

Generalized Matrix Completion Over A Higher-Order Finite Dimensional Algebra with Its Applications in Image Completion

Abstract

To increase the precision of tensor completion for image restoration, we introduce a recovery technique rooted in higher-order generalized scalars. We first extend the traditional second-order matrix model to a broader, higher-order matrix equivalent, termed the “t-matrix” model, applying a pixel neighborhood expansion strategy for image processing. This “t-matrix” model is then utilized to augment frequently employed matrix and tensor completion algorithms to their higher-order forms. We carry out comprehensive experiments on various algorithms using publicly accessible image sets, contrasting their performances. The results reveal that our generalized matrix completion models outperform their lower-order and conventional counterparts.

Keywords: Higher-Order Tensor Completion, Pixel Neighborhood Strategy, Generalized Matrix Model, Image Restoration

1. Introduction

Low Rank Matrix Completion (LRMC) constitutes a vital issue in the realms of data analysis and processing. Nonetheless, we often encounter high-order data with intricate structures, such as color images, video sequences, and hyperspectral images, in real-world applications. These multilinear data structures defy adequate representation via conventional vectors and matrices.

In order to characterize these complex high-dimensional data structures more accurately, we turn to tensors. As higher-order extensions of vectors and matrices, tensors are extensively utilized. Typically, high-dimensional data have a low intrinsic dimensionality, suggesting that they are low-rank or approximately low-rank [3]. Some elements may be lost during the collection of high-dimensional data. Low Rank Tensor Completion (LRTC) aims to recover these missing elements by leveraging information from known data elements. In contrast to LRMC, which only taps into the bidimensional information of the data, LRTC harnesses its higher-order information. Consequently, exploring low-rank tensor completion techniques holds substantial importance in data visualization.

Tensor completion techniques have proven successful in a variety of fields, including computer vision [5, 6, 7, 8], multi-class learning [9], and data mining [10, 11]. For instance, Liu et

al. [4] introduced the Sum of Matricization-based Nuclear Norms (SMNN) rooted in the tensor’s Tucker-rank [12] and proposed three optimization algorithms to address the tensor completion issue via SMNN minimization. Zhang et al. [5] presented the Tubal Nuclear Norm (Tubal-NN) based on tensor tubal-rank [13], putting forward a tensor nuclear norm penalized algorithm to tackle the tensor completion problem through Tubal-NN minimization. Lu et al. [14] offered a tensor completion model grounded in the Tensor Nuclear Norm (TNN) they proposed. Xue et al. [15] developed a tensor completion model rooted in their Truncated Tensor Nuclear Norm (T-TNN) definition.

To further extract neighborhood information and spatial structure from images, we employ a generalized matrix model rooted in higher-order circular convolution [16]. This model expands the well-received completion algorithm proposed by Lu et al. to higher orders and employs it for multi-way image recovery. Our evaluations indicate that this generalized high-order algorithm exhibits favorable recovery performance in comparison with existing algorithms.

2. Generalized Matrices

2.1. Basic Definition of Generalized Matrices

A generalized matrix (t-matrix) is a rectangular array composed of elements referred to as generalized scalars (t-scalars) [16]. As a generalized scalar forms an array in $\mathbb{C}^{I_1 \times \dots \times I_N}$, a generalized matrix with D_1 rows and D_2 columns can be represented by a multi-way complex array in $\mathbb{C}^{I_1 \times \dots \times I_N \times D_1 \times D_2}$. While various authors, including Kilmer et al. [13], categorize these generalized matrices as tensors, we employ the term “generalized matrix over higher-order scalars.” Utilizing this generalized matrix model provides an opportunity to extend numerous existing matrix algorithms.

2.2. Generalized Scalars

Let’s consider an N -order complex array to be an element of the set C , where $C \equiv \mathbb{C}^{I_1 \times \dots \times I_N}$. In parallel, an N -order real array is identified as an element of the set R , where $R \equiv \mathbb{R}^{I_1 \times \dots \times I_N}$. The sets R and C share the commutative ring structure, where the multiplication of their elements is defined by the N -order circular convolution, and addition corresponds to entry-wise array addition. Elements within C and R are termed generalized scalars. In this work, we primarily focus on C , as R is a subset of C . By defining the multiplication of a generalized scalar with a complex number through conventional complex multiplication, we can elevate the algebraic rings C to finite-dimensional commutative algebras.

Utilizing generalized scalars not only empowers us to construct novel matrices but also extends various classic matrix algorithms to the realm of generalized matrix algorithms. In 2011, Kilmer et al. pioneered the “t-product” model [13]. In this model, scalar elements in traditional matrices are replaced with fixed-size one-dimensional arrays, facilitating the extension of numerous classic matrix algorithms. Leveraging this expansion, new generalized matrices employ

elements from the commutative algebras R or C —the generalized scalars—thus are considered matrices built on the foundation of finite-dimensional commutative algebras.

In this paper, we adopt generalized matrices inspired by the work of Kilmer et al. [13], following the model by Liao and Maybank [16]. This research, premised on the fundamental notions introduced in the generalized matrix model [16], broadens Kilmer et al.’s order-one generalized scalars to higher orders through a neighborhood strategy, thereby extending original matrices into high-order generalized matrices. Following this, the multi-way circular convolution of generalized scalars in the spatial domain is translated to Hadamard multiplication in the Fourier domain via the Fourier transform, facilitating relevant computations. For instance, the subsequent definition is provided for order-two generalized scalars in the literature [16].

Definition 2.1 (Addition of Generalized Scalars) *Consider $\dot{x}, \dot{y} \in \mathbb{C}^{I_1 \times I_2}$ as general matrices of size $I_1 \times I_2$. The sum, $\dot{c} = \dot{x} + \dot{y}$, with $\dot{c} \in \mathbb{C}^{I_1 \times I_2}$, is computed element-wise, meaning the complex element of \dot{c} at position (i_1, i_2) is*

$$(\dot{c})_{i_1, i_2} = (\dot{x})_{i_1, i_2} + (\dot{y})_{i_1, i_2}, \quad \forall i_1, i_2.$$

Definition 2.2 (Multiplication of Generalized Scalars) *Let $\dot{x}, \dot{y} \in \mathbb{C}^{I_1 \times I_2}$ be generalized scalars of size $I_1 \times I_2$. Their product is defined as $\dot{c} = \dot{x} \circ \dot{y}$, where \dot{c} results from the order-two circular convolution of \dot{x} and \dot{y} . Specifically, we have*

$$(\dot{c})_{i_1, i_2} = \sum_{k_1=1}^{I_1} \sum_{k_2=1}^{I_2} (\dot{x})_{k_1, k_2} \cdot (\dot{y})_{k'_1, k'_2}, \quad \forall i_1, i_2$$

where $k'_1 = \text{mod}(i_1 - k_1, I_1) + 1$ and $k'_2 = \text{mod}(i_2 - k_2, I_2) + 1$.

While order-two generalized t-scalars share the data structure of an order-two numerical array, they are not matrices as their multiplication, per Definition 2.2, is always commutative. However, to describe linear transformations, it can be convenient to consider the underlying order-two arrays as matrices. We employ the notation $\text{Mat}(\dot{x})$ to elevate the underlying order-two array \dot{x} to a conventional matrix of identical size and entries.

Theorem 2.1 (Fourier Transform) *Let $\dot{x}, \dot{y} \in C$ be given generalized scalars with the product $\dot{c} = \dot{x} \circ \dot{y} \in C$. Define $\tilde{x} = F(\dot{x})$, $\tilde{y} = F(\dot{y})$, and $\tilde{c} = F(\dot{c})$ as their respective multi-way Fourier transforms. For any $\dot{x} \in C$, the multi-way Fourier transform is characterized by the following matrix multiplication, equivalently expressed as multi-mode multiplication:*

$$F(\dot{x}) = W_1 \cdot \text{Mat}(\dot{x}) \cdot W_2^T \equiv \text{Mat}(\dot{x}) \times_1 W_1 \times_2 W_2 \quad (1)$$

where $W_1 \in \mathbb{C}^{I_1 \times I_1}$ and $W_2 \in \mathbb{C}^{I_2 \times I_2}$ represent the Fourier matrices of appropriate sizes, with W_2^T being the transpose of W_2 and not its conjugate transpose. Consequently, for all i_1, i_2 , the following Hadamard product holds:

$$(\tilde{c})_{i_1, i_2} = (\tilde{x})_{i_1, i_2} \cdot (\tilde{y})_{i_1, i_2}.$$

An essential operation in any algebra is the multiplication of an algebraic element with a conventional scalar. This leads to the following definition:

Definition 2.3 (Scalar Multiplication) *Let $\dot{x} \in C \equiv \mathbb{C}^{I_1 \times I_2}$ be a generalized scalar and λ a complex number. Their multiplication, denoted as $\dot{y} \doteq \lambda \cdot \dot{x} \in C$, is defined for all i_1, i_2 as follows:*

$$(\dot{y})_{i_1, i_2} = \lambda \cdot (\dot{x})_{i_1, i_2}$$

With prior definitions as a foundation, we can readily identify the identity and zero elements within algebra C . These two distinct elements are defined as follows:

Definition 2.4 (Generalized Identity Scalar and Generalized Zero Scalar) *Consider a generalized scalar $\dot{e} \in \mathbb{C}^{I_1 \times I_2}$. In this case, $(\dot{e})_{i_1, i_2} = 1$ when $i_1 = i_2 = 1$, and $(\dot{e})_{i_1, i_2} = 0$ otherwise. The first element is 1, while all other elements are 0, characterizing the generalized identity scalar \dot{e} . Alternatively, when every element of \dot{x} is 0, we define this as the generalized zero scalar \dot{z} . Notably, every element of the generalized identity scalar is 1 in the Fourier domain, whereas the generalized zero scalar remains unchanged.*

2.3. Generalized Scalars as Finite-dimensional Linear Operators

Recognizing that each generalized scalar in algebra C operates as a finite-dimensional linear commutative operator, operator theory allows us to determine the spectrum of any element $\dot{x} \in C$. The spectrum, or the assortment of complex eigenvalues of \dot{x} , equates to the K entries of \tilde{x} (where $K \doteq I_1 \cdot I_2$), accounting for multiplicity.

Let us now delve into the forthcoming definitions.

Definition 2.5 (Conjugate) *A unique element \dot{y} in C is the conjugate of an element \dot{x} in C if every eigenvalue of \dot{y} is the complex conjugate of the corresponding eigenvalue of \dot{x} . The conjugate is denoted as $\dot{y} \doteq \dot{x}^*$.*

Definition 2.6 (Nonnegativity) *A generalized scalar \dot{x} is denoted as nonnegative if and only if all its complex eigenvalues are nonnegative real numbers.*

Definition 2.6 is pivotal as it facilitates the generalization of various nonnegativity concepts, including matrix rank, space dimension, norm, and distance, into nonnegative elements in C . We will explore these generalizations as required.

The nonnegativity stipulated in Definition 2.6 also establishes other important concepts. For instance, consider any two nonnegative generalized scalars \dot{x}, \dot{y} . Their subtraction is deemed self-conjugate, as the equation $(\dot{x} - \dot{y})^* = \dot{x} - \dot{y}$ invariably stands. Furthermore, the element $\dot{z} \in C$ represents the minimum nonnegative element. If $\dot{x} - \dot{y}$ are nonnegative, a partial order $\dot{x} \geq \dot{y} \geq \dot{z}$ is defined. If neither $\dot{x} - \dot{y}$ nor $\dot{y} - \dot{x}$ is nonnegative, \dot{x} and \dot{y} are termed noncomparable.

2.4. Generalized Matrices

In the algebra C , a generalized matrix is a rectangular array of generalized scalars. As these generalized scalars are arrays in $\mathbb{C}^{I_1 \times I_2}$, it is logical to represent the foundational data form of a generalized matrix in $C^{D_1 \times D_2}$ as an order-four array in $\mathbb{C}^{I_1 \times I_2 \times D_1 \times D_2}$. We designate this format as the little-endian representation of a generalized matrix. Conversely, some authors may arrange a generalized matrix in $C^{D_1 \times D_2}$ as an array in $\mathbb{C}^{D_1 \times D_2 \times I_1 \times I_2}$. We term this the big-endian representation, used in Kilmer et al's work [13]. The conversion between the little-endian and big-endian protocols is straightforward. Given this multi-way array structure, some authors refer to these generalized matrices as tensors [21], though they are distinct from canonical tensors with complex entries.

The operations of generalized matrices mirror those of traditional matrices. Specifically, if we have one generalized matrix in $C^{D_1 \times D_2} \equiv \mathbb{C}^{I_1 \times I_2 \times D_1 \times D_2}$ and another in $C^{D_2 \times D_3} \equiv \mathbb{C}^{I_1 \times I_2 \times D_2 \times D_3}$, their multiplication yields a generalized matrix in $C^{D_1 \times D_3} \equiv \mathbb{C}^{I_1 \times I_2 \times D_1 \times D_3}$. In a similar vein, constructs such as the conjugate transpose and diagonal matrix can be defined analogously, with brief elaboration provided as necessary. For more detailed discussions of these concepts, please refer to [16].

2.5. Singular Value Decomposition of A Generalized Matrix

To leverage the structure of a generalized matrix, it is often decomposed into a sequence of simpler components. Parallel to the compact Singular Value Decomposition (SVD) in traditional matrices, a crucial decomposition, called TSVD (Tensorial SVD), of a generalized matrix $\dot{X} \in C^{D_1 \times D_2}$ is represented as:

$$\dot{X} = \dot{U} \circ \dot{S} \circ \dot{V}^* \quad (2)$$

where $\dot{U} \in C^{D_1 \times D}$, $\dot{S} \in C^{D \times D}$, and $\dot{V} \in C^{D \times D}$, with $D \doteq \min(D_1, D_2)$. The symbol \dot{V}^* indicates the conjugate transpose of \dot{V} , and $\dot{S} \doteq \text{diag}(\dot{\sigma}_1, \dots, \dot{\sigma}_D)$ is a diagonal generalized matrix with nonnegative generalized scalars and the partial order as $\dot{\sigma}_1 \geq \dots \geq \dot{\sigma}_D \geq \dot{z}$.

Furthermore, the following generalized orthogonal constraints are maintained:

$$\dot{U}^* \circ \dot{U} = \dot{V}^* \circ \dot{V} = \dot{I} \doteq \text{diag}(\dot{e}, \dots, \dot{e}) \in C^{D \times D} \equiv \mathbb{C}^{I_1 \times I_2 \times D \times D} \quad (3)$$

Here, \dot{I} denotes the generalized identity matrix, which has diagonal entries of \dot{e} and off-diagonal entries of \dot{z} .

Equation (2) defines the tensorial singular value decomposition (TSVD) of a generalized matrix. Although its non-uniqueness persists, numerous theories expound on the computational and operational aspects of TSVD. Among these, one particularly practical method employs the mechanism of spectral slices.

Given a generalized matrix $\dot{X} \in C^{D_1 \times D_2}$, represented as an order-four complex array in $\mathbb{C}^{I_1 \times I_2 \times D_1 \times D_2}$, let $\text{Tensor}(\dot{X})$ map this array into a conventional tensor with identical size and entries. Complying with equation (1), the Fourier transform of \dot{X} can be expressed as the

following multi-modal multiplication:

$$\tilde{X} \doteq F(\dot{X}) = \text{Tensor}(\dot{X}) \times_1 W_1 \times_2 W_2. \quad (4)$$

Since all operations of generalized scalars in the Fourier domain are entry-wise, the subsequent definition can be employed to decompose generalized matrices in the Fourier domain and establish further constructs.

Definition 2.7 (Spectral Slice) *For any generalized matrix $\dot{X} \in C^{D_1 \times D_2}$ and its Fourier transform $\tilde{X} \in \mathbb{C}^{I_1 \times I_2 \times D_1 \times D_2}$ as defined in Equation (4), \tilde{X} can be partitioned into K spectral slices (where $K = I_1 \cdot I_2$). Each spectral slice, indexed by (i_1, i_2) , is a conventional complex matrix denoted by $\tilde{X}(i_1, i_2) \in \mathbb{C}^{D_1 \times D_2}$. This satisfies the following equation for all i_1, i_2, d_1, d_2 :*

$$(\tilde{X}(i_1, i_2))_{d_1, d_2} = (\tilde{X})_{i_1, i_2, d_1, d_2}. \quad (5)$$

Leveraging the concept of spectral slices, various constructs can be introduced. For instance, the Tensorial Singular Value Decomposition (TSVD) of a generalized matrix is outlined in Algorithm 1.

Algorithm 1 Tensorial Singular Value Decomposition via Spectral Slices

```

1: procedure TSVD( $\dot{X}$ )
2:   Apply equation (4) to compute the transform  $\tilde{X}$  from  $\dot{X}$ .
3:   for  $(i_1, i_2) \in [I_1] \times [I_2]$  do
4:     Compute the compact SVD of each spectral slice, where  $\tilde{X}(i_1, i_2) = U \cdot S \cdot V^*$ .
5:     Store the resulting  $U$ ,  $S$ , and  $V$  in  $\tilde{U}(i_1, i_2)$ ,  $\tilde{S}(i_1, i_2)$ , and  $\tilde{V}(i_1, i_2)$  respectively.
6:   end for
7:   Apply the inverse transform  $F^{-1}$  to each of  $\tilde{U}$ ,  $\tilde{S}$ , and  $\tilde{V}$  to obtain  $\dot{U}$ ,  $\dot{S}$ , and  $\dot{V}$ .
8: end procedure

```

Spectral slices facilitate the extension of numerous conventional algorithms, including the acclaimed Singular Value Thresholding (SVT) algorithm, reported in [1, 2], and integral to Low Rank Matrix Completion (LRMC) issues. The particular implementation of Singular Value Thresholding on generalized scalars is presented in Algorithm 2.

Algorithm 2 Tensorial Singular Value Thresholding via Spectral Slices

```

1: procedure  $\dot{Y} = \text{TSVT}(\dot{X}, \tau)$  where  $\tau$  is a small positive constant
2:   Use equation (4) to calculate the transform  $\tilde{X}$  of  $\dot{X}$ .
3:   for  $(i_1, i_2) \in [I_1] \times [I_2]$  do
4:     Compute the compact SVD on each spectral slice, such that  $\tilde{X}(i_1, i_2) = U \cdot S \cdot V^*$ .
5:     Store the singular value thresholding result  $U \cdot C_\tau(S) \cdot V^*$  in  $\tilde{Y}(i_1, i_2)$ .
6:   end for
7:   Apply the inverse transform  $F^{-1}$  to  $\tilde{Y}$  to obtain  $\dot{Y}$ , the approximation of  $\dot{X}$ .
8: end procedure

```

Significantly, these definitions apply when all generalized scalars are arrays of size $I_1 \times I_2$. For higher-order arrays of size $I_1 \times I_2 \times \dots \times I_N$, the replacement of I_1, I_2 and i_1, i_2 with I_1, I_2, \dots, I_N and i_1, i_2, \dots, i_N respectively suffices for these order- N generalized scalars.

3. Low-Rank Matrix Completion and Its Generalizations

3.1. Low-Rank Matrix Completion

A variant of the matrix completion problem involves determining the minimum rank matrix $X \in \mathbb{R}^{D_1 \times D_2}$ that aligns with the desired matrix M for all observed elements within the index set Ω [2]. This problem can be mathematically expressed as:

$$\underset{X}{\text{minimize}} \text{rank}(X) \text{ subject to } (X)_{i,j} = (M)_{i,j} \quad \forall (i,j) \in \Omega. \quad (6)$$

Considering the NP-hard nature of the initial minimization problem, the solution to equation (6) can, with high probability, be reformulated as a solution to the following convex optimization problem in most practical scenarios:

$$\underset{X, E}{\text{minimize}} \|X\|_* \text{ subject to } X + E = M, \quad G_\Omega(E) = \text{array of zeros} \quad (7)$$

where $G_\Omega : \mathbb{R}^{D_1 \times D_2} \rightarrow \mathbb{R}^{D_1 \times D_2}$ represents the linear operator preserving entries within the set Ω and setting those outside of Ω to zero.

The augmented Lagrange Multiplier function for the previously defined minimization problem is formalized as:

$$L(X, E, Y, \tau) = \|X\|_* + \langle Y, M - X - E \rangle + \frac{1}{2\tau} \|M - X - E\|_F^2 \quad (8)$$

with Y representing the dual variable and $\tau > 0$.

The Alternating Direction Method of Multipliers (ADMM) [18] is applied to iteratively refine the optimization variables X and E , as detailed in Algorithm 3. Here, $M \in \mathbb{R}^{D_1 \times D_2}$, and Ω represents a random nonempty proper subset of the Cartesian product $[D_1] \times [D_2]$. Moreover, D_τ in Line 9 denotes the SVT operator with the threshold τ .

3.2. Matrix Completion over Higher-order Generalized Scalars

Following the application of ADMM for optimizing equation (7), numerous authors have proposed methods for extending the completion process to third-order arrays. For instance, utilizing Kilmer et al.’s “t-product” model, Lu et al extended the aforementioned completion approach to third-order tensors [14].

Though termed tensor algorithms, the approach by Lu et al. [14] and other variants [5, 15, 17] are essentially matrix completion algorithms operating on first-order generalized scalars. However, as previously noted in Section 2, the order of generalized scalars can indeed be defined

Algorithm 3 ADMM for the solution of equation (7)

```

1: procedure  $X_{COMP}$  = MATRIXCOMPLETION( $M, \Omega$ )
2:   Initialization:  $k \leftarrow 0$ ,  $Y_0 = E_0 \leftarrow$  array of zeros,  $\alpha \leftarrow 0.9$ ,  $\tau_0 \leftarrow 10^4$ ,  $\tau_{\min} \leftarrow 10^{-6}$ 
3:   Set the missing entries of  $M$ , i.e.,  $(i, j) \in \Omega^c$ , to zeros
4:   while neither convergence nor predefined maximum iterations achieved do
5:      $X_{k+1} \leftarrow \operatorname{argmin}_X \|X\|_* + \frac{1}{2\tau_k} \|X + E_k - M - \tau_k \cdot Y_k\|_F^2 \equiv D_{\tau_k}(M - E_k + \tau_k \cdot Y_k)$ 
6:      $E_{k+1} \leftarrow G_{\bar{\Omega}}(M - X_{k+1} + \tau_k \cdot Y_{k+1})$ 
7:      $Y_{k+1} \leftarrow Y_k + \frac{1}{\tau_k} \cdot (M - X_{k+1} - E_{k+1})$ 
8:      $\tau_{k+1} \leftarrow \max(\alpha \cdot \tau_k, \tau_{\min})$  and  $k \leftarrow k + 1$ 
9:   end while
10:   $X_{COMP} \leftarrow X_k$ 
11: end procedure

```

as higher order.

Given that higher-order arrays encapsulate more structural information than their lower-order counterparts in real-world scenarios, we leverage this aspect by elevating the order of arrays via a pixel neighborhood strategy, initially introduced in [22], yet broadly overlooked by the research community.

Specifically, Figure 1 showcases the application of a “ 3×3 pixel neighborhood” strategy to increase the order of a 4×4 pixel grayscale image. Note that the figure depicts the order-four array outcome as a two-dimensional array of two-dimensional blocks.

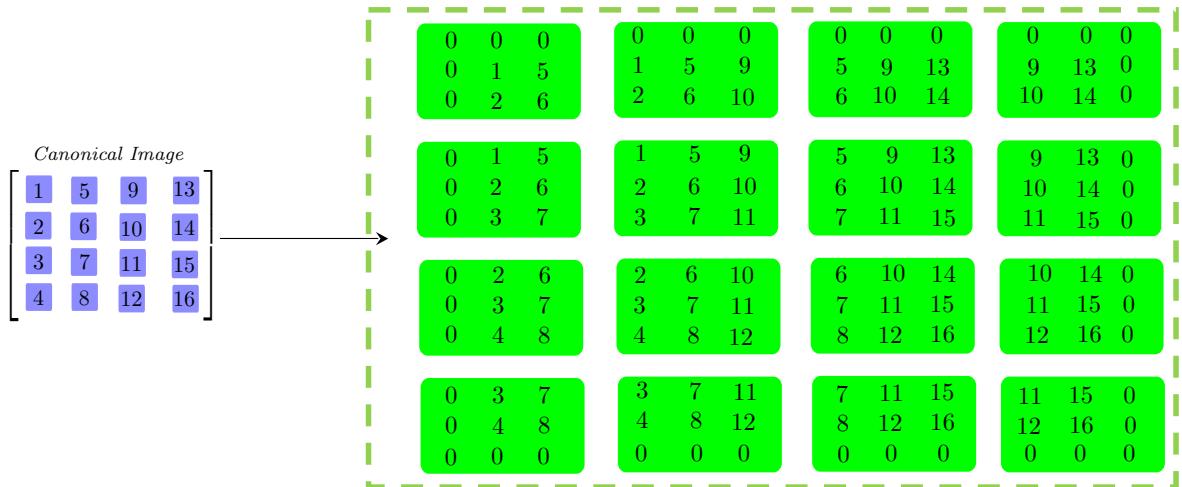


Figure 1: Elevating a 4×4 grayscale image to a fourth-order array using a central 3×3 pixel neighborhood strategy.

The employment of this pixel neighborhood strategy leads to an order-four array of size $3 \times 3 \times 4 \times 4$, which is interpreted as a 4×4 matrix with generalized scalars of size 3×3 .

Also note that someone might prefer the underlying array format in the shape of $4 \times 4 \times 3 \times 3$ rather than $3 \times 3 \times 4 \times 4$. We call the construct $3 \times 3 \times 4 \times 4$ the little-endian array format of generalized matrices, and the other the big-endian format of generalized matrices.

The endian protocol is not integral to the algebraic definitions of generalized matrices. Various array programming languages, including MATLAB and Python (NumPy), incorporate convenient tools to facilitate the conversion between endian protocols. While Kilmer et al.’s “t-product” model employs big-endianness, we utilize little-endianness, which boasts the advantage of aligning the multi-mode multiplication formulation of the Fourier transform of generalized matrices (as per equation (4)) with that of generalized scalars (as per equation (1)).

Given that each pixel value can be elevated to a generalized scalar, enabling the conversion of a traditional matrix into a generalized one with higher-order, fixed-size arrays as entries, the extension of Algorithm 3 for generalized matrix completion is straightforward. The generalization of Line 9 in Algorithm 3 can be achieved using TSVD, as outlined in Algorithm 2. Meanwhile, Line 10 in Algorithm 3 is extended by the elevated linear operator $G_{\bar{\Theta}}$, which preserves entries within the set $\bar{\Theta}$ and setting those outside of $\bar{\Theta}$ to \dot{z} .

We propose a generalized matrix completion algorithm for the recovery of multi-spectral images containing missing values, as depicted in Algorithm 4 wherein $M \in \mathbb{R}^{D_1 \times D_2 \times D_3}$, Ω represents a random nonempty proper subset of the Cartesian product $[D_1] \times [D_2] \times [D_3]$, the outcome Θ is a proper subset of $[I_1] \times [I_2] \times [D_3] \times [D_1] \times [D_2]$, and the generalized scalars with size $I_1 \times I_2 \times D_3$ are assumed where I_1 and I_2 are both odd numbers.

Algorithm 4 Higher-order TNN: ADMM for recovering an image with missing values

```

1: procedure  $X_{COMP} = \text{SEPCTRALIMAGECOMPLETION}(M, \Omega)$ 
2:   Assign an unlikely value, such as  $-1$ , to the missing entries indexed by  $\Omega$ .
3:   Construct  $M_1, M_2, \dots, M_{D_3} \in \mathbb{R}^{I_1 \times I_2 \times D_1 \times D_2}$  using the  $I_1 \times I_2$  neighborhood strategy for
   every frontal slice of  $M$ .
4:   Construct  $M_{UP} \in \mathbb{R}^{I_1 \times I_2 \times D_1 \times D_2 \times D_3}$  by aligning  $M_1, M_2, \dots, M_{D_3}$  along the mode-5 .
5:   Convert  $M_{UP}$  into a generalized matrix  $\dot{M} \in C^{D_1 \times D_2} \equiv \mathbb{R}^{I_1 \times I_2 \times D_3 \times D_1 \times D_2}$  by permut-
   ing of the indices of array.
6:   Store the positions of the entries “ $-1$ ” of  $\dot{M}$  within  $\Theta$ .
7:   Initialization:  $k \leftarrow 0$ ,  $\dot{Y}_0 \leftarrow$  array of zeros,  $\dot{E}_0 \leftarrow$  array of zeros,  $\alpha \leftarrow 0.9$ ,  $\tau_{\min} \leftarrow 10^{-6}$ 
8:   while neither convergence nor predefined maximum iterations achieved do
9:      $\dot{X}_{k+1} \leftarrow \text{TSVT}(\dot{M} - \dot{E}_k + \tau_k \cdot \dot{Y}_k, \tau_k)$ 
10:     $\dot{E}_{k+1} \leftarrow G_{\bar{\Theta}}(\dot{M} - \dot{X}_{k+1} + \tau_k \cdot \dot{Y}_{k+1})$ 
11:     $\dot{Y}_{k+1} \leftarrow Y_k + \frac{1}{\tau_k}(\dot{M} - \dot{X}_{k+1} - \dot{E}_{k+1})$ 
12:     $\tau_{k+1} \leftarrow \max(\alpha \cdot \tau_k, \tau_{\min})$  and  $k \leftarrow k + 1$ 
13:   end while
14:   Employ a column-wise (MATLAB compliant) protocol to reshape  $\dot{X}_k$  into an  $I_1 I_2 \times$ 
    $D_3 D_1 D_2$  matrix, extract the central row, and subsequently reshape it column-wise into
   an array  $X_{DOWN} \in \mathbb{R}^{D_3 \times D_1 \times D_3}$ .
15:   Permute the indices of  $X_{DOWN}$  to convert it into an array in  $\mathbb{R}^{D_1 \times D_3 \times D_3}$ .
16:   Adjust the entries of  $X_{DOWN}$  to nonnegative integers and preserve the adjusted array
   in  $X_{COMP}$ , constituting the recovered multi-spectral image.
17: end procedure

```

Lines 3, 4, and 5 up-convert the input multi-spectral image M , an initial $D_1 \times D_2 \times D_3$

array, into a $D_1 \times D_2$ generalized matrix M_{UP} , with generalized scalars of size $I_1 \times I_2 \times D_3$. Conversely, Line 14 down-converts the optimal generalized matrix \dot{X}_k into a third-order array X_{DOWN} .

Algorithm 4 builds upon Lu et al.’s tensor completion algorithm [14]. While Algorithm 4 can be elegantly derived through the generalized matrix model, superseding Kilmer et al.’s “t-product” model, the scope of this article preclude such an exhaustive investigation. Hence, this paper primarily concentrates on empirical validation.

4. Rank Considerations

4.1. Tubal Rank and Average Rank

An RGB image, a specific instance of multispectral images, comprises three monochromatic channels. Each channel in real-world RGB imagery can be adequately approximated using lower-rank matrices. Yet, when viewed as a third-order tensor, the rank of an RGB image, defined by the minimal quantity of rank-one tensor addends, becomes computationally intractable. Consequently, the traditional tensor rank is unsuitable for modeling the optimal recovery of an RGB image.

Kilmer et al.’s approach is the introduction of a novel rank concept termed tubal rank for a third-order array. Specifically, for a given third-order array \dot{X} , with its TSVD defined as $\dot{X} = \dot{U} \circ \dot{S} \circ \dot{V}^*$, the tubal rank of \dot{X} corresponds to the count of non-zero (i.e., not equal to \dot{z}) diagonal generalized scalars in \dot{S} . Nonetheless, under this definition, a generalized matrix of full tubal rank can consist of a full-rank matrix as one of its spectral slices, with all other spectral slices being zero matrices.

Addressing this issue, Lu et al. proposed defining the average of all spectral slice ranks as the “average rank” of a generalized matrix [17]. This “average rank” definition is more fitting than tubal rank but is solely utilized in Lu et al.’s TRPCA (Tensor Robust Principal Component Analysis) algorithm, rather than for the higher-order array recovery problem presented in [14]. Moreover, despite the potential for the “average rank” to be fractional, the mathematical justification for the “average rank” has not been adequately addressed.

4.2. Higher-order Rank and Its Trace Variant

In addition to Kilmer et al.’s tubal rank and Lu et al.’s average rank, another pertinent concept is the higher-order rank introduced by Liao and Maybank in the appendix of [16]. Specifically, given a generalized matrix \dot{X} with its tensor singular value decomposition (TSVD) $\dot{X} = \dot{U} \circ \dot{S} \circ \dot{V}^*$, the higher-order rank of \dot{X} is a nonnegative generalized scalar. It is calculated as the sum of the diagonal entries of the product $\dot{S}^\dagger \circ \dot{S}$, where \dot{S}^\dagger denotes the pseudo-inverse of \dot{S} .

The aforementioned definition aligns with its analogous concept in traditional matrices. The pseudo-inverse of a generalized matrix can be established using spectral slices, similar to

Algorithms 1 and 2. Specifically, the pseudo-inverse \dot{X}^\dagger of an input generalized matrix \dot{X} is defined by assigning each spectral slice’s pseudo-inverse as its corresponding slice in the result. It is not difficult to verify that the above-defined defined pseudo-inverse \dot{X}^\dagger corresponds to the product $\dot{V} \circ \dot{S}^\dagger \circ \dot{U}^*$, obtained using conventional approach.

By the preceding definition, it is easy to follow that the higher-order rank of any generalized matrix is a nonnegative generalized scalar. It can be sorted alongside comparable nonnegative counterparts using the partial order introduced in Section 2.3. However, sometimes, we favor a more efficient rank system akin to the fully ordered ones proposed by Kilmer et al. and Lu et al., as contrasted with the partially ordered, higher-order counterparts.

Reassuringly, the Szpilrajn Extension Theorem affirms that the partially ordered rank system, proposed in [16], can invariably be extended to a fully ordered construct.

Multiple strategies exist for transitioning the higher-order rank system to its entirely ordered equivalents. Considering any higher-order rank of a generalized matrix, all spectral points (i.e., Fourier entries) are nonnegative integers. Consequently, Kilmer et al.’s tubal rank denotes the maximum value among these spectral points, whereas Lu et al.’s average rank equates to their arithmetic mean.

Typically, in most scenarios, Lu et al.’s average rank is regarded as a superior statistic for a higher-order rank. Nevertheless, to avoid fractional rank values, we propose the use of the sum, rather than the arithmetic mean, of the spectral points of a higher-order rank to define its corresponding fully ordered rank. Furthermore, given that any generalized scalar also functions as a finite-dimensional linear endomorphic operator, the previously defined “sum rank” of a generalized matrix equates to the trace of the higher-order rank, hence it would be appropriate to formally label it as “trace rank”.

5. Experiments

Section 5 presents experimental validation and performance analysis of the related algorithms.

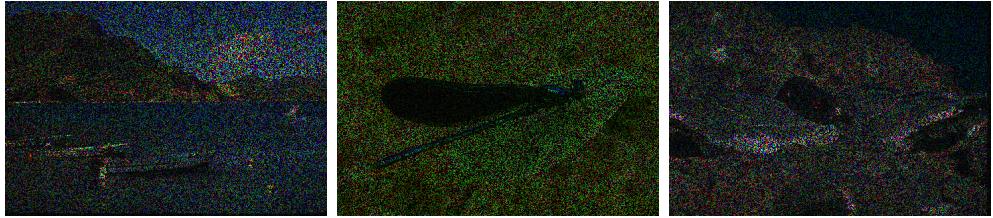
In our initial experiments, we utilize the Berkeley Segmentation dataset as the benchmark to evaluate the performance of four pertinent algorithms: Tubal-TNN [5], T-TNN [15], TNN [14], and our Higher-order TNN. Three RGB images, namely “Resort”, “Insect”, and ”Gulls” are selected for the first experiment. These images are represented as $321 \times 481 \times 3$ unsigned integer arrays.

To compare completion performances, we randomly select 70% of each image’s pixel values as “missing” entries. The uncompleted observed images, with missing values set to zero, provide a visual representation of the results. Figure 3 displays the original images alongside their incomplete versions.

We employ the three rival algorithms and our Higher-order TNN, delineated in Algorithm 4, to obtain an optimal, complete RGB image of identical size. Image completion quality is



(a) Original Images



(b) Observed Images with 70% Entries Missing

Figure 2: Three original and observed images: “Resort”, “Insect”, and “Gulls” from the Berkeley Segmentation Dataset.

quantified using the PSNR (Peak Signal-to-Noise Ratio), defined as:

$$PSNR = 10 \cdot \log_{10} \frac{D_1 \cdot D_2 \cdot D_3}{\|X_{COMP} - M\|_F^2} \quad (9)$$

Here, $D_1 \times D_2 \times D_3$ signifies the size of the RGB image. Notably, before computing the PSNR, values of both X_{COMP} and M are normalized to 1, adjusting the ”peak” value involved in the PSNR computation from 255 to 1.

Figure 3 presents visual and quantitative comparisons of four competing algorithms’ performance in completing the observed images: “Resort”, “Insect”, and “Gulls” from the Berkeley Segmentation Dataset. The proposed Higher-Order TNN outperforms its rivals in the term of PSNRs by at least 1 dB, 1.4 dB, and 1.6 dB respectively.

In our second experiment, we utilize three distinct images — “Temple”, “Chapel”, and “Grass-flower”, each with 50% entries randomly missing, to conduct completions analogous to the first experiment. Figure 5 depicts the original images alongside their incomplete versions.

Figure 5 presents the visual and quantitative comparisons of the four related algorithms for image completion. Consistent with the results from the initial experiment, the Higher-Order TNN significantly outperforms the other algorithms, achieving gains of at least 1.5 dB, 1.2 dB, and 2.2 dB, respectively.

After conducting initial experiments on six RGB images, we broadened our investigation to encompass 100 randomly chosen training RGB images from the Berkeley Segmentation Dataset.

Figure 6 presents the PSNR comparison of four competing algorithms for completing the 100 images with 70% of entries missing. Figure 7 illustrates the PSNR comparison of four competing algorithms for completing the 100 images with 50% of entries missing. As shown in both figures, our Higher-Order TNN significantly outperforms its rivals, including Tubal-TNN, T-TNN, and TNN.



(a) Results of Tubal-NN [5]



(b) Results of T-TNN [15]



(c) Results of TNN [14]



(d) Results of Higher-Order TNN (ours)

PSNR Comparison of Four Competing Algorithms

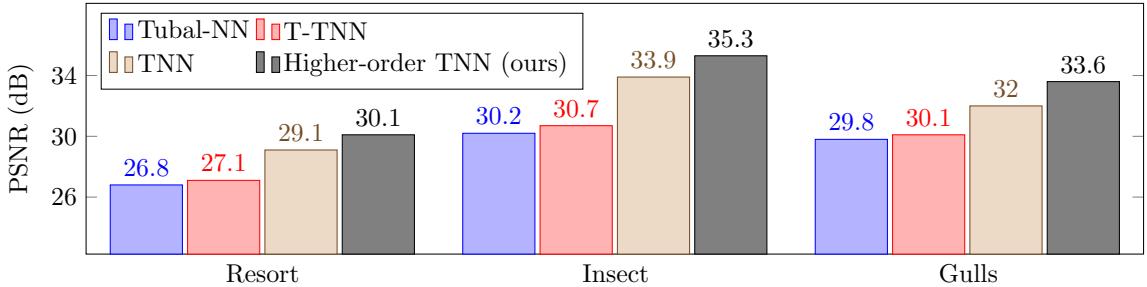


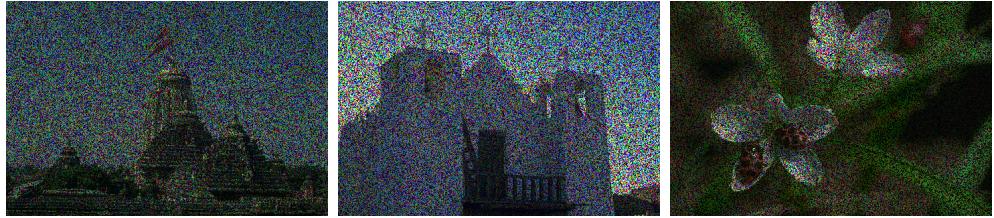
Figure 3: Visual and quantitative comparisons of the performance of four related algorithms in completing the images “Resort”, “Insect”, and “Gulls”

6. Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus



(a) Original Images



(b) Observed Images with 50% Entries Missing

Figure 4: Three original and observed images: “Temple”, “Chapel”, and “Grass-flower” from the Berkeley Segmentation Dataset.

eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Acknowledgments

This work was partially supported by

References

- [1] Candès, Emmanuel J., Li, Xiaodong, Ma, Yi, and Wright, John. 2011. “Robust principal component analysis?.” *Journal of the ACM (JACM)* 58 (3): 1–37. New York: ACM.
- [2] Candès, Emmanuel J., and Benjamin Recht. 2012. “Exact Matrix Completion via Convex Optimization.” *Communications of the ACM* 55 (6): 111–119. New York: ACM.
- [3] Kolda, Tamara G., and Brett W. Bader. 2009. “Tensor Decompositions and Applications.” *SIAM Review* 51 (3): 455–500. Philadelphia: Society for Industrial and Applied Mathematics.
- [4] Liu, Ji, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. 2012. “Tensor Completion for Estimating Missing Values in Visual Data.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (1): 208–220. Piscataway, NJ: IEEE.



(a) Tubal-NN [5]



(b) T-TNN [15]



(c) TNN [14]



(d) Higher-Order TNN (ours)

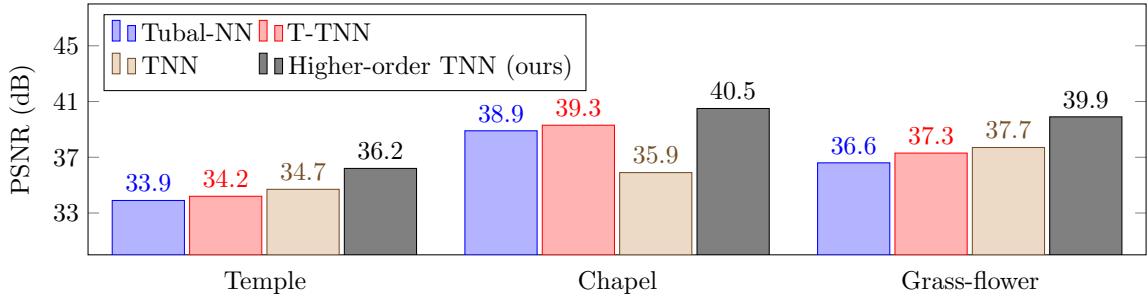


Figure 5: Visual and quantitative comparisons of the performance of four related algorithms in completing the images “Temple”, “Chapel”, and “Grass-flower”

- [5] Zhang, Zemin, Gregory Ely, Shuchin Aeron, Ning Hao, and Misha Kilmer. 2014. “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, 3842–3849.
- [6] Acar, Evrim, Daniel M. Dunlavy, Tamara G. Kolda, and Morten Mørup. 2011. “Scalable Tensor Factorizations for Incomplete Data.” Chemometrics and Intelligent Laboratory Systems 106 (1): 41–56. Amsterdam: Elsevier.
- [7] Gandy, Silvia, Benjamin Recht, and Isao Yamada. 2011. “Tensor Completion and Low-n-

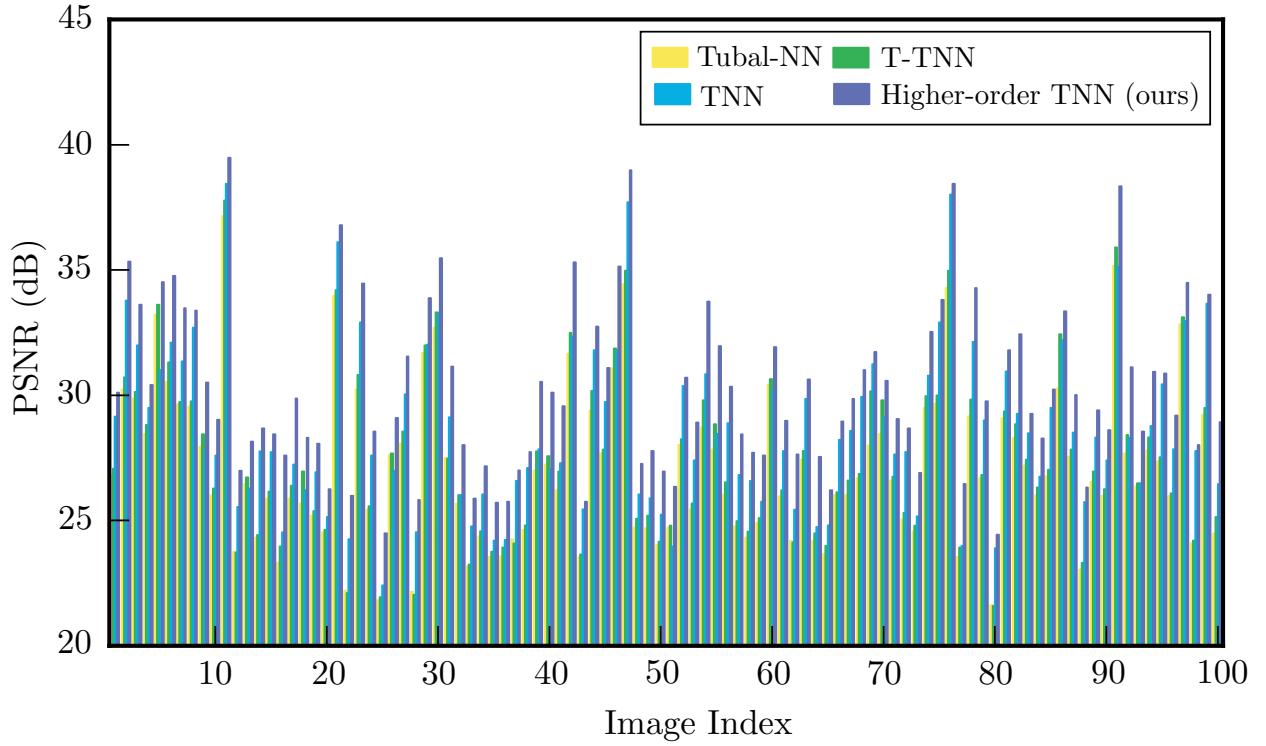


Figure 6: PSNR Comparison of Four Algorithms for RGB Image Completion with 70% Missing Entries

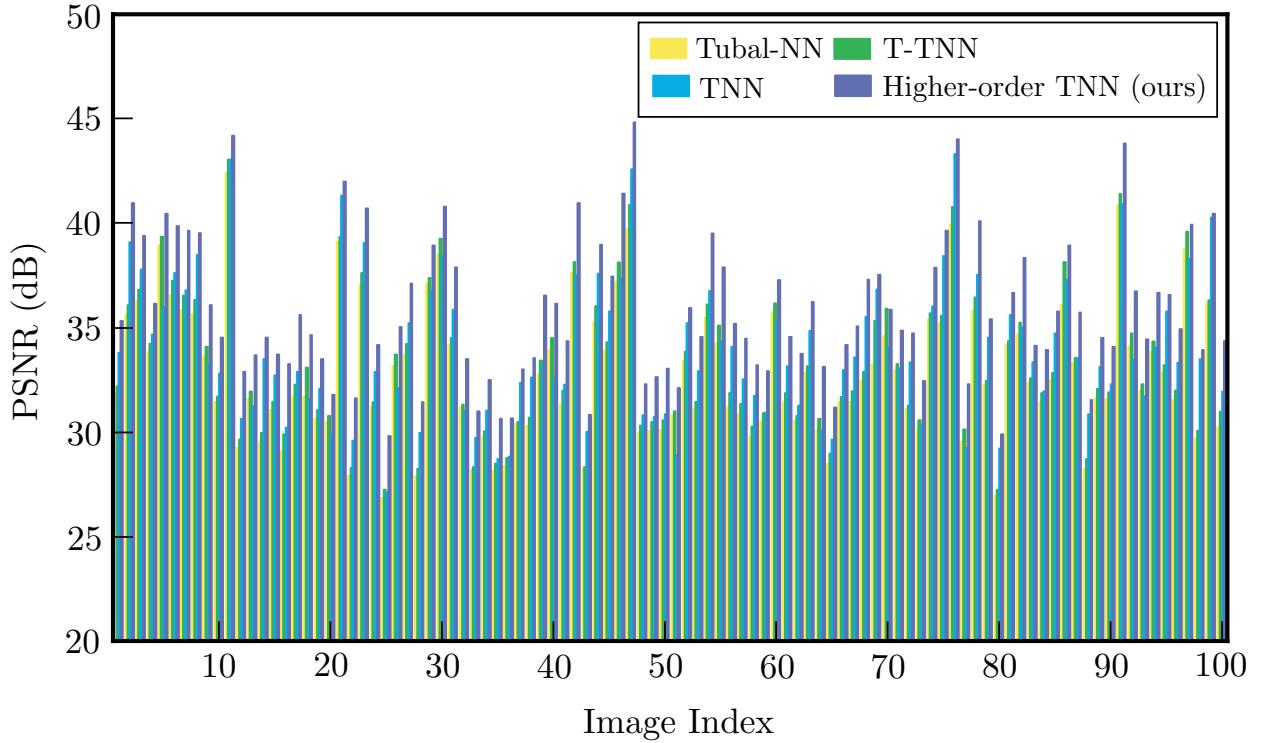


Figure 7: PSNR Comparison of Four Algorithms for RGB Image Completion with 50% Missing Entries

rank Tensor Recovery via Convex Optimization.” Inverse Problems 27 (2): 025010. Bristol, England: IOP Publishing.

[8] Signoretto, Marco, Van de Plas, Raf, De Moor, Bart, and Suykens, Johan AK. “Tensor

- versus Matrix Completion: A Comparison with Application to Spectral Data.” IEEE Signal Processing Letters 18, no. 7 (2011): 403-406. IEEE.
- [9] Obozinski, Guillaume, Taskar, Ben, and Jordan, Michael I. “Joint Covariate Selection and Joint Subspace Selection for Multiple Classification Problems.” Statistics and Computing 20 (2010): 231-252. Springer.
- [10] Kolda, Tamara G, and Sun, Jimeng. “Scalable Tensor Decompositions for Multi-aspect Data Mining.” In 2008 Eighth IEEE International Conference on Data Mining, 363-372. IEEE.
- [11] Sun, Jimeng, Papadimitriou, Spiros, Lin, Ching-Yung, Cao, Nan, Liu, Shixia, and Qian, Weihong. “Multivis: Content-based Social Network Exploration through Multi-way Visual Analysis.” In Proceedings of the 2009 SIAM International Conference on Data Mining, 1064-1075. SIAM.
- [12] Tucker, Ledyard R. “Some Mathematical Notes on Three-mode Factor Analysis.” Psychometrika 31, no. 3 (1966): 279-311. Springer.
- [13] Kilmer, Misha E, and Martin, Carla D. “Factorization Strategies for Third-order Tensors.” Linear Algebra and its Applications 435, no. 3 (2011): 641-658. Elsevier.
- [14] Lu, Canyi, Feng, Jiashi, Lin, Zhouchen, and Yan, Shuicheng. “Exact Low Tubal Rank Tensor Recovery from Gaussian Measurements.” arXiv preprint arXiv:1806.02511 (2018).
- [15] Xue, Shengke, Qiu, Wenyuan, Liu, Fan, and Jin, Xinyu. “Low-rank Tensor Completion by Truncated Nuclear Norm Regularization.” In 2018 24th International Conference on Pattern Recognition (ICPR), 2600-2605. IEEE.
- [16] Liao, Liang, and Maybank, Stephen John. “Generalized Visual Information Analysis via Tensorial Algebra.” Journal of Mathematical Imaging and Vision 62, no. 4 (2020): 560-584. Springer.
- [17] Lu, Canyi, Feng, Jiashi, Chen, Yudong, Liu, Wei, Lin, Zhouchen, and Yan, Shuicheng. “Tensor Robust Principal Component Analysis with a New Tensor Nuclear Norm.” IEEE Transactions on Pattern Analysis and Machine Intelligence 42, no. 4 (2019): 925-938. IEEE.
- [18] Boyd, Stephen, Parikh, Neal, Chu, Eric, Peleato, Borja, Eckstein, Jonathan, and others. “Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers.” Foundations and Trends in Machine learning 3, no. 1 (2011): 1-122. Now Publishers, Inc.
- [19] Lu, Canyi, Tang, Jinhui, Yan, Shuicheng, and Lin, Zhouchen. “Generalized Nonconvex Nonsmooth Low-rank Minimization.” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4130-4137. IEEE, 2014.

- [20] Martin, David, Fowlkes, Charless, Tal, Doron, and Malik, Jitendra. “A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics.” In Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, vol. 2, 416-423. IEEE, 2001.
- [21] Kilmer, Misha E., Braman, Karen, Hao, Ning, and Hoover, Randy C. “Third-Order Tensors as Operators on Matrices: A Theoretical and Computational Framework with Applications in Imaging.” SIAM Journal on Matrix Analysis and Applications 34, no. 1 (2013): 148-172. SIAM.
- [22] Liao, Liang, and Stephen John Maybank. 2020. “General data analytics with applications to visual information analysis: A provable backward-compatible semisimple paradigm over t-algebra.” arXiv preprint arXiv:2011.00307.