Tensor Completion using Kronecker Rank-1 Tensor Train with Application to Visual Data Inpainting

SUN, Weize; CHEN, Yuan; SO, Hing Cheung

Published in:
IEEE Access

Published: 01/01/2018

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

Publication record in CityU Scholars:
Go to record

Published version (DOI):
10.1109/ACCESS.2018.2866194

Publication details:

Citing this paper
Please note that where the full-text provided on CityU Scholars is the Post-print version (also known as Accepted Author Manuscript, Peer-reviewed or Author Final version), it may differ from the Final Published version. When citing, ensure that you check and use the publisher's definitive version for pagination and other details.

General rights
Copyright for the publications made accessible via the CityU Scholars portal is retained by the author(s) and/or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights. Users may not further distribute the material or use it for any profit-making activity or commercial gain.

Publisher permission
Permission for previously published items are in accordance with publisher's copyright policies sourced from the SHERPA RoMEO database. Links to full text versions (either Published or Post-print) are only available if corresponding publishers allow open access.

Take down policy
Contact lbscholars@cityu.edu.hk if you believe that this document breaches copyright and provide us with details. We will remove access to the work immediately and investigate your claim.

Download date: 19/12/2022
Tensor Completion Using Kronecker Rank-1 Tensor Train With Application to Visual Data Inpainting

WEIZE SUN, (Member, IEEE), YUAN CHEN, (Member, IEEE), AND HING CHEUNG SO, (Fellow, IEEE)

1Shenzhen University, Shenzhen 518060, China
2University of Science and Technology Beijing, Beijing 100081, China
3Department of Electronic Engineering, City University of Hong Kong, Hong Kong

Corresponding author: Yuan Chen (qchenyuan00@126.com)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61501300, Grant 61701021, and Grant U1501253, and in part by the Fundamental Research Funds for the Central Universities under Grant FRF-TP-17-017A2.

ABSTRACT The problem of data reconstruction with partly sampled elements under a tensor structure, which is referred to as tensor completion, is addressed in this paper. The properties of the rank-1 tensor train decomposition and the tensor Kronecker decomposition are introduced at first, and then the tensor Kronecker rank as well as Kronecker rank-1 tensor train decomposition are defined. The general tensor completion idea is presented following the criterion of minimizing the number of Kronecker rank-1 tensors, which is relaxed to the thresholding problem and the solution is derived. Furthermore, the number of Kronecker rank-1 tensors that the proposed algorithm can retrieve and its complexity order are analyzed. Computer simulations are carried out on real visual data sets and demonstrate that our method yields a superior performance over the state-of-the-art approaches in terms of recovery accuracy and/or computational complexity.

INDEX TERMS Image reconstruction, multidimensional signal processing, tensor completion, tensor train, Kronecker rank-1 decomposition.

I. INTRODUCTION

Sparsity is a common information used in various fields such as user interest prediction [1], signal restoration [2], gene categorization [3], face modeling [4], image/video compressive sensing [5], [6] and inpainting [7], [8]. The problem of data recovery from a sparse sample under a tensor structure, which is referred to as tensor completion, is tackled in this article. Tensor completion is an important problem which has drawn increasing attention in the recent decade, and is widely used in many science and engineering problems such as data mining [9], machine learning [10], hyperspectral imaging [11], computer vision, parameter estimation [12], remote sensing [13] and visual data recovery [14], [15].

It is well known that the sparsity or low-rankness of a matrix can be rationally measured by the number of nonzero entries or nonzero singular values. Such sparsity measures have been shown to be helpful to finely encode the data sparsity in applications such as matrix completion, which minimize the rank or summation of singular values for recovering the elements of the matrix that are not observed [16], [17]. However, when data from many real world applications follow a multi-way or multi-dimensional structure, these matrix rank based methods might not be applicable. Therefore, the tensor rank optimization schemes are proposed. This tensor rank minimization problem can then be further relaxed to a low-$n$-rank tensor recovery problem, which first unfolds an $R$-dimensional ($R$-D) tensor to $R$ matrices along the $R$ dimensions and then minimizes the sum of the ranks or singular values of all the matrices for data recovery [18], [19]. However, this type of methods may fail to exploit the tensor structure with the unfolding operations and thereby leads to a suboptimal result [20].

In order to make a better use of the tensor structure, several methods based on different types of tensor decompositions are suggested. Some researchers propose to minimize the number of minimum rank-1 tensors from the CANDECOMP/PARAFAC (CP) decomposition [21], [22] for completion. Many methods based on the CP decomposition [23]– [25] are proposed, which provide a satisfactory performance even in high missing rate. However, the
CP decomposition is very costly and suffers from the ill-posedness property [26] in some cases. Other researchers, on the other hand, minimize the Tucker rank as the tensor rank from the Tucker decomposition and achieve good reconstruction results. However, these algorithms require the knowledge of tensor rank [27] [28] or a rank estimate according to some specific data model [29] therefore is only valid for some special applications. On the other hand, some researchers propose to calculate the Tucker ranks using optimization toolbox [30] and give good estimation performance, but it results in a much higher complexity comparing to the low-n-rank type of methods [31]. Furthermore, inspired by the property of Fourier transform along the third dimension of a 3-D tensor, the Fourier domain singular value decomposition (SVD) or tensor-SVD (t-SVD) is proposed [32]. This method operates the 3-D tensor as a matrix thus the resulting algebra and analysis are very close to those of matrix [33]. However, when the number of dimensions is higher than 3, such approach will simply fold all the dimensions except the first two into the third dimension thus fails to make use of this higher-order structure or the low rankness implied in such high dimensional structure.

Recently, new types of tensor decomposition methods are proposed. In order to make use of the CP structure, a rank-1 Tensor Train (r1TT) decomposition is suggested to decompose a high dimensional tensor into a group of orthogonal rank-1 sub-tensors, which can be further represented by several vectors using rank-1 CP factorization. This approach is shown to be much faster than the CP decomposition [34]. Furthermore, the tensor Kronecker structure of the images is studied [35] and a decomposition algorithm is proposed for image compression. In this work, a new tensor completion approach making use of the advantages of Kronecker representation of tensor data as well as the fast r1TT decomposition method is proposed for the task of accurate and efficient tensor data reconstruction.

The rest of the paper is organized as follows. In Section II, the notations and the required definitions are presented. In Section III, our tensor recovery approaches are developed with analysis. Experimental results are provided in Section IV, and finally, conclusions are drawn in Section V.

II. NOTATIONS, DEFINITIONS AND PRELIMINARIES
We first present the notations and preliminaries in this paper. Scalars, vectors, matrices and tensors are denoted by italic, bold lower-case, bold upper-case and bold calligraphic symbols, respectively. The transpose of a vector or matrix is written as $^T$, and the i × i identity matrix is symbolized as $I_i$. To refer to the $(m_1, m_2, \ldots, m_g)$ entry of an R-D tensor $A \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_R}$, we use $A_{(m_1, m_2, \ldots, m_g)}$. Furthermore, the Frobenius-norm of a tensor $A$ is defined as $||A||_F = \sqrt{\sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \cdots \sum_{m_g=1}^{M_g} A_{(m_1, m_2, \ldots, m_g)}^2}$. The outer product of $A \in \mathbb{C}^{M_1 \times M_2 \times \cdots \times M_P}$ and $B \in \mathbb{C}^{N_1 \times N_2 \times \cdots \times N_Q}$ is written as $\odot$ and $C = A \odot B \in \mathbb{C}^{M_1 \times M_2 \times \cdots \times M_P \times N_1 \times N_2 \times \cdots \times N_Q}$ where $C_{(m_1, m_2, \ldots, m_g, n_1, n_2, \ldots, n_Q)} = A_{(m_1, m_2, \ldots, m_g)} B_{(n_1, n_2, \ldots, n_Q)}$.

Now we go on to the definitions and preliminaries.

Definition 1: The L way folding of an R-D tensor $A \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_R}$ by a pattern $K = [K_1, K_2, \ldots, K_R] \in \mathbb{Z}^{L \times R}$, where $K_r = [K_{r,1}, K_{r,2}, \ldots, K_{r,L}]^T, M_r = \prod_{l=1}^{L} K_{r,l}$, for $r = 1, 2, \ldots, R$ and $l = 1, 2, \ldots, L$, is $B = \text{fold}_{K_r}(A) \in \mathbb{R}^{K_{r,1} \times \cdots \times K_{r,L} \times \cdots \times K_{r,L}}$ where $B_{(k_{r,1}, k_{r,2}, \ldots, k_{r,L})} = A_{(m_1, m_2, \ldots, m_g)}$ with $m_r = \sum_{l=1}^{L} K_{r,1,l} - (k_{r,1} - 1) + 1, K_{r,1},1 = 1 \prod_{l=1}^{L} K_{r,l}'$ and $k_{r,1}' = 1, 2, \ldots, K_{r,1}$. This folding procedure in fact folds the r-th dimension of $A$ into L dimensions, namely, the $(r+(r-1)L)$-th to (rL)-th dimensions of the (LR)-D tensor $B$. Furthermore, we define the opposite operation of this L way folding as $A = \text{unfold}_{L,J}(B)$. Note that when $R = 2$, it becomes the general unfolding defined in [36].

Reordering the dimensions of the (LR)-D tensor $B$ as $C = \text{fold}_{K_r}(A), B = \text{fold}_{K_r}(\mathcal{A}), \mathcal{C} = \text{fold}_{K_r}(\mathcal{A})$ and $\mathcal{Q} = \text{fold}_{K_r}(\mathcal{C})$, and we write it as $\mathcal{Q} = \text{str}_{K_r}(\mathcal{A})$ [35].

Definition 2: The rank $F$ of CP decomposition of an L-D tensor $\mathcal{Q} \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_L}$ is [24]

$$\mathcal{Q} = \sum_{f=1}^{F} \mathcal{Q}_f = \sum_{f=1}^{F} \alpha_f \mathcal{Q}_{f,1} \odot \mathcal{Q}_{f,2} \odot \cdots \odot \mathcal{Q}_{f,L}$$ (1)

where $\alpha_f$ is the amplitude of the f-th term $\mathcal{Q}_f$ and $\mathcal{Q}_{f}^T \mathcal{Q}_{f} = 1$. This decomposition factorizes the tensor $\mathcal{Q}$ into $F$ sub-tensors, and each sub-tensor can be further divided into $L$ vectors $\{\mathcal{q}_{f,l}\}_{l=1}^{L}$.

Now we go on to define the Kronecker operation between tensors. By writing the tensor Kronecker product (TKP) [35] $\otimes$ of two R-D tensors $\mathcal{A}_i \in \mathbb{R}^{K_{i,1} \times K_{i,2} \times \cdots \times K_{i,R}}$, $i = 1, 2$, as $\mathcal{A} = \mathcal{A}_2 \otimes \mathcal{A}_1 \in \mathbb{R}^{K_1 \times K_2 \times \cdots \times K_R}$ with $\mathcal{A}_{(k_1, k_2, \ldots, k_R)} = \mathcal{A}_{2(k_1, k_2, \ldots, k_R)} \mathcal{A}_{1(k_1, k_2, \ldots, k_R)}$, $k_r = (k_{r,1} - 1)K_{r,1} + k_{r,1}', l = [1, K_r]$ and $k_{r,1}' = 1, 2, \ldots, K_{r,1},$ we get

Definition 3: The L way Tensor Kronecker Decomposition (TKD) [35] of an R-D tensor $\mathcal{A} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_R}$ with a given pattern $K = [K_1', K_2', \ldots, K_L']^T \in \mathbb{Z}^{L \times R}$ is

$$\mathcal{A} = \sum_{f=1}^{F} \mathcal{A}_f = \sum_{f=1}^{F} \alpha_f \mathcal{A}_{f,L} \otimes \cdots \otimes \mathcal{A}_{f,2} \otimes \mathcal{A}_{f,1}$$ (2)

where $\mathcal{Q} = \text{str}_{K_r}(\mathcal{A}) = \sum_{f=1}^{F} \alpha_f \mathcal{Q}_{f,1} \odot \mathcal{Q}_{f,2} \odot \cdots \odot \mathcal{Q}_{f,L}$ and $\mathcal{A}_{f,l} = \text{fold}_{K_r}(\mathcal{q}_{f,l})$ (3)

The relationship between the TKD and CP decomposition with an example of $R = 2, F = 1, L = 3$ and $K = [2, 2, 2, 2, 2, 2, 2]^T$ is shown in Figure 1. Similar to the definition of CP rank, we define F as the Tensor Kronecker Rank (TKR).

Remark 1: By defining $\mathcal{A}_{f,l} = \mathcal{A}_{f,L} \otimes \mathcal{A}_{f,L-1} \otimes \cdots \otimes \mathcal{A}_{f,1}$ we get a $\mathcal{A}_{f,l}$ as a ‘compressed’ data comparing to $\mathcal{A}_f$ in (2). For example, if $\mathcal{A}_f$ is an image, then $\mathcal{A}_{f,l}$ is an image...
with the whole background information but smaller or lower ‘resolution’, and the minimum ‘resolution’ this procedure can achieve is $K_T^T$, the last row of the pattern $K$ or the dimensions of $A_{r,L}$. Therefore, we name $A_{r,L}$ as ‘background tensor’ and the remaining $\{A_{1,l}\}_{l=1}^{L-1}$ as ‘detail tensors’.

**Remark 2:** When $L = 2$, (2) becomes the ‘single Kronecker Tensor Decomposition (KTD)’ defined in [37]:

$$A = \sum_{f=1}^{F} \sum_{g=1}^{G} \alpha_{fg} A_{f,2} \otimes A_{f,1}$$  \( (4) \)

Furthermore, in order to make use of the information of multiple patterns or ‘background images’, some researchers proposed to decompose the data tensor as ‘multiple KTD’ [38]:

$$A = \sum_{g=1}^{G} A_g = \sum_{g=1}^{G} \sum_{f_g=1}^{F_g} \alpha_{fg} A_{f_g,2} \otimes A_{f_g,1}$$  \( (5) \)

where $G$ is the pattern number, $Q_g = \text{str}(K_g)(A_g) = \sum_{f_g=1}^{F_g} \alpha_{fg} q_{fg,2} \otimes q_{fg,1}$ and $A_{f_g,1} = \text{fold}(K_g)(q_{fg,1})$ for $l = 1, 2$. The $K_g \in \mathbb{R}^{2 \times R}$ is the matrix for the $g$-th pattern, and $K_i \neq K_j$ for $i \neq j$. Note that there are several types of methods to calculate (5) [38]. For each $A_g$, it is compressed into several background tensors $A_{f_g,2}$ together with detail tensors $A_{f_g,1}$. Therefore, the ‘multiple KTD’ method employs several types of background information of the original data tensor as well as the low rank characteristic, and is suitable for the problem of visual data recovery [38]. Now we show the relation between the ‘multiple KTD’ and $L$ way TKD as follows.

**Remark 3:** By defining $A_g = \beta_g A$, $g = 1, 2, \ldots, G$ and $\sum_{g=1}^{G} \beta_g = 1$, (2) can be written as:

$$A = \sum_{f=1}^{F} \gamma_{fg} A_{f,L} \otimes \cdots \otimes A_{f,2} \otimes A_{f,1}$$  \( (6) \)

where $\gamma_{fg} = \beta_g \alpha_{fg}$. Furthermore, assigning $A_{f_g,2} = A_{f,L} \otimes \cdots \otimes A_{f,L-g+2} \otimes A_{f,L-g+1}, A_{f_g,1} = A_{f,L-g} \otimes \cdots \otimes A_{f,2} \otimes A_{f,1}$ and $\gamma_{fg} = \alpha_{fg}$, we eventually get (5). Note that when $g = L$, we have $A_{fg,2} = A_{f,L} || A_{f,L} || F$ and $A_{fg,1} = 1$. This means that for a given $L$ and $K$, once the TKD of $A$ is appropriately calculated, we can generate the ‘multiple KTD’ (5) of $A$ for at most $L$ terms. In other words, at most $L$ ‘background’ information can be computed from one single TKD decomposition, which is attractive. However, the calculation of this decomposition requires a CP decomposition of an $L$-D tensor, which is of high cost [34]. Therefore, we propose to apply the r1TT technique [34] for the decomposition (2) using the mapping relationship shown in Figure 1 as follows:

**Definition 4:** A tensor $A$ can be factorized according to (2) using the **Kronecker rank-1 Tensor Train (Kr1TT)** decomposition under a pattern $K$ with the following steps:

**Step 1:** Generate $Q = \text{str}(K)(A) \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_L}$. Then we can decompose the tensor $Q$ into

$$Q = \sum_{d_1=1}^{D_1} \cdots \sum_{d_{L-1}=1}^{D_{L-1}} \sum_{d_L=1}^{D_L} \alpha_{d_1 \cdots d_L} q_{d_1 \cdots d_L}$$  \( (7) \)

where $D_l = \min\{N_l, \prod_{i=1}^{l-1} N_i\}$ for $l = 2, 3, \ldots, L$, using the r1TT approach [34] as shown in Steps 2 to 4 as follows.

**Step 2:** Unfold the tensor $Q$ into a matrix $Q^L \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_L}$ by stacking the last dimension $\mathbf{q}$ of $Q$ into the first one of $Q^L$ and the remaining dimensions of $Q$ into the second one of $Q^L$. Performing the SVD, we get

$$Q^L = U_{D_L} S_{D_L} V_{D_L}^T = U_{D_L} H_{D_L}$$  \( (8) \)

where $S_{D_L} = \text{diag}[[\alpha_1, \alpha_2, \ldots, \alpha_{D_L}]]$ and $H_{D_L} = V_{D_L} S_{D_L}$. Combining (7) and (8), we have $U_L = [\mathbf{q}_1, L, \mathbf{q}_2, L, \ldots, \mathbf{q}_{D_L}, L]$. Furthermore, defining $H_{D_L} = [\mathbf{q}^T_1, L-1, \mathbf{q}^T_2, L-1, \ldots, \mathbf{q}^T_{D_L}, L-1]$ and notice that the vector $\mathbf{q}^T_{D_L,L-1}$ contains the information of the remaining dimensions, thus can be folded back to a tensor $Q^L_{d_{L-1},L-1} = \text{fold}(\mathbf{q}^T_{d_{L-1},L})(\mathbf{q}^T_{d_{L-1},L}) \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_{L-1}}$ where $\mathbf{q}^T_{d_{L-1},L} = [N_1, N_2, \ldots, N_{L-1}]$ and $d_l = 1, 2, \ldots, D_L$.

**Step 3:** Assume that a sub-tensor $Q^L_{d_1 \cdots d_L-1, L-1}$ for $l = 2, 3, \ldots, L$ is obtained, following the above Step 2 we get the new $U_{d_1 \cdots d_{L-1}, L} = [\mathbf{q}_{d_1 \cdots d_{L-1}, L+1}, \mathbf{q}_{d_1 \cdots d_{L-1}, L+2}, \ldots, \mathbf{q}_{d_1 \cdots d_{L-1}, L+D_{L-1}}, L]$, a set of sub-tensors $Q^L_{d_1 \cdots d_{L-1}, L+1}$ and

$$S_{d_1 \cdots d_{L-1}, L} = \text{diag}[[\alpha_{d_1 \cdots d_{L-1}}, \ldots, \alpha_{d_1 \cdots d_{L-1}, D_{L-1}}]]$$  \( (9) \)

**Step 4:** When $L = 2$, the $Q^L_{d_1 \cdots d_2}$ becomes a vector thus we have $\alpha_{d_1 \cdots d_2} = || Q^L_{d_1 \cdots d_2} ||$ and

$$Q^L_{d_1 \cdots d_2} = Q^L_{d_1 \cdots d_2} / \alpha_{d_1 \cdots d_2}$$  \( (10) \)

Finally, we get (7). Folding all the vectors in (7) into $R$-D tensors by (3) and setting $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_F$ by a descending order from $[\alpha_{d_1 \cdots d_1}]$, we eventually get (2), and the number of sub-tensors or Kronecker rank-1 terms is

$$F \leq \prod_{l=2}^{L} D_l$$  \( (11) \)


\[ \text{vec}(\mathcal{A}_i)^T \text{vec}(\mathcal{A}_j)^T = 0 \quad \text{for} \quad i \neq j \quad \text{and vec}() \quad \text{is the vectorization operation because there exists at least one orthogonal vector pairs vec}(\mathcal{A}_i) \quad \text{and vec}(\mathcal{A}_j). \]

Furthermore, when \( \alpha_f = 0 \) for \( f > P \), the tensor is called low Kronecker tensor rank according to the Kr1TT decomposition with rank \( P \).

When \( P = F = \prod_{l=2}^L D_l \), it is of full Kronecker tensor rank. Note that when \( L \) is sufficiently large, \( F \) in (11) will be a large number thus the maximum rank according to the Kr1TT decomposition is also large. In next section, we will propose to solve the tensor completion problem using this decomposition.

### III. MAIN RESULTS

In this section, the tensor recovery algorithm is derived. The task is to recover the \( R-D \) tensor \( \mathcal{X} \in \mathbb{R}^{M_1 \times \cdots \times M_R} \) given a finite number of observation \( \mathcal{M}_\Omega \) where \( \Omega \) being the set indicating the observed entries. Similar to the matrix based one, the tensor completion problem can be formulated as

\[
\min_{\mathcal{X}} \text{tensor rank}(\mathcal{X}) \quad \text{s.t.} \quad \mathcal{X}_\Omega = \mathcal{M}_\Omega
\]

(12)

The basic motivation here is to find a tensor such that it has low tensor rank and its entries over the sample set \( \Omega \) are consistent with the available observations. For visual data, it is well known that the adjacent elements or pixels are related to each others, therefore we can represent the tensor rank by tensor Kronecker rank according to a pre-defined pattern \( \mathbf{K} = [\mathbf{K}_1, \mathbf{K}_2, \cdots, \mathbf{K}_R] \in \mathbb{Z}^{L \times R} \), \( \mathbf{K}_f = [K_{f,1}, K_{f,2}, \cdots, K_{f,L}]^T \), \( M_f = \prod_{l=1}^L K_{f,l} \) and write (12) as

\[
\min_{\mathcal{X}} \text{TKR}(\mathcal{X}) \quad \text{s.t.} \quad \mathcal{X}_\Omega = \mathcal{M}_\Omega
\]

(13)

with the Kr1TT and CP decomposition

\[
\mathcal{X} = \sum_{f=1}^F \sigma_f \mathcal{X}_{f,L} \otimes \cdots \otimes \mathcal{X}_{f,2} \otimes \mathcal{X}_{f,1}
\]

(14)

\[
\mathcal{X}' = \sum_{f=1}^F \sigma_f \mathcal{X}_{f,1} \otimes \mathcal{X}_{f,2} \otimes \cdots \otimes \mathcal{X}_{f,L}
\]

(15)

where \( \mathcal{X}' = \text{str}(R_{(K)}) \), \( \mathcal{X}_{f,l} = \text{fold}_{(K_f)}(\mathcal{X}_{f,l}) \) and \( K_f = [K_{f,1}, K_{f,2}, \cdots, K_{f,R}]^T \). Now we go on to develop the steps of solving (13).

### A. TENSOR COMPLETION USING KRONECKER RANK-1 TENSOR TRAIN

Similar to [39], we define two sets \( S_r := \{ \mathcal{X} | \text{TKR}(\mathcal{X}) \leq r \} \) and \( S_c := \{ \mathcal{X} | \mathcal{X}_\Omega = \mathcal{M}_\Omega \} \) as the tensor rank constraint set and observation deviation constraint set, respectively, with \( r \leq F \) where \( F \) is the maximum tensor Kronecker rank of the tensor \( \mathcal{X} \). Then the problem (13) can be changed into a finding problem finding solution between two sets:

\[
\text{find} \quad \mathcal{X} \in S_r \cap S_c
\]

(16)

By denoting the projection of a point \( \mathcal{Y} \notin S \) onto the set \( S \) as \( \prod_k (\mathcal{Z}) := \arg \min_{(\mathcal{X})} \| \mathcal{X} - \mathcal{Y} \|_F^2 \), (16) can be calculated by alternating projection of a common point onto \( S_r \) and \( S_c \) as:

\[
\mathcal{Y}^{(k)} = \prod_{S_r} (\mathcal{X}^{(k-1)}) \quad \text{(17)}
\]

\[
\mathcal{X}^{(k+1)} = \prod_{S_c} (\mathcal{Y}^{(k)}) \quad \text{(18)}
\]

until convergence. Note that the solution for (18) is \( \mathcal{X}^{(k+1)} = \mathcal{M}_\Omega \), and \( \mathcal{X}^{(k+1)} = \mathcal{Y}^{(k)} \) and \( \mathcal{O} \) indicates the set of entries that are not observed. Therefore we mainly focus on solving (17).

According to [39], when \( L = 2 \), the projection of \( \mathcal{X} \notin S_r \) onto \( S_r \) can be computed via truncated SVD of the matrix \( \mathcal{X}' = \mathcal{X}' = \sum_{f=1}^r \sigma_f \mathcal{X}_{f,1} \otimes \mathcal{X}_{f,2} \otimes \mathcal{X}_{f,1} \) where \( \sigma_f \) are the \( r \) largest singular values. Combining (14) and (15) we get

\[
\mathcal{X} = \sum_{f=1}^r \sigma_f \mathcal{X}_{f,2} \otimes \mathcal{X}_{f,1} \quad \text{(19)}
\]

Similarly, for the general \( L > 2 \) case, we propose to perform the computation via truncated Kr1TT of the matrix \( \mathcal{X}' \) and get

\[
\mathcal{X} = \sum_{f=1}^r \mathcal{X}_{f,L} \otimes \cdots \otimes \mathcal{X}_{f,2} \otimes \mathcal{X}_{f,1} \quad \text{(20)}
\]

However, in order to get the \( r \) largest tensor Kronecker terms, we need to first calculate all \( F = \prod_{l=2}^L D_l \) terms according to Definition 2.7, which is highly inefficient. Therefore, the criterion

\[
\mathcal{X} = \sum_{f=1}^F \mathcal{X}_{f,L} \otimes \cdots \otimes \mathcal{X}_{f,2} \otimes \mathcal{X}_{f,1} \quad \text{(21)}
\]

is used instead of the rank constraint with \( \tau(x) = x \) when \( x > \tau \) and 0 otherwise, where \( \tau \) is the thresholding parameter [16]. It is worth noting that if \( \alpha_{d_1 \ldots d_{L-1} d_{L+1}} < \tau \), then for \( d_l = 1, 2, \cdots, D_L \), we have \( \alpha_{d_1 \ldots d_{L-1} d_{L+1}} \leq \tau \) in (9) thus these terms are not required in the computation process, therefore the complexity is reduced. The steps of this decomposition are shown in Figure 1, where \( L = 3 \) is employed as an example. The steps of the Tensor Completion via Kronecker rank-1 Tensor Train (TC-Kr1TT) approach are summarized in Algorithm 1.

### B. ANALYSIS OF THE NUMBER OF RANK-1 TERMS

In this subsection, the maximum number of tensor Kronecker rank, or the maximum number of \( r \) in (20), denoted as \( r_{\max} \), is studied. Note that when \( \tau = 0 \) in (21), we have \( r = F \) for (20) and (21). We have the following lemma.

Lemma 1: The Kr1TT algorithm of the tensor \( \mathcal{X} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_R} \) can generate at most

\[
r_{\max} = \prod_{p=2}^P D_p
\]

(22)
Algorithm 1 TC-Kr1TT Algorithm

Input: $K, \mathcal{M}_\Omega = \text{str}_{K} (\mathcal{M}_\Omega)$ and $\tau$.
Step 1: Initialize $\mathcal{Q}^{(0)} = \mathcal{M}_\Omega$.
do
Step 2: Compute $\mathcal{Q}^{(k),L} = \text{uf}(\mathcal{Q}^{(k)}) = \mathbf{U}_{D_L} \mathbf{H}_{D_L}^T$, with
$\mathbf{H}_{D_L} = \mathbf{V}_{D_L} \mathbf{S}_{D_L}$ and $\mathbf{S}_{D_L} = \tau (\mathbf{S}_{D_L})$ according to (8).
Note that $D_L' \leq D_L$.
Step 3: Fold all the $D_L$ vectors of $\mathbf{H}_{D_L} = \{q_i(1), \ldots, q_{D_L'}(1)\}$
into new sub-tensors $\mathcal{Q}_{d_1', d_2', \ldots, d_L', L}$ for $d_i' = 1, 2, 3, \ldots, L$.
Step 4: Execute Steps 2 and 3 for all the sub-tensors
$\mathcal{Q}_{d_1', d_2', \ldots, d_L', L}$ until (10) is met.
Step 5: Compute $\mathcal{X}^{(k+1)}$ according to (7) by substituting $D_L$ and $d_L$ by $D_L'$ and $d_L'$
for $l = 2, 3, \ldots, L$.
Step 6: Calculate $\mathcal{X}_{r}^{(k+1)}$ using (18).
until a stopping criterion is reached.
Step 7: Output $\text{str}_{\text{KTT}}(\mathcal{X})$ as recovery result.

Property 1: For an arbitrary tensor $\mathcal{A} \in \mathcal{R}^{G_1 \times G_2 \times \cdots \times G_P}$ with $G_p \geq 1, p = 1, 2, \ldots, P$, the number of sub-tensors that
can be computed using the r1TT decomposition is
$$ r_p = \prod_{p=2}^{P} D_p $$
where $D_p = \min\{G_p, \prod_{i=1}^{p-1} G_i\}$.

Proof: It can be calculated according to the definition of the r1TT decomposition. Note that for another arbitrary tensor $\mathcal{A}^{p+1} \in \mathcal{R}^{G_1 \times G_2 \times \cdots \times G_P \times G_{p+1}}$, the number of rank-1 sub-tensors this decomposition can obtain is $r_{p+1} = D_{p+1} r_p$.

Property 2: Following the definition in Property 1 and set $G_P \geq G_{P-1} \geq \cdots \geq G_1 \geq 1$ to get an $r_P$. Furthermore, assume that there exist another arbitrary tensor $\mathcal{A}' \in \mathcal{R}^{G_1' \times G_2' \times \cdots \times G_P'}$ with $\{G_p\}^P_{p=1}$ being a full permutation of $\{G_p\}^P_{p=1}$ defined in Property 1 but in a different order, $r_p \geq r_p'$ always holds.

Proof: When $P = 2$, we have $r_2 = r'_2$ and the property holds.

For $P \geq 3$, assume that the property holds for $P - 1$ but not $P$, which is, $r_{p-1} \geq r_{p-1}'$ but $r_P < r_P'$ defined as in Property 1. Then we have two cases: $G_P' = G_P$ and $G_P' \neq G_P$.

Case 1: For $G_P' = G_P$, we have $D_P' = D_P$ and $r_P = D_P r_{P-1}'$ thus $r_{p-1} < r_{p-1}'$ which is contradicted to the assumption.

Case 2: If $G_P' \neq G_P$, then we have $G_j' = G_j \geq G_1$ for some $j \leq P - 1, i \neq j$. Consider another arbitrary tensor $\mathcal{A}'' \in \mathcal{R}^{G_1' \times G_2'' \times \cdots \times G_P''}$ with $G_j'' = G_j' + 1 = G_j''$ and $G_j'' = G_j$ for the others, then we have $D_j' = D_j''$ for $i = 1, 2, \ldots, i - 1, i + 2, \ldots, P$ according to (24), and
$$ D_j' = \min\{G_j', Q\}, \quad D_j'' = \min\{G_j', QG_j'+1\} $$
$$ D_j' = \min\{G_j', Q\}, \quad D_j'' = \min\{G_j'+1, QG_j'+1\} $$
where $Q = \sum_{i=1}^{j-1} G_i$. When $G_j \geq QG_j'+1 \geq Q$, we have $D_j' = D_j'' = QG_j'+1 \min\{G_j', Q\} \geq QG_j'+1 = D_j' + D_j''$. On the other hand, if $G_j < QG_j'+1$, then $D_j' = G_j' \min\{G_j'+1, Q\} \geq G_j' \min\{G_j', Q\} = D_j' + D_j''$ because $G_j' \geq G_j$. Eventually, we get $r_P \geq r_P'$. We can repeat this process and finally get $G_P' = G_P$, which is the first case and the assumption is violated, thus Property 2 is proved.

This property means that among all the P-D tensors that have the lengths of the dimensions being non-repeatedly selected from the set $\{G_p\}^P_{p=1}$, a maximum possible number of Kronecker rank-1 terms is generated when the tensor follows the structure $\mathcal{A} \in \mathcal{R}^{G_1 \times G_2 \times \cdots \times G_P}$ with $G_p \geq G_{P-1} \geq \cdots \geq G_1 \geq 1$, therefore this maximum possible number is written as $r_{p, \text{max}}$. Similarly, we have $r_{P+1, \text{max}} = D_P r_{P, \text{max}}$.

Property 3: Following the definition in Property 2 and assuming that there exist another arbitrary tensor $\mathcal{A}' \in \mathcal{R}^{G_1' \times G_2' \times \cdots \times G_P'}$ where $G_j G_j' = G_j$ for only one $j \in [1, P], G_j' = G_j$ for $i = 1, 2, \ldots, j - 1$ and $G_{k+1} = G_k$ for $k = j + 1, j + 2, \ldots, P$, then we have $r_{p+1} \geq r_{\text{max}, p}$.

Proof: For this case, we have $D_j' = D_j$ for $i = 1, 2, \ldots, j - 1, D_{k+1}' = D_k$ for $k = j + 1, j + 2, \ldots, P$,
$D_j^r D_j^{r+1} = \min\{G_j, \prod_{i=1}^{r-1} G_i^T \} \min\{G_j^r, \prod_{i=1}^{r-1} G_i^T \}$ and $D_j = \min\{G_j, \prod_{i=1}^{r-1} G_i \} G_j^T \prod_{i=1}^{r-1} G_i^T$ thus $D_j D_j^{r+1} \geq D_j$, leading to $r_{p+1}^r \geq r_{\text{max}, p}$ according to (24).

**Remark 4:** Consider another arbitrary tensor $\mathbf{A}'' \in R^{G_{p+1}^r \times \cdots \times G_{p+r}^1}$ where $\{G_{p+1}^r\}^{p+1}$ is the full permutation of $\{G_{p+1}^r\}$ and $G_{p+1}^r \geq G_p \geq \cdots \geq G_1 \geq 1$, according to **Property 2**, we have $r_{\text{max}, p+1} = r_{p+1}^r \geq r_{\text{max}, p}$, which means that if the total number of elements of the data tensor is constrained as $N_i = \prod_{p=1}^P G_p$, folding it into the $(P + 1)$-D tensor $\mathbf{A}''$ will enlarge the maximum number of Kronecker rank-1 terms according to the $P$-D tensor $\mathbf{A}$.

**Remark 5:** If $G_{i+1}^r = G_i$ for $i \geq 1$ and $G_1 = 1$, we have $r_{\text{max}, p+1} = r_{\text{max}, p}$. Therefore, in order to get as many rank-1 terms as possible, it is suggested to select the number of tensor $\mathbf{A}$ when all prime numbers, and then the number $P$ is named as the maximum number of dimensions that $N_i = \prod_{p=1}^P G_p$ elements can be spanned or folded into.

**Remark 6:** Following the definition in **Remark 5**, an arbitrary tensor $\mathbf{A} \in R^{M_1 \times M_2 \times \cdots \times M_R}$ can be spanned into an $L$-D tensor $\mathbf{A}$ where $L = \sum_{r=1}^R P_r$, $P_r = \prod_{p=1}^P G_{p,r}$ for $r = 1, 2, \ldots, R$ and $G_{p,r} \geq G_{p-1,r}, \cdots \geq G_1,r > 1$. Furthermore, the maximum of $L$, denoted as $L_{\text{max}}$, is found when all $G_{p,r}$ are prime numbers.

There are several types of methods to construct this tensor $\mathbf{A}$ and here we present one of them, which is $\mathbf{A} = \text{str}(\mathbf{K})(\mathbf{J})$ where $\mathbf{K} = [K_1, K_2, \ldots, K_R] \in Z_{\text{max}}^{\times R}$, $K_r = [1, \sum_{p=1}^P G_p, 1, \sum_{p=1}^P G_p, \ldots, G_p, 1]$, and $G_r = [G_1, r, G_2, r, \ldots, G_p, r]$ for $r = 1, 2, \ldots, R$. Furthermore, considering a matrix $\mathbf{K}^r = \mathbf{J} \mathbf{K}$ where $\mathbf{J} \in Z^{L \times L}$ is the sparse selection matrix with one and only one along each row and column and 0 elsewhere, the tensor $\mathbf{A}' = \text{str}(\mathbf{K})(\mathbf{J})$ also meets the requirements in **Remark 6**. Among all the possible matrices $\mathbf{K}'$, the one that folds the tensor $\mathbf{A}$ into the $P$-D tensor $\mathbf{A}_{\text{max}} = \text{str}(\mathbf{K}_{\text{max}})(\mathbf{J}) \in R^{G_1^r \times G_2^r \times \cdots \times G_P^r}$ with $L_{\text{max}} = P$ and $G_r \geq G_{r-1} \geq \cdots \geq G_1 > 1$ gives the maximum possible rank-1 terms using Kr1TT decomposition and thus is written as $r_{\text{max}}$ according to **Property 2**. Furthermore, as the total number of elements is $N_i = \prod_{p=1}^P G_p = \prod_{p=1}^P P_r$ and $(G_{p+1}^p)^{p+1}$ are all prime numbers, such structure gives the maximum possible Kronecker rank-1 terms comparing to all other folding structures with $l < L_{\text{max}}$, that is, for any pattern $\mathbf{K}_l = \in Z^{L \times R}$ and $\mathbf{A}_l = \text{str}(\mathbf{K}_l)(\mathbf{J})$, the maximum possible Kronecker rank-1 terms using the Kr1TT approach is $r_{\text{max}}$ and **Lemma 1** is proved.

It is worth noting that although $r_{\text{max}}$ can be obtained accordingly, the pattern matrix $\mathbf{K}$ is not unique. For example, when $R = 2$ and $M_1 = M_2 = 4$, we have $\mathbf{K}_{\text{max}} = [2, 2, 1, 1; 1, 1, 2, 2]^T$ that gives $r_{\text{max}} = 2 \times 2 \times 2 = 8$. However, $\mathbf{K}' = [1, 2, 2, 1, 1, 2, 2]^T$ will lead to a tensor $\mathbf{A}' = \text{str}(\mathbf{K}')(\mathbf{J})$ which can be at most factorized into $4 \times 2 \times 8$ rank-1 terms according to (7), thus also giving 8 Kronecker rank-1 sub-tensors.

**C. DISCUSSION OF PATTERN SELECTION**

It is shown in **Remark 2.3.3** that in order to get as many different types of ‘resolutions’ or background information the pattern $\mathbf{K} \in Z^{L \times R}$ for the Kr1TT decomposition should be defined with a large $L$. This is also supported by **Property 3**. However, as $L$ increases, the number of SVDs to be performed increases, and the complexity of the TC-Kr1TT algorithm will also increase.

In this article, we propose to follow the idea in [35] and set the pattern $\mathbf{K} = [K_1, K_2, \ldots, K_R]^{T}$ with $K_l$ being all small integers for $l = 1, 2, \ldots, L - 1$. For example, to recover an image of dimensions $64 \times 64 \times 3$, a $\mathbf{K} = [2, 2, 16, 2, 2, 16; 1, 1, 3]^T$ could be used, and the TC-Kr1TT method will recover the image with 3 different resolutions, namely, $16 \times 16 \times 3, 32 \times 32 \times 3$ and $64 \times 64 \times 3$. Note that this procedure is also shown to be efficient for generating a large number of Kronecker rank-1 terms according to **Property 2**.

**D. COMPLEXITY ANALYSIS**

In this subsection, the complexity of the proposed algorithm is evaluated. As the major part of complexity comes from the SVDs in (21), we mainly analyze the number of flops of this process. Assume the tensor to be recovered is $\mathbf{X} \in R^{M_1 \times M_2 \times \cdots \times M_R}$, we use $\{\mathbf{A}|_{(M_1, M_2, \ldots, M_R)}\}$ with a pattern $\mathbf{K}$, and $\mathbf{Q} = \text{str}(\mathbf{X})(\mathbf{J}) \in R^{M_1 \times M_2 \times \cdots \times M_R}$. In the first layer, or Step 2 of Algorithm 1, the tensor is folded into an $N_i \times \prod_{i=1}^{L-1} N_i$ matrix for SVD and the complexity is approximately $T_1 = 14 N_i^2 \prod_{i=1}^{L-1} N_i + 8 N_i^3$ flops. For the $l$-th layer, $l = 2, 3, \ldots, L - 1$, the complexity is approximately $T_l = 14 N_i^2 \prod_{i=1}^{L-1} N_i + 8 N_i^3$ for each sub-tensor, and the maximum total number of sub-tensors is $T_{l,2} = \prod_{i=1}^{L} N_i$. Therefore, the total number of flops is approximately $T = \sum_{l=2}^{L} T_l = T_{l,1} T_{l,2}$ for $l = 2, 3, \ldots, L - 1$. However, thanks to the thresholding parameter $\tau$ in Figure 2, the number of sub-tensors $T_{l,2}$ in each layer can be reduced greatly.

According to Section III-C, the pattern with $N_l \geq N_{L-l} \approx \cdots \approx N_1$ as well as $N_i \ll \prod_{i=1}^{L-1} N_i$ is usually suggested. In this case, the complexity order of each SVD in the $l$-th layer becomes $T_{l,1} = O(\kappa N_i^3 \prod_{i=1}^{L-1} N_i) \approx O(\kappa N_i^3)$ except the first layer and $\kappa$ is a constant. In the $l$-th layer, the maximum total number of sub-tensors to be decomposed using SVD according to Step 4 in Algorithm 1 is $T_{l,2} = \prod_{i=1}^{L} N_i \approx N_L N_L^{L-1}$ for $l = 2, 3, \ldots, L - 1$. Therefore, the maximum total complexity of the whole layer is

$$T_l = T_{l,1} T_{l,2} \approx O(\kappa N_i^3 N_L N_L^{L-1}) = O(\kappa N_L N_L^{L-1})$$

thus the total complexity order for all the layers in each iteration is not larger than

$$T = T_L + \sum_{l=2}^{L-1} T_l \approx O(\kappa N_L^2 N_L^{L-1}) + O(\kappa (L - 2) N_L N_L^{L-1})$$

$$= O(\kappa [N_L + (L - 2)N_L] M)$$

where $M = \prod_{r=1}^{R} M_r = N_L N_L^{L-1}$.
It is worth mentioning that the technique of reducing $\tau$ iteratively by $\tau \leftarrow \tau \rho$ with a $\rho \in (0, 1)$ [16] can be adopted to balance the accuracy of tensor data recovery and computational complexity. In this case, the $\tau$ for the first number of iterations is usually large thus can help reducing the complexity and speeding up the process.

IV. PERFORMANCE EVALUATION

In this section, we conduct several experiments based on image and video data, and experimental results demonstrate the effectiveness of the proposed Kr1TT scheme over the state-of-the-art tensor recovery approaches.

A. EXPERIMENT SETUP

The problem of visual data completion is tackled. For the image recovery problem, six images, namely, ‘Lena’, ‘Giant’, ‘House’, ‘Pencils’, ‘Peppers’ and ‘Baboon’, are used, and two situations of sampling or missing, namely, random sampling and row sampling, are employed. These images and typical examples of these two sampling methods of the ‘Lena’ image with 50% observed entries are shown in Figure 3 (upper two rows). On the other hand, the ‘Basketball(B)’ [32] and ‘Gun Shooting(G)’ [40] [41] video of dimensions $144 \times 256 \times 40$ and $100 \times 260 \times 3 \times 80$, respectively, are employed for video inpainting problem, and the first and last frames of these two videos are shown in Figure 3 (bottom two rows).

The proposed methods are compared with KTD with both the orthogonal patterns (‘KTD-orth’) [37] and non-negative patterns (‘KTD-nn’) [38], CP Weighted OPTimization (‘CP-WOPT’) [25], smooth PARAFAC tensor completion with quadratic variation (‘SPC-QV’) [24], tensor completion via tensor singular value decomposition (‘t-SVD’) [32], and the Fixed point Low Rank Tensor Recovery (‘FP-LRTC’) [19] in terms of peak signal-to-noise ratio (PSNR) combined with the structural similarity index (SSIM) [42]. PSNR is most commonly used to measure the quality of reconstruction of image and videos and a higher PSNR generally indicates that the reconstruction is of higher quality, while the SSIM is used for measuring the similarity between the original data and the recovered one. All results are based on an average of 100 Monte Carlo trials.

B. EXPERIMENTAL RESULTS

In the first test, the convergence of the proposed algorithm is analyzed. The ‘Lena’ image of dimensions $256 \times 256 \times 3$ under 30% observed entries are used and four different patterns as

$$K^4 = \begin{bmatrix} 2 & 2 & 2 & 2 & 4Q \\ 2 & 2 & 2 & 4Q \\ 1 & 1 & 1 & 3 \end{bmatrix}^T$$

(29)

$$K^5 = \begin{bmatrix} 2 & 2 & 2 & 2 & 2Q \\ 2 & 2 & 2 & 2Q \\ 1 & 1 & 1 & 1 & 3 \end{bmatrix}^T$$

(30)

with $Q = 8$ are tested for the proposed method, and they are named as ‘TC-Kr1TT-4’, ‘TC-Kr1TT-5’, ‘TC-Kr1TT-6’ and ‘TC-Kr1TT-42’, respectively. For all the methods, we set $\tau = 0.025$ $P$ with $P = ||M_Q||_2$ is the $l_2$ norm of the observed entries. For the reason of fast convergence, a $\rho = 0.95$ for the first 70 iterations and 0.99 for the rest is suggested. The results of PSNR versus number of iterations are shown in Figure 4. For the first three patterns $K^L$ for $L = 4$, $L = 5$ and $L = 6$, as $L$ increases, the resolution of ‘background image’ or ‘background tensor’ decreases, and the original image is decomposed into smaller Kronecker parts, therefore the more Kr1TT terms can be achieved according to Section III-B. It is shown that for a larger $L$, the proposed method gives a better performance but requires more iterations to converge. Furthermore, comparing the results under patterns $K^4$ and $K^{42}$ that share the same $L$, making the ‘background image’ larger will lead to a similar reconstruction performance but a smaller number of iterations to converge. The CPU times of the ‘TC-Kr1TT-4’, ‘TC-Kr1TT-5’, ‘TC-Kr1TT-6’ and ‘TC-Kr1TT-42’ methods are 2.01s, 4.18s, 13.33s and 8.88s for a total number of 100 iterations, indicating that when $L$ increases with a same structure of ‘detail tensor’, which is $2 \times 2 \times 1$ for the first 3 cases, the complexity increases, partly due to the reason that the total number of SVDs required for each iteration increases.

In the second test, the image recovery problem is tackled. The proposed algorithm is compared with

\[ \mathbf{K}^6 = \begin{bmatrix} 2 & 2 & 2 & 2 & 2Q \\ 2 & 2 & 2 & 2Q \\ 1 & 1 & 1 & 1 & 3 \end{bmatrix} \]  

(31)

\[ \mathbf{K}^{42} = \begin{bmatrix} 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 3 \end{bmatrix} \]  

(32)
FIGURE 4. PSNR versus number of iterations with 30% observed entries.

FIGURE 5. PSNR versus observation percentage.

FIGURE 6. SSIM versus observation percentage.

The patterns used for the ‘TC-Kr1TT’ approach are $K^3$ and $K^6$ in (30) and (31) with $Q = 16$. Furthermore, two types of patterns, namely, \(\{K^3_{g=1}\}\) and \(\{K^7_{g=1}\}\), are considered for the ‘KTD’ based algorithms, and they are named as ‘KTD-orth/nn-G’ for $G = 3$ and 7. We set \(\{K^3_{g=1}\} = [[8 8 1; 64 64 3], [16 16 1; 32 32 3], [32 32 1; 16 16 3]],\) and \(\{K^7_{g=1}\} = [[2 1 3; 256 512 1], [4 2 3; 128 256 1], [2 4 3; 256 128 1], [4 8 3; 128 64 1], [8 4 3; 64 128 1], [8 8 1; 64 64 3], [16 16 1; 32 32 3]],\) which are the default patterns suggested in [38]. For all the thresholding based methods, including the ‘KTD-orth’, ‘t-SVD’ and the proposed approaches, the $\tau$ and $\rho$ are the same as those in the first test, while for the ‘KTD-nn’ and ‘CP-WOPT’ methods, as the unknown tensor rank $R$ is required for the completion, we test all the results for $R \in [1, 20]$ and select the best results only. Furthermore, we follow the default settings for ‘SPC-QV’ and ‘FP-LRTC’ methods suggested in [19] and [24]. The PSNR and SSIM results of all the methods for the 6 images as well as their average are shown in Figures 5 and 6. It is shown that for small observation percentage, the ‘SPC-QV’ approach gives the best performance as it utilizes not only the low-rankness but also the smoothness [24], and followed by the ‘TC-Kr1TT-6’ approach. When the ratio of the observed entries is sufficiently large, the ‘KTD-nn-7’ performs the best, followed by the ‘TC-Kr1TT-6’ approach. Note that according to Remark 2.3.3, the ‘TC-Kr1TT-6’ approach covers all the patterns of that the ‘KTD-orth-3’ and ‘KTD-nn-3’ algorithms used, which partly explain the performance gain of the proposed method over the traditional one. The average computational complexities for all the algorithms are provided in Table 1, which clearly indicates the higher efficiency of the proposed approach.

In the third experiment, we test the image recovery performance under the row sampling case. The proposed methods are compared with the ‘KTD-orth-3’, ‘KTD-nn-7’ and ‘SPC-QV’ approaches. Note that the ‘t-SVD’ and ‘FP-LRTC’ methods fail to give reasonable...
Table 1. Image recovery complexity of different algorithms

<table>
<thead>
<tr>
<th>Method</th>
<th>TC-Kr1TT-5</th>
<th>TC-Kr1TT-6</th>
<th>KTD-orth-6</th>
<th>KTD-orth-7</th>
<th>KTD-nn-6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity(s)</td>
<td>17.44</td>
<td>59.25</td>
<td>63.89</td>
<td>11.89</td>
<td>79.80</td>
</tr>
<tr>
<td>Method</td>
<td>KTD-nn-7</td>
<td>CP-WOPT</td>
<td>SPC-QV</td>
<td>t-SVD</td>
<td>PP-LRTC</td>
</tr>
<tr>
<td>Complexity(s)</td>
<td>82.47</td>
<td>179.91</td>
<td>158.38</td>
<td>35.48</td>
<td>68.18</td>
</tr>
</tbody>
</table>

Figure 7. PSNR versus observation percentage.

Figure 8. SSIM versus observation percentage.

Figure 9. One typical example of the completion results of the image ‘peppers’ under 20% observed entries.

Table 2. Result of video completion

<table>
<thead>
<tr>
<th>Pattern</th>
<th>SPC-QV</th>
<th>KTD-orth-3</th>
<th>KTD-nn-7</th>
</tr>
</thead>
<tbody>
<tr>
<td>K_B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K_G</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The results under the row sampling situation thus the results are not shown here. As this row sampling situation is more severe than the random sampling case, the pattern $K_7^T = [2 2 2 2 2 2 8; 2 2 2 2 2 8; 1 1 1 1 1 1 3]T$ is also used for the proposed method and named as ‘TC-Kr1TT-7’. Furthermore, the maximum number of iterations for the ‘SPC-QV’ method and the others are increased to 1000 and 200. The other parameters are the same as those in previous test except $\tau = 0.2 P$. The PSNR and SSIM results are shown in Figures 7 and 8. We can see that the proposed ‘Tc-Kr1TT-7’ algorithm gives the best performance when the observation percentage is small, followed by the ‘Tc-Kr1TT-6’, ‘SPC-QV’, ‘KTD-orth-3’ and ‘KTD-nn-7’ approaches. The average running times of the ‘TC-Kr1TT-6’, ‘TC-Kr1TT-7’, ‘SPC-QV’, ‘KTD-orth-3’ and ‘KTD-nn-7’ methods are 46.01s, 41.82s, 100.64s, 86.90s and 75.60s, respectively, indicating the computational efficiency of the proposed approaches. A typical example of the completion results of the image ‘peppers’ under 20% observed entries is provided in Figure 9, and it is shown that the proposed method can recover more details than the state-of-the-art methods when the sampling rate is small.

For video data, we test the video completion result based on the ‘Basketball’ and ‘Gunshooting’ videos and the results of the ‘SPC-QV’, ‘KTD-orth’, ‘KTD-nn’, ‘t-SVD’ and the proposed methods are shown in Table 2. The patterns used for ‘Basketball’ and ‘Gunshooting’ videos of the proposed method are $K_B^{TC} = [2 2 2 18; 2 2 2 32; 2 2 2 5]T$ and

$K_G^{TC} = [2 2 5 5; 2 2 5 13; 1 1 1 3; 2 2 N_1 5]T$ with $N_1 = 4$. For the KTD based methods, the corresponding patterns are $K_B^{KTD} = [[12 16 5; 12 16 8], [6 8 5; 24 32 8], [9 16 5; 16 16 8]]$ and $K_G^{KTD} = [[4 4 1 4; 25 65 3 N_2], [5 5 1 4; 20 52 3 N_2], [10 10 1 4; 10 26 3 N_2]]$ with $N_2 = 20$. The maximum numbers of iterations are 500 for the ‘SPC-QV’ algorithm and 150 for the others. Note that for video data, the smoothing factor in the ‘SPC-QV’ algorithm along the dimension of different frames is set to 0.5, which is the same value as that along the first and second dimensions as suggested in [24]. The other parameters are the same as those in Figure 9 except $\tau = 0.1 ||M_{\Omega}||_2$ for ‘KTD-orth’, ‘t-SVD’ and the proposed methods. It is shown that the ‘TC-Kr1TT’ approach can achieve a similar recovery result of the ‘SPC-QV’ method with a much smaller computational complexity, and outperforms the other algorithms.

In the end, we test the performance of video inpainting under different number of frames used. The ‘GunShooting’
TABLE 2. Video completion results for ‘Basketball’ and ‘Gunshooting’ videos.

<table>
<thead>
<tr>
<th></th>
<th>Basketball with 20% random sampled entries</th>
<th>Gunshooting with 5% random sampled entries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>PSNR(dB)</td>
<td>SSIM</td>
</tr>
<tr>
<td>TC-Kr1TT</td>
<td>21.98</td>
<td>0.951</td>
</tr>
<tr>
<td>SPC-QV</td>
<td>21.87</td>
<td>0.959</td>
</tr>
<tr>
<td>KTD-orth</td>
<td>21.97</td>
<td>0.961</td>
</tr>
<tr>
<td>KTD- nn</td>
<td>20.69</td>
<td>0.944</td>
</tr>
<tr>
<td>t-SVD</td>
<td>1.87</td>
<td>0.958</td>
</tr>
</tbody>
</table>

FIGURE 10. PSNR, SSIM and computational complexity versus number of frames.

FIGURE 11. A typical example of the completion results of 1 frame of the ‘Gunshooting’ video.

A video with dimensions of 100 × 260 × 3 × N, where N is randomly selected from 20 to 80, is used. It is assumed that only 8 frames out of N original frames are fully observed randomly, and the remaining frames are highly missing with an observation percentage of 3%. Note that when the video size N increases, the ratio of the frames fully observed, 8/N, as well as the actual observation percentage of the whole video, (3 + 97 × 8/N)% will both decrease, thus the PSNR and SSIM results of the recovered videos of all the algorithms might be degraded. The parameters are the same as those in the previous test with \[N_1, N_2\] being \[1, 5\], \[2, 10\], \[3, 15\] and \[4, 20\] for 20, 40, 60 and 80 frames, and the results of PSNR, SSIM and computational time are shown in Figure 10. Furthermore, a typical example of 1 random selected frame of the completion results under a total number of 80 frames is provided in Figure 11.

As the actual ratio of observed entries increases, or the number of total frames decreases, the recovery accuracy of the 4-D data as well as the computational complexity increase for all the methods. When the ratio of observation percentage is sufficiently small, or the number of total frames is sufficiently large, the proposed method gives the best performances with moderate complexities comparing to the state-of-the-art ones, indicating the robustness of the proposed methods against this severe environment.

V. Conclusion

In this paper, a novel tensor completion algorithm combining the ideas of tensor Kronecker and rank-1 tensor train decompositions is applied to image and video inpainting. The complexity order and number of rank-1 sub-tensors this factorization can retrieve are also given. Experimental results show that the proposed methods outperform the state-of-the-art algorithms in terms of PSNR, SSIM and/or computational complexity under some circumstances.

REFERENCES


[41] Pistol Shot Recorded at 73, 000 Frames Per Second


WEIZE SUN was born in Guangdong, China. He received the B.Sc. degree in electronic information science and technology from Sun Yat-sen University, Guangdong, in 2009, and the Ph.D. degree in electronic engineering from the City University of Hong Kong in 2013. Since 2015, he has been with the Department of Information Engineering, Shenzhen University, where he is currently an Assistant Professor. His research interests include statistical signal processing, matrix and tensor recovery, and machine learning and their applications.

YUAN CHEN was born in Shandong, China. She received the B.Sc. degree from the School of Information Science and Engineering, Shandong University, in 2010, and the Ph.D. degree in electronic engineering from the City University of Hong Kong in 2015. Since 2015, she has been with the School of Computer and Communication Engineering, University of Science and Technology Beijing, where she is currently an Assistant Professor. Her research interests include statistical signal processing, frequency estimation, and machine learning and their applications.

HING CHEUNG SO (S’90–M’95–SM’07–F’15) was born in Hong Kong. He received the B.Eng. degree in electronic engineering from the City University of Hong Kong, in 1990 and the Ph.D. degree in electronic engineering from The Chinese University of Hong Kong in 1995. From 1990 to 1991, he was an Electronic Engineer with the Research and Development Division, Everex Systems Engineering Ltd., Hong Kong. From 1995 to 1996, he was a Post-Doctoral Fellow with The Chinese University of Hong Kong. From 1996 to 1999, he was a Research Assistant Professor with the Department of Electronic Engineering, City University of Hong Kong, where he is currently a Professor. His research interests include detection and estimation, fast and adaptive algorithms, multidimensional harmonic retrieval, robust signal processing, source localization, and sparse approximation.

Dr. So was an Elected Member of the Signal Processing Theory and Methods Technical Committee (2011–2016), IEEE Signal Processing Society, where he was the Chair of the Awards Subcommittee (2015–2016). He was on the Editorial Boards of the IEEE TRANSACTIONS ON SIGNAL PROCESSING (2010–2014) and the IEEE SIGNAL PROCESSING MAGAZINE (2014–2017). He has been on the Editorial Boards of Signal Processing (since 2010) and Digital Signal Processing (since 2011). He was also a Lead Guest Editor of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, the special issue on Advances in Time/Frequency Modulated Array Signal Processing, in 2017.