

Accurate and Computational Efficient Joint Multiple Kronecker Pursuit for Tensor Data Recovery

Weize Sun¹, Peng Zhang^{1,*}, Jingxin Xu² and Huochao Tan³

¹Guangdong Key Laboratory of Intelligent Information Processing, College of Electronics and Information Engineering, Shenzhen University, Shenzhen, 518061, China

²Department of Housing and Public Works, Queensland, Australia

³Customer Service Centre, Guangdong Power Grid Corporation, Guangzhou, China

*Corresponding Author: Peng Zhang, Email: pzhang66@126.com

Received: 12 January 2021; Accepted: 17 February 2021

Abstract: This paper addresses the problem of tensor completion from limited samplings. Generally speaking, in order to achieve good recovery result, many tensor completion methods employ alternative optimization or minimization with SVD operations, leading to a high computational complexity. In this paper, we aim to propose algorithms with high recovery accuracy and moderate computational complexity. It is shown that the data to be recovered contains structure of Kronecker Tensor decomposition under multiple patterns, and therefore the tensor completion problem becomes a Kronecker rank optimization one, which can be further relaxed into tensor Frobenius-norm minimization with a constraint of a maximum number of rank-1 basis or tensors. Then the idea of orthogonal matching pursuit is employed to avoid the burdensome SVD operations. Based on these, two methods, namely iterative rank-1 tensor pursuit and joint rank-1 tensor pursuit are proposed. Their economic variants are also included to further reduce the computational and storage complexity, making them effective for large-scale data tensor recovery. To verify the proposed algorithms, both synthesis data and real world data, including SAR data and video data completion, are used. Comparing to the single pattern case, when multiple patterns are used, more stable performance can be achieved with higher complexity by the proposed methods. Furthermore, both results from synthesis and real world data shows the advantage of the proposed methods in term of recovery accuracy and/or computational complexity over the state-of-the-art methods. To conclude, the proposed tensor completion methods are suitable for large scale data completion with high recovery accuracy and moderate computational complexity.

Keywords: Tensor completion; tensor Kronecker decomposition; Kronecker rank-1 decomposition



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

The problem of data recovery from limited number of observed entries, referred to as matrix completion or tensor recovery problem, had attracted significant attention in the pass decade and had been applied in various fields such as recommendation system [1,2], Bioinformatics data [3], computer vision [4,5] and image data [6,7].

In practice, the matrix data we observed is often of low rank structure. Therefore, the matrix completion problem can be solved by finding a low rank matrix to approximate the data matrix to be recovered. However, the process of rank minimization is NP-hard [8]. In order to relax the non-convex NP-hard problem to a traceable and convex one, [9,10] proposed to minimize the matrix nuclear norm instead of matrix rank. Based on this concept, many algorithms, such as the singular values thresholding (SVT) [11] and singular value projection (SVP) [12] approaches. Unfortunately, a great number of singular value decomposition (SVD) operations or iterations are required for these methods, leading to high computational and storage complexity.

In real world systems, the observed data is sometimes of high dimensional structure. The recovery of these tensor data, which can be regarded as a generalization of that of matrix completion to the high dimensional case, is becoming more and more important. It is assumed that the observed data tensor contains a low rank structure and thus the missing entries can be recovered by the minimizing the tensor rank. In fact, several tensor decomposition methods and definitions of tensor rank are employed for the completion problem. Reference [13] proposed to minimize the number of rank-1 tensors from CANDECOMP/PARAFAC (CP) decomposition to recover the original tensor data. However, the decomposition is very costly, and it might fail to give reasonable results in some cases because of the ill-posedness property [14]. Other than the CP decomposition, the Tucker decomposition that yields a core tensor and corresponding subspace matrices of each dimension, are employed for tensor completion [15,16]. However, these methods require the knowledge of tensor rank, and might be not applicable in some applications.

Recently, some new types of tensor decomposition methods are proposed. By operating the 3-dimensional (3-D) tensors as 2-D matrices using the tensor product, the tensor singular value decomposition (t-SVD) is proposed [17]. Then the corresponding tensor rank and tensor nuclear norm [18] are defined and used for tensor data recovery. This method can transform the decomposition of one 3-D tensor to slice-wise matrix ones. However, for tensor with number of dimension higher than 3, it will fold all the dimensions higher than 3 into the third one, resulting in a lost of higher order information. Other researchers, proposed to employ the Kronecker tensor decomposition (KTD) based on the Kronecker structure of general R-D tensor to solve the tensor completion problem [19]. This approach employed the idea that folding an R-D tensor into a high-order tensor and then unfolding it into a matrix by general unfolding [20]. By reducing the tensor computation to matrix computation, a similar matrix rank can be defined, and good recovery results can be obtained. However, the computational complexity of the KTD based methods are still unsatisfying because of the SVD operation of large matrices.

Furthermore, in order to improve the efficiency of data recovery, some orthogonal rank-1 matrix and tensor pursuit approaches had been proposed [21,22]. These methods transfer the minimization of the matrix rank or tensor rank to an iterative rank-1 pursuit problem, and achieve a moderate data recovery result with low computational complexity. In this work, we will propose two novel tensor completion methods based on the minimization of tensor Kronecker rank as well as technique of matching pursuit, and apply the tensor data recovery methods to several real world data completion applications.

The remainder of the paper is organized as follows. Section 2 presents the notations and definitions. In Section 3, the tensor recovery algorithms are developed in details. Experiments are provided in Section 4 and conclusions are drawn in Section 5.

2 Notations and Definitions

The notations used in this paper is first defined. Scalars, vectors, matrices and tensors are denoted by italic, bold lower-case, bold upper-case and bold calligraphic symbols, respectively.

The Frobenius-norm of a tensor \mathcal{A} is defined as $\|\mathcal{A}\|_F = \sqrt{\sum_{m_1=1}^{M_1} \sum_{m_2=1}^{M_2} \cdots \sum_{m_R=1}^{M_R} \mathcal{A}_{(m_1, m_2, \dots, m_R)}^2}$. Furthermore, by stacking the first F -th dimensions of an R -D tensor $\mathcal{A} \in \mathbb{C}^{M_1 \times M_2 \times \cdots \times M_R}$ one by one into one dimension and the remaining dimensions into the other [23], the tensor \mathcal{A} can be unfolded into a matrix $\mathbf{A} = \{\mathcal{A}\}_{1 \rightarrow F} \in \mathbb{C}^{N_F \times M/N_F}$ where $N_f = \prod_{f=1}^F M_f$ and $M = \prod_{r=1}^R M_r$. The $i \times i$ identity matrix is symbolized as \mathbf{I}_i , and the transpose and conjugate transpose of a vector or matrix are written as T and H . Furthermore, we define $\text{vec}(\mathcal{A})$ as the vectorized result of the tensor \mathcal{A} , and $\mathbf{a} = \text{vec}(\mathcal{A}_\Omega)$ be the vectorized of \mathcal{A}_Ω , where Ω is the set indicating the location of tensor \mathcal{A} . Now we go on to the definitions.

Definition 2.1. The **2-way folding** of an R -D tensor $\mathcal{A} \in \mathbb{C}^{M_1 \times M_2 \times \cdots \times M_R}$ by a pattern $\{\mathbf{K}\}$ where $\mathbf{K} = [K_1, K_2, \dots, K_R]$, $M_r = K_r L_r$ for $r = 1, 2, \dots, R$ with integers K_r and L_r , is $\mathcal{B} = \text{fold}(\mathcal{A}) \in \mathbb{C}^{K_1 \times \cdots \times K_R \times L_1 \times \cdots \times L_R}$ where $\mathcal{B}_{(k_1, k_2, \dots, k_R, l_1, l_2, \dots, l_R)} = \mathcal{A}_{(m_1, m_2, \dots, m_R)}$ with $m_r = l_r(K_r - 1) + k_r$, $k_r = 1, 2, \dots, K_r$ and $l_r = 1, 2, \dots, L_r$. This folding procedure in fact folds the r -th and $(r + R)$ -th dimensions of \mathcal{B} .

We can now go on to define the Kronecker tensor product [19] of two tensors $\mathcal{A} \in \mathbb{C}^{K_1 \times K_2 \times \cdots \times K_R}$ and $\mathcal{B} \in \mathbb{C}^{L_1 \times L_2 \times \cdots \times L_R}$ as $\mathcal{X} = \mathcal{B} \otimes \mathcal{A} \in \mathbb{C}^{M_1 \times M_2 \times \cdots \times M_R}$ where $M_r = K_r L_r$, $\mathcal{Y} = \text{fold}_{\{\mathbf{K}\}}(\mathcal{X})$, $\{\mathcal{Y}\}_{1 \rightarrow R} = \mathbf{a}\mathbf{b}^T$, $\mathbf{a} = \{\mathcal{A}\}_{1 \rightarrow R}$ and $\mathbf{b} = \{\mathcal{B}\}_{1 \rightarrow R}$.

Definition 2.2. The $(\mathbf{K} \times L)$ tensor Kronecker unfolding [19] of an R -D tensor $\mathcal{A} \in \mathbb{C}^{M_1 \times M_2 \times \cdots \times M_R}$ with $M_r = K_r L_r$ under a pattern $\{\mathbf{K}\}$ is $\mathbf{Y}(k, l) = \text{unfold}_{\{\mathbf{K}\}}(\mathcal{Y})$ of the dimensions $\prod_{r=1}^R K_r \times \prod_{r=1}^R L_r$ whose entries (k, l) are given by $\mathbf{Y}(\mathbf{k}, \mathbf{l}) = \mathcal{Y}(\mathbf{m})$ for all $\mathbf{k} = [k_1, k_2, \dots, k_R]$, $k_r = 1, 2, \dots, K_r$, $\mathbf{l} = [l_1, l_2, \dots, l_R]$, $l_r = 1, 2, \dots, L_r$ and $\mathbf{m} = [m_1, m_2, \dots, m_R]$, $m_r = (k_r - 1)L_r$. This unfolding can be operated by first compute $\mathcal{Y} = \text{fold}_{\{\mathbf{K}\}}(\mathcal{X}) \in \mathbb{C}^{K_1 \times \cdots \times K_R \times L_1 \times \cdots \times L_R}$ and then calculate $\mathbf{Y} = \{\mathcal{Y}\}_{1 \rightarrow R}$. Then the Kronecker tensor decomposition (KTD) [19] can be defines as the **Definition 2.3.**

Definition 2.3. The KTD of a tensor $\mathcal{X} \in \mathbb{C}^{M_1 \times M_2 \times \cdots \times M_R}$ under G patterns $\{\mathbf{K}_g\}$, $g = 1, 2, \dots, G$ is

$$\mathcal{X} = \sum_{g=1}^G \mathcal{X}_g = \sum_{g=1}^G \sum_{f_g=1}^{F_g} \alpha_{g, f_g} \mathcal{B}_{g, f_g} \otimes \mathcal{A}_{g, f_g} \quad (1)$$

where $\mathcal{Y}_g = \text{fold}_{\{\mathbf{K}_g\}}(\mathcal{X}_g)$, $\{\mathcal{Y}_g\}_{1 \rightarrow R} = \sum_{f_g=1}^{F_g} \alpha_{g, f_g} \mathbf{a}_{g, f_g} \mathbf{b}_{g, f_g}^T$, $\mathbf{a}_{g, f_g} = \{\mathcal{A}_{g, f_g}\}_{1 \rightarrow R}$, $\mathbf{b}_{g, f_g} = \{\mathcal{B}_{g, f_g}\}_{1 \rightarrow R}$, and $\|\mathbf{a}_{g, f_g}\|_2 = \|\mathbf{b}_{g, f_g}\|_2 = 1$ the α_{g, f_g} is the amplitude of the f_g -th term, and the \mathbf{K}_g is the g -th pattern. This factorization can be calculated using truncated SVD or non-negative matrix factorization [19] on \mathcal{X}_g for $g = 1, 2, \dots, G$. It is worth noting that the word ‘pattern’ here refers to using different shape to explore different structure information of the tensor data.

Following this definition, the tensor rank of \mathcal{X} can be defined as the total number of Kronecker terms the joint KTD approach generates, which is $\sum_{g=1}^G F_g$. Furthermore, when $G=1$ and $F_G=1$, it is called Kronecker rank-1 tensor under the pattern $\{\mathbf{K}_G\}$ [19].

3 Main Result

The task is to recover the R-D tensor $\mathcal{X} \in \mathbb{C}^{M_1 \times M_2 \times \dots \times M_R}$ from the partly observed tensor \mathcal{X} , denoted as \mathcal{X}_Ω where Ω is the 1/0 set indicating the location of the observed entries. By writing the tensor \mathcal{X} as a linear combination of several Kronecker rank-1 tensors under a given set of patterns $\{\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_G\}$, we get:

$$\mathcal{X} = \sum_{g=1}^G \mathcal{X}_g = \sum_{g=1}^G \sum_{f=1}^F \theta_{g,f} \mathcal{M}_{g,f} \quad (2)$$

where $\mathcal{M}_{g,f}$ is the f -th Kronecker rank-1 tensor with the g -th pattern \mathbf{K}_g and $\theta_{g,f}$ is the magnitude of $\mathcal{M}_{g,f}$ for $g=1, 2, \dots, G$ and $f=1, 2, \dots, F$. Note that here we assume that the ranks of \mathcal{X}_g for $g=1, 2, \dots, G$ are F . By writing $\boldsymbol{\theta}_f = [\theta_{1,f}, \theta_{2,f}, \dots, \theta_{G,f}]^T$ for $f=1, 2, \dots, F$ and $\boldsymbol{\theta} = [\boldsymbol{\theta}_1; \boldsymbol{\theta}_2; \dots; \boldsymbol{\theta}_G]$, the low rank tensor completion problem becomes:

$$\min_{\boldsymbol{\theta}} \|\boldsymbol{\theta}\|_0 \quad s.t. \quad \left(\sum_{g=1}^G \sum_{f=1}^F \theta_{g,f} \mathcal{M}_{g,f} \right)_{\Omega} = \mathcal{X}_{\Omega} \quad (3)$$

where $\|\boldsymbol{\theta}\|_0$ denotes the number of nonzero elements of $\boldsymbol{\theta}$. For noisy scenario, the optimization problem is $\|\mathcal{X}_{\Omega} - \left(\sum_{g=1}^G \sum_{f=1}^F \theta_{g,f} \mathcal{M}_{g,f} \right)_{\Omega}\| \leq \epsilon$, where ϵ is a small value. The (3) can be transformed into:

$$\min_{\boldsymbol{\theta}} \left\| \mathcal{X}_{\Omega} - \left(\sum_{g=1}^G \sum_{f=1}^F \theta_{g,f} \mathcal{M}_{g,f} \right)_{\Omega} \right\|_F^2 \quad s.t. \quad \|\boldsymbol{\theta}\|_0 \leq F \quad (4)$$

where F is the maximum number of Kronecker rank. We can solve this problem by a greedy matching pursuit type method.

3.1 Iterative Multiple Kronecker Tensor Pursuit

In this Iterative multiple Kronecker Tensor Pursuit (IKTP) method, we update one Kronecker rank-1 basis tensor \mathcal{M}_n and one magnitude θ_n with one pattern $\mathbf{K}_h \in \{\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_G\}$, where $h = (n-1)\%G + 1$ in an iteration. Therefore, the problem (4) can be rewritten as

$$\min_{\boldsymbol{\theta}} \left\| \mathcal{X}_{\Omega} - \left(\sum_{f=1}^N \theta_f \mathcal{M}_f \right)_{\Omega} \right\|_F^2 \quad s.t. \quad \|\boldsymbol{\theta}\|_0 \leq N \quad (5)$$

where $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_N]^T$ and $N = GF$.

Supposing that after the $(k-1)$ -th iteration, $(k-1)$ Kronecker rank-1 basis tensors $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_{k-1}$ and their weights $\boldsymbol{\theta}^{k-1} = [\theta_1, \theta_2, \dots, \theta_{k-1}]^T$ were obtained. In the k -th iteration, the target is to pursue a new Kronecker rank-1 basis tensor \mathcal{M}_k from the residual term

$\mathcal{R}^k = (\mathcal{X})_\Omega - (\mathcal{Y}^{k-1})_\Omega$, where $\mathcal{Y}^{k-1} = \sum_{n=1}^{k-1} \theta_n \mathcal{M}_n$ is the recovered tensor in the $(k-1)$ -th iteration. According to **Definition 2.2**, we define $\mathcal{R}^k = \text{unfold}_{\{\mathbf{K}_h\}}(\mathbf{R}^k)$ and the $\mathcal{M}_k = \text{unfold}_{\{\mathbf{K}_h\}}(\mathcal{M}_k)$, where $h = (k-1)\%G + 1$, then the \mathbf{M}_k can be solved by optimizing:

$$\max_{\mathbf{M}_k} \left\{ \langle \mathcal{M}_k, \mathcal{R}^k \rangle : \mathbf{M}_k = \mathbf{u}_k \mathbf{v}_k^T, \|\mathbf{M}_k\|_F = 1 \right\} \quad (6)$$

where $\|\mathbf{u}_k\| = \|\mathbf{v}_k^T\| = 1$. The \mathbf{u}_k and \mathbf{v}_k are pair of top left and right singular vectors of \mathbf{R}^k , which can be solved using the power method [23,24]. Then the tensor \mathcal{M}_k can be folded form $\mathbf{M}_k = \mathbf{u}_k \mathbf{v}_k^T$ according to **Definition 2.2**.

After solving the new Kronecker rank-1 basis tensor \mathcal{M}_k , the weight vector θ^k for all currently available basis tensor $\{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_k\}$ can be updated by:

$$\min_{\theta^k} \left\| \mathcal{X}_\Omega - \left(\sum_{f=1}^k \theta_f \mathcal{M}_f \right) \right\|_\Omega \quad (7)$$

By reshaping the tensor \mathcal{X}_Ω and $(\mathcal{M}_f)_\Omega$ into vectors $\dot{\mathbf{x}} = \text{vec}(\mathcal{X}_\Omega)$, $\dot{\mathbf{m}}_f = \text{vec}((\mathcal{M}_f)_\Omega)$, and letting $\overline{\mathbf{M}}_k = [\dot{\mathbf{m}}_1, \dot{\mathbf{m}}_2, \dots, \dot{\mathbf{m}}_k]$, the optimal solution θ^k of (7) is calculated as

$$\theta^k = \left(\overline{\mathbf{M}}_k^T \overline{\mathbf{M}}_k \right)^{-1} \left(\overline{\mathbf{M}}_k \right)^T \dot{\mathbf{x}} \quad (8)$$

Then we can go on to the $(k+1)$ -th iteration until all N ranks are solved, and this iterative multiple Kronecker Tensor Pursuit (IKTP) method is now summarized in [Tab. 1](#).

Table 1: Iterative Kronecker tensor pursuit

Algorithm 1: Iterative Multiple Kronecker Tensor Pursuit (IKTP)
Input: The tensor data \mathbf{X} , observed set Ω and patterns $\{\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_G\}$
Step 1: Initialize $\mathcal{R}^1 = (\mathcal{X})_\Omega$
Repeat
Step 2: update $(\mathbf{u}_k, \mathbf{v}_k)$ from \mathbf{R}^k according to (6).
Step 3: Compute the optimal weights θ^k according to linkexeqn-8(8).
Step 4: Assign $\mathcal{Y}^{k-1} = \sum_{n=1}^{k-1} \theta_n \mathcal{M}_n$, and set $\mathcal{R}^{k+1} = (\mathcal{X})_\Omega - (\mathcal{Y}^k)_\Omega$.
Until a stopping criterion is satisfied. And give \mathbf{Y} as the output.

3.2 Joint Multiple Kronecker Tensor Pursuit

The proposed IKTP approach updates one Kronecker rank-1 basis tensor according to one of the G patterns alternatively. When the number of ranks to be recovered is large, a lot of iteration is required, making the proposed approach highly computational inefficient. To overcome this problem, we propose a Joint multiple Kronecker Tensor Pursuit (JKTP) method that updates all the G Kronecker rank-1 basis tensors in one iteration under the given patterns.

According to (4), suppose that after the $(k-1)$ -th iteration, the Kronecker rank-1 basis tensors $\mathcal{M}_{1,1}, \dots, \mathcal{M}_{G,1}, \mathcal{M}_{1,2}, \dots, \mathcal{M}_{G,2}, \dots, \mathcal{M}_{1,k-1}, \dots, \mathcal{M}_{G,k-1}$ and their weights

$\boldsymbol{\theta}^{k-1} = [\boldsymbol{\theta}_1; \boldsymbol{\theta}_2; \dots; \boldsymbol{\theta}_{k-1}]$, where $\boldsymbol{\theta}_f = [\theta_{1,f}, \theta_{2,f}, \dots, \theta_{G,f}]^T$ for $f = 1, 2, \dots, k-1$, have already been computed. In the k -th iteration, the target becomes to calculate G new Kronecker rank-1 basis tensors $\mathcal{M}_{1,k}, \dots, \mathcal{M}_{G,k}$ with unit Frobenius norm, from the recover tensor and the residual term are $\mathcal{Y}^{k-1} = \sum_{g=1}^G \sum_{f=1}^{k-1} \theta_{g,f} \mathcal{M}_{g,f}$ and $\mathcal{R}^k = \sum_{g=1}^G \mathcal{R}_g^k = (\mathcal{X})_{\Omega} - (\mathcal{Y}^{k-1})_{\Omega}$, respectively.

Under the given patterns $\{\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_G\}$, we define $\mathcal{R}_g^k = \text{unfold}_{\{\mathbf{K}_g\}}(\mathcal{R}_g^k)$ and $\mathcal{M}_{g,k} = \text{unfold}_{\{\mathbf{K}_g\}}(\mathcal{M}_{g,k})$ for $g = 1, 2, \dots, G$, and proposed to solve the $\mathcal{M}_{g,k}$, $g = 1, 2, \dots, G$, by:

$$\max_{\mathbf{M}_{g,k}} \left\{ \langle \mathbf{M}_{g,k}, \mathbf{R}_g^k \rangle : \mathbf{M}_{g,k} = \mathbf{u}_{g,k} \mathbf{v}_{g,k}^T, \|\mathbf{M}_{g,k}\|_F = 1 \right\} \quad (9)$$

where $\|\mathbf{u}_{g,k}\| = \|\mathbf{v}_{g,k}\| = 1$. The $\mathbf{u}_{g,k}$ and $\mathbf{v}_{g,k}$ are pair of top left and right singular vectors of \mathcal{R}_g^k , which can be solved using the power method [25,26]. The new $\mathbf{M}_{g,k}$ can be available by computing $\mathbf{M}_{g,k} = \mathbf{u}_{g,k} \mathbf{v}_{g,k}^T$. Then the $\mathcal{M}_{g,k}$ can be retrieve from the updated $\mathbf{M}_{g,k}$ for $g = 1, 2, \dots, G$.

After solving the new Kronecker rank-1 basis tensors $\mathcal{M}_{1,k}, \dots, \mathcal{M}_{G,k}$, we update the weights $\boldsymbol{\theta}^k = [\boldsymbol{\theta}_1; \boldsymbol{\theta}_2; \dots; \boldsymbol{\theta}_k]$ for all currently available basis tensors $\mathcal{M}_{1,1}, \dots, \mathcal{M}_{G,1}, \mathcal{M}_{1,2}, \dots, \mathcal{M}_{G,2}, \dots, \mathcal{M}_{1,k}, \dots, \mathcal{M}_{G,k}$ by solving the following least squares regression problem:

$$\min_{\boldsymbol{\theta}^k} \left\| \mathbf{X}_{\Omega} - \left(\sum_{g=1}^G \sum_{f=1}^k \theta_{g,f} \mathcal{M}_{g,f} \right)_{\Omega} \right\|_2 \quad (10)$$

Then the optimal solution $\boldsymbol{\theta}^k$ of (10) is given by

$$\boldsymbol{\theta}^k = \left(\overline{\mathbf{M}}_k^T \overline{\mathbf{M}}_k \right)^{-1} \left(\overline{\mathbf{M}}_k \right)^T \dot{\mathbf{x}} \quad (11)$$

where $\dot{\mathbf{x}} = \text{vec}(\mathbf{X}_{\Omega})$, $\overline{\mathbf{M}}_k = [\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_G]$, and $\mathbf{M}_g = [\dot{\mathbf{m}}_{g,1}, \dot{\mathbf{m}}_{g,2}, \dots, \dot{\mathbf{m}}_{g,k}]$, $\dot{\mathbf{m}}_{g,f} = \text{vec}((\mathcal{M}_{g,f})_{\Omega})$ for $f = 1, 2, \dots, k$, $g = 1, 2, \dots, G$. The proposed JKTP approach is now summarized in Tab. 2.

Table 2: Joint Kronecker tensor pursuit

Algorithm 2: Joint Kronecker Tensor Pursuit (JKTP)

Input: The tensor data \mathbf{X} , observed set Ω and patterns $\{\mathbf{K}_1, \mathbf{K}_2, \dots, \mathbf{K}_G\}$.

Step 1: Initialize $\mathcal{R}^1 = (\mathbf{X})_{\Omega}$.

Repeat

Step 2: for $g = 1, 2, \dots, G$:

update $(\mathbf{u}_{g,k}, \mathbf{v}_{g,k})$ from \mathbf{R}_g^k according to (9).

Step 3: Compute the optimal weights $\boldsymbol{\theta}^k$ using (11).

Step 4: Assign $\mathcal{Y}^{k-1} = \sum_{g=1}^G \sum_{f=1}^{k-1} \theta_{g,f} \mathcal{M}_{g,f}$, and set $\mathcal{R}^{k+1} = \sum_{g=1}^G \mathcal{R}_g^{k+1} = (\mathcal{X})_{\Omega} - (\mathcal{Y}^k)_{\Omega}$.

Until a stopping criterion is satisfied. And give \mathbf{Y} as the output.

3.3 Economic Algorithms

In each iteration, the proposed IKTP algorithm will track all pursued bases and save them in the memory, leading to a high requirement of storage space. Furthermore, when the number of iteration is large, the LS solution (8) will compute the inverse of a large matrix, making it be rather computationally unattractive. To overcome these problems, an economic updating scheme

which update $\alpha = [\alpha_1, \alpha_2]$ instead of the magnitude term θ can be used in Algorithm 1, where the α can be figured by

$$\alpha^k = \min_{[\alpha_1, \alpha_2]} \left\| \mathcal{X} - \alpha_1 \mathcal{Y}^{k-1} - \alpha_2 \mathcal{M}_k \right\|_2 \quad (12)$$

And the updated recovered tensor becomes $\mathcal{Y}^k = \alpha_1 \mathcal{Y}^{k-1} + \alpha_2 \mathcal{M}_k$.

Similarly, the economic updating technique can be used to update $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_{G+1}]$ for the JKTP approach instead of (10), where α is calculated as

$$\alpha^k = \min_{[\alpha_1, \alpha_2, \dots, \alpha_{G+1}]} \left\| \mathcal{X} - \alpha_1 \mathcal{Y}^{k-1} - \sum_{g=1}^G \alpha_{g+1} \mathcal{M}_{g,k} \right\|_2 \quad (13)$$

And the recovered tensor can be obtained by $\mathcal{Y}^k = \alpha_1 \mathcal{Y}^{k-1} + \sum_{g=1}^G \alpha_{g+1} \mathcal{M}_{g,k}$.

4 Performance Evaluation

In this section, we conduct experiments based on both visual data and SAR imaging data completion. The proposed IKTP and JKTP methods as well as they economic realization, IKTP-econ and JKTP-econ, are evaluated, and the state-of-the-art algorithms, including Tucker [15], KTD [19], R1TP [21] and R1MP [22] are employed for comparison. For all algorithms, the stop criterion is met if the two conditions are reached: $\|\mathcal{X}^k - \mathcal{X}^{k-1}\|_F < \varepsilon = 10^{-6}$ or a maximum number of i_{\max} is satisfied. All the experiments are performed using MATLAB running on Inter(R) Core(TM) i7-8700@3.2 GHz for 100 Monte Carlo trials.

4.1 Image Recovery

First, the problem of image recovery by JKTP, JKTP-econ, IKTP, IKTP-econ, Tucker [15], KTD [19], and R1TP [22] methods from randomly sampled entries is tackled. Six images of dimensions $256 \times 256 \times 3$ and $1024 \times 1024 \times 3$ that separated into 2 groups are used for testing and they are shown in Fig. 1, we can infer that the completion result of the images from the first group might be better than that from the second one, as the pixels in the image form the former group is more related to each other.



Figure 1: Images for testing: group 1 contain some patterns and those in group 2 are not

In the first test, the convergence performances of the proposed algorithms are verified. The images of dimensions $256 \times 256 \times 3$ are used for the proposed methods. The patterns used in this test are $\{\mathbf{K}_1\} = [8, 8, 1]$, $\{\mathbf{K}_2\} = [16, 16, 1]$, $\{\mathbf{K}_3\} = [32, 32, 1]$, $\{\mathbf{K}_4\} = [64, 64, 1]$, and the observation rate is set to be 0.2. Note that in order to achieve a good performance, the patterns are suggested to be set as square as possible. By writing ‘JKTP-i,’ ‘IKTP-i,’ ‘JKTP-econ-i’ and ‘IKTP-econ-i,’ $i = 1, 2, 3, 4$, we mean that the proposed approach will use i patterns out of all 4 patterns for data recovery, and the patterns used are randomly selected in each Monte Carlo trial. The PSNR results of the two groups images data under numbers of iteration in Figs. 2 and 3, respectively. It is shown that the JKTP and JKTP-econ algorithms can converge within 15 iterations and the IKTP and IKTP-econ approaches will converge within 25 iterations when the number of patterns is larger than 3. In the remainder image and video experiments, the maximum numbers of iteration are set to be 20 and 30 for these methods, respectively, and we will use at least 3 patterns for proposed algorithms.

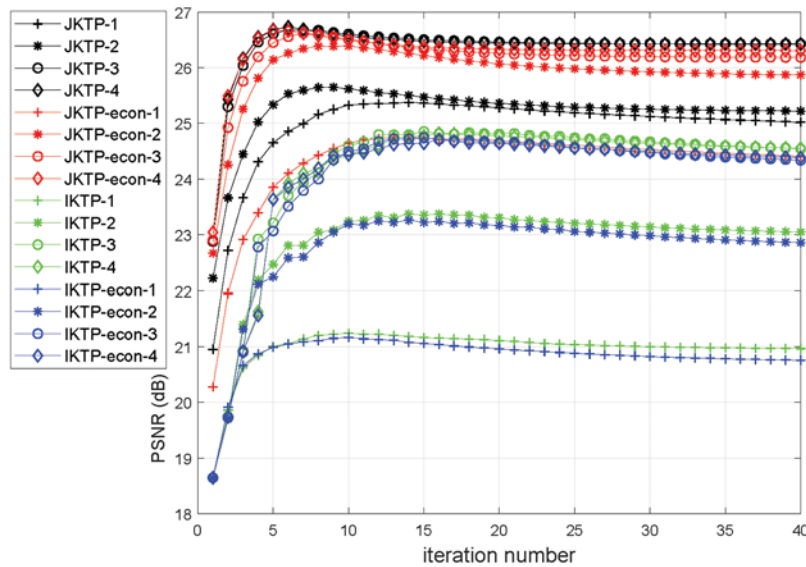


Figure 2: PSNR vs. numbers of iterations with 20% observed entries for group 1 images

In the second test, the completion performances of the proposed approaches under different number of patterns and the state-of-the-art methods for image recovery under 10% observation rate are tested. The four patterns are the same as those in the first test. The average PSNR and SSIM results as well as the computational time in seconds of different methods on the first image, namely, the ‘Group 1: Beans’ under the resolution of $256 \times 256 \times 3$ are shown in Tab. 3. Similar performance can be observed for other images in Fig. 1, and thus their recovery results are not shown here. It is shown that when more pattern used, better performance the proposed approached can achieve. Tucker and R1MP methods as well as the KTD approach with $i = 1, 2, 3, 4$ patterns are also included for comparison. It is shown that the JKTP and JKTP-econ methods give inferior results. For $i \geq 2$, all the proposed algorithms can give better performance than the Tucker and R1MP approaches. Furthermore, the computational complexity of the proposed methods are much smaller than KTD, making them more attractive in dealing image data.

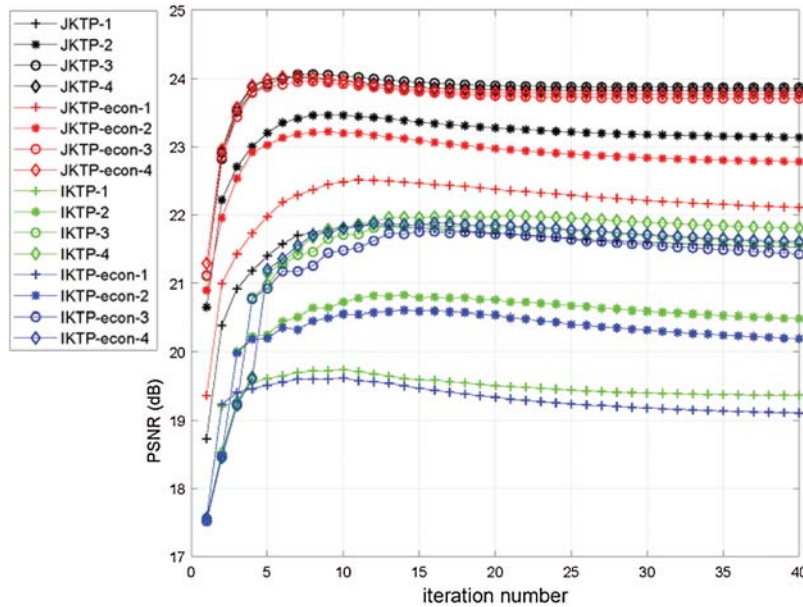


Figure 3: PSNR vs. numbers of iterations with 20% observed entries for group 2 images

Table 3: Image recovery performance of different algorithms

Group 1: Beans

Methods	PSNR ¹	SSIM	Complexity ²	Methods	PSNR ¹	SSIM	Complexity ²
JKTP-1	21.89	0.764	1.107	IKTP-1	21.01	0.708	1.619
JKTP-2	23.52	0.843	1.249	IKTP-2	21.68	0.750	1.636
JKTP-3	23.87	0.848	1.343	IKTP-3	22.48	0.793	1.631
JKTP-4	24.14	0.857	1.514	IKTP-4	22.56	0.795	1.588
JKTP-econ-1	21.76	0.754	0.285	KTD-1	21.77	0.694	0.789
JKTP-econ-2	23.72	0.842	0.428	KTD-2	21.80	0.715	1.338
JKTP-econ-3	23.87	0.848	0.532	KTD-3	22.94	0.787	2.059
JKTP-econ-4	24.14	0.857	0.597	KTD-4	23.01	0.801	2.637
IKTP-econ-1	21.01	0.708	0.441	Tucker	20.65	0.663	1.064
IKTP-econ-2	21.87	0.759	0.429	R1MP	21.93	0.763	0.204
IKTP-econ-3	22.44	0.792	0.398				
IKTP-econ-4	22.47	0.793	0.374				

The units are ¹: dB and ²: second

In the third test, the performance under different percentage of observation percentages, ranging from 2% to 10 %, is evaluated. The images with resolution $1024 \times 1024 \times 3$ are used. We test the proposed methods with 4 patterns $\{\mathbf{K}_1\} = [16, 16, 1]$, $\{\mathbf{K}_2\} = [32, 32, 1]$, $\{\mathbf{K}_3\} = [64, 64, 1]$ and $\{\mathbf{K}_4\} = [16, 16, 1]$. The other parameters for the proposed approaches are the same as those in the previous test. The PSNR and SSIM results of the two groups images are shown in Figs. 4 and 5, respectively. Generally speaking, the proposed methods can give 3 to 7dB gain over the other algorithms in all cases. The average computational time of the JKTP, JKTP-econ, IKTP,

IKTP-econ, KTD, R1MP and Tucker methods are 54.41, 30.54, 14.75, 9.51, 1.57×10^3 , 17.54 and 77.13 s. It is shown that the proposed methods run slightly slower than the R1MP methodology because the R1MP is matrix based rank-1 pursuit method while the proposed methods are tensor based ones. Comparing to the other methods, the computational complexity of proposed algorithms are much smaller, indicating the efficiency of the proposed approaches especially when the dimensions of the data to be recovered is large.

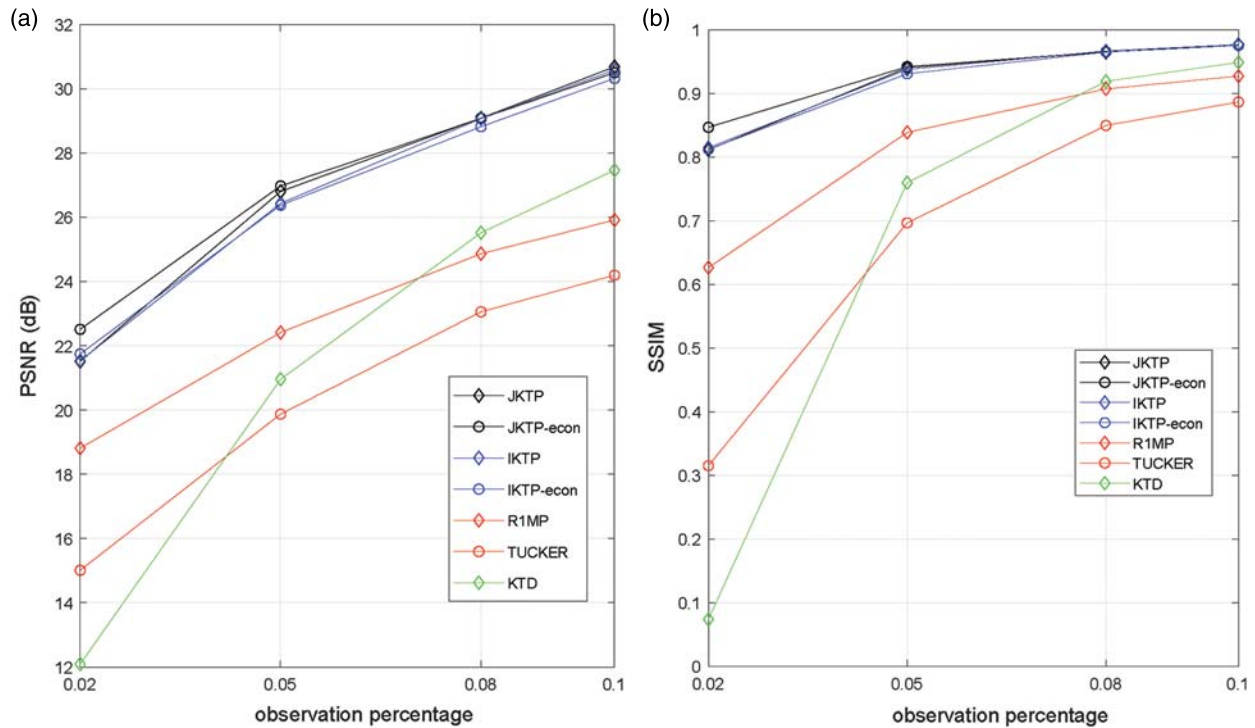


Figure 4: Average PSNR and SSIM vs. observation percentage of group 1 images (a) PSNR (b) SSIM

4.2 Video Inpainting

In this part, we test the algorithms JKTP, JKTP-econ, IKTP, IKTP-econ, Tucker [15], R1MP [19] and R1TP [21] under ‘Gun Shooting’ video [25,26] completion from randomly sampled entries, and the first and last frames of the video are shown in Fig. 6. Note that the KTD approach is not include for comparison because of its high computational complexity. The video is of dimensions $200 \times 520 \times 3 \times 80$.

And the situations of different number of frames ranging from 10 to 80 are tested. The observation rate is set to 0.03, and for all the methods, a maximum number of 50 iterations is used. The patterns for the proposed methods are $\{\mathbf{K}_1\} = [20, 52, 3, 5]$, $\{\mathbf{K}_2\} = [40, 104, 3, 5]$, $\{\mathbf{K}_3\} = [50, 130, 3, 5]$ and $\{\mathbf{K}_4\} = [20, 52, 1, 5]$, $\{\mathbf{K}_5\} = [40, 104, 1, 5]$, $\{\mathbf{K}_6\} = [50, 130, 1, 5]$. The PSNR and SSIM results are shown in Fig. 7 and the computational complexity results are shown in Fig. 8. Note that for the proposed methods, ‘-3’ means that only the first 3 patterns are used, while ‘-6’ means that all 6 patterns are employed in data recovery. Generally speaking, the proposed methods give the best PSNR performance when the number of frames used is higher

than 20. It is worth noting that although the proposed methods with 3 patterns perform worse than R1MP method when the number of frames used is less than 20, when more frames are used, the recovery performance of the proposed approaches increase rapidly, and far outperform the other methods.

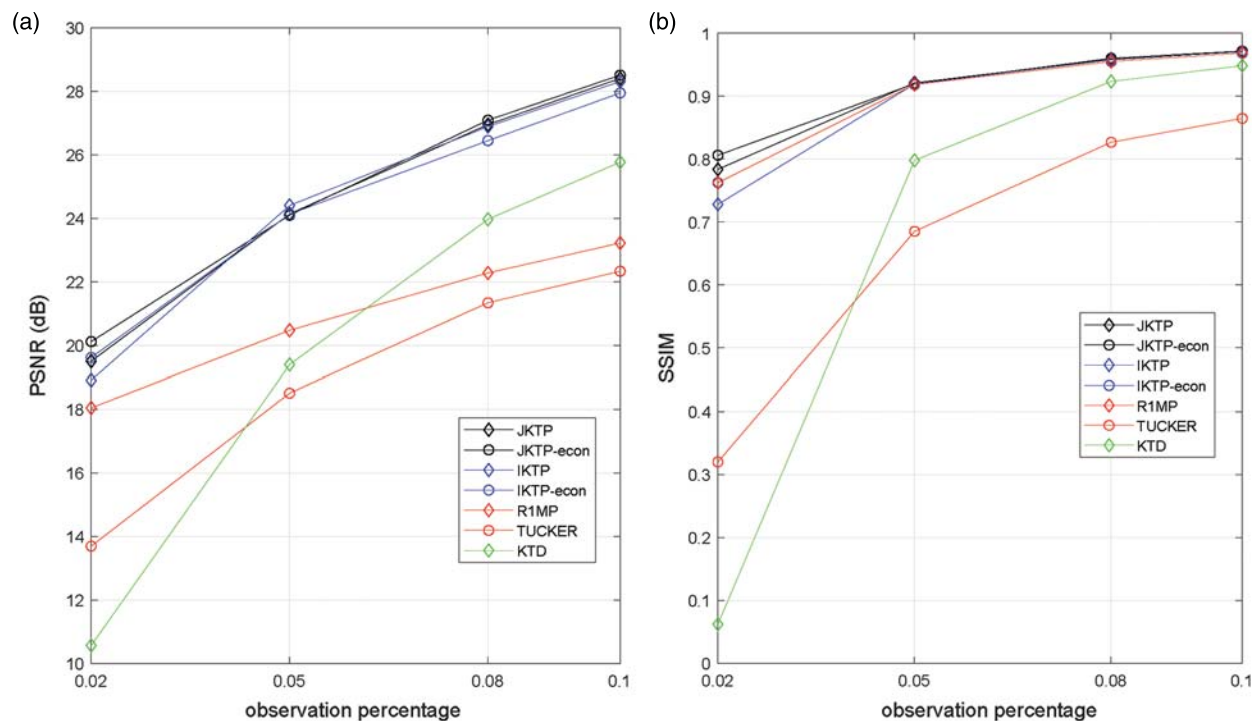


Figure 5: Average PSNR and SSIM vs. observation percentage of group 2 images (a) PSNR (b) SSIM

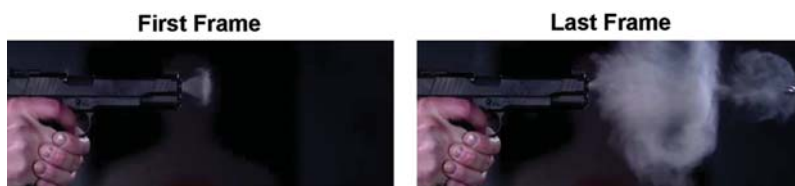


Figure 6: video data for testing

4.3 High Resolution SAR Imaging

In the end, we evaluate the algorithms with under-sampled high resolution SAR imaging data [27,28]. We employ the model shown in Fig. 9 to conduct our experiments as follows. First, scan the test scene to generate the data \mathcal{X} by using the SAR system, and the scanning parameters follow those in [27]. Note that the true image I can be computed from the data \mathcal{X} by the FFT process. Furthermore, the data \mathcal{X} can be sampled randomly under an observation rate to obtain \mathcal{M}_Ω . Finally, the tensor recovery methods can be applied for the recovery of the

original data as $\hat{\mathcal{X}}$, and the recovered image can be computed as $\hat{\mathbf{I}}$. The mean square error (MSE) between \mathcal{X} and $\hat{\mathcal{X}}$ is used to evaluate the performance of the proposed four methods, R1MP [27], Tucker [18] and KTD [22] methods. The tested high resolution SAR imaging scene with the size 6200×12000 is shown in Fig. 10a.

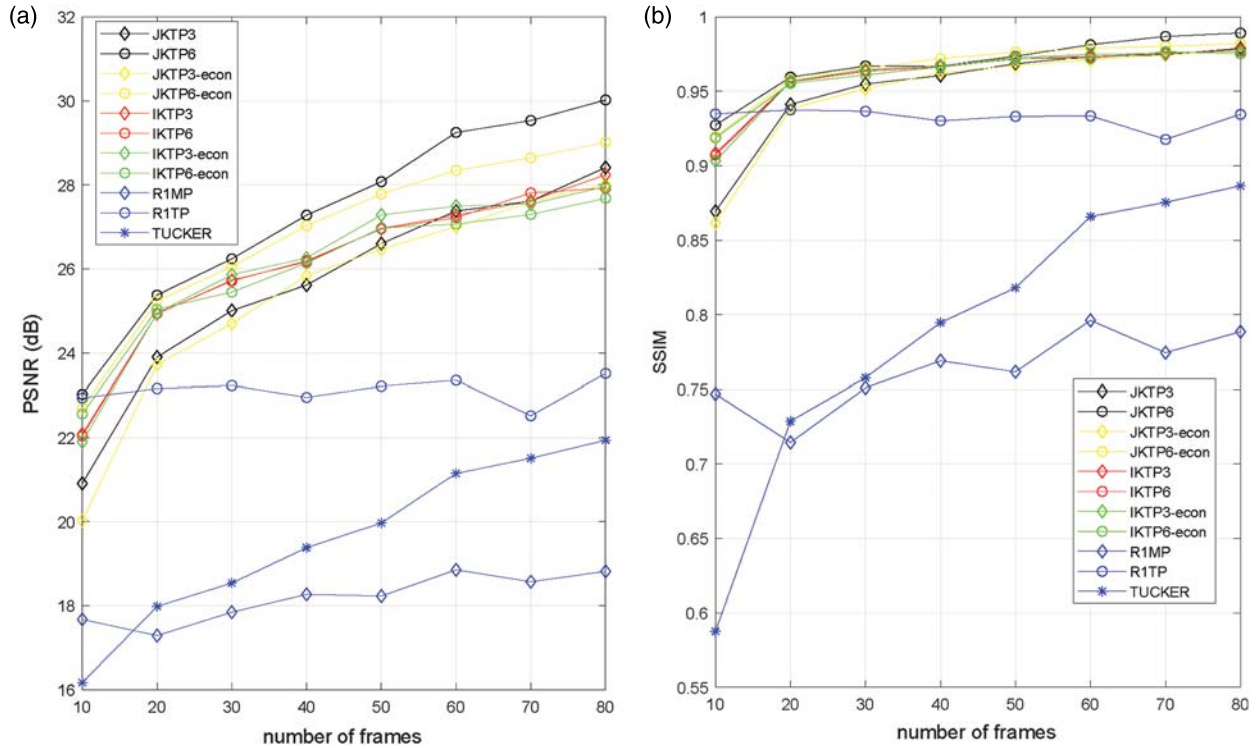


Figure 7: PSNR and SSIM results vs. number of frames (a) PSNR (b) SSIM

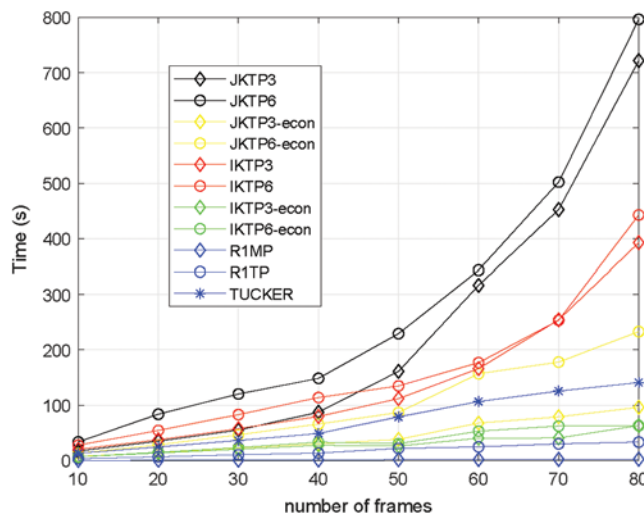


Figure 8: Computational complexity vs. number of frames

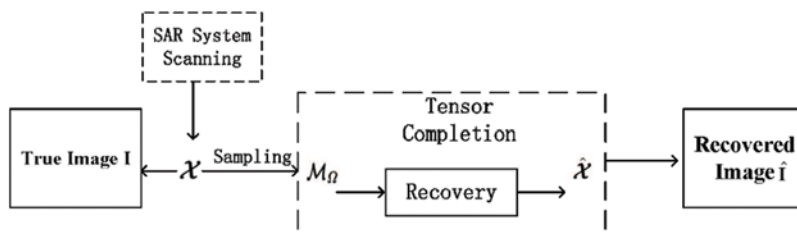


Figure 9: Model of SAR image recovery

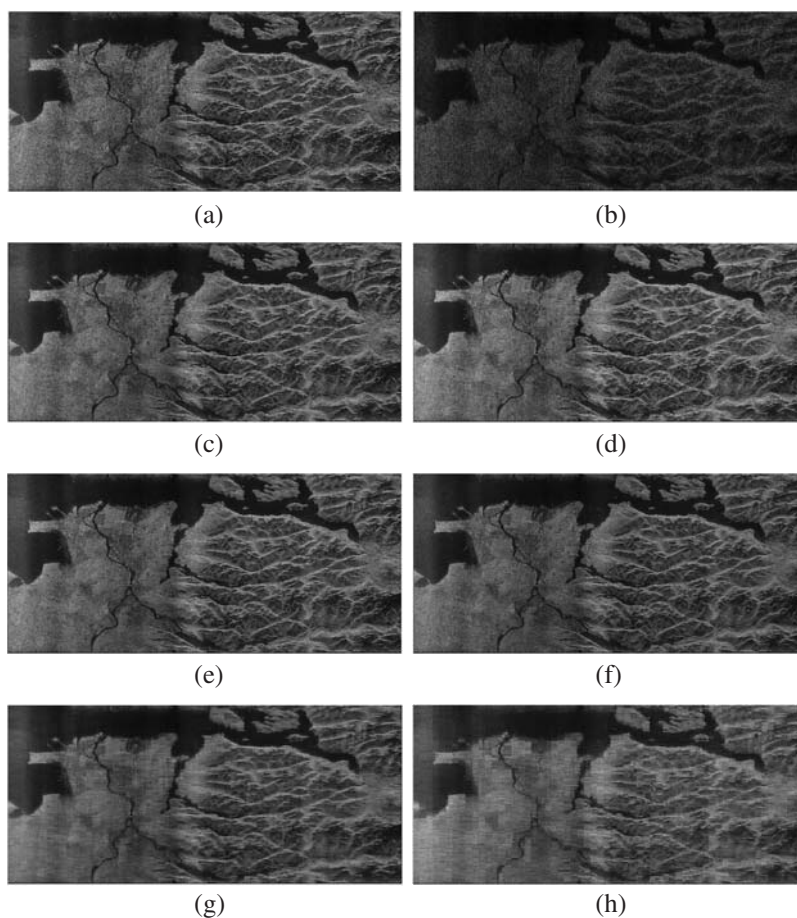


Figure 10: The original, sampled and recovered SAR images (a) Original (b) Sampled (c) JKTP (d) JKTP-econ (e) IKTP (f) IKTP-econ (g) R1MP (h) Tucker

Considering that the size 6200×12000 of the SAR imaging data is too large to be computed effectively by one computer, we will not complete the whole SAR data in one tune. We complete small subdata whose data dimensions are 620×50 , 620×200 and 620×400 in one time and repeat the process for 2400, 600 and 300 times, respectively, for the whole image. The patterns for the proposed methods with different completed data size are $\{\mathbf{K}_1^{50}\} = [124, 5]$, $\{\mathbf{K}_2^{50}\} = [62, 10]$, $\{\mathbf{K}_3^{50}\} = [155, 5]$, $\{\mathbf{K}_4^{50}\} = [155, 2]$,

$\{\mathbf{K}_5^{50}\} = [124, 10]$, $\{\mathbf{K}_6^{50}\} = [62, 5]$, $\{\mathbf{K}_7^{50}\} = [124, 2]$, $\{\mathbf{K}_8^{50}\} = [620, 1]$, $\{\mathbf{K}_1^{200}\} = [60, 20]$,
 $\{\mathbf{K}_2^{200}\} = [62, 25]$, $\{\mathbf{K}_3^{200}\} = [124, 10]$, $\{\mathbf{K}_4^{200}\} = [155, 10]$, $\{\mathbf{K}_5^{200}\} = [155, 8]$, $\{\mathbf{K}_6^{200}\} = [124, 8]$,
 $\{\mathbf{K}_7^{200}\} = [124, 20]$, $\{\mathbf{K}_8^{200}\} = [155, 20]$, $\{\mathbf{K}_1^{400}\} = [124, 20]$, $\{\mathbf{K}_2^{400}\} = [124, 16]$, $\{\mathbf{K}_3^{400}\} = [155, 16]$,
 $\{\mathbf{K}_4^{400}\} = [62, 40]$, $\{\mathbf{K}_5^{400}\} = [124, 25]$, $\{\mathbf{K}_6^{400}\} = [155, 20]$, $\{\mathbf{K}_7^{400}\} = [155, 10]$, $\{\mathbf{K}_8^{400}\} = [62, 20]$.
 The situations of different observation percentages ranging from 10% to 30% are tested and a maximum number of 40 iterations is used. The SAR imaging scene sampled under the observation percentage 10% is shown in Fig. 10b, and the SAR imaging scenes with the size of 620×200 recovered by all the algorithms under 10% observation rate are shown in Figs. 10c–10f. The completion results of the four methods including MSE performances and CUP times are listed in Tab. 4. Generally speaking, the proposed methods can give the best performance with moderate computational complexity among all the algorithms.

Table 4: Completion results with different step-sizes of completion under 10% to 30% observed entries

Methods		JKTP		JKTP-econ		IKTP		IKTP-econ	
Observation rate	Sub-data size	MSE	TIME	MSE	TIME	MSE	TIME	MSE	TIME
0.1	620×50	-9.43	37.04	-9.53	8.797	-7.43	14.72	-7.73	1.219
	620×200	-11.35	41.67	-11.35	12.57	-8.95	15.75	-9.18	1.827
	620×400	-11.39	43.07	-11.88	12.35	-9.65	15.14	-9.81	1.714
0.2	620×50	-13.20	38.42	-13.32	9.494	-11.31	17.71	-11.21	1.502
	620×200	-14.52	43.38	-14.69	13.43	-12.97	19.69	-12.75	2.492
	620×400	-14.65	42.04	-14.85	13.97	-13.71	21.73	-13.52	2.974
0.3	620×50	-15.90	41.45	-15.93	11.57	-14.37	15.14	-14.16	1.714
	620×200	-17.08	49.05	-17.10	14.84	-15.97	14.96	-15.70	2.305
	620×400	-17.24	47.83	-17.19	15.96	-16.40	18.54	-16.40	2.689

Methods		R1MP		TUCKER		KTD	
Observation rate	Sub-data size	MSE	TIME	MSE	TIME	MSE	TIME
0.1	620×50	-7.31	1.191	-7.63	4.163	/	/
	620×200	-8.14	1.940	-5.38	4.661	/	/
	620×400	-8.11	3.276	-6.24	4.534	/	/
0.2	620×50	-11.14	1.359	-11.80	4.318	/	/
	620×200	-11.37	2.296	-10.68	4.481	/	/
	620×400	-11.38	3.919	-11.36	2.795	/	/
0.3	620×50	-14.21	1.508	-13.37	4.197	/	/
	620×200	-14.28	2.504	-13.76	4.691	/	/
	620×400	-13.76	4.389	-13.53	4.827	/	/

5 Conclusion

In this paper, the novel tensor completion algorithms combining the ideas of tensor Kronecker Decomposition and rank-1 tensor pursuit, named IKTP and JKTP, are proposed and applied to color image, video and SAR image inpainting. A novel weight update algorithm, to reduce the time and storage complexity, is also derived. Experimental results show that the proposed methods outperform the state-of-the-art algorithms in terms of PSNR, SSIM and MSE. For the

tensor based algorithms, the results also show the advantage of the proposed methods in terms of computational complexity.

Funding Statement: The work described in this paper was supported in part by the Foundation of Shenzhen under Grant JCYJ20190808122005605, and in part by National Science Fund for Distinguished Young Scholars under grant 61925108, and in part by the National Natural Science Foundation of China (NSFC) under Grant U1713217 and U1913203.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] L. Gongshen, M. Kui, D. Jiachen, J. P. Nees, G. Hongyi *et al.*, “An entity-association-based matrix factorization recommendation algorithm,” *Computers, Materials & Continua*, vol. 58, no. 1, pp. 101–120, 2019.
- [2] H. Bai, X. Li, L. He, L. Jin, C. Wang *et al.*, “Recommendation algorithm based on probabilistic matrix factorization with adaboost,” *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1591–1603, 2020.
- [3] N. Li and B. Li, “Tensor completion for on-board compression of hyper-spectral images,” in *17th IEEE Int. Conf. on Image Processing*, Hong Kong, China, pp. 517–520, 2010.
- [4] J. Bengua, H. Phien, H. Tuan and M. Do, “Efficient tensor completion for color image and video recovery: Low-Rank tensor train,” *IEEE Transaction on Image Processing*, vol. 26, no. 5, pp. 2466–2479, 2017.
- [5] S. Gandy, B. Recht and I. Yamada, “Tensor completion and low-n-rank tensor recovery via convex optimization,” *Inverse Problem*, vol. 27, no. 2, pp. 25–43, 2011.
- [6] H. Yang, J. Yin and Y. Yang, “Robust image hashing scheme based on low-rank decomposition and path integral LBP,” *IEEE Access*, vol. 7, pp. 51656–51664, 2019.
- [7] R. Cheng and D. Lin, “Based on compressed sensing of orthogonal matching pursuit algorithm image recovery,” *Journal of Internet of Things*, vol. 2, no. 1, pp. 37–45, 2020.
- [8] L. Vandenberghe and S. Boyd, “Semidefinite programming,” *SIAM Review*, vol. 38, no. 1, pp. 49–95, 1996.
- [9] M. Fazel, H. Hindi and S. P. Boyd, “A rank minimization heuristic with application to minimum order system approximation,” in *Proc. of the American Control Conf.*, Arlington, VA, USA, pp. 4734–4739, 2001.
- [10] M. Fazel, “Matrix rank minimization with applications,” Stanford University: Department of Electrical Engineering, Ph.D. thesis, 2002.
- [11] J. F. Cai, E. J. Candes and Z. Shen, “A singular value thresholding algorithm for matrix completion,” *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1983, 2010.
- [12] P. Jain, R. Meka and I. S. Dhillon, “Guaranteed rank minimization via singular value projection,” *Advanced in Neural Information Processing*, vol. 20, no. 4, pp. 1956–1982, 2008.
- [13] T. G. Kolda and B. W. Bakder, “Tensor decompositions and applications,” *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [14] V. D. Silva and L. H. Lim, “Tensor rank and the ill-posedness of the best low-rank approximation problem,” *SIAM Journal on Matrix Analysis Applications*, vol. 30, no. 32, pp. 1084–1127, 2008.
- [15] W. Sun and H. C. So, “Accurate and computationally efficient tensor-based subspace approach for multi-dimensional harmonic retrieval,” *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5077–5088, 2012.
- [16] T. Thieu, H. Yang, T. Vu and S. Kim, “Recovering incomplete data using tucker model for tensor with low-n-rank,” *International Journal of Bioscience, Biochemistry and Bioinformatics*, vol. 12, no. 3, pp. 34–40, 2016.

- [17] M. Kilmer and C. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and Its Application*, vol. 465, no. 3, pp. 641–658, 2011.
- [18] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Transaction on Signal Processing*, vol. 65, no. 6, pp. 1511–1526, 2017.
- [19] A. H. Phan, A. Cichocki, P. Tichavsky, D. P. Mandic and K. Matsuoka, "On revealing replicating structures in multiway data: A novel tensor decomposition approach," in *Proc. of Latent Variable Analysis and Signal Separation International Conf.*, Tel Aviv, Israel, pp. 297–305, 2012.
- [20] M. Wang, K. D. Duc, J. Fischer, G. Luta and A. Brockmeier, "Operator norm inequalities between tensor unfoldings on the partition lattice," *Linear Algebra and Its Application*, vol. 520, no. 1, pp. 44–66, 2017.
- [21] W. Sun, L. Huang, H. C. So and J. J. Wan, "Orthogonal tubal rank-1 tensor pursuit for tensor completion," *Signal Processing*, vol. 157, no. 2, pp. 213–224, 2019.
- [22] Z. Wang, M. J. Lai, L. Lu, W. Fan, H. Davulcu *et al.*, "Orthogonal rank one matrix pursuit for low rank matrix completion," *SIAM Journal on Scientific Computing*, vol. 37, no. 1, pp. A488–A514, 2014.
- [23] M. Jaggi and M. Sulovsk, "A simple algorithm for nuclear norm regularized problems," in *Int. Conf. on Machine Learning*, Haifa, Isreal, 2010.
- [24] M. Dudik, Z. Harchaoui and J. Malick, "Lifted coordinate descent for learning with trace-norm regularization," in *AISTATS*, La Palma, Canary Islands, Spain, 2012.
- [25] "Pistol shot recorded at 73,000 frames per second," Published by Discovery, 2015. [Online]. Available: <https://youtu.be/7y9apnbI6GA>.
- [26] W. Wang, V. Aggarwal and S. Aeron, "Efficient low rank tensor ring completion," in *IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 5698–5706, 2017.
- [27] D. Yang, G. S. Liao, S. Q. Zhu, X. Yang and X. P. Zhang, "SAR imaging with undersampled data via matrix completion," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, pp. 9, 2014.
- [28] S. Zhang, G. Dong and G. Kuang, "Matrix completion for downward-looking 3-D SAR imaging with a random sparse linear array," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 4, pp. 1994–2006, 2018.