Hierarchical Tensor Ring Completion

Abdul Ahad, Zhen Long, Ce Zhu, Fellow, IEEE and Yipeng Liu, Senior Member, IEEE

Abstract

Tensor completion can estimate missing values of a high-order data from its partially observed entries. Recent works show that low rank tensor ring approximation is one of the most powerful tools to solve tensor completion problem. However, existing algorithms need predefined tensor ring rank which may be hard to determine in practice. To address the issue, we propose a hierarchical tensor ring decomposition for more compact representation. We use the standard tensor ring to decompose a tensor into several 3-order sub-tensors in the first layer, and each sub-tensor is further factorized by tensor singular value decomposition (t-SVD) in the second layer. In the low rank tensor completion based on the proposed decomposition, the zero elements in the 3-order core tensor are pruned in the second layer, which helps to automatically determine the tensor ring rank. To further enhance the recovery performance, we use total variation to exploit the locally piece-wise smoothness data structure. The alternating direction method of multiplier can divide the optimization model into several subproblems, and each one can be solved efficiently. Numerical experiments on color images and hyperspectral images demonstrate that the proposed algorithm outperforms state-of-the-arts ones in terms of recovery accuracy.

Index Terms

low rank tensor approximation, total variation, tensor completion, tensor singular value decomposition, tensor network

I. Introduction

Tensor, which is the higher-order generalization of vector and matrix, provides a natural form to represent higher-order data. For example, a color image has three indices that can be represented by a 3-order tensor. The tensor for multidimensional data processing has attracted much attention in different fields such as signal and image processing [1], [2], computer vision [3], quantum chemistry [4], [5] and data mining [6], [7]. However, in some applications, a part of entries of multidimensional data are missing during data acquisition or transmission, which has a significant influence on subsequent processing. Tensor completion can recover missing entries from its observed entries by exploit the coherence in multi-linear space. The low-rank method is one of the most powerful ones for tensor completion problems [3], [8]–[16].

The low-rank tensor completion methods are mainly divided into two categories according to different tensor rank formats. The first one is based on tensor factorization. In this group, the tensor rank is predefined and the goal is to optimize the factors of tensor decomposition. For example, in [9] and [10], the missing values of data are recovered with given CP rank in advance and each factor is updated by alternating least square (ALS) and gradient methods. Following it, Tucker-ALS [11] recovers the missing entries with known Tucker rank. The second one is to directly minimize the tensor rank. However, the rank is a non-convex function and solving the rank problem is NP-hard. Most existing methods use different nuclear norms as convex surrogates to solve this problem. For instance, HaLRTC [12] has been proposed to recover missing elements, which directly minimizes Tucker rank, and uses the nuclear norm of the unfolding matrix to solve the Tucker rank problem. Besides, a series of works [13], [17] are developed to achieve a better recovery performance with Tucker decomposition. Apart from CP rank and Tucker rank in the two groups [12], [13], [18]–[26], some other tensor ranks are used too, such as tensor train rank [27], [28], tensor tree rank [29]–[32], tensor ring rank [16], [33], [34] and tubal rank [35]–[37].

Among all the tensor decompositions, tensor networks can capture more correlations than the rest ones. The corresponding tensor network based completion methods show superior performance, such as STTC [8], TMac-TT [38] and TR-ALS [34]. However, tensor train rank has its entries large for the middle factors and small for border factors, which leads to an unbalanced decomposition. To alleviate the drawbacks of tensor-train, the generalization...
of tensor-train decomposition named tensor-ring decomposition has been proposed in \cite{33}. Low-rank tensor-ring decomposition is a powerful tool to recover missing data. In \cite{34}, the authors directly optimize tensor ring factors with predefined tensor ring rank. Following it, in \cite{39}, the authors add total variation (TV) regularization to model the local structure of image for low rank tensor ring completion on remote sensing image reconstruction. However, these two algorithms need predefined tensor ring rank which is hard to determine in practice. In \cite{40}, the authors propose another low rank tensor ring completion method which applies rank minimization regularization on 3-order factors. It results in high computational cost when data are in large-scale.

In this paper, to improve the tensor-ring based works, we propose a hierarchical low-rank tensor ring decomposition. For the first layer, we use the traditional tensor-ring decomposition model to factorize a tensor into many 3-order sub-tensors. For the second layer, each 3-order tensor is further decomposed by the tensor singular value decomposition (t-SVD) \cite{35}. For each step, we prune the zero elements in the 3-order core tensor, which achieves automatically rank determination. We use this advanced tensor network for exploiting the global multilinear data structure in low rank tensor completion. In the proposed optimization model for tensor completion, we additionally employ total variation to exploit the local similarity of data in the form of piecewise smoothness. The alternating direction method of multipliers (ADMM) is used to solve the optimization problem. For each subproblem, we calculate the corresponding variable by fixing the rest ones. In particular, for the hierarchical tensor ring decomposition problem, we update the variable from the first layer to the second layer. Experimental results on color images and hyperspectral images (HSI) show that our method outperforms state-of-the-art algorithms in terms of peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), relative square error (RSE) and spectral angle mapper (SAM).

The main contributions of this paper can be summarized as follows:

1) We propose a hierarchical tensor-ring decomposition. The tensor ring factors are further decomposed by t-SVD in the new tensor network. This is the first tensor network using multiple kinds of operators as connections for factors. In fact, the t-SVD for factors in the second layer can be applied to other tensor networks, e.g., tensor train, and the general hierarchical tensor networks can be obtained.

2) We adopt the newly proposed tensor network based rank and total variation terms to simultaneously utilize the multidimensional global data structure and the locally piecewise smoothness data structure.

3) Experimental results on color images and HSI show that the proposed method outperforms state-of-the-art algorithms in terms of recovery accuracy.

The rest of this paper is organized as follows. In Section II, we give the notations and preliminaries about tensor ring decomposition. The proposed model for tensor completion is given in Section III. In Section IV, we present detailed solutions. In Section V, numerical results are demonstrated. The conclusion is drawn in Section VI.

II. NOTATIONS AND PRELIMINARIES

A. Notations

In this paper, a scalar is denoted by standard lower case letter or uppercase letter, e.g., $a \in \mathbb{R}$, and a vector is denoted by boldface lowercase letter e.g., $a \in \mathbb{R}^l$. A matrix is denoted by boldface capital letter, e.g., $A \in \mathbb{R}^{l_1 \times l_2}$. A tensor of order $N \geq 3$ is denoted by boldface calligraphic letters, e.g., $\mathcal{A} \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_N}$. The elements of tensor $\mathcal{A}$ is defined by $a_{i_1,i_2,\cdots,i_N}$, where $i_n$ shows $n^{th}$ index of tensor $\mathcal{A}$.

The Frobenius norm of $\mathcal{A}$ is defined by $\|\mathcal{A}\|_F = \sqrt{\mathcal{A}^\top \mathcal{A}}$. The inner product of two tensors $\mathcal{A}, \mathcal{B}$ with the same size $\mathbb{R}^{l_1 \times l_2 \times \cdots \times l_N}$ can be defined as $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} a_{i_1,i_2,\cdots,i_N} b_{i_1,i_2,\cdots,i_N}$. Letting $\mathcal{A} \in \mathbb{R}^{l_1 \times l_2 \times \cdots \times l_N}$ be an $N$-order tensor, the standard mode-$n$ unfolding of $\mathcal{A}$ can be defined as $\mathcal{A}_{(n)} \in \mathbb{R}^{l_n \times l_1 l_2 \cdots \hat{l}_n \cdots l_N}$. Another mode-$n$ unfolding of tensor is often used in tensor ring, which is defined as $\mathcal{A}_{\text{enr}} \in \mathbb{R}^{l_1 \times l_2 \cdots l_{n-1} l_{n+1} \cdots l_N}$.

The matrix nuclear norm of $A$ is denoted as $\|A\|_* = \sum_{n=1}^N \sigma_n(A)$, where $\sigma_n(A)$ is the singular value of matrix $A$. $d^*$ denotes the conjugate of $d$.

The 3D total variation of $\mathcal{A} \in \mathbb{R}^{l_1 \times l_2 \times l_3}$ can be formulated as follows:

$$\|\mathcal{A}\|_{TV} = \sum_{i_1=1}^{l_1} \sum_{i_2=1}^{l_2} \sum_{i_3=1}^{l_3} \|\nabla_{x} \mathcal{A}\|_1 + \|\nabla_{y} \mathcal{A}\|_1 + \|\nabla_{z} \mathcal{A}\|_1$$

where $(\nabla_{x} \mathcal{A}) = \mathcal{A}_{i_1,i_2,i_3} - \mathcal{A}_{i_1+1,i_2,i_3}$, $(\nabla_{y} \mathcal{A}) = \mathcal{A}_{i_1,i_2,i_3} - \mathcal{A}_{i_1,i_2+1,i_3}$, $(\nabla_{z} \mathcal{A})\mathcal{A}_{i_1,i_2,i_3} - \mathcal{A}_{i_1,i_2,i_3+1}$.
B. Preliminaries on tensor ring decomposition

Definition 1. (tensor ring decomposition) \[33\]

Tensor ring decomposition represents a high-order tensor into multilinear products of low-order tensors in a circular form, where low-order tensors are called TR factors. The element-wise relationship of TR decomposition and generated tensors can be defined as follows:

\[
\mathcal{A}(i_1, i_2, \ldots, i_N) = \text{Trace} \left( \prod_{n=1}^{N} G_n(i_n) \right)
\]

where \( \text{Trace} \{ \cdot \} \) is the matrix trace operation, \( G_n(i_n) \in \mathbb{R}^{R_n \times R_n} \) is the \( i_n \)-th mode-2 slice matrix of \( G_n \), which can also be denoted by \( G_n(:, i_n,:) \) according to MatLab notation. Fig. 1 gives a more intuitive representation of tensor ring decomposition. For simplification, we use the notion \( \mathcal{F}(G_1, \ldots, G_N) \) to represent the tensor ring decomposition of an \( N \)-order tensor.

Definition 2. (tensor ring rank) \[33\]

In a tensor ring, the tensor ring rank is a vector \([R_1; R_2; \ldots; R_N]\). We set all TR-ranks to be equal in this paper, i.e. \( R_n = R, \forall n = 1, \ldots, N \).

Definition 3. (tensor permutation) \[34\]

The \( k \)-th tensor permutation of an tensor \( \mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \) can be defined as \( \mathcal{A}^{P_k} \in \mathbb{R}^{I_1 \times I_{k+1} \times \cdots \times I_N} \) such that \( \forall k, i_k \in [1, I_k] \).

\[
\mathcal{A}^{P_k}(i_k, \ldots, i_N, i_1, \ldots, i_{k-1}) = \mathcal{A}(i_1, \ldots, i_N)
\]

and we can have the following result:

Lemma 1. If \( \mathcal{A} = \mathcal{F}(G_1, \ldots, G_N) \), \( \mathcal{A}^{P_k} = \mathcal{F}(G_k G_{k+1} \cdots G_N G_1 \cdots G_{k-1}) \).

Definition 4. (tensor connect product) \[34\] Assuming \( G_n \in \mathbb{R}^{R_n \times I_n} \), the tensor connect product between \( G_n \) and \( G_{n+1} \) can be defined as:

\[
G_n G_{n+1} \in \mathbb{R}^{R_n \times (I_n \times R_n)}
\]

\( = \text{reshape} (\mathbf{L}(G_n) \times \mathbf{R}(G_{n+1})) \)
Thus, tensor connect product can be formulated as follows:
\[ \mathcal{G} = \mathcal{G}_1 \cdots \mathcal{G}_N \in \mathbb{R}^{R_0 \times (l_1-1) \times R_N}. \]

Fig. 2: Illustration of t-SVD of \( \mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3} \)

C. Preliminaries on tensor singular value decomposition

Definition 5. (t-SVD) \[41]\]
The t-SVD of a 3-order tensor \( \mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3} \) can be represented as follows:
\[ \mathcal{A} = \mathcal{U} \ast \mathcal{S} \ast \mathcal{V}^T \tag{2} \]
Where \( \mathcal{U} \in \mathbb{R}^{I_1 \times I_1 \times I_3} \) and \( \mathcal{V} \in \mathbb{R}^{I_2 \times I_2 \times I_3} \) are the orthogonal tensors, and \( \mathcal{S} \in \mathbb{R}^{I_1 \times I_2 \times I_3} \) is an f-diagonal tensor. Fig. 2 illustrates the t-SVD of tensor \( \mathcal{A} \).

Definition 6. (tensor tubal-rank) \[41]\] Let \( \mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3} \) be a 3-order tensor. The tubal-rank of tensor \( \mathcal{A} \) denoted as \( \text{rank}_{\text{tubal}}(\mathcal{A}) \), is defined as the number of non-zero tubes of \( \mathcal{S} \). Where \( \mathcal{S} \) is defined in equation (2).

Definition 7. (tensor nuclear norm (TNN)) \[41]\] The nuclear norm of a tensor \( \mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3} \) is defined as the sum of singular values of each frontal slice of \( \mathcal{A} \) as follows:
\[ \|\mathcal{A}\|_{\text{TNN}} := \sum_{i=1}^{I_3} \|\mathcal{A}^{(i)}\|_*, \]
where \( \mathcal{A}^{(i)} \) is the n-th frontal slice of \( \mathcal{A} \), and \( \mathcal{A} = \text{ftt}(\mathcal{A}, [], 3) \).

III. Optimization Model

In this section, we propose a smooth low rank hierarchical tensor ring approximation (SHTRA) for image completion. First, we develop a hierarchical tensor ring decomposition for more compact multi-way data representation. In this new decomposition, for the first layer, traditional tensor ring decomposition is used to factorize a tensor into several 3-order tensors. For the second layer, each 3-order tensor can be further decomposed by t-SVD. Fig. 3 shows the details of hierarchical tensor-ring decomposition.

The low rank approximation based the proposed hierarchical tensor-ring decomposition of \( \mathcal{X} \) can be formulated as follows:
\[ \min_{\mathcal{G}_{n=1, \cdots, N}} \frac{1}{2} ||\mathcal{X} - \mathcal{F}(\mathcal{G}_1, \cdots, \mathcal{G}_N)||^2_F + \sum_{n=1}^{N} \text{rank}_{\text{tubal}}(\mathcal{G}_n) \tag{3} \]

To further enhance the recovery performance in many data processing, we can add TV term to exploit the piece-wise smoothness structure. Thus, the optimization model of smooth low rank hierarchical tensor-ring approximation for image completion can be written as:
\[ \min_{\mathcal{X}, \mathcal{G}_{n=1, \cdots, N}} \frac{1}{2} ||\mathcal{X} - \mathcal{F}(\mathcal{G}_1, \cdots, \mathcal{G}_N)||^2_F + \sum_{n=1}^{N} \text{rank}_{\text{tubal}}(\mathcal{G}_n) + \lambda \text{TV}(\mathcal{X}) \]
\[ \text{s. t. } \mathcal{P}(\mathcal{X}) = \mathcal{P}^{(n)}(\mathcal{T}) \tag{4} \]
where $X$ is the recovered low-rank tensor, $T$ is the tensor for measurements, and $\lambda$ is the trade-off parameter between hierarchical tensor ring term and TV term. $\mathcal{P}_O(X)$ represents the observed entries.

The tubal rank is non-convex, and one of its convex surrogate is tensor nuclear norm. $\text{TV}(X)$ is a total variation of tensor $X$ which can be denoted as $\|\mathcal{D}(X)\|_1$, i.e. the tensor total variation is the $\ell_1$ norm of all the differences along all the modes, where $\mathcal{D}(X)$ takes the differences. In this way, the convex optimization model for (4) can be formulated as

$$
\min_{X, G_{n=1,\ldots,N}} \frac{1}{2} \|X - \mathcal{F}(G_1, \ldots, G_N)\|^2_F + \lambda \|\mathcal{D}(X)\|_1 + \sum_{n=1}^{N} \|G_n\|_{\text{TNN}},
$$

s. t. $\mathcal{P}_O(X) = \mathcal{P}_O(T), \tag{5}$

where $\|\|_{\text{TNN}}$ denotes the tensor nuclear norm. To solve the optimization model (5), we introduce additional tensor variables $Z$ and $M$ with the same size as $X$, and $Y$ for tensor difference. The equivalence of (5) can be obtained as follows:

$$
\min_{X, G_{n=1,\ldots,N}} \frac{1}{2} \|X - \mathcal{F}(G_1, \ldots, G_N)\|^2_F + \lambda \|Y\|_1 + \sum_{n=1}^{N} \|G_n\|_{\text{TNN}},
$$

Fig. 3: Graphical representation of a hierarchical tensor-ring (TR) model.
\[ \mathcal{P}_0 (X) = \mathcal{P}_0 (T) , \quad \mathcal{Z} = X, Y = D (Z), \quad \sum_{n=1}^{N} M_n = G_n \]

Under the ADMM framework, this problem further can be converted into following form:

\[
\begin{align*}
\min_{X, G, Z, M, Y} & \quad \frac{1}{2} \| X - \mathcal{F} (G_1, \ldots, G_N) \|_F^2 + \lambda \| Y \|_1 + \sum_{n=1}^{N} \| M_n \|_{\text{TN}} \\
+ & \quad \langle \Lambda_1, Z - X \rangle + \frac{\beta_1}{2} \| Z - X \|_F^2 + \langle \Lambda_2, Y - D (Z) \rangle + \frac{\beta_2}{2} \| Y - D (Z) \|_F^2 \\
+ & \quad \frac{\beta_2}{2} \| Y - D (Z) \|_F^2 + \sum_{n=1}^{N} \left( \langle \Lambda_3^{(n)}, M_n - G_n \rangle + \frac{\beta_3}{2} \| M_n - G_n \|_F^2 \right) \\
\text{s.t.} & \quad \mathcal{P}_0 (X) = \mathcal{P}_0 (T)
\end{align*}
\]

where \( \beta_1, \beta_2, \beta_3 \) are positive penalty scalars, and \( \Lambda_1, \Lambda_2, \Lambda_3 \) are the dual variables. This problem can be solved by updating each variable with others fixed.

The first sub-problem optimizes the variable \( G \) with other variables fixed, which can be written as:

\[
\begin{align*}
\min_{G} & \quad \frac{1}{2} \| X - \mathcal{F} (G_1, \ldots, G_N) \|_F^2 \\
+ & \quad \sum_{n=1}^{N} \left( \langle \Lambda_3^{(n)}, M_n - G_n \rangle + \frac{\beta_3}{2} \| M_n - G_n \|_F^2 \right)
\end{align*}
\]

The second sub-problem optimizes the variable \( M \) and keeps other variables fixed, can be written as:

\[
\begin{align*}
\min_{M} & \quad \sum_{n=1}^{N} \| M_n \|_{\text{TN}} + \sum_{n=1}^{N} \langle \Lambda_3^{(n)}, M_n - G_n \rangle \\
+ & \quad \sum_{n=1}^{N} \frac{\beta_3}{2} \| M_n - G_n \|_F^2
\end{align*}
\]

The third sub-problem optimizes the variable \( Z \) while keeping the others fixed, can be written as:

\[
\begin{align*}
\min_{Z} & \quad \langle \Lambda_1, Z - X \rangle + \frac{\beta_1}{2} \| Z - X \|_F^2 + \langle \Lambda_2, Y - D (Z) \rangle + \frac{\beta_2}{2} \| Y - D (Z) \|_F^2
\end{align*}
\]

The fourth sub-problem on \( Y \) can be written as:

\[
\begin{align*}
\min_{Y} & \quad \lambda \| Y \|_1 + \langle \Lambda_2, Y - D (Z) \rangle + \frac{\beta_2}{2} \| Y - D (Z) \|_F^2
\end{align*}
\]

The fifth sub-problem on \( X \) can be written as:

\[
\begin{align*}
\min_{X} & \quad \frac{1}{2} \| X - \mathcal{F} (G_1, \ldots, G_N) \|_F^2 + \langle \Lambda_1, Z - X \rangle \\
+ & \quad \frac{\beta_1}{2} \| Z - X \|_F^2 \quad \text{s.t.} \quad \mathcal{P}_0 (X) = \mathcal{P}_0 (T)
\end{align*}
\]

The sub-problem on updating dual variables can be written as:

\[
\begin{align*}
\min_{\Lambda_1, \Lambda_2, \Lambda_3^{(n)}, \ldots, N} & \quad \langle \Lambda_1, Z - X \rangle + \frac{\beta_1}{2} \| Z - X \|_F^2 + \langle \Lambda_2, Y - D (Z) \rangle \\
+ & \quad \frac{\beta_2}{2} \| Y - D (Z) \|_F^2 + \sum_{n=1}^{N} \langle \Lambda_3^{(n)}, M_n - G_n \rangle \\
+ & \quad \sum_{n=1}^{N} \frac{\beta_3}{2} \| M_n - G_n \|_F^2
\end{align*}
\]
IV. Solution

A. The solution of subproblem (6)

The sub-problem (6) can be reformulated as:

$$\min_{\theta_n} \frac{1}{2} \| X_{<n>} - \langle \mathcal{G}_n \rangle_{<2>} \left( \mathcal{G}_{<2>}^{(\hat{\mathcal{G}})} \right)^T \|_F^2 + \sum_{n=1}^{N} \left( \left( \Lambda_{3}^{(n)} \right)^{\top} M_n - \mathcal{G}_n \right) + \frac{\beta_3}{2} \| M_n - \mathcal{G}_n \|_F^2 \right)$$

(12)

where \( \mathcal{G}_{<2>}^{(\hat{\mathcal{G}})} \in \mathbb{R}^{1 \times I_1 \times R_{i-1} R_{i}} \) is a sub-chain dimension-mode unfolded matrix generated by merging all the core unfolded matrices except the \( k \)-th core. \( \langle \mathcal{G}_n \rangle_{<2>} \in \mathbb{R}^{I_1 \times R_{i-1} \times R_i} \) is a \( k \)-th core dimension-mode unfolded matrix.

The solution for (12) is as follows:

$$\langle \mathcal{G}_n \rangle_{<2>} = \left( X_{<n>} \mathcal{G}_{<2>}^{(\hat{\mathcal{G}})} + \Lambda_{3}^{(n)} + \beta_3 (M_n)_{<2>} \right) \left( \left( \mathcal{G}_{<2>}^{(\hat{\mathcal{G}})} \right)^{\top} (\mathcal{G}_n)_{<2>} + \beta_3 I \right)^{-1}$$

(13)

where \( I \) is an identity matrix.

B. The solution of sub-problem (7)

The optimization problem (7) can be equivalent to:

$$\sum_{n=1}^{N} \left( \frac{1}{\beta_3} \| M_n \|_{TN} + \frac{1}{2} \| M_n - \left( \mathcal{G}_n - \frac{\Lambda_{3}^{(n)}}{\beta_3} \right) \|_F^2 \right)$$

(14)

Let \( \tau = \frac{1}{\beta_3} \), \( \mathcal{L}_n = \mathcal{G}_n - \frac{\Lambda_{3}^{(n)}}{\beta_3} \) and \( M_n \in \mathbb{R}^I \times I_2 \times I_3 \) be a 3rd-order core tensor. Therefore, (14) can be rewritten as:

$$\sum_{n=1}^{N} \left( \tau \| M_n \|_{TN} + \frac{1}{2} \| M_n - \mathcal{L}_n \|_F^2 \right)$$

(15)

(15) can be computed by using tensor singular value thresholding (t-SVT) as follows [42]:

$$M_n = \text{SVT}_\tau(\mathcal{L}_n)$$

(16)

where \( n = 1, \cdots, N \) and for each \( \tau > 0 \), tensor singular value thresholding (t-SVT) operator can be defined as:

$$\text{SVT}_\tau(\mathcal{L}) = \mathcal{U} * \mathcal{S}_\tau * \mathcal{V}^\top$$

(17)

where

$$\mathcal{S}_\tau = \text{ifft} \left( \left( \tilde{\mathcal{S}} - \tau \right)_+, \left[ 1, 3 \right] \right)$$

(18)

where \( \tilde{\mathcal{S}} \) is a real tensor. \( t_+ \) indicates the positive part i.e. \( t_+ = \max(t, 0) \). This operator applies the soft-thresholding rule to singular values \( \tilde{\mathcal{S}} \) of each frontal slice of \( \mathcal{L} \). The detailed solutions are concluded in Algorithm 1.

Fig. 4: Testing color images with the size $256 \times 256 \times 3$. 
**Algorithm 1 Tensor Singular Value Thresholding (t-SVT)**

**Input:** \( L_n \in \mathbb{R}^{R_{i-1} \times R_o \times I_o}, \tau > 0 \)

**Output:** SVT\(_\tau\)(\( L \)) as defined in [17]

Compute \( \hat{L} = \text{fft}(L, [], 3) \).

Perform Matrix SVT on each frontal slice of \( \hat{L} \) by

\[
\text{for } k = 1, \cdots, \frac{L+1}{2} \text{ do } \\
\quad [U, S, V] = \text{SVD}(\hat{I}^{(k)}) \\
\quad \hat{W} = U (S - \tau)_+ V^*; \\
\text{end for}
\]

\[
\text{for } k = \left\lfloor \frac{L+1}{2} \right\rfloor + 1, \cdots, I_n \text{ do } \\
\quad \hat{W}^{(k)} = \text{conj}\left(\hat{W}^{(L-k+2)}\right) \\
\text{end for}
\]

Compute SVT\(_\tau\)(\( L \)) = ifft(\( \hat{W} \), [ ], 3)

---

**Fig. 5:** Testing color images with the size \( 321 \times 481 \times 3 \).

---

**C. The solution of sub-problem (8)**

This sub-problem can be solved by differentiable function with respect to \( Z \). The minimization condition is equivalent to the following linear equation [13]:

\[
Z = \frac{\mathcal{D}^* (\Lambda_2 - \beta_2 Y) + \beta_1 X - \Lambda_1}{\beta_1 I + \beta_2 \mathcal{D}^* \mathcal{D}}
\]

(19)

where \( \mathcal{D}^* \) is adjoint of \( \mathcal{D} \). As the structure of operator \( \mathcal{D}^* \mathcal{D} \) is block circulant, it can be transformed into the Fourier domain and fast calculated. By using the off-the-shelf conjugate gradient technique, and the fast computation of \( Z \) can be written as:

\[
Z = \text{ifftn}\left(\frac{\text{fftn}(\mathcal{J})}{\beta_1 I + \beta_2 (\text{fftn}(\mathcal{D}^* \mathcal{D}))}\right),
\]

(20)

where \( \mathcal{J} = \mathcal{D}^* (\Lambda_2 - \beta_2 Y) + \beta_1 X - \Lambda_1 \), fftn and ifftn are 3D Fast Fourier transform and 3D inverse Fast Fourier transform, respectively. It is noticed that operator \( \mathcal{D}^* \mathcal{D} \) can be computed outside the loop to decrease the computational cost.

---

**Fig. 6:** Hyperspectral images (10th band grayscale) (a) Hyperspectral image 'Indianpines' with the size \( 145 \times 145 \times 224 \). (b) Hyperspectral image 'Salinas' with size \( 512 \times 217 \times 224 \).
Fig. 7: Comparison of different methods on color images with the size 256×256×3 when different sampling ratios change from 10% to 60% in terms of PSNR, SSIM and RSE.
Algorithm 2 SHTRA

**Input:** Zero-filled observed tensor $\mathcal{T}$, observed index $\Omega$

**Initialization:** $X_\Omega = \mathcal{T}_\Omega$. For $n = 1, \cdots, N$ random sample $G_n$, iter = 0, maxiter, residual bound $\epsilon$, $\kappa = 1.01$.

**while** iter < maxiter **do**

iter = iter + 1.

$X_{\text{last}} = X$

For $n = 1, \cdots, N$, Update $G_n$, by [13]

For $n = 1, \cdots, N$ Update $M_n$ by [16]

Update $Z$ by [20]

Update $Y$ by [22]

Update $X$ by [24]

Update Dual Variables $\Lambda_1, \Lambda_2, \Lambda_3$ by [25]

$\beta = \min(\kappa \times \beta, 10)$

If $\|X-X_{\text{last}}\|_F/\|X_{\text{last}}\|_F < \epsilon$, **Break**

**end while**

**Output:** recovered tensor $X$

---

D. The solution of sub-problem [9]

Considering the an-isotropic TV, the sub-problem [9] can be the written as:

$$\min_Y \lambda \|Y\|_1 + \frac{\beta_2}{2} \|Y - \left(D(Z) - \frac{\Lambda_2}{\beta_2}\right)\|_F^2$$

(21)

The sub-problem (21) can solved by soft-thresholding operator as follows:

$$Y = \text{sth}\left(D(Z) - \frac{\Lambda_2}{\beta_2}, \frac{\lambda}{\beta_2}\right)$$

(22)
Fig. 9: Comparison of different methods on color images with the size $321 \times 481 \times 3$ when different sampling ratios change from 10% to 60% in terms of PSNR, SSIM and RSE.
Fig. 10: Examples on image completion using different methods with different sampling ratios.

Fig. 11: Examples on Indianpines completion using different methods with different sampling ratios.

where \( \text{sth} \) is the soft-thresholding operator defined as follows:

\[
\text{sth}(a, \tau) = \text{sgn}(a) \max(|a| - \tau, 0)
\]

(23)

where \( a = D(Z) - \frac{\Delta_i}{\rho_i} \) and \( \tau = \frac{\lambda}{\rho_i} \).
E. The solution of sub-problem (10)

The sub-problem (10) is a convex optimization with equality constraint. We can update $X$ as follows:

$$X_{i_1, \cdots, i_d} = \begin{cases} 
    \frac{R + \Lambda_1 - \beta_1 Z}{1 - \beta_1}, & i_1, \ldots, i_d \notin \emptyset \\
    T_{i_1, \cdots, i_d}, & i_1, \ldots, i_d \in \emptyset
\end{cases}$$

(24)

where $R_g = \mathcal{F}(G_1, \cdots, G_N)$.

F. The solution of sub-problem (11)

According to ADMM, the dual variables can be updated by:

$$\Lambda_1 = \Lambda_1 + \beta_1 (Z - X)$$
$$\Lambda_2 = \Lambda_2 + \beta_2 (Y - D(Z))$$
$$\Lambda_3^{(r)} = \Lambda_3^{(r)} + \beta_3 (M_0 - G_n), n = 1, \cdots N.$$  

(25)
The penalty vector $\beta = [\beta_1, \beta_2, \beta_3]^T$ can be updated as follows [44], [45]:

$$\beta^{(k)} = \begin{cases} \eta \beta^{(k-1)}, & \text{if } \zeta^{(k)} > \eta_2 \beta^{(k-1)} \\ \beta^{(k-1)}, & \text{otherwise} \end{cases}$$

where $\zeta^{(k)} = \|Z - X\|, \|Y - D(Z)\|, \sum_{n=1}^{N} (\|M_n - G_n\|) \|^T$ in $k$-th iteration, $\eta_1, \eta_2$ are scalar factors.

The pseudo-codes of the SHTRA are given in Algorithm 2. The convergence condition is set to be $\|X_{\text{last}} - X_{\text{till last}}\| < \epsilon$, where $X$ is the recovered tensor and $X_{\text{last}}$ is the recovered tensor of last iteration and $\epsilon$ is the residual bound.

### G. Computational Complexity

The main computational complexity of the proposed algorithm comes from the update of $G_n$. For an $N$-order tensor $X^{I_1 \times I_2 \times \cdots \times I_N}$, the computational complexity of $G_n$ is mainly from the inversion of matrix $\left( (G_{<2>}^n)^T (G_{<2>}^n) + \beta_3 I \right)$ or the matrix multiplication of $\left( (G_{<2>}^n)^T \right)$ and $(G^n_{<2>})$.

By Assuming $I_n = I$ and $R_n = R$ for $n = 1, \cdots, N$, the computational complexity of $G_n$ is $O(\min(R^6, I^N R^4))$. In the proposed algorithm, $R \ll I$. Therefore, the overall complexity is $O(NTI^N R^4)$, where $T$ is the number of iterations in the proposed algorithm.

### V. Numerical Experiments

In this section, we have conducted several groups of experiments and compared our method with the state-of-art ones including TR-ALS [34], TRLRF [40], STTC [43], TMAC-TT [28], SPC [46], and LRTV-PDS [47].

To quantitatively measure the missing degree, we define the sampling ratio (SR) as follows:

$$SR = \frac{O}{\prod_{n=1}^{N} I_n}$$

where $O$ is the number of observed entries.

The peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and relative square error (RSE) are used to measure the recovery accuracy. The PSNR is the peak signal-to-noise ratio between two images which is defined as: $10 \log_{10} \left( \frac{\max \left( A \right)}{\text{MSE}} \right)$, where max is the maximum fluctuation of input image data and MSE is the cumulative squared error between reconstructed and original image. MSE can be defined as $\frac{\| \hat{A} - A \|^2}{N}$. The higher the PSNR, the better the quality of the recovered image. The SSIM can measure the intensity of light (i.e. luminance) and the contrast of two images. It shows how closely these features vary together between two images. The higher value for SSIM indicates better results. The RSE determines the performance by computing relative error between original tensor $A$ and recovered tensor $\hat{A}$, which can be defined as $\frac{\| \hat{A} - A \|^2}{\|A\|^2}$. The lower the value of RSE, the better the performance. Besides, for the HSI, we chose MPSNR, MSSIM and spectral angle mapper (SAM) [48] for evaluation. Spectral angle mapper (SAM) measures the spectral similarity by calculating the angle between the spectra $\hat{a}_i$ and $a_i$ over the whole spatial domain, which can be defined as $\text{SAM} = \frac{1}{I} \sum_{j=1}^{I} \arccos \left( \frac{\hat{a}_i \cdot a_j}{\| \hat{a}_i \| \| a_j \|} \right)$, where $A \in \mathbb{R}^{I_1 \times I_2 \times K}$ is the HSI and $\hat{A} \in \mathbb{R}^K$ is the pixel of $A$ and $a \in \mathbb{R}^K$ is the pixel of $A$.

All the experiments have been conducted by using MATLAB R2018b on the desktop computer with the specification Intel(R) Core(TM) i5-4590 CPU, 3301 MHz, 4 Core(s) and 8GB RAM. Two types of datasets are used for images recovery, i.e. 1) color image dataset, 2) HSI dataset. In color image dataset, we use images with two different sizes. One group consists of standard color images with the size 256 $\times$ 256 $\times$ 3 and can be seen in Fig. 4, the other images are of the size 321 $\times$ 481 $\times$ 3 as can be seen in Fig. 5. For HSI dataset, we use two different types. One is the Indian pines with the size 145 $\times$ 145 $\times$ 224, and other is Salinas with the size 512 $\times$ 217 $\times$ 224, as can be seen in Fig. 6.

https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/
http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes
A. Color image with the size $256 \times 256 \times 3$

In this group of experiments, we apply the proposed SHTRA method along with STTC, TRLRF, TR-ALS, SPC, LRTV-PDS and TMac-TT on four color images with the size of $256 \times 256 \times 3$. We set the trade-off parameters between smoothness and low-rank terms as 0.0005 for STTC. We set the weight of direction total variation as $\beta$ for LRTV-PDS and TMac-TT on four color images with the size of $256 \times 256$.

A. Color image with the size $321 \times 481 \times 3$

In this group of experiments, we choose some color images with the size of $321 \times 481 \times 3$ for comparison of the proposed method with TR-ALS, TRLRF, TMac-TT, SPC, LRTV-PDS, and STTC. We set the parameters the same as those in last subsection. We also set the condition for convergence as $\epsilon = 5 \times 10^{-4}$. For fairness, we set the TR rank to be 15 for all TR-factors for the proposed method.

1) Experimental Results: Fig. 7 shows the recovery performance of 4 color images in terms of PSNR, RSE, and SSIM with SR from 10% to 60%. From 7, it can be observed that the Debye-Waller factors include TRLRF, TR-ALS, and TMac-TT. On the other side, it can be seen that the method with smoothness term such as SPC, STTC, and SHTRA show better performance compared with non-smooth methods. Therefore, it may prove that smoothness constraints can effectively enhance the recovery performance.

In Fig. 7, we can also see that the proposed algorithm exhibits superior recovery performance in most cases compared to state-of-art-algorithms. Compared with other tensor-ring based methods such as TR-ALS and TRLRF, SHTRA has achieved far better performance in every index. Fig. 8 shows the recovered images with different sample ratios ranging from 10% to 40%. When the sampling ratio is very low, TRLRF has recovered the image with the features hard to identify. However, the proposed SHTRA shows better effectiveness for the reconstruction of missing entries in most cases compared with the others.

B. Color image with the size $321 \times 481 \times 3$

In this group of experiments, we use two hyperspectral images Indianpines and Salinas with the size $145 \times 145 \times 224$ and $512 \times 217 \times 224$ for comparison of the proposed algorithm with STTC, SPC, LRTV-PDS, TR-ALS, and TRLRF. We tune the parameters to get the optimal performance. We set the TR rank to be 10 for TR-ALS, TRLRF, and the proposed SHTRA. We set the weights for total variation as $\mathbf{w} = [4; 4; 10]$, $\mathbf{w} = [2; 2; 10]$ for STTC and SHTRA, respectively. We also set the trade-off parameter between low-rank term and smoothness as 0.003 and 0.0005 for STTC and SHTRA, respectively. The trade-off parameter in SHTRA is set as $\rho = 12$, and we also tune the penalty factors for the optimal performance with $\mathbf{B} = [0.001; 0.001; 0.8]$. For convergence, we set the condition $\epsilon = 10^{-4}$, and the maximum number of iterations is set to 300.

1) Indianpines with the size $145 \times 145 \times 224$: Indianpines with the size $145 \times 145$ and 224 refers to spectral reflectance bands. Due to computational cost, we only chose the first 30 bands, resulting in the size $145 \times 145 \times 30$. Table II shows the quantitative evaluation of recovery with sample ratios ranging from 10% to 50%. It shows that the proposed algorithm recovers the missing
image Indianpines with superior performance compared to state-of-the-art ones in terms of MSSIM, MPSNR and SAM.

Fig. 11 shows the reconstruction with the sample ratios ranging from 10% to 40% with different spectral band in grayscale color-map. We can see that the proposed algorithm successfully recovers the missing entries for each SR, and the performance is better than those of the others. In addition, it can be seen that when SR = 10%, TRLRF, STTC, and LRTV-PDS fail to reconstruct the image. However, with SR increasing, the performance of TRLRF and SPC improves along with the proposed algorithm.

2) Salinas with the size $512 \times 217 \times 224$: We resize Salinas by sampling its spatial resolution to $256 \times 109$, and keep the first 30 bands due to computational cost. The original image is resized to $256 \times 109 \times 30$.

Table I shows that the proposed algorithm has achieved better performance in most cases. When SR = 20% and SR = 40%, TRLRF has slightly good performance compared to other algorithms. Fig. 12 shows the recovered Salinas for each algorithm with different SRs. As can be seen, LRTV-PDS and STTC have the worst performance when SR = 10%. In contrast, the recovery performance of SPC and SHTRA is good. Besides, SHTRA is superior to the other methods in terms of recovery resolution when SR = 50%.

### Table I: Quantitative comparison of different algorithms for Salinas and Indianpines completion.

| Datasets | SR | measure indexes | STTC | SPC | LRTV-PDS | TR-ALS | TRLRF | SHTRA |
|----------|----|-----------------|------|-----|----------|--------|-------|-------|
| Indianpines | 10% | M-PSNR (dB) | 39.779 | 49.595 | 24.528 | 47.182 | 42.680 | **50.319** |
| | | M-SSIM | 0.143 | 0.791 | 0.004 | 0.626 | 0.429 | **0.804** |
| | | SAM | 0.142 | 0.045 | 1.205 | 0.085 | 0.124 | **0.045** |
| | | CPU time (sec) | 52.68 | 58.37 | 1352 | 92.89 | 190.6 | 444.3 |
| | 20% | M-PSNR (dB) | 39.942 | 50.443 | 25.040 | 51.806 | 52.438 | **52.558** |
| | | M-SSIM | 0.231 | 0.831 | 0.010 | 0.838 | 0.052 | **0.880** |
| | | SAM | 0.141 | 0.042 | 1.113 | 0.046 | 0.046 | **0.038** |
| | | CPU time (sec) | 71.09 | 111.8 | 93.32 | 27.18 | 192.2 | 440.1 |
| | 30% | M-PSNR (dB) | 40.332 | 51.144 | 25.620 | 52.660 | 54.046 | **54.351** |
| | | M-SSIM | 0.305 | 0.857 | 0.017 | 0.872 | 0.094 | **0.916** |
| | | SAM | 0.137 | 0.040 | 0.995 | 0.041 | 0.039 | **0.034** |
| | | CPU time (sec) | 52.92 | 97.44 | 60.01 | 20.01 | 177.2 | 310.4 |
| | 40% | M-PSNR (dB) | 42.095 | 54.399 | 31.968 | 52.037 | 48.313 | **53.615** |
| | | M-SSIM | 0.130 | 0.845 | 0.008 | 0.663 | 0.489 | **0.790** |
| | | SAM | 0.265 | 0.032 | 1.209 | 0.101 | 0.180 | 0.040 |
| | | CPU time (sec) | 181.2 | 276.6 | 325.8 | 85.27 | 255.2 | 561.1 |
| Salinas | 10% | M-PSNR (dB) | 42.095 | 54.399 | 31.968 | 52.037 | 48.313 | **53.615** |
| | | M-SSIM | 0.130 | 0.845 | 0.008 | 0.663 | 0.489 | **0.790** |
| | | SAM | 0.265 | 0.032 | 1.209 | 0.101 | 0.180 | 0.040 |
| | | CPU time (sec) | 181.2 | 276.6 | 325.8 | 85.27 | 255.2 | 561.1 |
| | 20% | M-PSNR (dB) | 40.332 | 51.144 | 25.620 | 52.660 | 54.046 | **54.351** |
| | | M-SSIM | 0.305 | 0.857 | 0.017 | 0.872 | 0.094 | **0.916** |
| | | SAM | 0.137 | 0.040 | 0.995 | 0.041 | 0.039 | **0.034** |
| | | CPU time (sec) | 52.92 | 97.44 | 60.01 | 20.01 | 177.2 | 310.4 |
| | 30% | M-PSNR (dB) | 51.333 | 55.942 | 32.480 | 60.467 | 61.962 | **55.790** |
| | | M-SSIM | 0.639 | 0.889 | 0.022 | 0.910 | 0.933 | **0.904** |
| | | SAM | 0.143 | 0.029 | 1.113 | 0.025 | 0.026 | **0.026** |
| | | CPU time (sec) | 181.2 | 276.6 | 325.8 | 85.27 | 255.2 | 561.1 |
| | 40% | M-PSNR (dB) | 53.448 | 56.946 | 33.057 | 61.523 | 63.359 | **53.615** |
| | | M-SSIM | 0.710 | 0.910 | 0.039 | 0.935 | 0.952 | **0.954** |
| | | SAM | 0.122 | 0.027 | 0.995 | 0.022 | 0.019 | **0.018** |
| | | CPU time (sec) | 66.95 | 94.82 | 31.13 | 14.41 | 49.26 | 78.02 |
| | 50% | M-PSNR (dB) | 55.268 | 57.911 | 33.727 | 61.547 | 64.686 | **64.169** |
| | | M-SSIM | 0.761 | 0.926 | 0.058 | 0.932 | 0.952 | **0.954** |
| | | SAM | 0.104 | 0.026 | 0.887 | 0.021 | 0.017 | **0.017** |
| | | CPU time (sec) | 66.95 | 94.82 | 31.13 | 14.41 | 49.26 | 78.02 |

---

*Note: The table above shows the quantitative comparison of different algorithms for Salinas and Indianpines completion. The entries are measured in decibels (dB) for PSNR, with higher values indicating better performance, and in terms of SSIM, with values closer to 1 indicating better performance.*
VI. Conclusion

In this paper, we develop a low rank hierarchical tensor ring approximation for image completion. The newly proposed hierarchical tensor ring can result in more compact representation for multidimensional images in applications, in comparison with the other decompositions. By low rank approximation in both layers of the hierarchical tensor ring, automatically tuned TR ranks can alleviate overfitting when the rank is set to be large and there are a limited number of observations. To enhance the recovery performance, total variation is taken in the optimization model for exploiting local piece-wise smoothness additionally. It can be solved by ADMM. Experimental results on color image and HSI show that the proposed algorithm outperforms the state-of-the-art ones.

References

[1] A. Momeni, H. Rajabalipanah, A. Abdolali, and K. Achouri, “Generalized optical signal processing based on multioperator metasurfaces synthesized by susceptibility tensors,” Physical Review Applied, vol. 11, no. 6, p. 064042, 2019.
[2] B. Madathil, S. V. M. Sagheer, V. Rahiman, A. J. Tom, J. Francis, S. N. George, et al., “Tensor low rank modeling and its applications in signal processing,” arXiv preprint arXiv:1912.03435, 2019.
[3] C. Lu, X. Peng, and Y. Wei, “Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5996–6004, 2019.
[4] E. Mutlu, K. Kowalski, and S. Krishnamoorthy, “Toward generalized tensor algebra for ab initio quantum chemistry methods,” in Proceedings of the 6th ACM SIGPLAN International Workshop on Libraries, Languages and Compilers for Array Programming, pp. 46–56, 2019.
[5] J. H. Li, W. J. Huang, T. Xu, S. R. Kirk, and S. Jenkins, “Stress tensor eigenvector following with next-generation quantum theory of atoms in molecules,” International Journal of Quantum Chemistry, vol. 119, no. 7, p. e25847, 2019.
[6] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, “Tensors for data mining and data fusion: Models, applications, and scalable algorithms,” ACM Transactions on Intelligent Systems and Technology (TIST), vol. 8, no. 2, pp. 1–44, 2016.
[7] L. Kong, X.-Y. Liu, H. Sheng, P. Zeng, and G. Chen, “Federated tensor mining for secure industrial internet of things,” IEEE Transactions on Industrial Informatics, 2019.
[8] Y. Liu, Z. Long, and C. Zhu, “Image completion using low tensor tree rank and total variation minimization,” IEEE Transactions on Multimedia, vol. 21, no. 2, pp. 338–350, 2019.
[9] S. Leurgans, R. Ross, and R. Abel, “A decomposition for three-way arrays,” SIAM Journal on Matrix Analysis and Applications, vol. 14, no. 4, pp. 1064–1083, 1993.
[10] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, “Scalable tensor factorizations for incomplete data,” Chemometrics and Intelligent Laboratory Systems, vol. 106, no. 1, pp. 41–56, 2011.
[11] C. A. Andersson and R. Bro, “Improving the speed of multi-way algorithms:Part I Tucker3,” Chemometrics and Intelligent Laboratory Systems, vol. 42, no. 1-2, pp. 93–103, 1998.
[12] J. Liu, P. Musialski, P. Wonka, and J. Ye, “Tensor completion for estimating missing values in visual data,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 1, pp. 208–220, 2013.
[13] S. Gandy, B. Recht, and I. Yamada, “Tensor completion and low-n-rank tensor recovery via convex optimization,” Inverse Problems, vol. 27, no. 2, p. 025010, 2011.
[14] X.-Y. Liu, S. Aeron, V. Aggarwal, and X. Wang, “Low-tubal-rank tensor completion using alternating minimization,” IEEE Transactions on Information Theory, 2019.
[15] Z. Long, Y. Liu, L. Chen, and C. Zhu, “Low rank tensor completion for multiway visual data,” Signal Processing, vol. 155, pp. 301–316, 2019.
[16] H. Huang, Y. Liu, J. Liu, and C. Zhu, “Provable tensor ring completion,” Signal Processing, vol. 171, p. 107486, 2020.
[17] H. Tan, B. Cheng, W. Wang, Y.-J. Zhang, and B. Ran, “Tensor completion via a multi-linear low-n-rank factorization model,” Neurocomputing, vol. 133, pp. 161–169, 2014.
[18] B. Romera-Paredes and M. Pontil, “A new convex relaxation for tensor completion,” in Advances in Neural Information Processing Systems, pp. 2967–2975, 2013.
[19] D. Kressner, M. Steinlechner, and B. Vandereycken, “Low-rank tensor completion by Riemannian optimization,” BIT Numerical Mathematics, vol. 54, no. 2, pp. 447–468, 2014.
[20] Y. Xu, R. Hao, W. Yin, and Z. Su, “Parallel matrix factorization for low-rank tensor completion,” Inverse Problems and Imaging, vol. 9, no. 2, pp. 601–624, 2015.
[21] L. Yang, J. Fang, H. Li, and B. Zeng, “An iterative reweighted method for tucker decomposition of incomplete tensors,” IEEE Transactions on Signal Processing, vol. 64, no. 18, pp. 4817–4829, 2016.
[22] Y. Yang, Y. Feng, X. Huang, and J. A. Suykens, “Rank-1 tensor properties with applications to a class of tensor optimization problems,” SIAM Journal on Optimization, vol. 26, no. 1, pp. 171–196, 2016.
[23] M. Signoretto, R. Van de Plas, B. De Moor, and J. A. Suykens, “Tensor versus matrix completion: A comparison with application to spectral data,” IEEE Signal Processing Letters, vol. 18, no. 7, pp. 403–406, 2011.
[24] C. Mu, B. Huang, J. Wright, and D. Goldfarb, “Square deal: Lower bounds and improved relaxations for tensor recovery,” in International conference on machine learning, pp. 73–81, 2014.
[25] Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, and S.-I. Amari, “Bayesian robust tensor factorization for incomplete multiway data,” IEEE transactions on neural networks and learning systems, vol. 27, no. 4, pp. 736–748, 2015.
[26] Y. Yang, Y. Feng, and J. A. Suykens, “A rank-one tensor updating algorithm for tensor completion,” IEEE Signal Processing Letters, vol. 22, no. 10, pp. 1633–1637, 2015.
[27] I. V. Oseledets, “Tensor-train decomposition,” SIAM Journal on Scientific Computing, vol. 33, no. 5, pp. 2295–2317, 2011.

[28] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do, “Efficient tensor completion for color image and video recovery: Low-rank tensor train,” IEEE Transactions on Image Processing, vol. 26, pp. 2466–2479, May 2017.

[29] W. Hackbusch and S. Kühn, “A new scheme for the tensor representation,” Journal of Fourier Analysis and Applications, vol. 15, no. 5, pp. 706–722, 2009.

[30] J. Ballani, L. Grasedyck, and M. Kluge, “Black box approximation of tensors in hierarchical tucker format,” Linear algebra and its applications, vol. 438, no. 2, pp. 639–657, 2013.

[31] C. Da Silva and F. J. Herrmann, “Efficient tensor completion for color image and video recovery: Low-rank tensor train,” IEEE Transactions on Image Processing, vol. 26, pp. 2466–2479, May 2017.

[32] W. Hackbusch and S. Kühn, “A new scheme for the tensor representation,” Journal of Fourier Analysis and Applications, vol. 15, no. 5, pp. 706–722, 2009.

[33] J. Ballani, L. Grasedyck, and M. Kluge, “Black box approximation of tensors in hierarchical tucker format,” Linear algebra and its applications, vol. 438, no. 2, pp. 639–657, 2013.

[34] C. Da Silva and F. J. Herrmann, “Efficient tensor completion for color image and video recovery: Low-rank tensor train,” IEEE Transactions on Image Processing, vol. 26, pp. 2466–2479, May 2017.

[35] W. Hackbusch and S. Kühn, “A new scheme for the tensor representation,” Journal of Fourier Analysis and Applications, vol. 15, no. 5, pp. 706–722, 2009.

[36] J. Ballani, L. Grasedyck, and M. Kluge, “Black box approximation of tensors in hierarchical tucker format,” Linear algebra and its applications, vol. 438, no. 2, pp. 639–657, 2013.

[37] C. Da Silva and F. J. Herrmann, “Efficient tensor completion for color image and video recovery: Low-rank tensor train,” IEEE Transactions on Image Processing, vol. 26, pp. 2466–2479, May 2017.

[38] W. Hackbusch and S. Kühn, “A new scheme for the tensor representation,” Journal of Fourier Analysis and Applications, vol. 15, no. 5, pp. 706–722, 2009.

[39] J. Ballani, L. Grasedyck, and M. Kluge, “Black box approximation of tensors in hierarchical tucker format,” Linear algebra and its applications, vol. 438, no. 2, pp. 639–657, 2013.

[40] C. Da Silva and F. J. Herrmann, “Efficient tensor completion for color image and video recovery: Low-rank tensor train,” IEEE Transactions on Image Processing, vol. 26, pp. 2466–2479, May 2017.

[41] W. Hackbusch and S. Kühn, “A new scheme for the tensor representation,” Journal of Fourier Analysis and Applications, vol. 15, no. 5, pp. 706–722, 2009.

[42] J. Ballani, L. Grasedyck, and M. Kluge, “Black box approximation of tensors in hierarchical tucker format,” Linear algebra and its applications, vol. 438, no. 2, pp. 639–657, 2013.

[43] C. Da Silva and F. J. Herrmann, “Efficient tensor completion for color image and video recovery: Low-rank tensor train,” IEEE Transactions on Image Processing, vol. 26, pp. 2466–2479, May 2017.

[44] W. Hackbusch and S. Kühn, “A new scheme for the tensor representation,” Journal of Fourier Analysis and Applications, vol. 15, no. 5, pp. 706–722, 2009.

[45] J. Ballani, L. Grasedyck, and M. Kluge, “Black box approximation of tensors in hierarchical tucker format,” Linear algebra and its applications, vol. 438, no. 2, pp. 639–657, 2013.

[46] C. Da Silva and F. J. Herrmann, “Efficient tensor completion for color image and video recovery: Low-rank tensor train,” IEEE Transactions on Image Processing, vol. 26, pp. 2466–2479, May 2017.

[47] W. Hackbusch and S. Kühn, “A new scheme for the tensor representation,” Journal of Fourier Analysis and Applications, vol. 15, no. 5, pp. 706–722, 2009.

[48] J. Ballani, L. Grasedyck, and M. Kluge, “Black box approximation of tensors in hierarchical tucker format,” Linear algebra and its applications, vol. 438, no. 2, pp. 639–657, 2013.

[49] C. Da Silva and F. J. Herrmann, “Efficient tensor completion for color image and video recovery: Low-rank tensor train,” IEEE Transactions on Image Processing, vol. 26, pp. 2466–2479, May 2017.