal{O})+\dfrac{\beta}{2}\|\mathcal{Y}^{k+1}-\mathcal{X}+\dfrac{\mathcal{M}^k}{\beta}\|_F^2,\\ \mathcal{M}^{k+1}=\mathcal{M}+\beta(\mathcal{Y}-\mathcal{X}).\end{cases} \tag{2.32}$$

Commonly, the Frobenius norm is used to construct the data fidelity term, i.e., $f(\mathcal{A}(\mathcal{X})-\mathcal{O})=\frac{1}{2}\|\mathcal{A}(\mathcal{X})-\mathcal{O}\|_F^2$. Thus, the updating of $\mathcal{X}^{k+1}$ is described by

$$\mathcal{X}^{k+1}=\left(\mathcal{A}^*\mathcal{A}+\beta\mathcal{I}\right)^{-1}\left(\mathcal{A}^*(\mathcal{O})+\beta\mathcal{Y}^{k+1}+\mathcal{M}^k\right), \tag{2.33}$$

where $\mathcal{A}^*$ is the adjoint of $\mathcal{A}$ and $\mathcal{I}$ is the identical mapping.

Practically, this TNN minimization model outperforms the matrix-based method on the completion of videos [21] and denoising of HSIs [27]. Specifically, when the underlying low-tubal rank tensor satisfies the tensor incoherent condition (Eq. (25) in [22]), it can be exactly recovered by minimizing the TNN with high probability as long as the number of samples is of a certain order with respect to both the tensor size and tubal rank [22]. Additionally, Lu et al. [24] established the theoretical guarantee for the exact recovery of the tensor robust component analysis problem via minimizing the TNN. Zhang et al. [28] proposed a modified tensor principal component pursuit incorporating the prior subspace information to recover the low tubal rank and

the sparse components under a significantly weaker incoherence assumption. Furthermore, the t-SVD framework together with the TNN have been investigated for many other applications on multidimensional images, such as tensor recovery from binary measurements [29], robust tensor completion [30], hyperspectral image fusion [31], and subspace clustering [32,33].

2.2.4 Extensions

In this section, we will introduce some recent advances based on the t-SVD framework.

2.2.4.1 Nonconvex surrogates

For the matrix case, it has been discussed that the minimization of the matrix nuclear norm with the SVT operator will cause unavoidable biases in some situations. For example, it will cause rank deficiency, namely, the estimated results will be of lower rank than the original ground truth data. Meanwhile, the variance of the estimated data corresponding to the magnitude of singular values will also be smaller than that of the original data. The main origin of these effects is that the nuclear norm is a relaxation of the matrix rank, and all of the singular values are decreased equally when minimizing the nuclear norm with the SVT operator. To avoid this issue, many low-rank matrix recovery methods tailor weighted forms of the matrix nuclear norm or other nonconvex surrogates for the rank function. Below, we will give a brief review of similar advances under the t-SVD framework.

Let us first examine two typical examples. In [34], Oh et al. proposed to minimize the partial sum of singular values (PSSV). For a matrix $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$, its PSSV is defined as

$$\|\mathbf{X}\|_{p=N} \triangleq \sum_{i=N+1}^{\min(n_1,n_2)} \sigma_i(\mathbf{X}), \tag{2.34}$$

where $\sigma_i(\mathbf{X})$ denotes the i-th largest singular value of $\mathbf{X}$ for $i = 1, \ldots, \min(m, n)$. Meanwhile, Hu et al. [35] proposed the truncated nuclear norm of a matrix that is given by the matrix nuclear norm subtracted from the sum of the largest few singular values. Given a matrix $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$, the truncated nuclear norm has an definition equivalent to that of PSSV, while the authors minimized

$$\|\mathbf{X}\|_* - \mathrm{Tr}(\mathbf{A}_l \mathbf{X} \mathbf{B}_l^{\mathrm{T}}), \tag{2.35}$$

where $\mathrm{Tr}(\cdot)$ is the trace and the rows of $\mathbf{A}_l \in \mathbb{R}^{r \times n_1}$ and $\mathbf{B}_l \in \mathbb{R}^{r \times n_2}$ are taken from the singular vectors corresponding to the first r largest singular values of $\mathbf{X}$ at the l-th iteration. There are two obvious advantages of these two strategies in [34,35]: (i) rank deficiency is avoided by keeping the first N singular values; and (ii) the variance of the data is well preserved because large singular values are well retained.

In [36,37], Jiang et al. generalized the PSSV to the partial sum of TNN (PSTNN) for third-order tensors while Xue et al. generalized the truncated nuclear norm to the

truncated TNN. For a third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its PSTNN is given by

$$\|\mathcal{X}\|_{\text{PSTNN}} \triangleq \sum_{i=1}^{n_3} \|\widehat{\mathcal{X}}^{(i)}\|_{p=N}. \tag{2.36}$$

Meanwhile, the truncated TNN of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as

$$\|\mathcal{X}\|_{\text{TNN}} - \text{Tr}(\mathcal{A}_l * \mathcal{X} * \mathcal{B}_l^{\text{T}}), \tag{2.37}$$

where $\mathcal{A}_l \in \mathbb{R}^{r \times n_1 \times n_3}$ and $\mathcal{B}_l \in \mathbb{R}^{r \times n_2 \times n_3}$ come from the t-SVD of $\mathcal{X}$ at the l-th iteration. In [38], Zhang et al. proposed the corrected TNN based on a similar idea. The experimental results of [36–38] show that these generalizations are superior to the TNN for color image and video recovery.

More generally, for a matrix $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$, its weighted nuclear norm (WNN) [39] can be defined as

$$\|\mathbf{X}\|_{w,*} = \sum_i w_i \sigma_i(\mathbf{X}), \tag{2.38}$$

where $w_i \geq 0$ $(i = 1, 2, \cdots, \min(n_1, n_2))$ is a nonnegative weight. Then, the PSSV and the truncated nuclear norm fall into a special case of WNN with setting $w_i = 0$ for $i \leq r$ and $w_i = 1$ for $i > r$. In [39], the authors suggested an effective and efficient reweighting strategy by setting $w_i = \frac{c}{\sigma_i(\mathbf{X}_l)+\epsilon}$ at the $(l+1)$-th iteration, where c is the nonnegative regularization parameter and ϵ is a small positive number and is used to avoid division by zero. Huang et al. [40] and Liu et al. [41] extended the WNN to the third-order tensor case and proposed the weighted TNN. In their approaches, each singular value of $\widehat{\mathcal{X}}^{(i)}$ $(i = 1, 2, \cdots, n_3)$ is assigned a weight.

Many other nonconvex surrogates of the rank function (or the ℓ_0) have been considered for the horizontal slices of $\widehat{\mathcal{X}}$ (or the singular values of $\widehat{\mathcal{X}}$). Thus, these nonconvex surrogates are extended for third-order tensors under the t-SVD framework [42]. Next, some popular surrogates are listed. The nonconvex logdet function for a matrix in [43] has been applied for $\texttt{blockdiag}(\widehat{\mathcal{X}})$ by Ji et al. [44]. The logarithmic function [45] and the Laplace function [46] have been introduced to the singular values of $\texttt{blockdiag}(\widehat{\mathcal{X}})$ in [47] and [48], respectively. Additionally, the ℓ_p-norm $(0 < p < 1)$ of the matrix's singular values, i.e., the Schatten-p norm,[6] has been extended for third-order tensors in [49]. Moreover, Gao et al. [50] extended the weighted form of the Schatten-p norm [51] for third-order tensors. Wang et al. [52] utilized the prior knowledge about the frequency spectrum in the t-SVD framework and assigned different weights on the matrix nuclear norm of different frontal slices. Apart from the nonconvex surrogates of the rank function, low-rank matrix factorization techniques are also extended to low-tubal rank tensor factorization [53–55] in which the computational burden due to SVD computation is reduced.

[6] It is a quasinorm when $0 < p < 1$.

Other approaches tailoring the special structure of tensor singular values are reported in [56,57].

2.2.4.2 Additional prior knowledge

Enhancing the low-rankness of the underlying tensor by minimizing the convex TNN or the above nonconvex surrogates has been demonstrated to be effective for multidimensional image recovery. However, this approach may fail in two cases. First, many real-world multidimensional images consist of abundant fine details and are not strictly of low tubal rank. Second, when the sampled elements are extremely limited or structured, the rank of the observation will be low. Actually, being low-rank and sufficient samples are the basic assumptions in the theoretical guarantee for exact recovery when optimizing (2.4). When minimizing the nonconvex surrogates, those criteria may be relaxed but still hold. Thus, these two challenges are unavoidable in practical scenarios.

To address these issues, we can introduce additional prior knowledge for multidimensional image recovery. Multidimensional images can be viewed as a set of 2D natural images, such as a video consisting of a series of frames. Thus, the widely used piece-wise smoothness can be utilized via adding the total variation (TV) or framelet regularizer [58,59]. Moreover, the well-known nonlocal self-similarity has also been employed [60,61]. The TV regularizer was incorporated into the TNN-based multidimensional image recovery models by exploiting the local smoothness [62,63]. Recently, Zhao et al. [64] introduced an implicit regularizer and adopted a denoising deep convolution neural network (CNN) as the solution mapping of its corresponding subproblem under the plug-and-play framework. Since the deep learning-based methods can learn data-driven priors from a large number of natural images [65–67], the performance of the method in [64] is promising.

2.2.4.3 Multiple directions and higher-order tensors

Although the t-SVD framework has achieved great success, it is limited in that it treats the dimensions of third-order tensors differently. As discussed in [68], when computing the t-prod, the selection of the dimension to apply FFT[7] will result in quite different performance. Another obvious limitation is that t-SVD is suitable only for third-order tensors. When the multidimensional image is of fourth order, such as color videos with two spatial dimensions, one temporal and one color dimension, or even higher order, the tensor is suggested to be reshaped into the third order to fit the framework of t-SVD.

An extension of this method is to simultaneously minimize the TNN of the tensors with three kinds of permutation of its dimensions [5,69], that is, to permute the original third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with the scale of the third dimension as n_3, n_2, and n_1, respectively, and minimize the summation of their TNNs. After adopting

[7] One can twist or permute the dimensions of a third-order tensor. Thus, when computing t-prod, the Fourier transformation is conducted along the specified third mode.

this strategy, the performance is improved compared with minimizing the TNN. Additionally, it can be interpreted as (i) applying FFT along one specified dimension of a third-order tensor, (ii) obtaining a transformed tensor such as $\widehat{\mathcal{X}}$, and (iii) computing the nuclear norm of the slices supported by the remaining two dimensions.

Furthermore, for an N-th-order ($N \geq 3$) tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_N}$, we can select two modes and reshape the data into a third-order tensor while concatenating the remaining $N-2$ dimensions. Then, we can compute its TNN and sum them for different selections of these two modes [70,71]. However, since it is necessary to compute C_N^2 (the number of 2-combinations from a given set of N elements) TNNs for an N-th-order tensor, this approach may be time consuming when the order is high. Meanwhile, one important advantage of t-SVD is that it treats the third-order tensor data integrally, avoiding the information loss inherent in matricization or flattening. Thus, generalization of the t-prod, which is fundamental in the t-SVD framework, for higher-order tensors as in [72,73] may be more attractive. Nevertheless, the corresponding definition of tensor rank or TNN is still under development.

2.2.5 Summary

The t-prod plays a fundamental role for the construction of the t-SVD framework. The novel notion of the tensor rank and TNN are subsequently deduced. The capability of TNN-based models has been demonstrated in the literature, for example in [20–22,24,74]. Additionally, the extensions of the approach mentioned above can substantially enhance the performance in specific applications. Several numerical examples are given in Section 2.4.1. Nonetheless, in the next section, we delve into the t-SVD framework and explore another possible approach.

2.3 Transform-based t-SVD

In this section, we focus on the Fourier transform in the t-SVD. When computing the t-prod, the 1D DFT is applied on the mode-3 fibers to convert the circular convolution to element-wise multiplication. Thus, the tensor low-rankness within t-SVD can be viewed as the low-rankness in the transformed domain. For multidimensional images, the mode-3 fibers can be temporal vectors of a video or spectral vector of an HSI. From the signal processing perspective, the Fourier transform has certain limitations. Basically, in the implementation of the DFT or FFT within t-SVD, the padding boundary is periodic or circular by default. The circular boundary is undesirable for the temporal or spectral vectors in multidimensional images. Furthermore, the computation of DFT gives rise to complex values, leading to high computational complexity. Moreover, the Fourier transform leads to information loss that varies with time for the temporal vectors in videos or the spectral vectors of HSIs. Therefore, it is natural to find some substitutes that are more suitable for multidimensional images. Thus, below, we start with replacing the DFT by linear invertible transforms and end with a discussion of the possible noninvertible cases. With many basic ingredients given above, this section will be much briefer.

2.3.1 Linear invertible transform-based t-SVD

In [75], Madathil and George employed the discrete cosine transform (DCT) within the t-SVD framework for video completion and denoising. Xu et al. [76] and Lu et al. [77] also adopted the DCT as a substitution of DFT for tensor completion. DCT shows two obvious advantages over DFT under the t-SVD framework. First, the computation of DCT does not involve complex numbers, reducing the computational cost. Second, in the implementation of the DCT, the boundary condition along the tubes is reflexive. The experimental results of [76] illustrate that the quality of the frames (or bands) at the beginning and the end is strongly improved when minimizing the DCT-induced TNN. The frames (or bands) at the beginning and the end correspond to the head and the tail of the tubes, respectively. This shows that the reflexive boundary generated by DCT is more reasonable than the circular boundary required by DFT for videos or MSIs.

The origin of the use of DCT instead of DFT within the t-SVD framework can be traced back to [78], in which Kernfeld et al. noted that the t-prod, together with the tensor decomposition scheme, can be defined via the DCT along the third mode or even with any invertible linear transform. As mentioned in (2.15), the t-prod between two tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is equivalent to

$$\texttt{blockdiag}(\widehat{\mathcal{A}}) \cdot \texttt{blockdiag}(\widehat{\mathcal{B}}), \tag{2.39}$$

where $\widehat{\mathcal{A}} = \mathcal{A} \times_3 \mathbf{F}_{n_3}$ is the transformed tensor and is equivalent to $\widehat{\mathcal{B}}$. Otherwise, for simplicity, given two tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, we denote

$$\mathcal{A} \bigcirc \mathcal{B} \tag{2.40}$$

as the frontal-slice-wise product,[8] implying that

$$(\mathcal{A} \bigcirc \mathcal{B})^{(i)} = \mathcal{A}^{(i)} \cdot \mathcal{B}^{(i)} \text{ for } i = 1, 2, \cdots, n_3. \tag{2.41}$$

Therefore, we can define the t-prod with a given linear invertible transform via replacing the DFT matrix $\mathbf{F}_{n_3}$ by a selected linear invertible transform matrix. We denote the linear invertible transform matrix as $\mathbf{L} \in \mathbb{C}^{n_3 \times n_3}$, and it satisfies

$$\mathbf{L}^{-1} \cdot \mathbf{L} = \mathbf{L} \cdot \mathbf{L}^{-1} = \mathbf{I}_{n_3}, \tag{2.42}$$

where $\mathbf{L}^{-1}$ is the inverse transform matrix of $\mathbf{L}$ and $\mathbf{I}_{n_3} \in \mathbb{R}^{n_3 \times n_3}$ is the identity matrix. Then, the linear invertible transform-based t-prod can be defined.

Definition 2.9 (Transform-based t-prod [78]). Let $\mathbf{L} \in \mathbb{C}^{n_3 \times n_3}$ be a linear invertible transform matrix. The linear invertible transform-based t-prod between two tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is defined as

$$\mathcal{C} \in \mathbb{R}^{n_1 \times n_4 \times n_3} = \mathcal{A} *_{\mathbf{L}} \mathcal{B} = ((\mathcal{A} \times_3 \mathbf{L}) \bigcirc (\mathcal{B} \times_3 \mathbf{L})) \times_3 \mathbf{L}^{-1}. \tag{2.43}$$

[8] In Definition 2.1 of [78], it is also called face-wise product.

Table 2.2 The algorithm for computing transform-based t-SVD.

Input: $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathbf{L} \in \mathbb{C}^{n_3 \times n_3}$.

1. $\widehat{\mathcal{X}} \leftarrow \mathcal{X} \times_3 \mathbf{L}$.
4. **for** $i = 1$ to n_3 **do**
5. $\quad [U, S, V] = \mathtt{svd}(\widehat{\mathcal{X}}^{(i)})$.
6. $\quad \widehat{\mathcal{U}}^{(i)} \leftarrow U, \widehat{\mathcal{S}} \leftarrow S, \widehat{\mathcal{V}}^{(i)} \leftarrow V$
8. **end for**
9. $\mathcal{U} \leftarrow \widehat{\mathcal{U}} \times_3 \mathbf{L}^{-1}, \mathcal{S} \leftarrow \widehat{\mathcal{S}} \times_3 \mathbf{L}^{-1}, \mathcal{V} \leftarrow \widehat{\mathcal{V}} \times_3 \mathbf{L}^{-1}$.

Output: $\mathcal{U}$, $\mathcal{S}$, and $\mathcal{V}$.

The corresponding definitions, such as tensor transpose, identity tensor, and orthogonal tensor, with the linear invertible transform $\mathbf{L}$-induced t-prod, can be defined in the same manner as Definitions 2.3–2.6. To save space, we omit their specific definitions here and note that the definition of the f-diagonal tensor is still the same as Definition 2.5 since it is defined in the original domain rather than in the transform domain. Moreover, we note that the tensor transpose should satisfy (2.17) when it is defined. Interested readers are suggested to refer to [78] for details. After obtaining the transform-based t-prod, we can obtain the transform-based t-SVD.

Theorem 2.2 (Transform-based t-SVD [78]). *Let $\mathbf{L} \in \mathbb{C}^{n_3 \times n_3}$ be a linear invertible transform matrix. For a third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, it can be decomposed as*

$$\mathcal{X} = \mathcal{U} *_{\mathbf{L}} \mathcal{S} *_{\mathbf{L}} \mathcal{V}^{\mathrm{H}}, \tag{2.44}$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are the orthogonal tensors defined by using the transform-based t-prod and $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor whose frontal slices are diagonal matrices.

The linear invertible transform $\mathbf{L}$-induced t-SVD of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ can be efficiently obtained by computing a series of matrix SVDs on the frontal slices of $\mathcal{X} \times_3 \mathbf{L}$ in the transform domain, as shown in Table 2.2. Consistent with the t-SVD framework based on DFT, we can define the tensor tubal rank, multirank, and average rank based on the transform-based t-SVD. Since this chapter focuses on the practical usage of the low-rank regularization for multidimensional image recovery, we omit the specific tensor rank definitions that resemble Definition 2.7, and we turn directly to the convex surrogate. We can obtain the linear invertible transform-induced TNN under the transform-based t-SVD.

Definition 2.10 (Transform-based TNN). Let $\mathbf{L} \in \mathbb{C}^{n_3 \times n_3}$ be a linear invertible transform matrix. The linear invertible transform-based TNN of a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$is defined as

$$\|\mathcal{X}\|_{\mathbf{L}-\mathrm{TNN}} := \sum_{i=1}^{n_3} \left\| (\mathcal{X} \times_3 \mathbf{L})^{(i)} \right\|_*. \tag{2.45}$$

After obtaining the transform-based TNN, we can replace the TNN term in (2.4) with any given transform $\mathbf{L}$ for solving the multidimensional image recovery problem. This is also a convex problem and can be optimized via ADMM iterations shown in (2.32). For brevity, we do not go deep into the details.

In [79], Song et al. considered the general unitary transformation. Let $\mathbf{U} \in \mathbb{C}^{n_3 \times n_3}$ be the unitary transform matrix with

$$\mathbf{U}^{\mathrm{H}}\mathbf{U} = \mathbf{U}\mathbf{U}^{\mathrm{H}} = \mathbf{I}_{n_3}. \tag{2.46}$$

Based on the unitary transform, they proposed the transformed TNN as shown in (2.45). Similar to the original t-SVD framework, the transformed TNN is the convex envelope of the sum of the rank of $(\mathcal{X} \times \mathbf{U})$'s frontal slices. They also provide the theoretical guarantee of the exact recovery from incomplete and sparsely corrupted observations. Further, Zhang et al. employed the unitary-transformed TNN for tensor completion with Poisson observations [80]. The original t-SVD framework is a special case of the work by Song et al. [79] if setting the transformed matrix as $\frac{1}{\sqrt{n_3}}\mathbf{F}_{n_3}$.

Additionally, Lu et al. [77] considered the general linear invertible transformation $\mathbf{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ that satisfies

$$\mathbf{L}^{\mathrm{T}}\mathbf{L} = \mathbf{L}\mathbf{L}^{\mathrm{T}} = l\mathbf{I}_{n_3}, \tag{2.47}$$

where $\cdot^{\mathrm{T}}$ denotes the matrix transpose and $l > 0$ is a constant. Based on this linear invertible transform, they deduced the new tensor tubal rank, tensor spectral norm, and TNN.[9] Under certain tensor incoherent conditions, one can exactly recover the underlying data from incomplete observations [77].

2.3.2 Beyond invertibility and data adaptivity

According to the theoretical results rigorously established in [77,79], the original t-SVD framework that is implemented with the DFT is a special case of the t-SVD framework based on the linear invertible transform. Thus, the extensions introduced in Section 2.2.3 can be directly generalized for the linear invertible transform-based t-SVD. For example, in [75], the authors proposed a 3D DCT-based TNN minimization model for multidimensional image denoising and completion. In [81], the transform-based t-prod between third-order tensors is generalized to N-th-order tensors ($N \geq 3$).

However, one key issue in the transform-based t-SVD framework, i.e., the design or choice of the transform is still challenging. Empirically, transformations that can make the original data lower-rank in the transformed domain are preferred. However, a linear invertible transform will not essentially change the tensor ranks defined in Definition 2.7. Therefore, we need to delve more deeply into the inner mechanism of minimizing transform-based TNN for multidimensional image recovery.

[9] In [77], the TNN based on linear transformation is similar to (2.45) but multiplied by a factor $\frac{1}{l}$.

In [79], Song et al. pointed out that with a suitable transformation that can redistribute tubal entries into the matrix slices (along the tube direction) of the tensors, low-rank transformed matrix slices can be generated. They illustrate the distribution of all the frontal slices' singular values from the transformed data under different transformations. More small singular values are generated after the wavelet transform than by the DFT. Thus, the wavelet-transformed (along the tubes) data are closer to lower-rank data.

Recently, Jiang et al. [82] adopted the tight wavelet frame (framelet) as the transformation. The framelet system used in [82] is generated from the B-spline and implemented in a multiresolution manner [83]. The transform matrix can be written as $\mathbf{W} \in \mathbb{R}^{wn_3 \times n_3}$, where $w \geq 1$ is an integer and $\mathbf{W}$ satisfies the unitary extension principle (UEP) [84], i.e.,

$$\mathbf{W}^{\mathrm{T}}\mathbf{W} = \mathbf{I}_{n_3}, \tag{2.48}$$

where $\mathbf{W}^{\mathrm{T}}$ is indeed the inverse transform matrix of $\mathbf{W}$. It is clear that $\mathbf{W}\mathbf{W}^{\mathrm{T}} \neq \mathbf{I}_{wn_3}$. This means that the framelet transform is semi-invertible. Nonetheless, Jiang et al. directly defined the framelet-based TNN (FTNN) as $\sum_{i=1}^{wn_3} \| (\mathcal{X} \times_3 \mathbf{W})^{(i)} \|_*$ and replaced the TNN term in (2.4) by the FTNN. Then, this model can be optimized via the ADMM. Their method can be understood as an iterative method with iterations consisting of: (i) performing the framelet transform; (ii) regularizing the nuclear norm of the transformed data's frontal slices; and (iii) carrying out the inverse framelet transform. Owing to the UEP, these iterations are not difficult.

In [82], the authors showed that the framelet transform along the third mode can generate more small singular values of the transformed data's frontal slices than DCT for different types of multidimensional imaging data. Consequently, their results are also better than those obtained by minimizing the DCT-deduced TNN or the original TNN obtained using DFT. Furthermore, they also computed the numerical rank of the transformed data's frontal slices. Given a tolerance for computing the numerical rank, the framelet-transformed data can be better approximated with lower rankness, generating better recovery results. Thus, we can deduce an empirical rule for choosing the transform. That is, the transform that can generate lower-rank frontal slices in the transformed domain for a specific type of data and for which the inverse transform is not difficult to obtain should be chosen.

In the above discussion, we only mentioned the predefined transform. If we intend to find the transform satisfying the empirical rule for different types of data, a more flexible approach is to use data adaptive transform. In addition to the wavelet transform, Song et al. [79] also adopted a data-dependent unitary transform. They first optimized (2.4) and set the obtained result as an initial guess $\mathcal{X}_0 \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Then, the first r left singular vectors of $\mathcal{X}_0$'s mode-3 unfolding matrix are extracted to constitute the transform matrix. The performance of this strategy is better than that obtained using the predefined wavelet transform.

Meanwhile, Kong et al. [85] constructed a principal component extraction matrix $\mathbf{Q}$ that maximizes the distribution variance of $\{\| (\mathcal{X} \times_3 \mathbf{Q})^{(i)} \|_F\}$. The matrix $\mathbf{Q}$ can be obtained from the left singular vectors of mode-3 unfolding of $\mathcal{X}$. Interestingly,

the transform matrix $\mathbf{Q}$ in [85] will be updated in their algorithm during the iterative process. In their experiments, they also use an oracle $\mathbf{Q}$ that is computed from the complete data and its performance is surprisingly good. This confirms that a data-dependent transform is powerful and flexible.

In [86], Jiang et al. replaced the inverse transform, i.e., $\mathbf{F}_{n_3}^{-1}$, with a data adaptive term. Their model can be written as

$$\min_{\mathcal{Z},\mathbf{D}} \sum_{i=1}^{n_3} \|\mathcal{Z}^{(i)}\|_* \quad \text{s.t. } (\mathcal{Z} \times_3 \mathbf{D})_\Omega = \mathcal{O}_\Omega, \tag{2.49}$$

where $\mathbf{D} \in \mathbb{R}^{n_3 \times d}$ is a data adaptive dictionary, $d \gg n_3$ is a positive integer, and the ℓ_2-norm of $\mathbf{D}$'s columns should be less than or equal to 1. From their learned dictionaries, it can be found that the atoms will vary for different types of data and adaptively fit the variation of tubes.

2.4 Numerical experiments

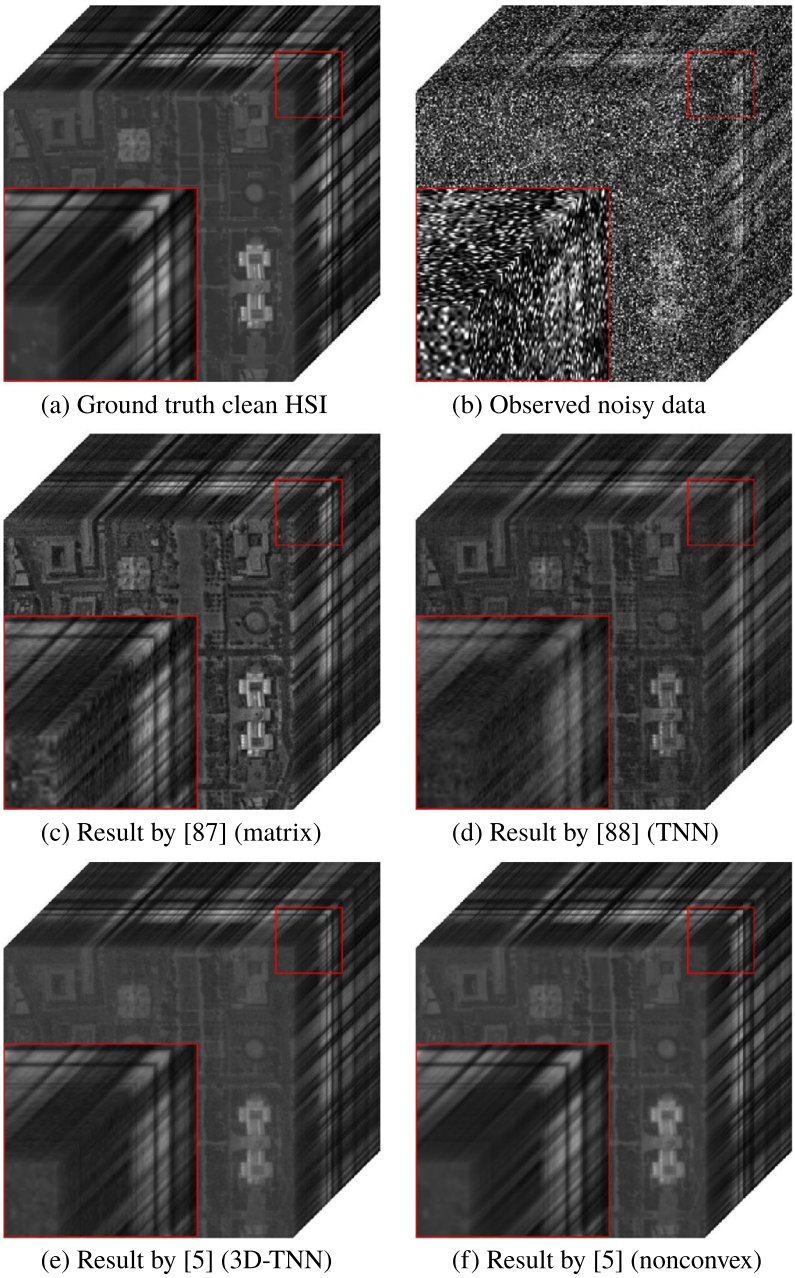
In this part, we provide several numerical examples on different kinds of applications, including hyperspectral image denoising & completion, color image demosaicing, and (hyperspectral) video completion. The traditional t-SVD-based methods and their extensions are discussed in Section 2.4.1. Section 2.4.2 reports the results from the transform-based t-SVD methods.

2.4.1 Examples within the t-SVD framework

As mentioned above, we show three numerical experiments with approaches within the traditional t-SVD framework. One experiment is simulated on an HSI that is a subimage of the *Washington DC Mall* dataset.[10] This HSI is of the size $256 \times 256 \times 191$, so that it contains 256×256 spatial pixels and 191 spectral bands. The clean image is normalized with entries in each band in the interval $[0, 1]$. The noisy HSI is simulated by adding Gaussian noise with standard deviation of 0.15 and 20% of the entries are corrupted by the salt and pepper noise. The compared methods are a low-rank matrix-based HSI denoising method in [87], a TNN minimization model [88], and a 3D TNN-based model and its nonconvex version [5]. We show the recovered results in Fig. 2.1. It is observed that the results obtained by minimizing the 3D TNN and its nonconvex surrogate are visually better than the results obtained by minimizing TNN or by using the low-rank matrix-based method.

In the second experiment, a hyperspectral video (HSV)[11] of the size $120 \times 120 \times 33 \times 31$ is selected. Specifically, this hyperspectral video has 31 frames and each

[10] http://lesun.weebly.com/hyperspectral-data-set.html.
[11] http://openremotesensing.net/knowledgebase/hyperspectral-video/.



(a) Ground truth clean HSI (b) Observed noisy data

(c) Result by [87] (matrix) (d) Result by [88] (TNN)

(e) Result by [5] (3D-TNN) (f) Result by [5] (nonconvex)

FIGURE 2.1

The recovered results on the noisy HSI.

frame has 33 bands from 400 nm to 720 nm wavelength with a 10-nm step [89]. In Fig. 2.3, we show two images located at different frames and different bands in the HSV recovered by the TNN minimization method [22] and the weighted sum of TNN (WSTNN) minimization method [70]. WSTNN is designed for the N-th-order ($N \geq 3$) tensor while we reshape the HSV to a third-order tensor of the size $120 \times 120 \times 1023$ by combining its spectral and temporal modes. We can observe that the WSTNN is evidently superior to the TNN, particularly in the recovery of texture information.

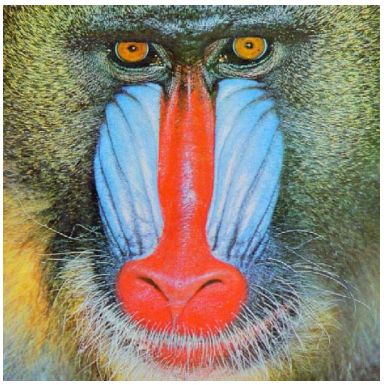
The third experiment is conducted on the color image named "*baboon*".[12] This color image contains 512×512 spatial pixels and three color channels. For the color image, we consider the typical demosaicing problem that aims to estimate the color image from the Bayer pattern sampling. That is, each two-by-two cell contains two green, one blue, and one red cell. We first use the SNN minimization model [2] and the TNN minimization model [22]. Then, we employ a TV regularized TNN minimization method in [62] and a CNN denoiser regularized method [64], in which an implicit regularization term is also added to input the CNN denoisers expressing the deep prior learned from a large amount of training images. Fig. 2.2 shows the results obtained by the different methods. It is observed that the SNN and TNN minimization models almost failed for this structural sampling case. After adding additional regularizers, "TV+TNN" and "CNN+TNN" successfully recovered this color image.

The first example shows that the performance of TNN can be strongly improved by considering multiple directions and using the nonconvex surrogate. As shown in the second example, the strategy of considering multiple directions can be generalized well for higher-order tensors. It is found from the third example that the capability of TNN will be limited when the sampling is structured and the introduction of additional prior knowledge is useful.

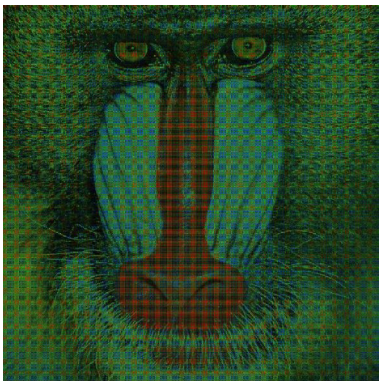
2.4.2 Examples of the transform-based t-SVD

In this part, we test methods that are deduced from the transform-based t-SVD. We display two experimental results on the HSI data "*Pavia City Center*"[13] of the size $200 \times 200 \times 80$ (height $\times$ width $\times$ band) and the video "*foreman*"[14] of the size $144 \times 176 \times 50$, respectively. Since the aforementioned transform-based TNN methods mainly concentrate on the low-rank tensor completion problem, we conduct the completion experiment in this part. The sampling rate for the video is 50% while 5% pixels are sampled for the HSI. The involved methods are the original TNN minimization model using DFT and its variants using DCT, framelet, and the dictionary. For a quantitative comparison between different methods, we select two widely used image quality assessments, namely, the peak signal-to-noise ratio (PSNR) and the structural

[12] http://sipi.usc.edu/database/database.php.
[13] http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.
[14] http://trace.eas.asu.edu/yuv/.

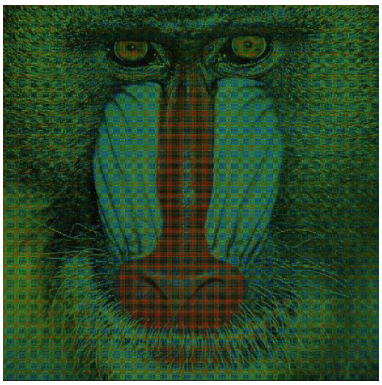


(a) Ground truth complete data



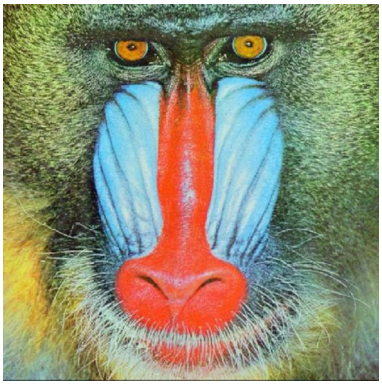
(b) Observed incomplete data

(c) Result by [2] (SNN)



(d) Result by [22] (TNN)

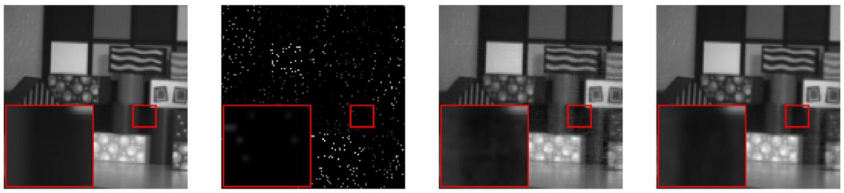
(e) Result by [62] (TV+TNN)



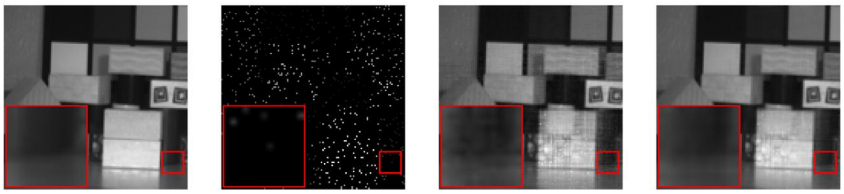
(f) Result by [64] (CNN+TNN)

FIGURE 2.2

The demosaicing results on the color image data "*baboon*".



(a) Ground truth (b) Observation (c) Result by [22] (TNN) (d) Result by [70] (WSTNN)



(e) Ground truth (f) Observation (g) Result by [22] (TNN) (h) Result by [70] (WSTNN)

FIGURE 2.3

The completion results of the HSV with SR = 5%. (a)–(d) The image located at the 15th band and the 7th frame. (e)–(h) The image located at the 25th band and the 30th frame.

similarity index (SSIM) [90]. Higher values indicate better recovery quality. They are computed on each frontal slice and the mean value is adopted.

We display the visual results together with the PSNR and SSIM values in Figs. 2.4 and 2.5. It is observed that for the video data and the HSI data, DCT-based TNN outperforms the original TNN that uses DFT. The framelet-based TNN achieves the second-best performance. The data-dependent dictionary-based TNN obtains the best visual performance and the highest PSNR and SSIM values.

2.5 Conclusions and new guidelines

In this chapter, focusing on the multidimensional imaging data recovery problem, we revisit the establishment of the t-SVD framework and the TNN minimization model. Some extensions such as nonconvex surrogates and adding additional regularization terms are introduced. Furthermore, we delve more deeply into the t-SVD framework, and replace the DFT with the linear invertible transform. Then, we discuss the cases of noninvertible transforms and data adaptive transform. Overall, the prevailing trend of the development based on the t-SVD framework can be summarized as:

- from the *convex* model to the *nonconvex* model,
- from the *single* low-rankness to the *multiple* regularization terms,
- from the *predefined* transform to the *data adaptive* transform.

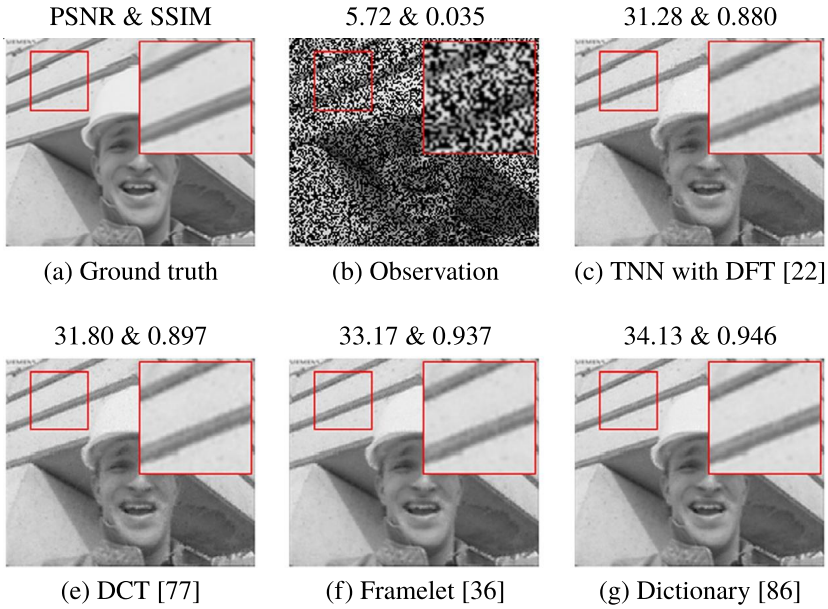


FIGURE 2.4

The 22nd frame of the results on the video data "*foreman*."

Thus, in the future, for the recovery of multidimensional images, there are several possible directions and challenges. The *first* is to establish the theoretical guarantee of the exact recovery for low-rank tensors via minimizing the weighted form of the TNN. Extending the leverage score [91] from matrices to tensors may be a feasible approach for achieving this goal. Although in this chapter we focus on the practical use of the TNN (or the transform-based TNN) and its variants, only the theoretical guarantee will make those approaches really practical without additional concerns. The *second* direction lies in exploiting reasonable prior knowledge of the multidimensional images under the maximum a posteriori estimation framework. It is expected to analyze the distinct low-rank structures along different dimensions [92], or even some artificially defined dimensions, e.g., the nonlocal self-similarity dimension. Furthermore, data-driven priors can also be simultaneously considered. *Third*, as in [82,86], additional transforms or dictionaries can be considered to be embedded in the t-SVD framework for better representation of the data implicit low-rankness, such as the nonlinear transform and the convolutional dictionary. Moreover, the theoretical guarantee for noninvertible transform-based TNN minimization methods is still difficult and new techniques for their analysis are needed. *Last*, to face the challenges arising from massive data, acceleration techniques such as the randomized algorithm in [93–95] can be considered. Moreover, the combination of the above points is also promising. For example, in [4], the unitary transform-based TNN is adopted to characterize the similarity of nonlocal patches.

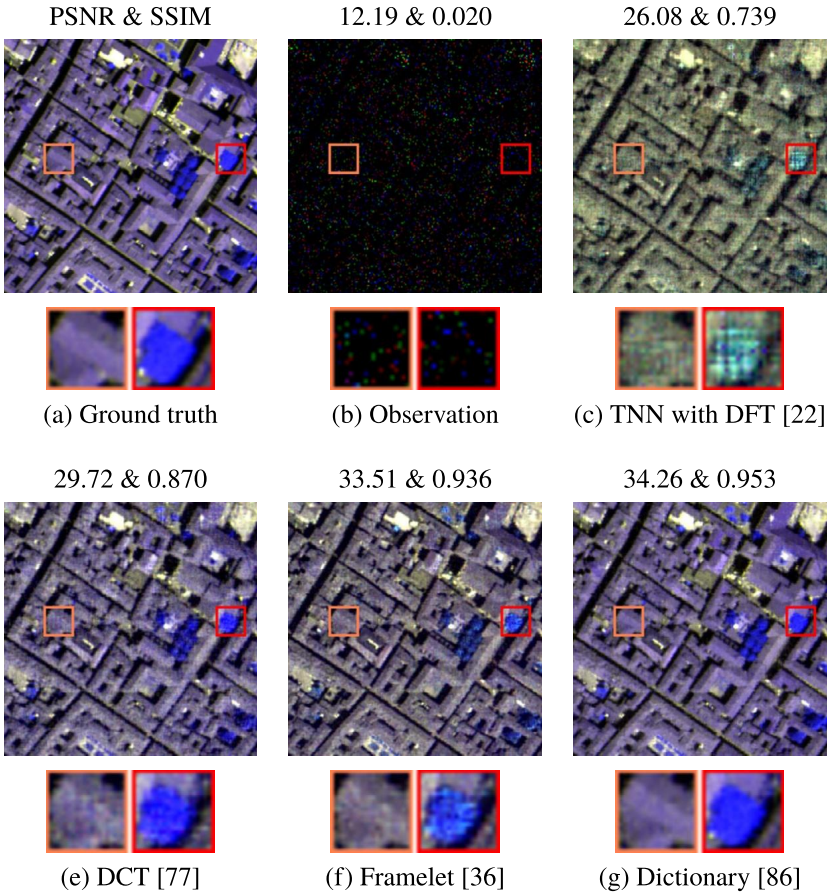


FIGURE 2.5

The pseudo-color images (R-4 G-12 B-68) and the corresponding enlarged areas of the results obtained by different methods on the HSI data "*Pavia City Center*."

References

[1] E.J. Candès, B. Recht, Exact matrix completion via convex optimization, Foundations of Computational Mathematics 9 (6) (2009) 717.

[2] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1) (2013) 208–220.

[3] E.J. Candes, Y. Plan, Matrix completion with noise, Proceedings of the IEEE 98 (6) (2010) 925–936.

[4] M.K. Ng, X. Zhang, X.-L. Zhao, Patched-tube unitary transform for robust tensor completion, Pattern Recognition 100 (2020) 107181.

[5] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-H. Ma, T.-Y. Ji, Mixed noise re-

moval in hyperspectral image via low-fibered-rank regularization, IEEE Transactions on Geoscience and Remote Sensing 58 (1) (2019) 734–749.

[6] G.T. Herman, A.R. De Pierro, N. Gai, On methods for maximum a posteriori image reconstruction with a normal prior, Journal of Visual Communication and Image Representation 3 (4) (1992) 316–324.

[7] J.M. Bioucas-Dias, J.M. Nascimento, Hyperspectral subspace identification, IEEE Transactions on Geoscience and Remote Sensing 46 (8) (2008) 2435–2445.

[8] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Review 51 (3) (2009) 455–500.

[9] Y. Wang, D. Meng, M. Yuan, Sparse recovery: from vectors to tensors, National Science Review 5 (5) (2017) 756–767.

[10] Z. Long, Y. Liu, L. Chen, C. Zhu, Low rank tensor completion for multiway visual data, Signal Processing 155 (2019) 301–316.

[11] N.D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E.E. Papalexakis, C. Faloutsos, Tensor decomposition for signal processing and machine learning, IEEE Transactions on Signal Processing 65 (13) (2017) 3551–3582.

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

[13] R.A. Harshman, et al., Foundations of the PARAFAC Procedure: Models and Conditions for an "Explanatory" Multimodal Factor Analysis, University of California at Los Angeles, Los Angeles, CA, 1970.

[14] C. Zeng, T.-X. Jiang, M.K. Ng, An approximation method of CP rank for third-order tensor completion, Numerische Mathematik (2021) 1–31.

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

[16] T.-Y. Ji, T.-Z. Huang, X.-L. Zhao, T.-H. Ma, L.-J. Deng, A non-convex tensor rank approximation for tensor completion, Applied Mathematical Modelling 48 (2017) 410–422.

[17] W. Cao, Y. Wang, C. Yang, X. Chang, Z. Han, Z. Xu, Folded-concave penalization approaches to tensor completion, Neurocomputing 152 (2015) 261–273.

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

[19] M.E. Kilmer, K. Braman, N. Hao, R.C. Hoover, Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging, SIAM Journal on Matrix Analysis and Applications 34 (1) (2013) 148–172.

[20] O. Semerci, N. Hao, M.E. Kilmer, E.L. Miller, Tensor-based formulation and nuclear norm regularization for multienergy computed tomography, IEEE Transactions on Image Processing 23 (4) (2014) 1678–1693.

[21] Z. Zhang, G. Ely, S. Aeron, N. Hao, M. Kilmer, Novel methods for multilinear data completion and de-noising based on tensor-SVD, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Columbus, OH, USA, 2014, pp. 3842–3849.

[22] Z. Zhang, S. Aeron, Exact tensor completion using t-SVD, IEEE Transactions on Signal Processing 65 (6) (2017) 1511–1526.

[23] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, S. Yan, Tensor robust principal component analysis: exact recovery of corrupted low-rank tensors via convex optimization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5249–5257.

[24] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, S. Yan, Tensor robust principal component analysis with a new tensor nuclear norm, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (4) (2020) 925–938.
[25] J.-F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on Optimization 20 (4) (2010) 1956–1982.
[26] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (1) (2011) 1–122.
[27] H. Fan, Y. Chen, Y. Guo, H. Zhang, G. Kuang, Hyperspectral image restoration using low-rank tensor recovery, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 10 (10) (2017) 4589–4604.
[28] F. Zhang, J. Wang, W. Wang, C. Xu, Low-tubal-rank plus sparse tensor recovery with prior subspace information, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020), https://doi.org/10.1109/TPAMI.2020.2986773.
[29] J. Hou, F. Zhang, H. Qiu, J. Wang, Y. Wang, D. Meng, Robust low-tubal-rank tensor recovery from binary measurements, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021), https://doi.org/10.1109/TPAMI.2021.3063527.
[30] Q. Jiang, M. Ng, Robust low-tubal-rank tensor completion via convex optimization, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, International Joint Conferences on Artificial Intelligence Organization, Macao, China, 2019, pp. 2649–2655.
[31] R. Dian, S. Li, Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization, IEEE Transactions on Image Processing 28 (10) (2019) 5135–5146.
[32] M. Cheng, L. Jing, M.K. Ng, Tensor-based low-dimensional representation learning for multi-view clustering, IEEE Transactions on Image Processing 28 (5) (2018) 2399–2414.
[33] M. Yin, J. Gao, S. Xie, Y. Guo, Multiview subspace clustering via tensorial t-product representation, IEEE Transactions on Neural Networks and Learning Systems 30 (3) (2019) 851–864.
[34] T. Oh, Y. Tai, J. Bazin, H. Kim, I.S. Kweon, Partial sum minimization of singular values in robust PCA: algorithm and applications, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (4) (2016) 744–758.
[35] Y. Hu, D. Zhang, J. Ye, X. Li, X. He, Fast and accurate matrix completion via truncated nuclear norm regularization, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (9) (2013) 2117–2130.
[36] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm, Journal of Computational and Applied Mathematics 372 (2020) 112680.
[37] S. Xue, W. Qiu, F. Liu, X. Jin, Low-rank tensor completion by truncated nuclear norm regularization, in: 2018 24th International Conference on Pattern Recognition (ICPR), IEEE, 2018, pp. 2600–2605.
[38] X. Zhang, M.K. Ng, A corrected tensor nuclear norm minimization method for noisy low-rank tensor completion, SIAM Journal on Imaging Sciences 12 (2) (2019) 1231–1273.
[39] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, L. Zhang, Weighted nuclear norm minimization and its applications to low level vision, International Journal of Computer Vision 121 (2) (2017) 183–208.
[40] Y. Huang, G. Liao, L. Zhang, Y. Xiang, J. Li, A. Nehorai, Efficient narrowband RFI mitigation algorithms for SAR systems with reweighted tensor structures, IEEE Transactions on Geoscience and Remote Sensing 57 (11) (2019) 9396–9409.

[41] M. Liu, X. Zhang, L. Tang, Real color image denoising using t-product-based weighted tensor nuclear norm minimization, IEEE Access 7 (2019) 182017–182026.
[42] H. Wang, F. Zhang, J. Wang, T. Huang, J. Huang, X. Liu, Generalized nonconvex approach for low-tubal-rank tensor recovery, IEEE Transactions on Neural Networks and Learning Systems (2021) 1–15, https://doi.org/10.1109/TNNLS.2021.3051650.
[43] W. Dong, G. Shi, X. Li, Y. Ma, F. Huang, Compressive sensing via nonlocal low-rank regularization, IEEE Transactions on Image Processing 23 (8) (2014) 3618–3632.
[44] T. Ji, T. Huang, X. Zhao, D. Sun, A new surrogate for tensor multirank and applications in image and video completion, in: 2017 International Conference on Progress in Informatics and Computing (PIC), 2017, pp. 101–107.
[45] J.H. Friedman, Fast sparse regression and classification, International Journal of Forecasting 28 (3) (2012) 722–738.
[46] J. Trzasko, A. Manduca, Highly undersampled magnetic resonance image reconstruction via homotopic ℓ_0-minimization, IEEE Transactions on Medical Imaging 28 (1) (2008) 106–121.
[47] L. Chen, X. Jiang, X. Liu, Z. Zhou, Robust low-rank tensor recovery via nonconvex singular value minimization, IEEE Transactions on Image Processing 29 (2020) 9044–9059.
[48] W.-H. Xu, X.-L. Zhao, T.-Y. Ji, J.-Q. Miao, T.-H. Ma, S. Wang, T.-Z. Huang, Laplace function based nonconvex surrogate for low-rank tensor completion, Signal Processing. Image Communication 73 (2019) 62–69, tensor Image Processing.
[49] H. Kong, X. Xie, Z. Lin, T-Schatten-p norm for low-rank tensor recovery, IEEE Journal of Selected Topics in Signal Processing 12 (6) (2018) 1405–1419.
[50] Q. Gao, P. Zhang, W. Xia, D. Xie, X. Gao, D. Tao, Enhanced tensor RPCA and its application, IEEE Transactions on Pattern Analysis and Machine Intelligence 43 (6) (2021) 2133–2140.
[51] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, L. Zhang, Weighted Schatten p-norm minimization for image denoising and background subtraction, IEEE Transactions on Image Processing 25 (10) (2016) 4842–4857.
[52] S. Wang, Y. Liu, L. Feng, C. Zhu, Frequency-weighted robust tensor principal component analysis, arXiv preprint, arXiv:2004.10068, 2020.
[53] P. Zhou, C. Lu, Z. Lin, C. Zhang, Tensor factorization for low-rank tensor completion, IEEE Transactions on Image Processing 27 (3) (2018) 1152–1163.
[54] M. Cheng, L. Jing, M.K. Ng, A weighted tensor factorization method for low-rank tensor completion, in: 2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM), 2019.
[55] X.-L. Lin, M.K. Ng, X.-L. Zhao, Tensor factorization with total variation and Tikhonov regularization for low-rank tensor completion in imaging data, Journal of Mathematical Imaging and Vision 62 (6) (2020) 900–918.
[56] Y. Liu, L. Chen, C. Zhu, Improved robust tensor principal component analysis via low-rank core matrix, IEEE Journal of Selected Topics in Signal Processing 12 (6) (2018) 1378–1389.
[57] L. Feng, Y. Liu, L. Chen, X. Zhang, C. Zhu, Robust block tensor principal component analysis, Signal Processing 166 (2020) 107271.
[58] X. Li, Y. Ye, X. Xu, Low-rank tensor completion with total variation for visual data inpainting, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI Press, 2017, pp. 2210–2216.
[59] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, T.-Y. Ji, L.-J. Deng, Matrix factorization for low-rank tensor completion using framelet prior, Information Sciences 436 (2018) 403–417.

[60] X.-T. Li, X.-L. Zhao, T.-X. Jiang, Y.-B. Zheng, T.-Y. Ji, T.-Z. Huang, Low-rank tensor completion via combined non-local self-similarity and low-rank regularization, Neurocomputing 367 (2019) 1–12.
[61] T. Xie, S. Li, L. Fang, L. Liu, Tensor completion via nonlocal low-rank regularization, IEEE Transactions on Cybernetics 49 (6) (2019) 2344–2354.
[62] F. Jiang, X.Y. Liu, H. Lu, R. Shen, Anisotropic total variation regularized low-rank tensor completion based on tensor nuclear norm for color image inpainting, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 1363–1367.
[63] H. Fan, C. Li, Y. Guo, G. Kuang, J. Ma, Spatial–spectral total variation regularized low-rank tensor decomposition for hyperspectral image denoising, IEEE Transactions on Geoscience and Remote Sensing 56 (10) (2018) 6196–6213.
[64] X.-L. Zhao, W.-H. Xu, T.-X. Jiang, Y. Wang, M.K. Ng, Deep plug-and-play prior for low-rank tensor completion, Neurocomputing 400 (2020) 137–149.
[65] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising, IEEE Transactions on Image Processing 26 (7) (2017) 3142–3155.
[66] K. Zhang, W. Zuo, L. Zhang, FFDNet: toward a fast and flexible solution for CNN-based image denoising, IEEE Transactions on Image Processing 27 (9) (2018) 4608–4622.
[67] L. Zhang, W. Zuo, Image restoration: from sparse and low-rank priors to deep priors, IEEE Signal Processing Magazine 34 (5) (2017) 172–179.
[68] W. Hu, D. Tao, W. Zhang, Y. Xie, Y. Yang, The twist tensor nuclear norm for video completion, IEEE Transactions on Neural Networks and Learning Systems 28 (12) (2016) 2961–2973.
[69] D. Wei, A. Wang, X. Feng, B. Wang, B. Wang, Tensor completion based on triple tubal nuclear norm, Algorithms 11 (7) (2018) 94.
[70] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-Y. Ji, T.-H. Ma, Tensor N-tubal rank and its convex relaxation for low-rank tensor recovery, Information Sciences 532 (2020) 170–189.
[71] A. Wang, C. Li, Z. Jin, Q. Zhao, Robust tensor decomposition via orientation invariant tubal nuclear norms, Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, AAAI Press, 2020, pp. 6102–6109.
[72] C.D. Martin, R. Shafer, B. LaRue, An order-p tensor factorization with applications in imaging, SIAM Journal on Scientific Computing 35 (1) (2013) A474–A490.
[73] X.-Y. Liu, X. Wang, Fourth-order tensors with multidimensional discrete transforms, arXiv preprint, arXiv:1705.01576, 2017.
[74] F. Zhang, W. Wang, J. Huang, J. Wang, Y. Wang, RIP-based performance guarantee for low-tubal-rank tensor recovery, Journal of Computational and Applied Mathematics 374 (2020) 112767.
[75] B. Madathil, S.N. George, Dct based weighted adaptive multi-linear data completion and denoising, Neurocomputing 318 (2018) 120–136.
[76] W.-H. Xu, X.-L. Zhao, M. Ng, A fast algorithm for cosine transform based tensor singular value decomposition, arXiv preprint, arXiv:1902.03070, 2019.
[77] C. Lu, X. Peng, Y. Wei, Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 5989–5997.
[78] E. Kernfeld, M. Kilmer, S. Aeron, Tensor–tensor products with invertible linear transforms, Linear Algebra and Its Applications 485 (2015) 545–570.

[79] G. Song, M.K. Ng, X. Zhang, Robust tensor completion using transformed tensor singular value decomposition, Numerical Linear Algebra with Applications 27 (3) (May 2020).
[80] X. Zhang, M.K.P. Ng, Low rank tensor completion with Poisson observations, IEEE Transactions on Pattern Analysis and Machine Intelligence (2021), https://doi.org/10.1109/TPAMI.2021.3059299.
[81] J. Han, p-order tensor products with invertible linear transforms, arXiv preprint, arXiv: 2005.11477, 2020.
[82] T.-X. Jiang, M.K. Ng, X.-L. Zhao, T.-Z. Huang, Framelet representation of tensor nuclear norm for third-order tensor completion, IEEE Transactions on Image Processing 29 (2020) 7233–7244.
[83] J.-F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on Optimization 20 (4) (2010) 1956–1982.
[84] A. Ron, Z. Shen, Affine systems in $L_2(R^d)$: the analysis of the analysis operator, Journal of Functional Analysis 148 (2) (1997) 408–447.
[85] H. Kong, C. Lu, Z. Lin, Tensor Q-rank: a new data dependent tensor rank, Machine Learning (2021), https://doi.org/10.1007/s10994-021-05987-8.
[86] T.-X. Jiang, X.-L. Zhao, H. Zhang, M.K. Ng, Dictionary learning with low-rank coding coefficients for tensor completion, IEEE Transactions on Neural Networks and Learning Systems (2021), https://doi.org/10.1109/TNNLS.2021.3104837.
[87] H. Zhang, W. He, L. Zhang, H. Shen, Q. Yuan, Hyperspectral image restoration using low-rank matrix recovery, IEEE Transactions on Geoscience and Remote Sensing 52 (8) (2013) 4729–4743.
[88] H. Fan, Y. Chen, Y. Guo, H. Zhang, G. Kuang, Hyperspectral image restoration using low-rank tensor recovery, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 10 (10) (2017) 16.
[89] A. Mian, R. Hartley, Hyperspectral video restoration using optical flow and sparse coding, Optics Express 20 (10) (2012) 10658–10673.
[90] ZhouWang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612.
[91] Y. Chen, S. Bhojanapalli, S. Sanghavi, R. Ward, Completing any low-rank matrix, provably, Journal of Machine Learning Research 16 (94) (2015) 2999–3034.
[92] Y. Chang, L. Yan, B. Chen, S. Zhong, Y. Tian, Hyperspectral image restoration: where does the low-rank property exist, IEEE Transactions on Geoscience and Remote Sensing (2020) 1–16.
[93] D.A. Tarzanagh, G. Michailidis, Fast randomized algorithms for t-product based tensor operations and decompositions with applications to imaging data, SIAM Journal on Imaging Sciences 11 (4) (2018) 2629–2664.
[94] J. Zhang, A.K. Saibaba, M.E. Kilmer, S. Aeron, A randomized tensor singular value decomposition based on the t-product, Numerical Linear Algebra with Applications 25 (5) (2018) e2179.
[95] M. Che, Y. Wei, H. Yan, The computation of low multilinear rank approximations of tensors via power scheme and random projection, SIAM Journal on Matrix Analysis and Applications 41 (2) (2020) 605–636.