

Article

Tensor Completion Based on Triple Tubal Nuclear Norm

Dongxu Wei ^{1,2,3,*}, Andong Wang ⁴ , Xiaoqin Feng ¹, Boyu Wang ⁵ and Bo Wang ⁶

¹ School of Physics and Electronic Electrical Engineering, Huaiyin Normal University, Huaian 223300, China; fengxq2003@126.com

² School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China

³ Jiangsu Province Key Construction Laboratory of Modern Measurement Technology and Intelligent System, Huaian 223300, China

⁴ School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; andong.wang.1990@gmail.com

⁵ Jiangsu Yaoshi Software Technology Co., Ltd., Nanjing 211103, China; antonw@163.com

⁶ Jiangsu Shuoshi Welding Technology Co., Ltd., Nanjing 211103, China; lukewb@163.com

* Correspondence: weidx@hytc.edu.cn

Received: 21 May 2018; Accepted: 19 June 2018; Published: 28 June 2018

Abstract: Many tasks in computer vision suffer from missing values in tensor data, i.e., multi-way data array. The recently proposed tensor tubal nuclear norm (TNN) has shown superiority in imputing missing values in 3D visual data, like color images and videos. However, by interpreting in a circulant way, TNN only exploits tube (often carrying temporal/channel information) redundancy in a circulant way while preserving the row and column (often carrying spatial information) relationship. In this paper, a new tensor norm named the triple tubal nuclear norm (TriTNN) is proposed to simultaneously exploit tube, row and column redundancy in a circulant way by using a weighted sum of three TNNs. Thus, more spatial-temporal information can be mined. Further, a TriTNN-based tensor completion model with an ADMM solver is developed. Experiments on color images, videos and LiDAR datasets show the superiority of the proposed TriTNN against state-of-the-art nuclear norm-based tensor norms.

Keywords: tensor completion; tensor SVD; ADMM; image inpainting; video inpainting

1. Introduction

In recent decades, the rapid progress in multi-linear algebra [1] has provided a firm theoretical foundation for many applications in computer vision [2], data mining [3], machine learning [4], signal processing [5], and many other areas. Benefiting from its multi-way nature, the tensor has power against the vector and matrix in exploiting multi-way information in multi-modal data, like color images [6], videos [7], hyper-spectral images [8], functional magnetic resonance imaging [9], traffic volume data [10], etc. In many computer vision tasks, the data, like color images or videos, may be moderately redundant, then it can be interpreted by fewer latent factors [11]. The low-rank tensor provides a suitable model for such data [12]. The two most well-known low-rank tensor models are the low-CP-rank model [13], which tries to interpret a tensor in the fewest rank-one tensors [14], and the low-Tucker-rank model [15], which seeks a tensor proxy that is simultaneously low-rank along each mode.

In many computer vision applications, like image or video inpainting, one has to tackle the missing values in the observed data tensor due to many circumstances [2,16], including failure of sensors, errors or loss in communication, occlusions or noise in the environment, etc. However, it is

obviously unable to fill in the missing entries perfectly since they can take arbitrary values without other priors taken into consideration. The most adopted prior is the low-rank prior assuming the underlying data tensor has low rank. Low-rank tensor completion [2,17] seeks a low-rank tensor to fit the underlying data tensor. It has been a hot research topic due its wide use [18]. In low-rank tensor recovery, the rank minimization problem (RMP) is often formulated [2]. However, the general rank minimization problem and most tensor problems are NP-hard [19,20]. To obtain polynomial-time algorithms, many different tensor rank surrogates have been proposed [2,7,17,21–24] to substitute the rank functions in RMP. Surrogates of the tensor CP rank and Tucker rank have been broadly studied [7,17,23,25–29].

Recently, a novel low-rank tensor model called the low-tubal-rank model was proposed [22,30]. The core of it is to model the 3D data as a tensor that has low tubal-rank [31], which is defined through a new tensor singular value decomposition (t-SVD) [1,32]. It has been successfully used in modeling multi-way real-world data, such as color images [6], videos [33], seismic data [34], WiFi fingerprint [35], MRI imaging [22], traffic volume data [36], etc. As pointed out in [37], compared with other tensor models, the low-tubal-rank tensor model is superior in capturing a “spatial-shifting” correlation, which is ubiquitous in real-world data arrays.

This paper focuses on low-tubal-rank models for tensor completion. The recently-proposed tensor tubal nuclear norm (TNN) [30] based on t-SVD has shown superiority in imputing missing values in 3D visual data, like color images and videos. Its power lies in exploiting tube (often carrying temporal/channel information) redundancy in a circulant way while preserving the row and column (often carrying spatial information) relationship. A simple and successful variant of TNN, dubbed the twist tubal nuclear norm (t-TNN) [16], instead exploits row redundancy in a circulant way while keeping the tube relationship. However, both of them only exploit one kind of redundancy in a circulant way. In this paper, a new tensor norm dubbed the tensor triple tubal nuclear norm (TriTNN) is proposed to simultaneously exploit the row, column and tube redundancy while preserving the relative tube, row and column relationship. Based on the proposed TriTNN, a tensor completion model is studied and optimized by alternating direction multiplier methods (ADMM) [38]. Experimental results on color images, videos and LiDAR datasets demonstrate that the proposed TriTNN has better performances than other state-of-the-art nuclear norm-based tensor norms.

The paper is organized as follows. Some notations and preliminaries are presented in Section 2. The TriTNN is proposed following the introductions of the most related works in Section 3. The problem formulation and the proposed ADMM algorithm are shown in Section 4. Experimental results are reported in Section 5. We conclude this work in Section 6.

2. Notations and Preliminaries

In this section, the notations and the basic definitions are introduced.

2.1. Notations

Vectors are denoted by bold lower case letters, e.g., $v \in \mathbb{R}^n$, matrices are denoted by bold upper case letters, e.g., $M \in \mathbb{R}^{n_1 \times n_2}$, and tensors are denoted by calligraphy letters, e.g., $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Given a third-order tensor, its fiber is defined as a 1D vector obtained by fixing all indices, but one, and its slice is a 2D matrix defined by fixing all but two indices. Given a 3D tensor \mathcal{T} , \mathcal{T}_{ijk} denotes the entry with index (i, j, k) ; $T^{(k)} := \mathcal{T}(:, :, k)$ denotes the k -th frontal slice. $\tilde{\mathcal{T}}$ denotes the tensor after performing the fast Fourier transformation along the tube fibers of \mathcal{T} . Notations $\text{dft3}(\cdot)$ and $\text{idft3}(\cdot)$ are used to represent the discrete Fourier transformation (DFT) and inverse discrete Fourier transformation (IDFT) along the tube fibers of 3D tensors.

Given a matrix $M \in \mathbb{R}^{n_1 \times n_2}$, its nuclear norm is defined as $\|M\|_* := \sum_{i=1}^p \sigma_i(M)$, where $p = \min\{n_1, n_2\}$ and $\sigma_1(M) \geq \dots \geq \sigma_p(M)$ are the singular values of M in non-ascending order. The inner product between two 3D tensors $\mathcal{T}_1, \mathcal{T}_2 \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined

as $\langle \mathcal{T}_1, \mathcal{T}_2 \rangle := \sum_{ijk} \mathcal{T}_1(i, j, k) \mathcal{T}_2(i, j, k)$. The Frobenius norm of a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as $\|\mathcal{T}\|_F := \sqrt{\sum_{ijk} \mathcal{T}_{ijk}^2}$. The l_∞ -norm of a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as $\|\mathcal{T}\|_\infty := \max_{ijk} |\mathcal{T}_{ijk}|$.

2.2. Tensor Singular Value Decomposition

Firstly, five block-based operators, i.e., `bvec`, `bvfold`, `bdiag`, `bdfold` and `bcirc` [1], are introduced. Given a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the block vectorizing and its opposite operation are defined as follows:

$$\text{bvec}(\mathcal{T}) := \begin{bmatrix} \mathcal{T}^{(1)} \\ \mathcal{T}^{(2)} \\ \vdots \\ \mathcal{T}^{(n_3)} \end{bmatrix}, \quad \text{bvfold}(\text{bvec}(\mathcal{T})) = \mathcal{T},$$

the block diag matrix and its opposite operation:

$$\text{bdiag}(\mathcal{T}) := \begin{bmatrix} \mathcal{T}^{(1)} & & & \\ & \ddots & & \\ & & \mathcal{T}^{(n_3)} & \\ & & & \end{bmatrix}, \quad \text{bdfold}(\text{bdiag}(\mathcal{T})) = \mathcal{T},$$

and the block circulant matrix as follows:

$$\text{bcirc}(\mathcal{T}) := \begin{bmatrix} \mathcal{T}^{(1)} & \mathcal{T}^{(n_3)} & \dots & \mathcal{T}^{(2)} \\ \mathcal{T}^{(2)} & \mathcal{T}^{(1)} & \dots & \mathcal{T}^{(3)} \\ \vdots & \ddots & \ddots & \vdots \\ \mathcal{T}^{(n_3)} & \mathcal{T}^{(n_3-1)} & \dots & \mathcal{T}^{(1)} \end{bmatrix}.$$

Based on the five operators defined above, we are able to give the definition of the tensor t-product.

Definition 1 (t-product [1]). Let $\mathcal{T}_1 \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{T}_2 \in \mathbb{R}^{n_2 \times n_4 \times n_3}$. The t-product of \mathcal{T}_1 and \mathcal{T}_2 is a tensor \mathcal{T} of size $n_1 \times n_4 \times n_3$:

$$\mathcal{T} = \mathcal{T}_1 * \mathcal{T}_2 =: \text{bvfold}\{\text{bcirc}(\mathcal{T}_1)\text{bvec}(\mathcal{T}_2)\}. \tag{1}$$

Viewing a 3D tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ as an $n_1 \times n_2$ matrix of tubes, the tensor t-product is analogous to the matrix multiplication by replacing scalar multiplication with the vector circular convolution between the tubes, as follows:

$$\mathcal{T}(i, j, :) = \sum_{k=1}^{n_2} \mathcal{T}_1(i, k, :) \bullet \mathcal{T}_2(k, j, :), \tag{2}$$

where \bullet denotes the circular convolution [1] between two tube vector $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n_3}$ defined as:

$$(\mathbf{x} \bullet \mathbf{y})_j = \sum_{k=1}^{n_3} \mathbf{x}_k \mathbf{y}_{1+(j-k) \bmod n_3}.$$

Due to the relationship between the circular convolution and the DFT, the t-product in the original domain is equivalent to matrix multiplication of the frontal slices in the Fourier domain [1], i.e.,

$$\widetilde{\mathcal{T}}^{(k)} = \widetilde{\mathcal{T}}_1^{(k)} \widetilde{\mathcal{T}}_2^{(k)}, \quad k = 1, \dots, n_3. \tag{3}$$

The tensor transpose, identity tensor, f-diagonal tensor and orthogonal tensor are further defined.

Definition 2 (Tensor transpose [1]). Let \mathcal{T} be a tensor of size $n_1 \times n_2 \times n_3$; then \mathcal{T}^\top is the $n_2 \times n_1 \times n_3$ tensor obtained by transposing each of the frontal slices and then reversing the order of transposed frontal Slices 2 through n_3 .

Definition 3 (Identity tensor [1]). The identity tensor $\mathcal{I} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is a tensor whose first frontal slice is the $n_1 \times n_1$ identity matrix, and all other frontal slices are zero.

Definition 4 (F-diagonal tensor [1]). A tensor is called *f-diagonal* if each frontal slice of the tensor is a diagonal matrix.

Definition 5 (Orthogonal tensor [1]). A tensor $\mathcal{Q} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is orthogonal if it satisfies the following relationship:

$$\mathcal{Q}^\top * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^\top = \mathcal{I}.$$

Based on the concepts defined above, the tensor singular value decomposition (t-SVD) and the tensor tubal rank are established as follows.

Definition 6 (Tensor singular value decomposition and tensor tubal-rank [31]). Any tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ can be decomposed as:

$$\mathcal{T} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top, \tag{4}$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors and \mathcal{S} is a rectangular *f-diagonal* tensor of size $n_1 \times n_2 \times n_3$.

The tensor tubal rank of \mathcal{T} is defined to be the number of non-zero tubes of \mathcal{S} in Equation (4), i.e.,

$$r_{\text{tubal}}(\mathcal{T}) := \sum_i \mathbf{1}(\mathcal{S}(i, i, :) \neq \mathbf{0}), \tag{5}$$

where $\mathbf{1}(\cdot)$ is an indicator function whose value is one if the input condition is satisfied, and zero otherwise.

The t-SVD is illustrated in Figure 1. It can be computed efficiently by FFT and IFFT in the Fourier domain according to Equation (3). For more details, see [1].

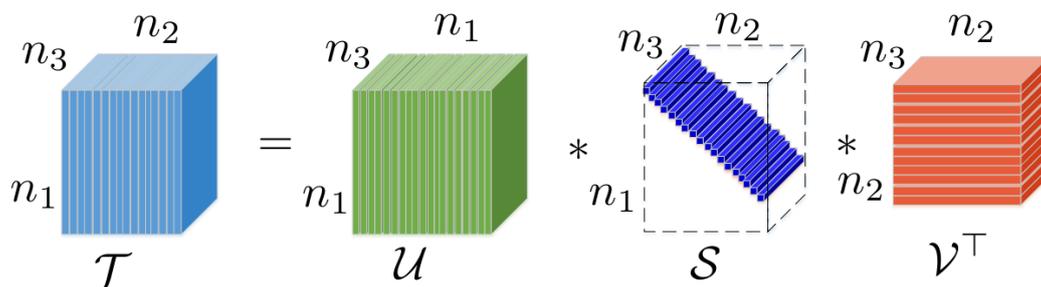


Figure 1. Illustration of tensor (t)-SVD.

3. The Triple Tubal Nuclear Norm

In this section, we will define the triple tensor tubal nuclear norm, before which the most related norms, i.e., tubal nuclear norm and twist tubal nuclear norm, are introduced first.

3.1. Tubal Nuclear Norm

Based on the preliminaries introduced in Section 2, the tubal nuclear norm is defined as follows:

Definition 7 (Tubal nuclear norm [31]). *The tubal nuclear norm (TNN) $\|\mathcal{T}\|_*$ of a 3D tensor \mathcal{T} is defined as the nuclear norm of the block diagonal matrix of $\tilde{\mathcal{T}}$ (the Fourier domain version of \mathcal{T}), i.e.,*

$$\|\mathcal{T}\|_* := \|\text{bdiag}(\tilde{\mathcal{T}})\|_* = \sum_{k=1}^{n_3} \|\tilde{\mathcal{T}}^{(k)}\|_* \tag{6}$$

From Definition 7, we can compute the TNN of a tensor efficiently through first conducting FFT along the tube direction and summing the nuclear norms of each frontal slice. Given a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the computation cost of $\|\mathcal{T}\|_*$ is $O(n_1 n_2 n_3 (n_3 + \log n_3))$. Since a block circulant matrix can be block diagonalized through the Fourier transform [1], we obtain:

$$\|\mathcal{T}\|_* = \|\text{bdiag}(\tilde{\mathcal{T}})\|_* = \|(\mathbf{F}_{n_3} \otimes \mathbf{I}_{n_1}) \text{bcirc}(\mathcal{T}) (\mathbf{F}_{n_3}^{-1} \otimes \mathbf{I}_{n_2})\|_* = \|\text{bcirc}(\mathcal{T})\|_*, \tag{7}$$

where \otimes denotes the Kronecker product [14], \mathbf{F}_n is the $n \times n$ discrete Fourier transform matrix and \mathbf{I}_n is an $n \times n$ identity matrix. Note that $(\mathbf{F}_{n_3} \otimes \mathbf{I}_{n_1}) / \sqrt{n_3}$ is a unitary matrix.

The tubal nuclear norm has been used as a convex relaxation of the tensor tubal-rank for tensor completion, tensor robust principle component analysis (TRPCA) and outlier robust tensor principle component analysis (OR-TPCA) [6,30,36,39]. In optimization over TNN, one often needs to compute the proximal operator [40] of TNN defined as:

$$\mathfrak{S}_\tau(\mathcal{T}_0) := \underset{\mathcal{T}}{\text{argmin}} \frac{1}{2} \|\mathcal{T}_0 - \mathcal{T}\|_{\mathbb{F}}^2 + \tau \|\mathcal{T}\|_* \tag{8}$$

In [3], a closed-form expression of $\mathfrak{S}_\tau(\cdot)$ is given as follows:

Definition 8 ([3]). *For a 3D tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with reduced t-SVD $\mathcal{T} = \mathbf{U} * \mathcal{S} * \mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\mathbf{V} \in \mathbb{R}^{n_2 \times r \times n_3}$ are orthogonal tensors and $\mathcal{S} \in \mathbb{R}^{r \times r \times n_3}$ is the f-diagonal tensor of singular tubes, the proximal operator $\mathfrak{S}_\tau(\cdot)$ at \mathcal{T}_0 can be computed through the following equation:*

$$\mathfrak{S}_\tau(\mathcal{T}) := \mathbf{U} * \text{idft3}(\max(\text{dft3}(\mathcal{S}) - n_3 \tau, 0)) * \mathbf{V}^\top \tag{9}$$

3.2. Twist Tubal Nuclear Norm

The twist tubal nuclear norm [16] is related to a pair of tensor operations named column twist and column squeeze as follows.

Definition 9 (Tensor column twist and column squeeze (here, the twist and squeeze operations in [1,16] are renamed as column twist and column squeeze, respectively, since we will define the row twist and row squeeze) [16]). *Let $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, then the column twist tensor $\mathcal{T}_1 = \vec{\mathcal{T}}$ is a tensor of size $n_1 \times n_3 \times n_2$ whose lateral slice $\mathcal{T}_1(:, k, :) = \mathcal{T}^{(k)}$. Correspondingly, the column squeeze tensor of \mathcal{T}_1 , i.e., $\mathcal{T} = \overleftarrow{\mathcal{T}}_1$, can be obtained by the reverse process, i.e., $\mathcal{T}^{(k)} = \text{ColSqueeze}(\mathcal{T}_1(:, k, :))$. See Figure 2.*

Then, we give the definition of twist nuclear norm as follows:

Definition 10 (Twist tensor nuclear norm [16]). *The twist tensor nuclear norm (t-TNN) based on the t-SVD framework is defined as follows:*

$$\|\mathcal{T}\|_{\vec{x}} := \|\vec{\mathcal{T}}\|_* = \|\text{bcirc}(\vec{\mathcal{T}})\|_* \tag{10}$$

From the above definition, t-TNN can be computed efficiently by first twisting the original tensor and then computing the TNN. Given a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the computation cost of $\|\mathcal{T}\|_{\vec{x}}$ is $O(n_1 n_2 n_3 (n_2 + \log n_2))$. The t-TNN has attained significant improvement against the TNN in video

inpainting [16]. Since the t-TNN is essentially a TNN, the proximal operator of it can be derived from the proximal operator of TNN as follows:

$$\text{ColSqueeze}(\mathfrak{S}_\tau(\vec{\mathcal{T}}_0)) = \underset{\mathcal{T}}{\text{argmin}} \frac{1}{2} \|\mathcal{T}_0 - \mathcal{T}\|_F^2 + \tau \|\mathcal{T}\|_\star.$$

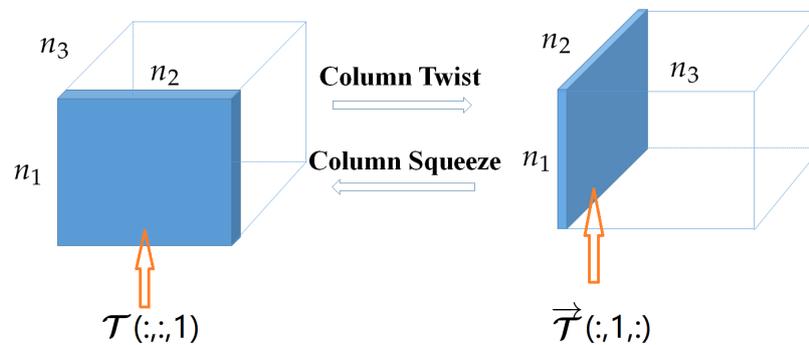


Figure 2. The column twist and column squeeze operations.

3.3. A Circular Interpretation of TNN and t-TNN

In this subsection, an illustration of TNN and t-TNN in a circular way [16], which motivates the proposal of TriTNN, will be given. For a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we define the operation called circulant block matricization of \mathcal{T} [41] in the following equation:

$$\text{circ}(\mathcal{T}) := \begin{bmatrix} \text{circ}(\mathcal{T}(1, 1, :)) & \text{circ}(\mathcal{T}(1, 2, :)) & \cdots & \text{circ}(\mathcal{T}(1, n_2, :)) \\ \text{circ}(\mathcal{T}(2, 1, :)) & \text{circ}(\mathcal{T}(2, 2, :)) & \cdots & \text{circ}(\mathcal{T}(2, n_2, :)) \\ \vdots & \vdots & \ddots & \vdots \\ \text{circ}(\mathcal{T}(n_1, 1, :)) & \text{circ}(\mathcal{T}(n_1, 2, :)) & \cdots & \text{circ}(\mathcal{T}(n_1, n_2, :)) \end{bmatrix} \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}, \quad (11)$$

where $\text{circ}(x)$ denotes an $n_3 \times n_3$ matrix with $x \in \mathbb{R}^{n_3}$ in the following way:

$$\text{circ}(x) := \begin{bmatrix} \text{circ}(x_1) & \text{circ}(x_{n_3}) & \cdots & \text{circ}(x_2) \\ \text{circ}(x_2) & \text{circ}(x_1) & \cdots & \text{circ}(x_3) \\ \vdots & \vdots & \ddots & \vdots \\ \text{circ}(x_{n_3}) & \text{circ}(x_{n_3-1}) & \cdots & \text{circ}(x_1) \end{bmatrix} \in \mathbb{R}^{n_3 \times n_3}. \quad (12)$$

By the permutation operation, there exist two so-called stride permutation matrices [42] P_1 and P_2 , such that the following relationship between $\text{circ}(\mathcal{T})$ and $\text{bcirc}(\mathcal{T})$ holds:

$$\text{circ}(\mathcal{T}) = P_1 \text{bcirc}(\mathcal{T}) P_2. \quad (13)$$

Since the matrix nuclear norm is permutation invariant, it holds that [16]:

$$\|\text{circ}(\mathcal{T})\|_\star = \|\text{bcirc}(\mathcal{T})\|_\star. \quad (14)$$

As an example, Figure 3a,b intuitively shows the relationships between the original tensor, the column twist tensor, the block circulant matrix and the circulant block matricization of a tensor $\mathcal{T} \in \mathbb{R}^{3 \times 3 \times 3}$. As illustrated in Subplots (a) and (b) of Figure 3, from the circulant perspective, TNN essentially exploits the tube redundancy in a circulant way while keeping the row and column relationship, and t-TNN essentially exploits the row redundancy in a circulant way while preserving the tube and column relationship [16]. In computer vision applications, the row and column of a data tensor (like color images or videos) often carry spatial information, and the tube often carries temporal

or channel information. From the computational perspective, FFT is operated along the tube direction to compute TNN, while t-TNN needs FFT along the row direction.

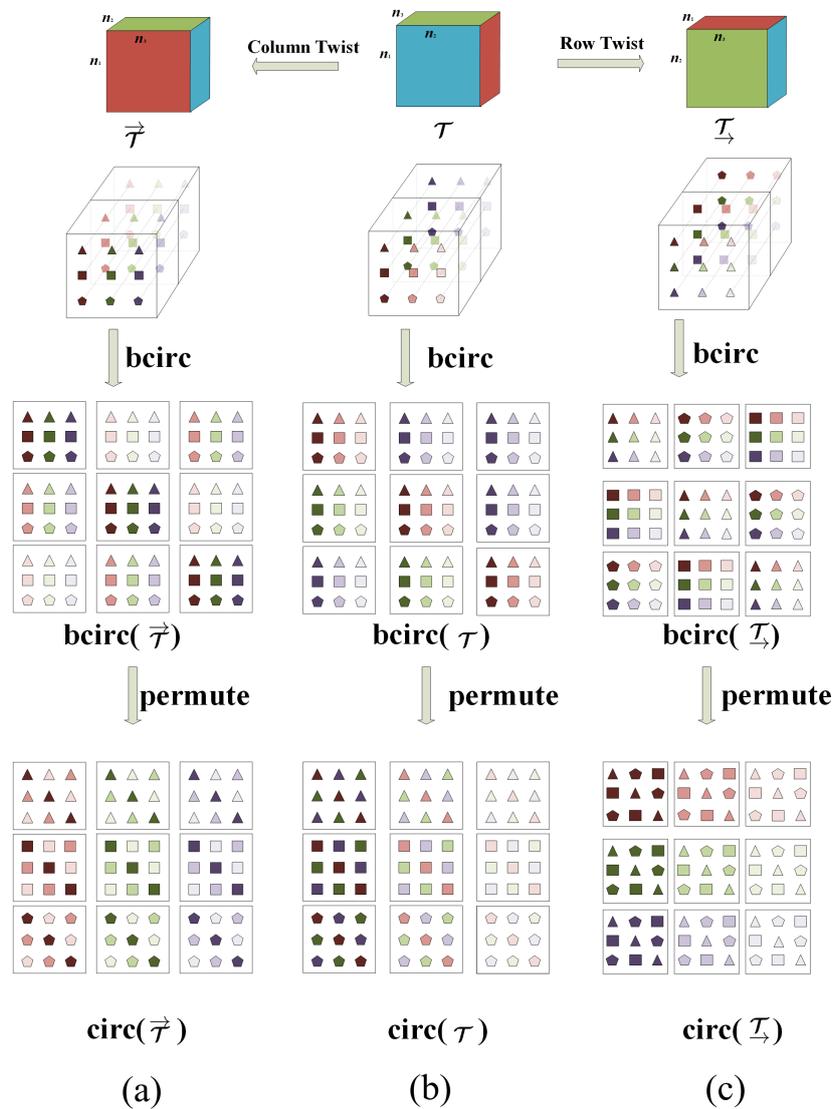


Figure 3. An intuitive illustration of relationships between the original tensor, the column twist tensor, the row twist tensor, the block circulant matrix and the circulant block matricization of a tensor $\mathcal{T} \in \mathbb{R}^{3 \times 3 \times 3}$. Subplots (a–c) show the operations on the column twist tensor, the original tensor and the row twist tensor, respectively. It can be seen that $\text{circ}(\mathcal{T})$ exploits the tube redundancy in a circulant way while keeping the row and column relationship, $\text{circ}(\vec{\mathcal{T}})$ exploits the row redundancy in a circulant way while keeping the tube and column relationship and $\text{circ}(\vec{\mathcal{T}})$ exploits the column redundancy in a circulant way while keeping the tube and row relationship.

3.4. The Proposed Row Twist Tubal Nuclear Norm and Triple Tubal Nuclear Norm

As discussed above, TNN and t-TNN need FFT along the tube and row direction, respectively. Note that, for real-world visual data, like color images, the row and column carry homogeneous information, and they are better treated equally. A simple operation similar to the column twist and column squeeze, called row twist and row squeeze, respectively, is defined.

Definition 11 (Tensor row twist and row squeeze). Let $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, then the row twist tensor $\mathcal{T}_1 = \underline{\mathcal{T}}_{\rightarrow}$ is a tensor of size $n_3 \times n_2 \times n_1$ whose horizontal slice $\mathcal{T}_1(k, :, :) = \mathcal{T}^{(k)}$. Correspondingly, the row squeeze tensor of \mathcal{T}_1 , i.e., $\mathcal{T} = \underline{\mathcal{T}}_1$, can be obtained by the reverse process, i.e., $\mathcal{T}^{(k)} = \text{RowSqueeze}(\mathcal{T}_1(k, :, :))$. See Figure 4.

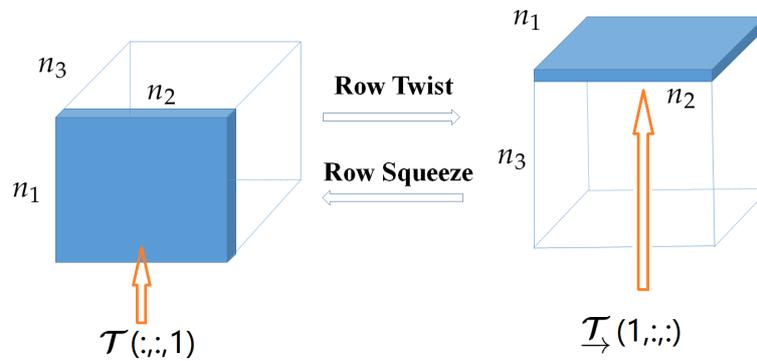


Figure 4. The row twist and row squeeze operations.

Definition 12 (Row twist tubal nuclear norm (rt-TNN)). The row twist tubal nuclear norm (rt-TNN) of a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as the tubal nuclear norm of its row twisted tensor, i.e.,

$$\|\mathcal{T}\|_{\underline{\star}} = \|\underline{\mathcal{T}}_{\rightarrow}\|_{\star} = \|\text{bcirc}(\underline{\mathcal{T}}_{\rightarrow})\|_{\star}. \tag{15}$$

As illustrated in Subplot (c) of Figure 3, from the circulant perspective, rt-TNN essentially exploits the column redundancy in a circulant way while keeping the row and tube relationship. From the computational perspective, FFT is operated along the column direction to compute rt-TNN. The proximal operator of rt-TNN can be derived from the proximal operator of TNN as follows:

$$\text{RowSqueeze}(\mathfrak{S}_{\tau}(\underline{\mathcal{T}}_0)) = \underset{\mathcal{T}}{\text{argmin}} \frac{1}{2} \|\mathcal{T}_0 - \mathcal{T}\|_{\mathbb{F}}^2 + \tau \|\mathcal{T}\|_{\underline{\star}}.$$

It should be noted that each of TNN, t-TNN and rt-TNN only exploits one type of redundancy, i.e., the tube, row and column redundancies, in a circulant way. Real-world data may have more than one type of redundancy, and it is beneficial to exploit such a property. To simultaneously exploit the tube, row and column redundancy in a circulant way while keeping other relationships, we simply combine the TNN, t-TNN and rt-TNN to get the triple tubal nuclear norm.

Definition 13 (Triple tubal nuclear norm). The triple tubal nuclear norm (TriTNN) is defined as a weighted sum of its tubal nuclear norm, column twist tubal nuclear norm and its row twist tubal nuclear norm, i.e.,

$$\|\mathcal{T}\|_{\star} = \lambda_1 \|\mathcal{T}\|_{\star} + \lambda_2 \|\mathcal{T}\|_{\overrightarrow{\star}} + \lambda_3 \|\mathcal{T}\|_{\underline{\star}}, \tag{16}$$

where λ_1, λ_2 and λ_3 are positive weights satisfying:

$$\lambda_1 + \lambda_2 + \lambda_3 = 1. \tag{17}$$

From the above definition, the computation of $\|\mathcal{T}\|_{\star}$ can be divided into computations of TNN, t-TNN and rt-TNN, which has the following computational complexity:

$$O(n_1 n_2 n_3 (n_1 + n_2 + n_3 + \log n_1 + \log n_2 + \log n_3)). \tag{18}$$

Due to the coupling of three tubal nuclear norms, it is very difficult to derive a closed-form expression of the proximal operator of TriTNN.

4. TriTNN-Based Tensor Completion

4.1. Problem Formulation

Let $\mathcal{L}^* \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ be the true, but unknown tensor to be completed. Suppose only a small fraction of its entries are observed and the observations are corrupted by small dense noise. Let $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ denote the observed noisy tensor of \mathcal{L}^* . Then, we have the following observation model:

$$\mathcal{T} = (\mathcal{L}^* + \mathcal{E}) \odot \mathcal{O}, \tag{19}$$

where $\mathcal{E} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the noise tensor with element-wisely i.i.d. Gaussian noise, \odot denotes the element-wise multiplication and $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ denotes the binary tensor whose entry $\mathcal{O}_{ijk} = 1$ if the (i, j, k) -th entry is observed, otherwise $\mathcal{O}_{ijk} = 0$. The goal is to estimate \mathcal{L}^* given noisy observation \mathcal{T} from observation Model (19).

We estimate \mathcal{L}^* by simultaneously exploiting the tube, row and column redundancy in a circular way through minimizing the proposed triple tubal nuclear norm. Specifically, we come up with the following problem:

$$\begin{aligned} \min_{\mathcal{L}} \quad & \|\mathcal{L}\|_{\star} \\ \text{s.t.} \quad & \|\mathcal{O} \odot (\mathcal{T} - \mathcal{L})\|_{\mathbb{F}} \leq \epsilon, \end{aligned} \tag{20}$$

where parameter $\epsilon > 0$ denotes the noise level. The motivation is to recover \mathcal{L}^* by choosing the tensor with the smallest TriTNN from a hyper-ball in $\mathbb{R}^{n_1 \times n_2 \times n_3}$ with radius ϵ . It is well known that Problem (20) in the form of convex minimization with a bounded norm constraint is equivalent to the following unconstrained problem [23]:

$$\min_{\mathcal{L}} \frac{1}{2} \|\mathcal{O} \odot (\mathcal{T} - \mathcal{L})\|_{\mathbb{F}}^2 + \tau \|\mathcal{L}\|_{\star}. \tag{21}$$

where $\tau > 0$ is the regularization parameter.

4.2. An ADMM Solver to Problem (21)

The alternative direction multiplier method (ADMM) [38] has been extensively used in solving composite convex problems like Problem (21). We will solve Problem (21) by using ADMM in this subsection.

Considering the definition of TriTNN, we introduce auxiliary variables $\mathcal{U}, \mathcal{V}, \mathcal{W} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and obtain the following constrained problem:

$$\begin{aligned} \min_{\mathcal{L}, \mathcal{U}, \mathcal{V}, \mathcal{W}} \quad & \frac{1}{2} \|\mathcal{O} \odot (\mathcal{T} - \mathcal{L})\|_{\mathbb{F}}^2 + \tau \lambda_1 \|\mathcal{U}\|_{\star} + \tau \lambda_2 \|\mathcal{V}\|_{\star} + \tau \lambda_3 \|\mathcal{W}\|_{\star} \\ \text{s.t.} \quad & \mathcal{U} = \mathcal{L}, \quad \mathcal{V} = \vec{\mathcal{L}}, \quad \mathcal{W} = \underline{\mathcal{L}}. \end{aligned} \tag{22}$$

First, the augmented Lagrangian of Problem (22) is:

$$\begin{aligned} L_{\rho}(\mathcal{L}, \mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3) \\ = \frac{1}{2} \|\mathcal{O} \odot (\mathcal{T} - \mathcal{L})\|_{\mathbb{F}}^2 + \tau \lambda_1 \|\mathcal{U}\|_{\star} + \tau \lambda_2 \|\mathcal{V}\|_{\star} + \tau \lambda_3 \|\mathcal{W}\|_{\star} + \langle \mathcal{Y}_1, \mathcal{U} - \mathcal{L} \rangle + \frac{\rho}{2} \|\mathcal{U} - \mathcal{L}\|_{\mathbb{F}}^2 \\ + \langle \mathcal{Y}_2, \mathcal{V} - \vec{\mathcal{L}} \rangle + \frac{\rho}{2} \|\mathcal{V} - \vec{\mathcal{L}}\|_{\mathbb{F}}^2 + \langle \mathcal{Y}_3, \mathcal{W} - \underline{\mathcal{L}} \rangle + \frac{\rho}{2} \|\mathcal{W} - \underline{\mathcal{L}}\|_{\mathbb{F}}^2, \end{aligned} \tag{23}$$

where $\mathcal{Y}_1, \mathcal{Y}_2$ and \mathcal{Y}_3 are Lagrangian multipliers and $\rho > 0$ is the penalty parameter.

Using the framework of ADMM, we update the variables alternatively by fixing others at the $k + 1$ -th iteration in the following way.

Update \mathcal{L} . We update \mathcal{L} by fixing the other variables as follows:

$$\begin{aligned} \mathcal{L}^{k+1} &= \underset{\mathcal{L}}{\operatorname{argmin}} L_\rho(\mathcal{L}, \mathcal{U}^k, \mathcal{V}^k, \mathcal{W}^k, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k) \\ &= \underset{\mathcal{L}}{\operatorname{argmin}} \frac{1}{2} \|\mathcal{O} \odot (\mathcal{T} - \mathcal{L})\|_{\mathbb{F}}^2 + \left\langle \mathcal{Y}_1^k, \mathcal{U}^k - \mathcal{L} \right\rangle + \frac{\rho}{2} \|\mathcal{U}^k - \mathcal{L}\|_{\mathbb{F}}^2 \\ &\quad + \left\langle \mathcal{Y}_2^k, \mathcal{V}^k - \overrightarrow{\mathcal{L}} \right\rangle + \frac{\rho}{2} \|\mathcal{V}^k - \overrightarrow{\mathcal{L}}\|_{\mathbb{F}}^2 + \left\langle \mathcal{Y}_3^k, \mathcal{W}^k - \overleftarrow{\mathcal{L}} \right\rangle + \frac{\rho}{2} \|\mathcal{W}^k - \overleftarrow{\mathcal{L}}\|_{\mathbb{F}}^2 \\ &= (\rho \mathcal{U}^k + \rho \overleftarrow{\mathcal{V}}^k + \rho \overrightarrow{\mathcal{W}}^k + \mathcal{Y}_1^k + \overleftarrow{\mathcal{Y}}_2^k + \overrightarrow{\mathcal{Y}}_3^k + \mathcal{O} \odot \mathcal{T}) \odot (\mathcal{O} + \rho \mathbb{I}), \end{aligned} \tag{24}$$

where \odot denotes element-wise division and \mathbb{I} denotes the tensor of all ones.

Update \mathcal{U}, \mathcal{V} and \mathcal{W} . Tensor \mathcal{U} is updated as follows:

$$\begin{aligned} \mathcal{U}^{k+1} &= \underset{\mathcal{U}}{\operatorname{argmin}} L_\rho(\mathcal{L}^{k+1}, \mathcal{U}, \mathcal{V}^k, \mathcal{W}^k, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k) \\ &= \underset{\mathcal{U}}{\operatorname{argmin}} \tau \lambda_1 \|\mathcal{U}\|_* + \left\langle \mathcal{Y}_1^k, \mathcal{U} - \mathcal{L}^{k+1} \right\rangle + \frac{\rho}{2} \|\mathcal{U} - \mathcal{L}^{k+1}\|_{\mathbb{F}}^2 \\ &= \mathfrak{S}_{\tau \lambda_1 / \rho}(\mathcal{L}^{k+1} - \frac{\mathcal{Y}_1^k}{\rho}), \end{aligned} \tag{25}$$

where $\mathfrak{S}(\cdot)$ is the proximal operator of TNN at point $\mathcal{L}^{k+1} - \mathcal{Y}_1^k / \rho$ with parameter $\tau \lambda_1 / \rho$ (see Equation (8)).

Tensors \mathcal{V} and \mathcal{W} are updated similarly to \mathcal{U} ,

$$\begin{aligned} \mathcal{V}^{k+1} &= \underset{\mathcal{V}}{\operatorname{argmin}} L_\rho(\mathcal{L}^{k+1}, \mathcal{U}^{k+1}, \mathcal{V}, \mathcal{W}^k, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k) \\ &= \operatorname{ColSqueeze}(\mathfrak{S}_{\tau \lambda_2 / \rho}(\overrightarrow{\mathcal{L}^{k+1}} - \frac{\mathcal{Y}_2^k}{\rho})), \end{aligned} \tag{26}$$

and:

$$\begin{aligned} \mathcal{W}^{k+1} &= \underset{\mathcal{W}}{\operatorname{argmin}} L_\rho(\mathcal{L}^{k+1}, \mathcal{U}^{k+1}, \mathcal{V}^{k+1}, \mathcal{W}, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k) \\ &= \operatorname{RowSqueeze}(\mathfrak{S}_{\tau \lambda_3 / \rho}(\overrightarrow{\mathcal{L}^{k+1}} - \frac{\mathcal{Y}_3^k}{\rho})). \end{aligned} \tag{27}$$

Update $\mathcal{Y}_1, \mathcal{Y}_2$ and \mathcal{Y}_3 . Using dual ascending, we update $\mathcal{Y}_1, \mathcal{Y}_2$ and \mathcal{Y}_3 as follows:

$$\begin{aligned} \mathcal{Y}_1^{k+1} &= \mathcal{Y}_1^k + \rho(\mathcal{U}^{k+1} - \mathcal{L}^{k+1}), \\ \mathcal{Y}_2^{k+1} &= \mathcal{Y}_2^k + \rho(\mathcal{V}^{k+1} - \overrightarrow{\mathcal{L}^{k+1}}), \\ \mathcal{Y}_3^{k+1} &= \mathcal{Y}_3^k + \rho(\mathcal{W}^{k+1} - \overleftarrow{\mathcal{L}^{k+1}}). \end{aligned} \tag{28}$$

We summarize the algorithm in Algorithm 1 and analyze the computational complexity as follows.

Complexity analysis: The main computational cost in each iteration rests in the singular tube thresholding operator, requiring the computation of FFT, IFFT and SVDs. Therefore, the time complexity in each iteration is:

$$O(n_1 n_2 n_3 (n_1 + n_2 + n_3 + \log(n_1 n_2 n_3))), \tag{29}$$

Algorithm 1 Solving Problem (21) using ADMM.

Input: The observed tensor \mathcal{T} , the parameters $\tau, \lambda_1, \lambda_2, \lambda_3, \rho$.

- 1: **while** not converged **do**
- 2: Update \mathcal{L} using Equation (24);
- 3: Update \mathcal{U}, \mathcal{V} and \mathcal{W} using Equations (25), (26) and (27), respectively;
- 4: Update $\mathcal{Y}_1, \mathcal{Y}_2$ and \mathcal{Y}_3 using Equations (28);
- 5: Check the convergence conditions:

$$\begin{aligned} \|\mathcal{L}^{k+1} - \mathcal{L}^k\|_\infty \leq \delta, \quad \|\mathcal{U}^{k+1} - \mathcal{U}^k\|_\infty \leq \delta, \\ \|\mathcal{V}^{k+1} - \mathcal{V}^k\|_\infty \leq \delta, \quad \|\mathcal{W}^{k+1} - \mathcal{W}^k\|_\infty \leq \delta. \end{aligned}$$

- 6: **end while**

Output: $\hat{\mathcal{L}}$.

4.3. Convergence of Algorithm 1

As Problem (21) has more than two variables, the convergence property of Algorithm 1 cannot be directly obtained from existing results on the convergence of ADMM [38]. Thus, we prove its convergence in terms of the objective function in the following theorem.

Theorem 1 (Convergence of Algorithm 1). *For any $\rho > 0$, if the unaugmented Lagrangian $L_0(\mathcal{L}, \mathcal{U}, \mathcal{V}, \mathcal{W}, \mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3)$ has a saddle point, then the iterations $(\mathcal{L}^k, \mathcal{U}^k, \mathcal{V}^k, \mathcal{W}^k, \mathcal{Y}_1^k, \mathcal{Y}_2^k, \mathcal{Y}_3^k)$ in Algorithm 1 satisfy the residual convergence, objective convergence and dual variable convergence of Problem (21).*

Proof. The key idea of the proof is to rewrite Problem (9) into a two-block ADMM problem. Since the *RowTwist* and *ColTwist* operations are linear, there exist two matrices $P_1, P_2 \in \mathbb{R}^{n_1 n_2 n_3 \times n_1 n_2 n_3}$, such that the constraints $\mathcal{V} = \text{ColTwist}(\mathcal{L})$ and $\mathcal{W} = \text{RowTwist}(\mathcal{L})$ are equivalent to the vectorization expressions:

$$\text{vec}(\mathcal{V}) = P_1 \text{vec}(\mathcal{L}), \text{ and } \text{vec}(\mathcal{W}) = P_2 \text{vec}(\mathcal{L}),$$

where $\text{vec}(\cdot)$ denotes the operation of tensor vectorization (see [14]).

For notational simplicity, let:

$$x = \text{vec}(\mathcal{L}), \quad y = \begin{bmatrix} \text{vec}(\mathcal{Y}_1) \\ \text{vec}(\mathcal{Y}_2) \\ \text{vec}(\mathcal{Y}_3) \end{bmatrix}, \quad z = \begin{bmatrix} \text{vec}(\mathcal{U}) \\ \text{vec}(\mathcal{V}) \\ \text{vec}(\mathcal{W}) \end{bmatrix}, \quad A = \begin{bmatrix} I \\ P_1 \\ P_2 \end{bmatrix},$$

and:

$$f(x) = \frac{1}{2} \|\mathcal{O} \odot (\mathcal{L} - \mathcal{T})\|_{\mathbb{F}}^2, \quad g(z) = \tau(\lambda_1 \|\mathcal{U}\|_* + \lambda_2 \|\mathcal{V}\|_* + \lambda_3 \|\mathcal{W}\|_*).$$

It is obvious that $f(\cdot)$ and $g(\cdot)$ are closed, proper and convex. Then, Problem (21) can be re-written as follows:

$$\begin{aligned} \min_{x,z} \quad & f(x) + g(z) \\ \text{s.t.} \quad & Ax - z = \mathbf{0}. \end{aligned}$$

According to the convergence analysis in [38], we have:

$$\begin{aligned} \text{objective convergence:} & \quad \lim_{k \rightarrow \infty} f(\mathbf{x}^k) + g(\mathbf{z}^k) = f^* + g^*, \\ \text{dual variable convergence:} & \quad \lim_{k \rightarrow \infty} \mathbf{y}^k = \mathbf{y}^*, \\ \text{constraint convergence:} & \quad \lim_{k \rightarrow \infty} \mathbf{A}\mathbf{x}^k - \mathbf{z}^k = \mathbf{0}, \end{aligned}$$

where f^*, g^* are the optimal values of $f(\mathbf{x}), g(\mathbf{z})$, respectively. Variable \mathbf{y}^* is a dual optimal point defined as:

$$\mathbf{y}^* = \begin{bmatrix} \text{vec}(\mathbf{Y}_1^*) \\ \text{vec}(\mathbf{Y}_2^*) \\ \text{vec}(\mathbf{Y}_3^*) \end{bmatrix},$$

where $(\mathbf{Y}_1^*, \mathbf{Y}_2^*, \mathbf{Y}_3^*)$ is the dual component of a saddle point $(\mathcal{L}^*, \mathcal{U}^*, \mathcal{V}^*, \mathcal{W}^*, \mathbf{Y}_1^*, \mathbf{Y}_2^*, \mathbf{Y}_3^*)$ of the unaugmented Lagrangian $L_0(\mathcal{L}, \mathcal{U}, \mathcal{V}, \mathcal{W}, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3)$. \square

4.4. Differences from Prior Work

First, we show the difference between the proposed model TriTNN and two mostly related models TNN [30] and t-TNN [16]. Although all of them are based on the tubal nuclear norm, the main difference lies in that TNN and t-TNN only use information of one orientation, whereas TriTNN uses information of three orientations.

Now, we compare the proposed model with Tubal-Alt-Min [37]. It is based on tensor tubal rank and employs the tensor factorization strategy for tensor completion. The differences between TriTNN and Tubal-Alt-Min are: (a) TriTNN preserves the low-rank structure by summing three tubal nuclear norms, whereas Tubal-Alt-Min adopts low-rank tensor factorization to characterize the low-rank property of a tensor. In this way, they are two different kinds of models for tensor completion (since Tubal-Alt-Min and the proposed TriTNN are quite different algorithms and the main goal of this paper is to improve upon TNN, we do not compare Tubal-Alt-Min in the experiment section.) (b) Since TriTNN is based on the tubal nuclear norm, it is formulated as a convex optimization problem (21). Benefiting from convexity, each local minimum of Problem (21) must be a global minimum. However, Tubal-Alt-Min is formulated as a non-convex optimization problem, thus it may produce sub-optimal solutions.

5. Experiments

In this section, extensive experiments will be conducted to explore the effectiveness of the proposed Algorithm 1. All the codes are implemented in MATLAB language, and all experiments are carried out in Windows 10 based on an Intel Core(TM) 2.60-GHz CPU with 12 G RAM.

To explore the effectiveness of the proposed TriTNN-based model, we compare with the following nuclear norm-based tensor completion models:

- The tensor nuclear norm-based model with ADMM solver: high accuracy low-rank tensor completion (HaLRTC, denoted by SNN in this paper) (code available: <http://www.cs.rochester.edu/u/jliu/publications.html>) [2], The tensor nuclear norm is defined as the weighted sum of nuclear norms of the unfolding matrices along each mode (thus, we denote this model as SNN):

$$\|\mathcal{T}\|_{SNN} := \alpha_1 \|\mathbf{T}_{(1)}\|_* + \alpha_2 \|\mathbf{T}_{(2)}\|_* + \alpha_3 \|\mathbf{T}_{(3)}\|_*,$$

where $\alpha_1, \alpha_2, \alpha_3$ are positive parameters and $\mathbf{T}_{(i)} \in \mathbb{R}^{n_i \times \prod_{j \neq i} n_j}$, $i = 1, 2, 3$, is the unfolding matrix of tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ along the i -th mode [2].

- The latent tensor nuclear norm-based model (LatentNN) (code available: <https://github.com/ryotat/tensor>) [21]. The latent tensor nuclear norm is defined as:

$$\|\mathcal{T}\|_{latent} := \inf_{\mathcal{X}+\mathcal{Y}+\mathcal{Z}=\mathcal{T}} \|\mathcal{X}_{(1)}\|_* + \|\mathcal{Y}_{(2)}\|_* + \|\mathcal{Z}_{(3)}\|_*,$$

where $\mathcal{X}_{(1)}$, $\mathcal{Y}_{(2)}$ and $\mathcal{Z}_{(3)}$ are the first-mode, second-mode and third-mode unfoldings of latent tensors \mathcal{X} , \mathcal{Y} and \mathcal{Z} , respectively.

- The square nuclear norm-based model (SquareNN) (code available: <https://sites.google.com/site/mucun1988/publi>) [23]. The square nuclear norm of a tensor is defined as the nuclear norm of the most balanced unfolding of a tensor (see [7,23]).
- The most related tubal nuclear norm-based model (TNN) (code available: <https://github.com/jamiezeminzhang/>) [30] and the twist tubal nuclear norm-based model (t-TNN) [16].

We conduct tensor completion experiments on color images, videos and a dataset for autonomous vehicle. For an estimation tensor $\hat{\mathcal{L}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its quality is evaluated by the peak signal-to-noise ratio (PSNR) computed by the definition:

$$PSNR = 10 \log_{10} \left(\frac{n_1 n_2 n_3 \|\mathcal{L}^*\|_{\infty}^2}{\|\hat{\mathcal{L}} - \mathcal{L}^*\|_F^2} \right),$$

where \mathcal{L}^* is the underlying tensor. The higher the PSNR value is, the better the recovery performance will be.

5.1. Color Image Inpainting

Color images in row \times column \times channel are naturally expressed in 3D tensor form. Image inpainting aims at reconstructing a color image from a small fraction of its entries. In this experiment, twelve test images of size $256 \times 256 \times 3$ are used; see Figure 5. Given an image \mathcal{M} of size $d_1 \times d_2 \times 3$, we randomly sample 30% of its pixels and add i.i.d. Gaussian noise with standard deviation $\sigma = 0.1\sigma_0$, where $\sigma_0 = \|\mathcal{M}\|_F / \sqrt{3d_1d_2}$ is the rescaled magnitude of \mathcal{M} .

The weight parameters α of SNN are chosen to satisfy $\alpha_1 : \alpha_2 : \alpha_3 = 1 : 1 : 0.01$ as suggested in [2]. Parameter $\tau = 8e3$ and $\lambda_1, \lambda_2, \lambda_3$ in Problem (22) are chosen to satisfy $\lambda_1 : \lambda_2 : \lambda_3 = 1 : 0.01 : 0.01$. Parameters of other algorithms are tuned for better performances in most cases. We also employ the structural similarity index measure (SSIM) [43] to measure the quality of inpainted color images. The higher the SSIM value is, the better the inpainting performance will be. Given a color image, we test ten times and report the averaged PSNR and SSIM values.

The inpainting results of five images are shown in Figure 6 for qualitative comparison. We can see that the proposed TriTNN-based model obtains better visual performances. For quantitative comparison, the PSNR and SSIM values on the twelve images of seven algorithms are reported in Figure 7. It can be seen that the proposed TriTNN-based outperforms the competitors in most cases.

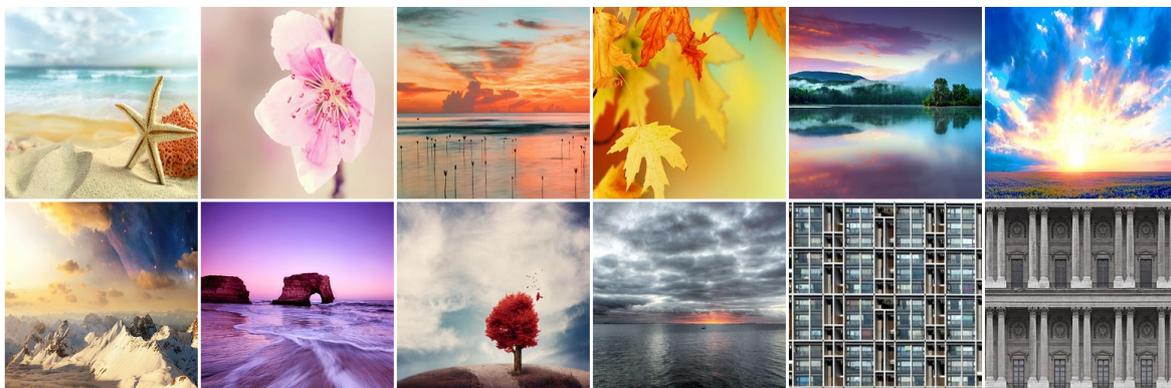
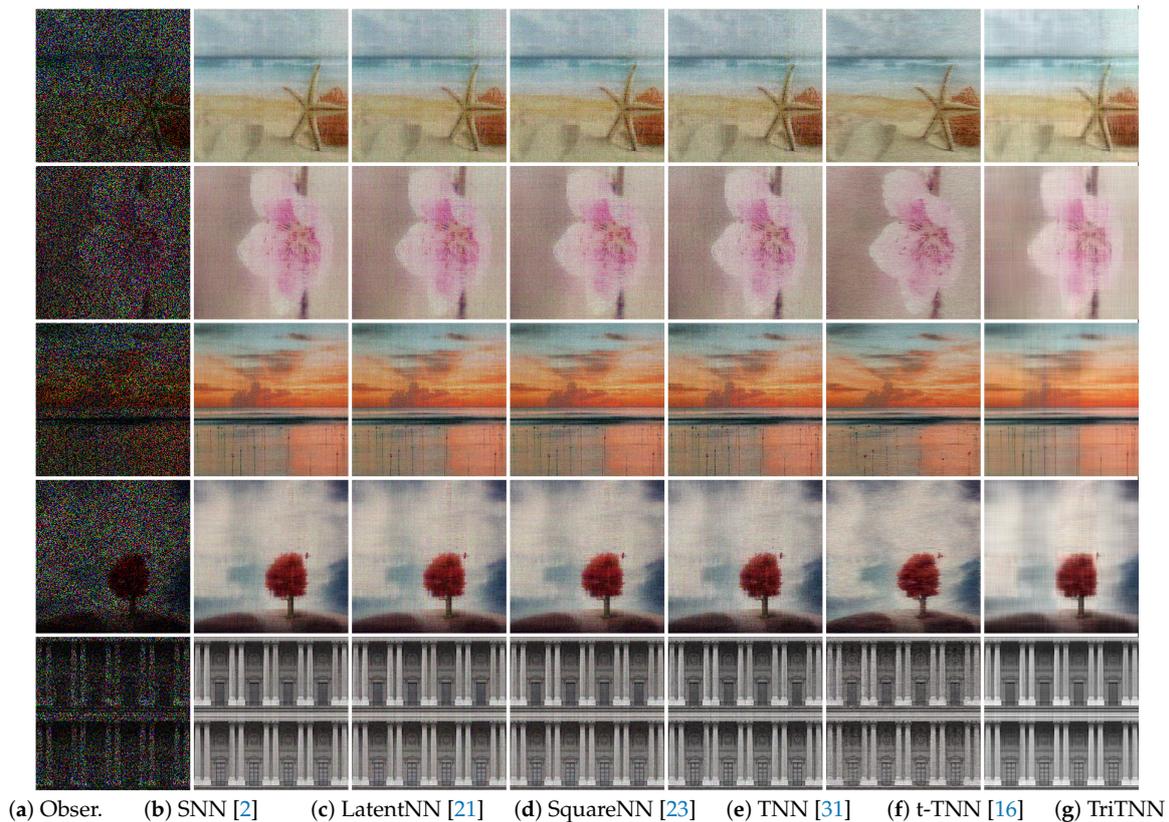


Figure 5. Twelve color images used in the experiments.



(a) Obser. (b) SNN [2] (c) LatentNN [21] (d) SquareNN [23] (e) TNN [31] (f) t-TNN [16] (g) TriTNN

Image	Algorithms						
	SNN	LatentNN	SquareNN	TNN	t-TNN	TriTNN	
Starfish	PSNR	22.36	21.41	21.62	21.63	22.88	23.40
	SSIM	0.5014	0.4573	0.4557	0.4461	0.5301	0.5865
Flower	PSNR	23.38	22.54	22.44	22.50	23.72	24.85
	SSIM	0.4670	0.4286	0.4109	0.4001	0.4692	0.6721
Sunset	PSNR	24.48	24.04	23.82	23.61	24.90	25.19
	SSIM	0.6381	0.6161	0.5982	0.5853	0.6650	0.7111
Tree	PSNR	24.13	23.27	23.19	23.25	24.39	25.35
	SSIM	0.5491	0.5120	0.4946	0.4850	0.5680	0.6932
Church	PSNR	25.81	25.21	25.01	25.66	24.70	26.87
	SSIM	0.8576	0.8466	0.8391	0.8511	0.8055	0.8359

(h) PSNR and SSIM values of the algorithms

Figure 6. Examples of color image inpainting. (a) is the observed noisy incomplete image (Obs.) with sampling ratio 0.3 and noise level $\sigma = 0.1\sigma_0$; (b–g) show the inpainting results of nuclear norm-based models: SNN [2], latent tensor nuclear norm-based model (LatentNN) [21], SquareNN [23], tensor tubal nuclear norm (TNN) [31], t-TNN [16] and the proposed model, respectively. The corresponding PSNR and SSIM values are listed in (k). The highest PSNR and SSIM indicating the best inpainting performance are highlighted in bold. It is suggested to be viewed in color as a pdf file with a 4× zoom in. TriTNN, triple tubal nuclear norm.

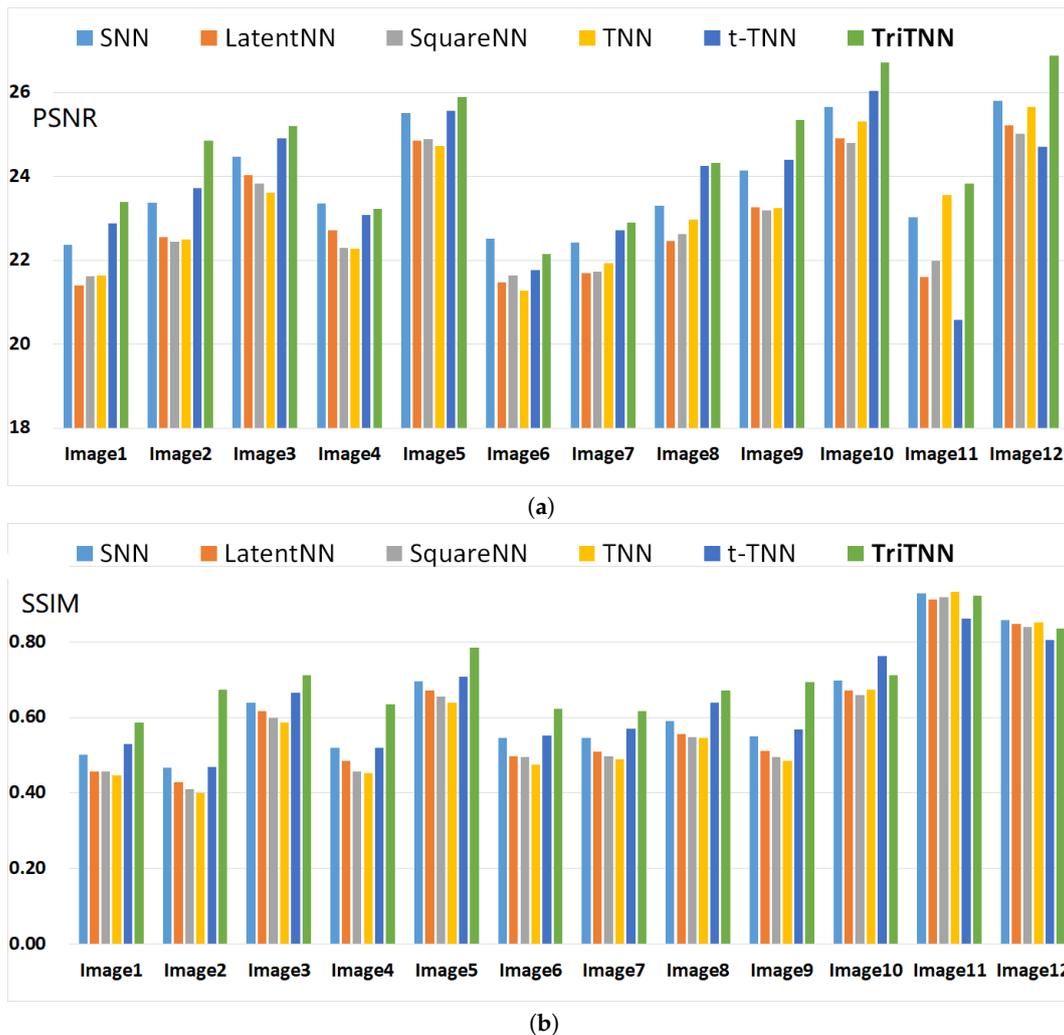


Figure 7. Quantitative evaluation of algorithms on color images for Uniform-0.3: the image is sampled uniformly with ratio $p = 0.3$ and corrupted with noise level $\sigma = 0.1\sigma_0$. (a) PSNR values; (b) SSIM values.

5.2. Video Inpainting

The video inpainting task aims at imputing the missing pixels of a video. The performance competition is carried out on five widely-used YUV videos (They are available from <https://sites.google.com/site/subudhibadri/fewhelpfuldownloads>): salesman_qcif, silent_qcif, suzie_qcif, tempete_cif and waterfall_cif. Due to the computational limitation, we use the first 30 frames of Y components in each video. This results in three tensors sized $144 \times 176 \times 32$ and two tensors sized $288 \times 352 \times 32$. For each video, we uniformly sample 10% of the entries and conduct the video inpainting experiments.

The weight parameters α of SNN are chosen to satisfy $\alpha_1 : \alpha_2 : \alpha_3 = 1 : 1 : 1$ as suggested in [2]. Parameters $\tau = 2, \rho = 5e-5$ and $\lambda_1, \lambda_2, \lambda_3$ are chosen to satisfy $\lambda_1 : \lambda_2 : \lambda_3 = 1 : 1 : 1$ for the proposed model. Parameters of other algorithms are tuned for better performances in most cases. Given a video, we test ten times and report the averaged PSNR value.

The qualitative comparison is shown in Figure 8. The PSNR values are reported in Table 1. It can be seen that the TriTNN-based model outperforms the others. The proposed TriTNN has better performances than TNN and t-TNN, since the TriTNN exploits the row, column and tube redundancy simultaneously, whereas TNN or t-TNN only exploit one type of redundancy. The superiority of TriTNN over SNN, LatentNN and SquareNN may be explained by the fact that the circulant block

matricization of a tensor coded in TriTNN makes use of more information than directly unfolding along each mode.

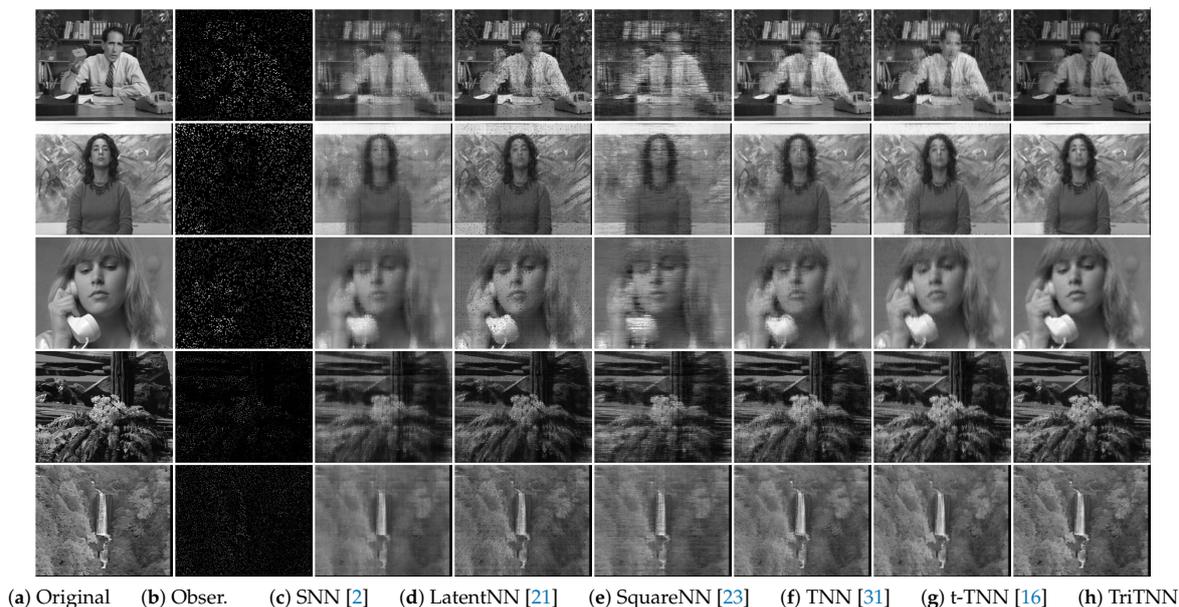


Figure 8. Examples of YUV video inpainting. (a) is the first frame of each video; (b) is the observed incomplete frame (Obs.) with 90% missing entries; (c–h) show the inpainting results of nuclear norm-based models: SNN [2], LatentNN [21], SquareNN [23], TNN [31], t-TNN [16] and the proposed model TriTNN-based model, respectively. It is suggested to be viewed as a pdf file with a 4× zoom in.

Table 1. Quantitative evaluation of algorithms in PSNR values for YUV video inpainting; each video is sampled uniformly with ratio $p = 0.1$.

Video	SNN [2]	LatentNN [21]	SquareNN [23]	TNN [31]	t-TNN [16]	TriTNN
salesman_qcif	21.10	24.21	20.31	25.56	25.73	26.18
silent_qcif	24.07	23.95	22.98	27.93	28.03	28.47
suzie_qcif	25.79	24.63	24.48	27.92	29.47	29.70
tempete_cif	19.10	18.45	19.23	20.65	21.21	21.46
waterfall_cif	24.10	22.39	24.06	26.71	27.44	28.22

5.3. A Dataset for Autonomous Driving

Environment perception for autonomous driving has attracted more and more attention in computer vision. In this subsection, experiments on a dataset collected for autonomous driving are performed.

The dataset (a collection of Frame No. 165–No. 244 in Scenario B and Scenario B; additional sensor data available at <http://www.mrt.kit.edu/z/publ/download/velodynetracking/dataset.html>) has 80 frames of gray images and LiDAR point cloud data acquired by a Velodyne HDL-64E LiDAR. The image sequence is resized to be a tensor of size $128 \times 256 \times 80$, and the LiDAR data are resampled, transformed and formatted to be two tensors of size $64 \times 436 \times 80$ representing the distance data and the intensity data, respectively.

Given a tensor to complete $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, experiments are carried out with seven different observation settings where the sampling ratio p varies from 0.1–0.7. The observed entries are further corrupted by i.i.d. Gaussian noise with standard deviation $\sigma = 0.2\sigma_0$, where $\sigma_0 = \|\mathcal{T}\|_F / \sqrt{d_1 d_2 d_3}$ is the normalized magnitude of \mathcal{T} . The weight parameters α of SNN are chosen to satisfy $\alpha_1 : \alpha_2 : \alpha_3 = 1 : 1 : 1$. Parameters $\tau = 2, \rho = 5e-5$ and $\lambda_1, \lambda_2, \lambda_3$ are chosen to satisfy $\lambda_1 : \lambda_2 : \lambda_3 = 1 : 1 : 1$ for the

proposed model. Parameters of other algorithm are tuned manually to achieve better performances in most cases. Given a sampling ratio p , we repeat ten trials and report the averaged PSNR value.

The quantitative performance comparison in terms of PSNR is shown in Tables 2–4 for gray image sequence, distance data and intensity data completion, respectively. The highest average PSNR indicating the best recovery performance is highlighted in bold. From Tables 2–4, we can see that the proposed TriTNN-based model yields better performances over the other five algorithms for noisy tensor completion.

Table 2. Comparison of the PSNR values for image sequence completion.

Sampling Ratio	SNN [2]	LatentNN [21]	SquareNN [23]	TNN [31]	t-TNN [16]	TriTNN
$p = 0.1$	14.58	15.50	17.27	17.23	17.29	17.45
$p = 0.2$	17.79	17.27	18.16	18.32	18.59	18.74
$p = 0.3$	18.74	18.36	18.80	19.04	19.12	19.53
$p = 0.4$	19.29	19.02	19.21	19.50	19.44	20.02
$p = 0.5$	19.61	19.43	19.49	19.82	19.64	20.28
$p = 0.6$	19.80	19.70	19.69	20.06	19.76	22.01
$p = 0.7$	19.90	19.83	19.84	20.23	19.85	22.41

Table 3. Comparison of the PSNR values for HDL-64E distance data completion.

Sampling Ratio	SNN [2]	LatentNN [21]	SquareNN [23]	TNN [31]	t-TNN [16]	TriTNN
$p = 0.1$	17.80	16.90	17.64	19.03	18.67	18.91
$p = 0.2$	19.26	18.48	18.94	20.04	19.60	20.09
$p = 0.3$	20.33	19.71	19.94	20.93	20.41	21.04
$p = 0.4$	21.24	20.67	20.89	21.74	21.17	21.92
$p = 0.5$	22.05	21.55	21.75	22.47	21.88	22.72
$p = 0.6$	22.82	22.39	22.56	23.16	22.61	23.68
$p = 0.7$	23.54	23.21	23.34	23.79	23.35	24.51

Table 4. Comparison of the PSNR values for HDL-64E intensity data completion.

Sampling Ratio	SNN [2]	LatentNN [21]	SquareNN [23]	TNN [31]	t-TNN [16]	TriTNN
$p = 0.1$	17.30	17.63	17.57	18.31	18.62	18.36
$p = 0.2$	18.79	19.30	18.92	19.50	19.61	19.63
$p = 0.3$	19.85	20.29	19.93	20.38	20.37	20.45
$p = 0.4$	20.84	21.11	20.90	21.24	21.08	21.30
$p = 0.5$	21.72	21.86	21.76	22.05	21.78	22.18
$p = 0.6$	22.51	22.62	22.53	22.79	22.50	22.94
$p = 0.7$	23.39	23.40	23.39	23.59	23.27	23.80

6. Conclusions

This paper studied the problem of completing a 3D tensor from incomplete observations. It has been used in broad applications of computer vision tasks like image and video inpainting. In this paper, a new tensor norm named the triple tubal nuclear norm (TriTNN) is proposed to simultaneously exploit tube, row and column redundancy in a circulant way by using a weighted sum of three TNNs. A TriTNN-based tensor completion model is presented. It is further solved by an ADMM-based algorithm with convergence guarantee. Experimental results on color images, videos and LiDAR datasets show the superiority of the proposed TriTNN against state-of-the-art nuclear norm-based tensor norms.

For the proposed TriTNN, the authors believe it can outperform many nuclear norm-based tensor completion models because more spatial-temporal information is exploited. However, generally speaking, it has the following two drawbacks:

1. Computational inefficiency: Compared to TNN and t-TNN, it is more time-consuming since it involves computing TNN, t-TNN and rt-TNN (see Equation (18)).
2. Sample inefficiency: Using the analysis of [23] and [44], to complete an incomplete tensor, TriTNN needs more observations than TNN and t-TNN (limited to the scope of this paper; the authors do not discuss it further).

In the future research, the authors are interested in efficient algorithms like [45] to tackle the problem of computational inefficiency. To decrease the sample complexity of TriTNN, it will be helpful to follow the suggestions in [44] to design new atomic norms like [46]. To get better visual completion performances, the authors would like to consider adding smoothness regularization in the model like [47–49] and adopting different tensorization methods like [50]. It is also helpful for studying new tensor completion models using deep neural networks [51]. For potential extensions of TriTNN, the authors would like to explore the p -th order ($p > 3$) extension [52] and extensions to other discrete transforms other than DFT like [53].

Author Contributions: Conceptualization, D.W. and A.W. Data curation, D.W. and X.F. Formal analysis, B.W. (Bo Wang). Methodology, D.W. and A.W. Resources, B.W. (Boyu Wang) and B.W. (Bo Wang). Software, X.F. and B.W. (Boyu Wang). Writing, original draft, D.W., A.W. and X.F. Writing, review and editing, B.W. (Boyu Wang) and B.W. (Bo Wang).

Acknowledgments: The authors are grateful to the anonymous referees for their insightful and constructive comments that led to an improved version of the paper. The authors would like to thank Fangfang Yang and Zhong Jin for their long-time support. The authors wish to thank the audiences of White Paper Tao (WPT) for their focus. This work is partially supported by the National Natural Science Foundation of China under Grant Nos. 61702262, 61703209, 91420201 and 61472187.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kilmer, M.E.; Braman, K.; Hao, N.; Hoover, R.C. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM J. Matrix Anal. Appl.* **2013**, *34*, 148–172. [[CrossRef](#)]
2. Liu, J.; Musialski, P.; Wonka, P.; Ye, J. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 208–220. [[CrossRef](#)] [[PubMed](#)]
3. Wang, A.; Jin, Z. Near-optimal Noisy Low-tubal-rank Tensor Completion via Singular Tube Thresholding. In Proceedings of the IEEE International Conference on Data Mining Workshop (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 553–560.
4. Signoretto, M.; Dinh, Q.T.; Lathauwer, L.D.; Suykens, J.A.K. Learning with tensors: a framework based on convex optimization and spectral regularization. *Mach. Learn.* **2014**, *94*, 303–351. [[CrossRef](#)]
5. Cichocki, A.; Mandic, D.; De Lathauwer, L.; Zhou, G.; Zhao, Q.; Caiafa, C.; Phan, H.A. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Process. Mag.* **2015**, *32*, 145–163. [[CrossRef](#)]
6. Lu, C.; Feng, J.; Chen, Y.; Liu, W.; Lin, Z.; Yan, S. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 5249–5257.
7. Wei, D.; Wang, A.; Wang, B.; Feng, X. Tensor Completion Using Spectral (k, p) -Support Norm. *IEEE Access* **2018**, *6*, 11559–11572. [[CrossRef](#)]
8. Liu, Y.; Shang, F. An Efficient Matrix Factorization Method for Tensor Completion. *IEEE Signal Process. Lett.* **2013**, *20*, 307–310. [[CrossRef](#)]
9. Song, X.; Lu, H. Multilinear Regression for Embedded Feature Selection with Application to fMRI Analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 2562–2568.

10. Tan, H.; Feng, G.; Feng, J.; Wang, W.; Zhang, Y.J.; Li, F. A tensor-based method for missing traffic data completion. *Transp. Res. Part C* **2013**, *28*, 15–27. [[CrossRef](#)]
11. Zhao, Q.; Zhang, L.; Cichocki, A. Bayesian CP Factorization of Incomplete Tensors with Automatic Rank Determination. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1751–1763. [[CrossRef](#)] [[PubMed](#)]
12. Cichocki, A.; Lee, N.; Oseledets, I.; Phan, A.H.; Zhao, Q.; Mandic, D.P. Tensor Networks for Dimensionality Reduction and Large-scale Optimization: Part 1 Low-Rank Tensor Decompositions. *Found. Trends Mach. Learn.* **2016**, *9*, 249–429. [[CrossRef](#)]
13. Harshman, R.A. *Foundations of the PARAFAC Procedure: Models and Conditions for an “Explanatory” Multi-Modal Factor Analysis*; University of California: Los Angeles, CA, USA, 1970.
14. Kolda, T.G.; Bader, B.W. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455–500. [[CrossRef](#)]
15. Tucker, L.R. Some mathematical notes on three-mode factor analysis. *Psychometrika* **1966**, *31*, 279–311. [[CrossRef](#)] [[PubMed](#)]
16. Hu, W.; Tao, D.; Zhang, W. The twist tensor nuclear norm for video completion. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2961–2973. [[CrossRef](#)] [[PubMed](#)]
17. Yuan, M.; Zhang, C.H. Incoherent Tensor Norms and Their Applications in Higher Order Tensor Completion. *IEEE Trans. Inf. Theory* **2017**, *63*, 6753–6766. [[CrossRef](#)]
18. Song, Q.; Ge, H.; Caverlee, J.; Hu, X. Tensor Completion Algorithms in Big Data Analytics. *arXiv* **2017**, arXiv:1711.10105.
19. Candès, E.J.; Tao, T. The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Inf. Theory* **2010**, *56*, 2053–2080. [[CrossRef](#)]
20. Hillar, C.J.; Lim, L. Most Tensor Problems Are NP-Hard. *J. ACM* **2009**, *60*, 45. [[CrossRef](#)]
21. Tomioka, R.; Suzuki, T. Convex tensor decomposition via structured Schatten norm regularization. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 1331–1339.
22. Semerci, O.; Hao, N.; Kilmer, M.E.; Miller, E.L. Tensor-Based Formulation and Nuclear Norm Regularization for Multienergy Computed Tomography. *IEEE Trans. Image Process.* **2014**, *23*, 1678–1693. [[CrossRef](#)] [[PubMed](#)]
23. Mu, C.; Huang, B.; Wright, J.; Goldfarb, D. Square Deal: Lower Bounds and Improved Relaxations for Tensor Recovery. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 73–81.
24. Zhao, Q.; Meng, D.; Kong, X.; Xie, Q.; Cao, W.; Wang, Y.; Xu, Z. A Novel Sparsity Measure for Tensor Recovery. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 271–279.
25. Tomioka, R.; Hayashi, K.; Kashima, H. Estimation of low-rank tensors via convex optimization. *arXiv* **2010**, arXiv:1010.0789.
26. Chretien, S.; Wei, T. Sensing tensors with Gaussian filters. *IEEE Trans. Inf. Theory* **2017**, *63*, 843–852. [[CrossRef](#)]
27. Ghadermarzy, N.; Plan, Y.; Yilmaz, Ö. Near-optimal sample complexity for convex tensor completion. *arXiv* **2017**, arXiv:1711.04965.
28. Ghadermarzy, N.; Plan, Y.; Yilmaz, Ö. Learning tensors from partial binary measurements. *arXiv* **2018**, arXiv:1804.00108.
29. Liu, Y.; Shang, F.; Fan, W.; Cheng, J.; Cheng, H. *Generalized Higher-Order Orthogonal Iteration for Tensor Decomposition and Completion*. In Proceedings of the Advances in Neural Information Processing Systems, Palais des Congrès de Montréal, Montréal, CANADA, 8–13 December 2014 ; pp. 1763–1771.
30. Zhang, Z.; Ely, G.; Aeron, S.; Hao, N.; Kilmer, M. Novel methods for multilinear data completion and de-noising based on tensor-SVD. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3842–3849.
31. Zhang, Z.; Aeron, S. Exact Tensor Completion Using t-SVD. *IEEE Trans. Signal Process.* **2017**, *65*, 1511–1526. [[CrossRef](#)]
32. Sun, W.; Chen, Y.; Huang, L.; So, H.C. Tensor Completion via Generalized Tensor Tubal Rank Minimization using General Unfolding. *IEEE Signal Process. Lett.* **2018**. [[CrossRef](#)]
33. Zhou, P.; Lu, C.; Lin, Z.; Zhang, C. Tensor Factorization for Low-Rank Tensor Completion. *IEEE Trans. Image Process.* **2018**, *27*, 1152–1163. [[CrossRef](#)] [[PubMed](#)]

34. Ely, G.T.; Aeron, S.; Hao, N.; Kilmer, M.E. 5D seismic data completion and denoising using a novel class of tensor decompositions. *Geophysics* **2015**, *80*, V83–V95. [[CrossRef](#)]
35. Liu, X.; Aeron, S.; Aggarwal, V.; Wang, X.; Wu, M. Adaptive Sampling of RF Fingerprints for Fine-grained Indoor Localization. *IEEE Trans. Mob. Comput.* **2016**, *15*, 2411–2423. [[CrossRef](#)]
36. Jiang, J.Q.; Ng, M.K. Exact Tensor Completion from Sparsely Corrupted Observations via Convex Optimization. *arXiv* **2017**, arXiv:1708.00601.
37. Liu, X.Y.; Aeron, S.; Aggarwal, V.; Wang, X. Low-tubal-rank tensor completion using alternating minimization. *arXiv* **2016**, arXiv:1610.01690.
38. Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **2011**, *3*, 1–122. [[CrossRef](#)]
39. Zhou, P.; Feng, J. Outlier-Robust Tensor PCA. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
40. Cai, J.F.; Candès, E.J.; Shen, Z. A singular value thresholding algorithm for matrix completion. *SIAM J. Opt.* **2010**, *20*, 1956–1982. [[CrossRef](#)]
41. Gleich, D.F.; Greif, C.; Varah, J.M. The power and Arnoldi methods in an algebra of circulants. *Numer. Linear Algebra Appl.* **2013**, *20*, 809–831. [[CrossRef](#)]
42. Granata, J.; Conner, M.; Tolimieri, R. The tensor product: a mathematical programming language for FFTs and other fast DSP operations. *IEEE Signal Process. Mag.* **1992**, *9*, 40–48. [[CrossRef](#)]
43. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
44. Oymak, S.; Jalali, A.; Fazel, M.; Eldar, Y.C.; Hassibi, B. Simultaneously structured models with application to sparse and low-rank matrices. *IEEE Trans. Inf. Theory* **2015**, *61*, 2886–2908. [[CrossRef](#)]
45. Mu, C.; Zhang, Y.; Wright, J.; Goldfarb, D. Scalable Robust Matrix Recovery: Frank-Wolfe Meets Proximal Methods. *SIAM J. Sci. Comput.* **2016**, *38*, A3291–A3317. [[CrossRef](#)]
46. Richard, E.; Obozinski, G.R.; Vert, J.P. Tight convex relaxations for sparse matrix factorization. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 3284–3292.
47. Yokota, T.; Zhao, Q.; Cichocki, A. Smooth PARAFAC Decomposition for Tensor Completion. *IEEE Trans. Signal Process.* **2016**, *64*, 5423–5436. [[CrossRef](#)]
48. Chen, Y.L.; Hsu, C.T.; Liao, H.Y.M. Simultaneous Tensor Decomposition and Completion Using Factor Priors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 577–591. [[CrossRef](#)] [[PubMed](#)]
49. Xutao Li, Y.Y.; Xu, X. Low-Rank Tensor Completion with Total Variation for Visual Data Inpainting. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 2210–2216.
50. Yokota, T.; Erem, B.; Guler, S.; Warfield, S.K.; Hontani, H. Missing Slice Recovery for Tensors Using a Low-rank Model in Embedded Space. *arXiv* **2018**, arXiv:1804.01736.
51. Fan, J.; Cheng, J. Matrix completion by deep matrix factorization. *Neural Netw.* **2018**, *98*, 34–41. [[CrossRef](#)] [[PubMed](#)]
52. Martin, C.D.; Shafer, R.; Larue, B. An Order- p Tensor Factorization with Applications in Imaging. *SIAM J. Sci. Comput.* **2013**, *35*, A474–A490. [[CrossRef](#)]
53. Liu, X.Y.; Wang, X. Fourth-order tensors with multidimensional discrete transforms. *arXiv* **2017**, arXiv:1705.01576.

