



Research Article

An Efficient Tensor Completion Method Combining Matrix Factorization and Smoothness

Leiming Tang , Xunjie Cao, Weiyang Chen , and Changbo Ye

Nanjing University of Aeronautics and Astronautics, Nanjing, China

Correspondence should be addressed to Weiyang Chen; weiweigenes@nuaa.edu.cn

Received 25 February 2021; Accepted 28 April 2021; Published 11 May 2021

Academic Editor: Fangqing Wen

Copyright © 2021 Leiming Tang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, the low-complexity tensor completion (LTC) scheme is proposed to improve the efficiency of tensor completion. On one hand, the matrix factorization model is established for complexity reduction, which adopts the matrix factorization into the model of low-rank tensor completion. On the other hand, we introduce the smoothness by total variation regularization and framelet regularization to guarantee the completion performance. Accordingly, given the proposed smooth matrix factorization (SMF) model, an alternating direction method of multiple- (ADMM-) based solution is further proposed to realize the efficient and effective tensor completion. Additionally, we employ a novel tensor initialization approach to accelerate convergence speed. Finally, simulation results are presented to confirm the system gain of the proposed LTC scheme in both efficiency and effectiveness.

1. Introduction

As a high-order generalization of vector and matrix, tensor can show complicated structures of high-order data more clearly. With the tensor form, we can understand the internal connection of data from a higher level perspective. Thus, tensor is widely applied in several fields like signal reconstruction [1], signal processing [2–5], image recovery [6], and video inpainting [7].

The purpose of low-rank tensor completion (LRTC) is to estimate missing entries and recover the incomplete tensor data, which takes advantage of the low rank prior to estimate the missing data. Optimization problem LRTC can be ordinarily formulated as follows:

$$\begin{aligned} \min_{\mathcal{X}} \text{rank}(\mathcal{X}), \\ \text{s.t. } \mathcal{X}_{\Omega} = \mathcal{T}_{\Omega}. \end{aligned} \quad (1)$$

Here, \mathcal{X} is the target tensor, and \mathcal{T} is the available data; Ω is observed entries index. The constraint $\mathcal{X}_{\Omega} = \mathcal{T}_{\Omega}$ is to keep the entries of \mathcal{X} in Ω consistent with \mathcal{T} .

It is obvious that minimizing the tensor rank is the fundamental problem of LRTC. However, there is no uniform definition of the tensor rank. CP-rank [8] and Tucker-rank [9]

are two commonly used definitions, which lie in the corresponding decompositions, CP decomposition [10], and Tucker decomposition [11]. Moreover, the problem of calculating CP-rank is NP-hard, and it has no relevant relaxation.

Although minimizing Tucker-rank is still NP-hard, it can be relaxed by sum of nuclear norm (SNN) since it is the tightest surrogate of matrix rank. Based on this property of Tucker-rank, Liu et al. [12] developed a theoretical framework for LRTC, and the definition of the nuclear norm for tensors is formed as follows:

$$\|\mathcal{X}\|_* = \sum_{i=1}^N \omega_i \|\mathbf{X}_{(i)}\|_*. \quad (2)$$

Here, $\mathbf{X}_{(i)}$ is the mode- i unfolding matrix of \mathcal{X} , $\|\mathcal{X}\|_*$ presents the nuclear norm of $\mathbf{X}_{(i)}$, and $\omega_i \geq 0$ ($\sum_{i=1}^N \omega_i = 1$, $i = 1, 2, \dots, N$) denotes the corresponding weight value of $\mathbf{X}_{(i)}$. Then, the LRTC model can be written as

$$\begin{aligned} \min_{\mathcal{X}} \sum_{i=1}^N \omega_i \|\mathbf{X}_{(i)}\|_*, \\ \text{s.t. } \mathcal{X}_{\Omega} = \mathcal{T}_{\Omega}. \end{aligned} \quad (3)$$

As shown in (3), the LRTC problem is divided into a series of matrix nuclear norm minimization. Through unfolding the target tensor, the matrices are formed to be optimized. In order to resolve (3), SiLRTC, HaLRTC, and FaLRTC were proposed by Liu as the solution for LRTC [12]. Unfortunately, all these algorithms require to compute singular value decomposition (SVD) iteratively, which is complexity exhausted. Considering this difficulty, Liu and Shang [13] applied matrix factorization to the SNN model to reduce the scale of matrices requiring SVD (see details in Section 2).

$$\begin{aligned} \min_{L_i, R_i, \mathcal{X}} \sum_{i=1}^N \omega_i \|R_i\|_*, \quad i = 1, \dots, N, \\ \text{s.t. } \mathbf{X}_{(i)} = L_i R_i, \mathcal{X}_\Omega = \mathcal{T}_\Omega, \end{aligned} \quad (4)$$

where $L_i \in St(I_i, s_i)$, $R_i \in \mathbb{R}^{s_i \times \prod_{j \neq i} I_j}$, and $St(I_i, s_i)$ denote the Stiefel manifold, and $s_i > r_i$ is a given upper bound of rank. In this paper, we use SNN combined with matrix factorization to utilize the low-rankness feature of tensors and improve the efficiency of LRTC.

Additionally, only considering low-rankness factor is not adequate for LRTC, which results in inevitable performance degradation with the decrement of the sampling rate [14, 15]. Apart from the low rankness of tensor, smoothness is also a significant attribute in real world data. For instance, spectral signals [16] and natural pictures and videos [17] generally have this property. Total variation (TV) [18] is commonly applied as constraints to ensure smoothness. The application of TV in image processing has been proved that it can effectively encourage the smooth prior [4, 19, 20].

For the purpose of improving the quality of tensor completion, Yokota and Hontani [21] proposed LRTV-PDS to minimize SNN and TV simultaneously as:

$$\begin{aligned} \min_{\mathcal{X}} \rho \|\mathcal{X}\|_* + (1 - \rho) \|\mathcal{X}\|_{\text{TV}}, \\ \text{s.t. } v_{\min} \leq \mathcal{X} \leq v_{\max}, \|\mathcal{X}_\Omega - \mathcal{T}_\Omega\|_F^2 \leq \delta, \end{aligned} \quad (5)$$

where ρ is the trade-off connecting the SNN and TV term and δ is the noise threshold parameter. The first constraint restricts the recovered data in a range, $[v_{\min}, v_{\max}]$. The second constraint means that the output tensor and original tensor are usually the same at observed entries. Considering the effect of noise, some deviation within a certain range is allowed.

As another commonly used definition of tensor rank, CP-rank is also widely utilized in LRTC. On the basis of CP decomposition, Yokota et al. [22] considered TV as the smoothness constraint along with the CP-rank and proposed the smooth PARAFAC tensor completion (SPC) method to achieve better result of tensor completion when the sampling rate is exceedingly low.

However, the TV regularization usually leads to staircase effect [23], which may cause the possible loss of information and geometric features in practice. For this reason, the framelet regularization can be further applied to avoid the possible performance degradation. More precisely, as a generation of

the orthogonal basis, framelet effectively relaxes the restriction of the orthogonality and linear independence so that the valuable information and geometric features can be well preserved by the introduced redundancy [24]. In this paper, the framelet is applied to exploit the smoothness and preserve more details.

Considering SNN, matrix factorization, framelet, and TV simultaneously, we propose a novel model for tensor completion, which is named smooth matrix factorization (SMF). The SMF model is formulated as

$$\begin{aligned} \min_{\mathcal{X}} \sum_{i=1}^N \omega_i \|R_{(i)}\|_* + \lambda_1 \|W X_{(3)}^T\|_{1,1} + \lambda_2 \|D_S X_{(3)}\|_{1,1}, \\ \text{s.t. } \mathcal{X}_\Omega = \mathcal{T}_\Omega, \mathbf{X}_{(i)} = L_i R_i, L_i \in St(I_i, s_i), i = 1, \dots, N. \end{aligned} \quad (6)$$

Here, λ_1 and λ_2 are adjustable parameters to balance SNN, framelet, and TV. Besides W indicates framelet transformation, D_S denotes the difference matrix, and $l_{1,1}$ -norm is the sum of absolute values of the matrix elements.

In addition, we notice that how to initialize the observed tensor can affect the efficiency and effectiveness of tensor completion. Most of these existing works only focus on optimizing the model and ignore initializing the tensor before iteration algorithm. There are two commonly used initialization methods: one is to set the unknown values as zero; the other is to set them as the average of all observed values. Different from the existing simple methods, our proposed initialization method considers the location of the known data and expands the known data around until the entire tensor is completed. Through the rough estimation of unknown data, we get a more accurate initial tensor, which can accelerate the convergence speed of solution.

In summary, the low-complexity tensor completion (LTC) scheme is proposed to improve the efficiency of LRTC. The proposed LTC mainly consists of two parts—the SMF model and the alternating direction method of multiple (ADMM-) based solution. Specifically, the matrix factorization is introduced into the SNN to exploit the low rankness of the entire tensor. By factorizing matrices, the computational complexity in calculating SVD can be significantly reduced, which leads to efficiency improvement. Then, to guarantee the effectiveness of tensor completion, the TV is used to exploit the smoothness globally. After that, the framelet is further applied to further ensure the smoothness and preserve the information due to its redundancy. With respect to the proposed SMF model, we also give an effective ADMM-based algorithm to solve it. Additionally, we propose a novel initialization approach to accelerate convergence speed. On the basis of natural data, such as color image and grayscale video, the experimental results show our LTC scheme can achieve a better trade-off between performance and complexity for LRTC.

Compared with the existing works, the contributions of this paper are mainly three folds:

- (i) By concurrently utilizing the low-rank and smooth properties, we propose an advanced model for low-

rank tensor completion. Besides, matrix factorization is introduced to this model to save calculation cost

- (ii) We propose an effective tensor initialization approach, which can get a better start point for tensor completion to accelerate the convergence speed
- (iii) An ADMM-based algorithm is developed to resolve the new SMF model. As shown in the result of numerical experiments, our method can clearly improve the effectiveness of LRTC

The rest of this paper is structured as follows: Firstly, we present some notations that this work needs in Section 2. Then, we propose the LTC scheme which consists of the SMF model and the corresponding ADMM-based solution in Section 3. Afterwards, the performance of LTC is evaluated and compared with other competing methods in Section 4. Finally, the conclusions are given in Section 5.

2. Preliminary

In this section, several basic notations and relevant definitions this work needs are shown [25].

2.1. Tensor Basics. Here, we use different fonts to distinguish data formats. For instance, we write vectors as \mathbf{x} , matrices as \mathbf{X} , and tensors as \mathcal{X} . The (i_1, i_2, \dots, i_N) -th component of $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is denoted as $x_{i_1 i_2 \dots i_N}$.

The inner product of two tensors \mathcal{A} and \mathcal{B} is formulated as

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1 i_2 \dots i_N} a_{i_1 i_2 \dots i_N} b_{i_1 i_2 \dots i_N}. \quad (7)$$

As the corresponding norm to the inner product, the definition of Frobenius norm of tensor \mathcal{A} is then given as

$$\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle} = \sqrt{\sum_{i_1 i_2 \dots i_N} |a_{i_1 i_2 \dots i_N}|^2}. \quad (8)$$

The fiber of the tensor \mathcal{X} is a vector obtained by fixing each index except one. The mode- n fiber is denoted as $\mathcal{X}(i_1, \dots, i_{n-1}, :, i_{n+1}, \dots, i_N)$. The mode- n unfolding of tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is represented as $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times \prod_{i \neq n} I_i}$. Additionally, we denote the inverse operator of unfolding as “fold.” For example, the operation of fold a matrix into a tensor is written as $\mathcal{X} = \text{fold}_n(\mathbf{X}_{(n)})$.

2.2. Matrix Factorization. Given a low-rank tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ with rank (r_1, r_2, \dots, r_N) , the mode- i unfolding $\mathbf{X}_{(i)} \in \mathbb{R}^{I_i \times \prod_{j \neq i} I_j}$ can be factorized into the form $\mathcal{X} = \mathbf{L}_i \mathbf{R}_i$, $i = 1, \dots, N$, where $\mathbf{L}_i \in \text{St}(I_i, s_i)$, $\mathbf{R}_i \in \mathbb{R}^{s_i \times \prod_{j \neq i} I_j}$, and $\text{St}(I_i, s_i) = \{\mathbf{B} \in \mathbb{R}^{I_i \times s_i} : \mathbf{B}_T \mathbf{B} = \mathbf{I}\}$ represent the Stiefel manifold, the collection of all orthonormal $I_i \times s_i$ matrices, and $s_i > r_i$ is a designated constraint of rank, $i = 1, \dots, N$. By introducing matrix factorization, we have the following property of tensor nuclear norm:

$$\|\mathbf{X}_{(i)}\|_* = \|\mathbf{L}_i \mathbf{R}_i\|_* = \|\mathbf{R}_i\|_*, \quad i = 1, \dots, N. \quad (9)$$

Thus, the SNN minimization problem (3) can be rewritten with smaller scale matrices as follows to reduce computational complexity:

$$\begin{aligned} \min_{\mathbf{L}_i, \mathbf{R}_i, \mathcal{X}} \quad & \sum_{i=1}^N \omega_i \|\mathbf{R}_i\|_*, \\ \text{s.t.} \quad & \mathbf{X}_{(i)} = \mathbf{L}_i \mathbf{R}_i, \mathbf{L}_i \in \text{St}(I_i, s_i), \\ & \mathcal{X}_\Omega = \mathcal{T}_\Omega, \quad i = 1, \dots, N. \end{aligned} \quad (10)$$

2.3. Framelet. In the discrete setting, the framelet transform is denoted as a linear operator $\mathbf{W} \in \mathbb{R}^{k \times mn}$. For instance, $\mathbf{W}\mathbf{f}$ means that the framelet transform operator \mathbf{W} is applied to the image data, which rearranged as a vector $\mathbf{f} \in \mathbb{R}^{mn}$. On the basis of the unitary extension principle, \mathbf{W} denotes the framelet transform matrix satisfying $\mathbf{W}^T \mathbf{W} = \mathbf{I}$. In this paper, the piecewise linear B-spline framelets constructed by [26] is applied to exploit the smoothness and preserve details in tensor completion.

3. The Proposed LTC Scheme

In this section, the LTC scheme is proposed, which consists of the SMF model, the tensor initialization method, and the ADMM-based algorithm.

3.1. The Proposed SMF Model. Considering a 3rd-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the proposed SMF model is as follows:

$$\begin{aligned} \min_{\mathcal{X}} \quad & \sum_{i=1}^3 \omega_i \|\mathbf{R}_{(i)}\|_* + \lambda_1 \|\mathbf{W}\mathbf{X}_{(3)}\|_{1,1} + \lambda_2 \|\mathbf{D}_S \mathbf{X}_{(3)}\|_{1,1}, \\ \text{s.t.} \quad & \mathcal{X}_\Omega = \mathcal{T}_\Omega, \mathbf{X}_{(i)} = \mathbf{L}_i \mathbf{R}_i, \mathbf{L}_i \in \text{St}(I_i, s_i), i = 1, 2, 3. \end{aligned} \quad (11)$$

Here, λ_1 and λ_2 are regularization parameters, \mathcal{X} is the object tensor, and \mathcal{T}_Ω is the incomplete input tensor, and Ω is the set of indices of available data in \mathcal{T} .

Typically, the SMF model contains two main terms—SNN with matrix factorization and smoothness constraints.

The first term, SNN with matrix factorization, can exploit the low-rank property. The goal of introducing SNN is to exploit the globally multidimensional structure, which is the basic of LRTC. Based on the introduced SNN, matrix factorization is further applied to save calculation cost. The purpose of introducing matrix factorization in SNN is to improve the efficiency. Given a low-rank tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ with rank (r_1, r_2, \dots, r_N) , the mode- i unfolding $\mathbf{X}_{(i)} \in \mathbb{R}^{I_i \times \prod_{j \neq i} I_j}$ can be factorized into the form $\mathcal{X} = \mathbf{L}_i \mathbf{R}_i$, $i = 1, \dots, N$. Then, the SNN problem can be rewritten with smaller scale matrices to reduce computational complexity. For example, the computational complexity of SVD of \mathbf{X} ($\mathbf{X} \in \mathbb{R}^{m \times n}$, $\text{rank}(\mathbf{X}) = r$) is $O(m^2 n + mn^2)$. By

introducing matrix factorization, the cost of computing SVD can be reduced to $O(m^2s + ms^2)$, $r < s \ll n$ [13].

The second term, smoothness constraints, contains the total variation regularization and the framelet regularization, which are used for a better performance of tensor completion.

The total variation regularization $\|\mathbf{D}_s \mathbf{X}_{(3)}\|_{1,1}$ is used to exploit piecewise smoothness along the mode-3 unfolding of \mathcal{X} , where \mathbf{D}_s is the difference matrix

$$\mathbf{D}_s = \begin{pmatrix} -1 & 1 & \cdots & 0 & 0 \\ 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & -1 & 1 \\ 1 & 0 & \cdots & 0 & -1 \end{pmatrix}. \quad (12)$$

The TV regularization is used to make the third dimension of the recovered tensor smooth to improve the efficiency of tensor completion.

The framelet regularization $\|\mathbf{W}\mathbf{X}_{(3)}^T\|_{1,1}$ retains details in spatial domain. Here, \mathbf{W} denotes the framelet transform matrix satisfying $\mathbf{W}^T \mathbf{W} = \mathbf{I}$. The framelet regularization can further ensure the output tensor smooth and preserve the details due to its redundancy.

In summary, SMF exploits the global low-rank property and the smoothness prior along both the spatial and third mode. As shown in experiments, SNN with matrix factorization term, total variation regularization, and framelet regularization, the three parts of SMF take advantage of these two properties to improve the efficiency and effectiveness of tensor completion.

3.2. The Tensor Initialization Method. In order to improve the efficiency of tensor completion, we initialize the incomplete tensor to get a better start point for subsequent data processing. Considering the location of known data, the main idea of our tensor initialization method is expanding the known data around until the entire tensor is completed.

Here, we use a specific example to illustrate our tensor initialization method. Considering an incomplete color image which is denoted as $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we make a rough estimate of missing data by assigning the value of each known pixel to the unknown pixels around it. In other words, by averaging the known values adjacent each unknown pixel, the value of the pixel is initialized. For example, if $x_{i,j,k}$ is unknown and two adjacent pixels, $x_{i-1,j,k}$ and $x_{i,j+1,k}$, are observed, we can set $x_{i,j,k}$ as $0.5(x_{i-1,j,k} + x_{i,j+1,k})$. All unknown pixels observed with adjacent pixels can be initialized in this way.

Here, we introduce the operator $\mathcal{H} = \mathcal{P}(\mathcal{X})$, which means

$$\mathcal{H}(i, j, k) = \begin{cases} 1, & \text{if } \mathcal{X}(i, j, k) \neq 0, \\ 0, & \text{if } \mathcal{X}(i, j, k) = 0, \end{cases} \quad (13)$$

which is used to project the original tensor to the binarized tensor.

In this step, we have completed an expansion of the known pixels, and the pixels initialized this time will be considered as known in next expansion. Then, we repeat this expansion process until the entire tensor is filled. The proposed tensor initialization method is detailed in Algorithm 1.

3.3. The Proposed ADMM-Based Algorithm. Here, we design an effective ADMM-based algorithm to resolve the convex problem. In particular, we introduce two additional variables \mathbf{M} , \mathbf{N} , and the model (11) can be equivalently transformed to the following formulation:

$$\begin{aligned} \min_{\mathbf{L}_i, \mathbf{R}_i, \mathcal{X}} \quad & \sum_{i=1}^3 \omega_i \|\mathbf{R}_i\|_* + \lambda_1 \|\mathbf{M}\|_{1,1} + \lambda_2 \|\mathbf{N}\|_{1,1}, \\ \text{s.t.} \quad & \mathbf{X}_{(i)} = \mathbf{L}_i \mathbf{R}_i, \mathbf{L}_i \in \text{St}(I_i, s_i), i = 1, 2, 3, \\ & \mathbf{M} = \mathbf{W}\mathbf{X}_{(3)}^T, \mathbf{N} = \mathbf{D}_s \mathbf{X}_{(3)}, \mathcal{X}_{\Omega} = \mathcal{F}_{\Omega}, \end{aligned} \quad (14)$$

where $\mathbf{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathbf{N} \in \mathbb{R}^{n_3 \times n_1 \times n_2}$. By introducing matrices \mathbf{M} and \mathbf{N} , we separate blocks of variables and the augmented Lagrangian function of (14) becomes:

$$\begin{aligned} L(\mathbf{L}_i, \mathbf{R}_i, \mathcal{X}, \mathbf{M}, \mathbf{N}) = & \sum_{i=1}^3 \left(\omega_i \|\mathbf{R}_i\|_* + \frac{\beta_1}{2} \|\mathbf{X}_{(i)} - \mathbf{L}_i \mathbf{R}_i\|_F^2 \right) \\ & + \lambda_1 \|\mathbf{M}\|_{1,1} + \langle \mathbf{W}\mathbf{X}_{(3)}^T - \mathbf{M}, \Psi \rangle \\ & + \frac{\beta_2}{2} \|\mathbf{W}\mathbf{X}_{(3)}^T - \mathbf{M}\|_F^2 + \lambda_2 \|\mathbf{N}\|_{1,1} \\ & + \langle \mathbf{D}_s \mathbf{X}_{(3)} - \mathbf{N}, \Theta \rangle + \frac{\beta_3}{2} \|\mathbf{D}_s \mathbf{X}_{(3)} - \mathbf{N}\|_F^2, \\ \text{s.t. } & \mathbf{L}_i \in \text{St}(I_i, s_i), i = 1, 2, 3, \mathcal{X}_{\Omega} = \mathcal{F}_{\Omega}, \end{aligned} \quad (15)$$

where Ψ and Θ are the Lagrange multipliers and β_1 , β_2 , and β_3 are the penalty parameters. Based on ADMM, we can divide the problem (15) into subproblems which are easier to deal with in smaller sizes.

For the purpose of facilitating the analysis of time complexity in the proposed ADMM-based algorithm, we assume that $s = \max(s_1, s_2, s_3)$, $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, and $n_1 \geq n_2 \geq n_3$.

The first subproblem optimizes the variable \mathbf{L}_i , which is presented as:

$$\begin{aligned} \mathbf{L}_i^{k+1} = & \arg \min_{\mathbf{L}_i} \left\| \mathbf{X}_{(i)}^k - \mathbf{L}_i \mathbf{R}_i^k \right\|_F^2, \\ \text{s.t. } & \mathbf{L}_i \in \text{St}(I_i, s_i). \end{aligned} \quad (16)$$

Through solving this subproblem with the orthogonality constraint, the optimal \mathbf{L}_i is calculated. Following [27, 28], the optimal solution is given by:

$$\mathbf{L}_i^{k+1} = \mathbf{Q}\mathbf{R} \left(\mathbf{X}_{(i)}^k \left(\mathbf{R}_i^k \right)^T \right). \quad (17)$$

The cost of computing L is $O(s^2 n_1)$.

```

Input: The observed tensor  $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ .
Output: The initialized tensor  $\mathcal{X}$ .
1:  $\mathcal{X} = \mathcal{T}$ ,  $\mathcal{X}_{next} = \mathcal{X}$ 
2: while 0 exists in  $\mathcal{X}$  do
3:    $\mathcal{H} = \mathcal{P}(\mathcal{X})$ 
4:   for  $k = 1$  to  $n_3$  do
5:     for  $i = 1$  to  $n_1$  do
6:       for  $j = 1$  to  $n_2$  do
7:         if  $\mathcal{X}(i, j, k) = 0$  then
8:            $\alpha_{sum} = \mathcal{H}(i-1, j, k) + \mathcal{H}(i, j+1, k) + \mathcal{H}(i+1, j, k) + \mathcal{H}(i, j-1, k)$ 
9:           if  $\alpha_{sum} \neq 0$  then
10:              $\mathcal{X}_{next}(i, j, k) = (h_{i-1,j,k}/\alpha_{sum})\mathcal{X}(i-1, j, k) +$ 
                $(h_{i,j+1,k}/\alpha_{sum})\mathcal{X}(i, j+1, k) + (h_{i+1,j,k}/\alpha_{sum})\mathcal{X}(i+1, j, k) +$ 
                $(h_{i,j-1,k}/\alpha_{sum})\mathcal{X}(i, j-1, k)$ 
11:           end if
12:         end if
13:       end for
14:     end for
15:   end for
16:    $\mathcal{X} = \mathcal{X}_{next}$ 
17:end while
18:return  $\mathcal{X}$ 

```

ALGORITHM 1: The Proposed Tensor Initialization Method.

The second subproblem optimizing the variable \mathbf{R}_i can be calculated as follows:

$$\mathbf{R}_i^{k+1} = \arg \min_{\mathbf{R}_i} \omega_i \|\mathbf{R}_i\|_* + \frac{\beta_1}{2} \left\| \mathbf{X}_{(i)}^k - \mathbf{L}_i^{k+1} \mathbf{R}_i \right\|_F^2. \quad (18)$$

Then, \mathbf{R}_i^{k+1} is the optimal solution to the subproblem (18) if and only if

$$0 \in \omega_i \partial \left\| \mathbf{R}_i^{k+1} \right\|_* + \beta_1 \left(\mathbf{L}_i^{k+1} \right)^T \left(\mathbf{L}_i^{k+1} \mathbf{R}_i^{k+1} - \mathbf{X}_{(i)}^k \right), \quad (19)$$

where $\partial \|\cdot\|_*$ is the subdifferential of $\|\cdot\|_*$. Considering the property of \mathbf{L}_i^{k+1} , $(\mathbf{L}_i^{k+1})^T \mathbf{L}_i^{k+1} = \mathbf{I}_s$, the formula can be transformed as

$$0 \in \omega_i \partial \left\| \mathbf{R}_i^{k+1} \right\|_* + \beta_1 \left(\mathbf{R}_i^{k+1} - \left(\mathbf{L}_i^{k+1} \right)^T \mathbf{X}_{(i)}^k \right). \quad (20)$$

Obviously, the formula (20) also needs to be satisfied by the optimal solution \mathbf{R}_i for the following convex problem:

$$\arg \min_{\mathbf{R}_i} \omega_i \|\mathbf{R}_i\|_* + \frac{\beta_1}{2} \left\| \mathbf{R}_i - \left(\mathbf{L}_i^{k+1} \right)^T \mathbf{X}_{(i)}^k \right\|_F^2. \quad (21)$$

Therefore, \mathbf{R}_i^{k+1} is also the solution to the subproblem (21), which has an explicit solution.

$$\mathbf{R}_i^{k+1} = \text{SVT}_{\omega_i/\beta_1} \left(\left(\mathbf{L}_i^{k+1} \right)^T \mathbf{X}_{(i)}^k \right). \quad (22)$$

Here, $\text{SVT}_\tau(\cdot)$ is a singular value thresholding operator defined by $\text{SVT}_\tau(\mathbf{X}) = \mathbf{U} \text{diag} [\max(\sigma - \tau, 0)] \mathbf{V}^T$, the SVD of \mathbf{X} is given by $\mathbf{Q} = \mathbf{U} \text{diag} (\{\sigma_i\}_{1 \leq i \leq r}) \mathbf{V}^T$. The complexity of computing \mathbf{R} is $O(sn_1^2 n_2^2)$.

The third subproblem about \mathbf{M} related to framelet regularization can be written as:

$$\begin{aligned} \mathbf{M}^{k+1} &= \arg \min_{\mathbf{M}} \lambda_1 \|\mathbf{M}\|_{1,1} + \left\langle \mathbf{W} \left(\mathbf{X}_{(3)}^k \right)^T - \mathbf{M}, \Psi \right\rangle \\ &\quad + \frac{\beta_2}{2} \left\| \mathbf{W} \left(\mathbf{X}_{(3)}^k \right)^T - \mathbf{M} \right\|_F^2 \\ &= \arg \min_{\mathbf{M}} \lambda_1 \|\mathbf{M}\|_{1,1} \\ &\quad + \frac{\beta_2}{2} \left\| \mathbf{W} \left(\mathbf{X}_{(3)}^k \right)^T - \mathbf{M} + \frac{\Psi}{\beta_2} \right\|_F^2. \end{aligned} \quad (23)$$

This problem has an explicit optimal solution

$$\mathbf{M}^{k+1} = \text{S}_{\lambda_1/\beta_2} \left(\mathbf{W} \left(\mathbf{X}_{(3)}^k \right)^T + \frac{\Psi}{\beta_2} \right), \quad (24)$$

where $\text{S}_\mu(\cdot)$ is a soft-thresholding operator written as:

$$\text{S}_\mu(x) = \begin{cases} 0, & \text{if } |x| \leq \mu, \\ \text{sign}(x)(|x| - \mu), & \text{if } |x| > \mu. \end{cases} \quad (25)$$

The cost of calculating $\mathbf{W} \mathbf{X}_{(3)}^T$ is $O(la^2 n_1 n_2 n_3)$. Here, l denotes the framelet level and a denotes filter number, then the cost of calculating \mathbf{M} is $O(lm^2 n_1 n_2 n_3)$.

Input: The initialized tensor \mathcal{T} , index set Ω , parameters $\lambda_1, \lambda_2, \beta_1, \beta_2$, and β_3 .
Output: The completed tensor \mathcal{X} .
 $\mathcal{X}^0 = \mathcal{T}, L_i^0 = \text{eye}(I_i, s_i), R_i^0 = \text{rand}(s_i, \Pi_{i \neq j} I_j)$,
1: **initialize:** $i = 1, 2, 3, M^0, N^0, \psi^0$ and
 Θ^0 initialized to 0, $k = 0, k_{\max} = 300$.
2: **while** not converged and $k < k_{\max}$ **do**
3: **for** $i = 1$ to 3 **do**
4: L_i^{k+1} via (17)
5: R_i^{k+1} via (22)
6: **end for**
7: M^{k+1} via (24)
8: N^{k+1} via (27)
9: \mathcal{X}^{k+1} via (29)
10: ψ^{k+1}, Θ^{k+1} via (30)
11: **end while**
12: **return** \mathcal{X}^{k+1}

ALGORITHM 2: The Proposed ADMM-based Algorithm for Solving the SMF Model in (11).

The fourth subproblem about N concerning total variation regularization can be calculated as follows:

$$\begin{aligned} N^{k+1} &= \arg \min_N \lambda_2 \|N\|_{1,1} + \langle D_S X_{(3)}^k - N, \Theta \rangle + \frac{\beta_3}{2} \|D_S X_{(3)}^k - N\|_F^2 \\ &= \arg \min_N \lambda_2 \|N\|_{1,1} + \frac{\beta_3}{2} \left\| D_S X_{(3)}^k - N + \frac{\Theta}{\beta_3} \right\|_F^2, \end{aligned} \quad (26)$$

which has an explicit solution

$$N^{k+1} = S_{\lambda_2/\beta_3} \left(D_S X_{(3)}^k + \frac{\Theta}{\beta_3} \right). \quad (27)$$

The cost of computing N is $O(n_1 n_2 n_3^2)$.

The final subproblem optimizing \mathcal{X} can be formulated as:

$$\begin{aligned} \mathcal{X}_{\Omega^c}^{k+1} &= \arg \min_{\mathcal{X}} \sum_{i=1}^3 \left(\frac{\beta_1}{2} \|X_{(i)} - L_i R_i\|_F^2 \right) \\ &\quad + \frac{\beta_2}{2} \left\| \mathbf{W} X_{(3)}^{kT} - \mathbf{M} + \frac{\Psi}{\beta_2} \right\|_F^2 \\ &\quad + \frac{\beta_3}{2} \left\| D_S X_{(3)}^k - N + \frac{\Theta}{\beta_3} \right\|_F^2. \end{aligned} \quad (28)$$

We can update \mathcal{X} as follows:

$$\begin{aligned} \mathcal{X}_{\Omega^c}^{k+1} &= \left(\sum_i \text{fold}_i (\text{inv}(\beta_1 \mathbf{I} + \beta_2 \mathbf{I} + \beta_3 D_s^T D_s)) * (\beta_1 L_i^{k+1} R_i^{k+1} \right. \\ &\quad \left. + \beta_2 \left[\mathbf{W}^T \left(\mathbf{M} - \frac{\Psi}{\beta_2} \right) \right]^T + \frac{\beta_3 D_s^T (N - (\Theta/\beta_3))}{N} \right)_{\Omega^c}, \\ \mathcal{X}_{\Omega}^{k+1} &= \mathcal{T}_{\Omega}. \end{aligned} \quad (29)$$

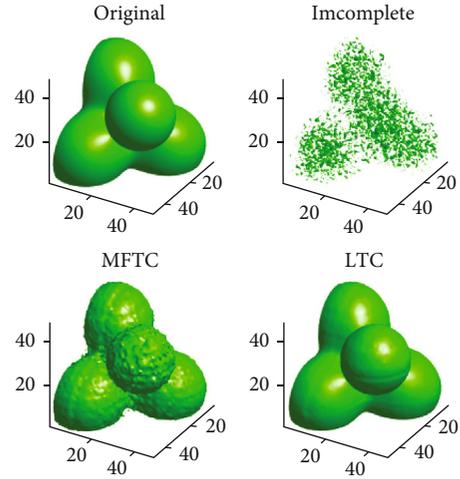


FIGURE 1: Isosurface of synthetic data and completed results of MFTC and LTC.

The cost of calculating \mathcal{X} is $O(sn_1 n_2 n_3 + lm^2 n_1 n_2 n_3 + n_1 n_2 n_3^2 + n_1^3)$.

According to ADMM, the multipliers Ψ and Θ are updated as

$$\Psi^{k+1} = \Psi^k + \beta_2 \left(\mathbf{W} \left(X_{(3)}^{k+1} \right)^T - \mathbf{M}^{k+1} \right), \quad (30)$$

$$\Theta^{k+1} = \Theta^k + \beta_3 \left(D_S X_{(3)}^{k+1} - N^{k+1} \right).$$

To summarize, the proposed iterative ADMM-based algorithm for solving the SMF model in (11) is outlined in Algorithm 2. The computing complexity of L, R, M, N , and \mathcal{X} at each iteration is $O(sn_1^2 n_2^2 + lm^2 n_1 n_2 n_3 + n_1^3)$.

TABLE 1: The PSNR and SSIM values of five LRTC algorithms on image data.

Image	MR	PSNR					SSIM				
		LTC	MFTC	HaLRTC	PDS	SPC	LTC	MFTC	HaLRTC	PDS	SPC
Lena	70%	28.36	22.46	24.96	28.11	28.83	0.8784	0.6738	0.7523	0.8656	0.8880
	80%	26.82	20.17	21.51	26.06	27.46	0.8274	0.5603	0.5834	0.8095	0.8473
	90%	24.21	16.90	17.54	23.11	25.44	0.7329	0.4043	0.3875	0.7083	0.7744
Giant	70%	24.16	20.59	20.36	23.09	25.26	0.8145	0.6949	0.7123	0.7762	0.8476
	80%	22.59	18.48	17.30	21.31	23.55	0.7359	0.5612	0.5780	0.6810	0.7781
	90%	20.41	15.26	13.91	19.22	21.52	0.5934	0.3578	0.3722	0.5340	0.6580
Pepper	70%	29.26	24.73	24.47	30.87	30.38	0.9426	0.8633	0.8440	0.9682	0.9521
	80%	27.64	22.05	19.80	28.74	29.08	0.9109	0.7709	0.6721	0.9445	0.9338
	90%	25.09	18.10	15.85	25.32	27.00	0.8420	0.5807	0.4811	0.8765	0.8944
Airplane	70%	27.55	25.03	26.16	27.53	28.10	0.9276	0.8630	0.9010	0.9476	0.9278
	80%	25.99	22.71	22.61	25.70	27.10	0.8857	0.7787	0.7938	0.9064	0.9042
	90%	23.59	19.51	19.11	23.07	25.49	0.8059	0.6256	0.6261	0.8069	0.8584

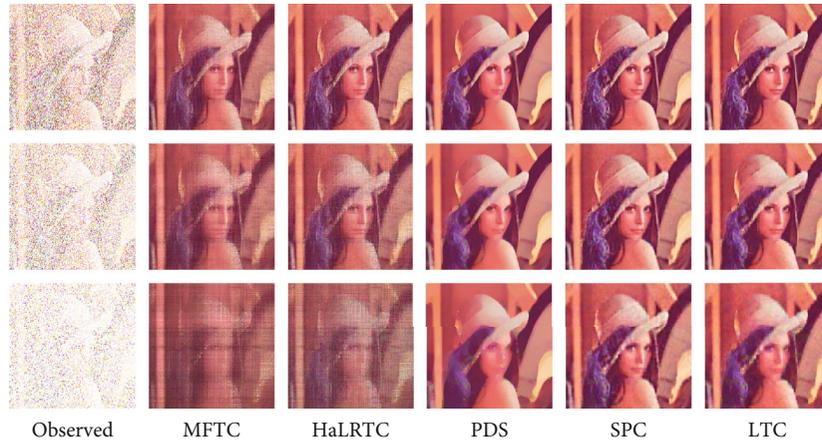


FIGURE 2: The recovered color image by five tensor completion methods with MR = 70%, 80%, 90%.



FIGURE 3: The different recovered color images by five tensor completion methods with MR = 80%.

4. Numerical Experiments

In this part, several experiments are conducted on one synthetic and some visual data to demonstrate the performance

of our proposed LTC scheme. We also compare it with another four state-of-the-art algorithms: MFTC [13], HaLRTC [12], LRTV-PDS [21], and SPC [22]. Here, we notice that MFTC and HaLRTC only use Tucker-rank to

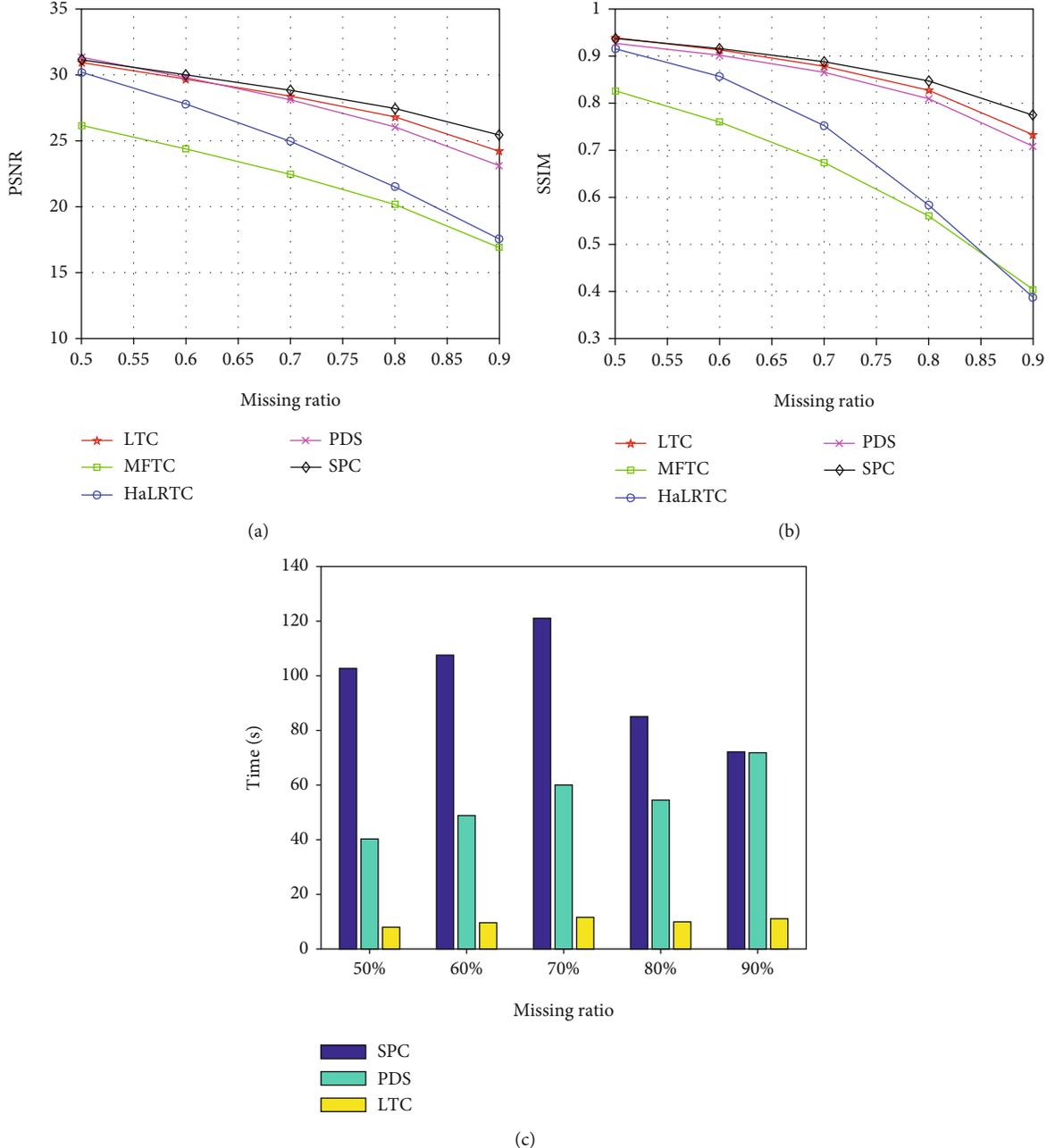


FIGURE 4: Comparison of performances by five LRTC methods on the color image. (a) Comparison of PSNR, (b) comparison of SSIM, and (c) comparison of running time.

exploit the low-rank prior, while LRTV-PDS and SPC use Tucker-rank and CP-rank, respectively. Besides, both of them are expected to achieve better results on visual data for considering TV term as the smoothness constraint. To measure the estimated accuracy of completed tensor data, peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) are employed as quality metrics. The PSNR is defined as $10 \log_{10}(\text{MAX}^2/\text{MSE})$. Here, MAX is the max of whole data, and MSE is formulated as $\|\hat{\mathcal{X}} - \mathcal{X}_0\|_F^2/N$. The SSIM estimates the resemblance of images in luminance, contrast, and structure. The relative change is defined as $\text{RelCha} = \|\mathcal{X}^{k+1} - \mathcal{X}^k\|_F / \|\mathcal{X}\|_F$, which is set as stopping cri-

terion of all five algorithms. In the following experiments, if the relative change is smaller than the tolerance 10^{-5} during the iteration, we terminate the iterative process and output \mathcal{X}^k as the recovered tensor.

4.1. Synthetic Data Completion. Here, we apply our LTC scheme and MFTC on synthetic data.

Figure 1 displays a visualization of a synthetic tensor and the corresponding results of tensor completion achieved by MFTC and LTC. We use four Gaussian functions to get the synthetic 3rd-order tensor. By randomly removing 80% of the voxels, we get the incomplete tensor. Compared to the

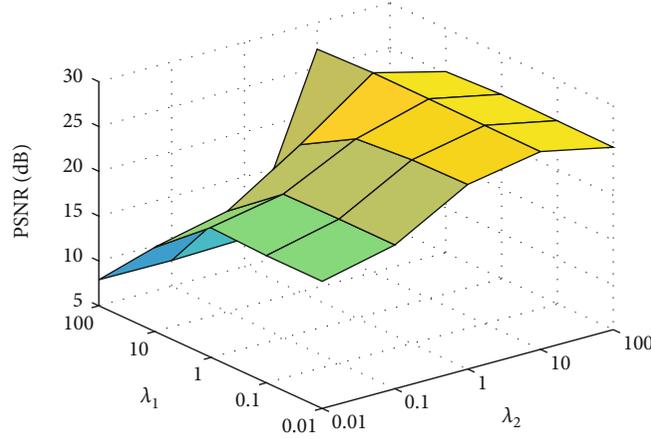


FIGURE 5: The PSNR values with different values of parameters λ_1 and λ_2 .

results of MFTC and LTC, it is obvious that only low-rank assumption is not enough for highly missing data, and smoothness constraints can effectively improve the performance of tensor completion.

4.2. Color Images Completion. In this part, we compare the performance of MFTC, HaLRTC, PDS, SPC, and the proposed LTC on four color images “Lena,” “Giant,” “Pepper,” and “Airplane.” The incomplete images are generated by randomly deleting elements. In color image data completion, the missing ratios (MRs) are set as 70%, 80%, and 90%. The quality metrics of recovered images completed by five methods are listed in Table 1. From Table 1, it is obvious that LTC achieves high quality metrics values and performs better than other LRTC algorithms except SPC.

Figure 2 presents the results of color image “Lena” recovered by MFTC, HaLRTC, PDS, SPC, and LTC with different missing ratios. The MRs are set as 70%, 80%, and 90%. From Figure 2, it can be observed that the quality of the completed images by MFTC and HaLRTC becomes worse as the missing ratio increases, while the results obtained by PDS, SPC, and LTC still contain most information of the original image.

Figure 3 presents the recovered color images “Lena,” “Giant,” “Pepper,” and “Airplane” completed by MFTC, HaLRTC, PDS, SPC, and LTC with MR = 80%. Obviously, we can see that methods with smoothness constraints (PDS, SPC, and LTC) perform better than those only considering low-rank prior (MFTC, HaLRTC) for different color images.

Figure 4 shows performance comparison of color image “Lena” by MFTC, HaLRTC, PDS, SPC, and LTC in PSNR, SSIM, and running time. The MRs are set as 50%, 60%, 70%, 80%, and 90%, respectively. We can see that PDS, SPC, and LTC are superior to MFTC and HaLRTC in terms of PSNR and SSIM. With MR increasing, the advantage of methods with smoothness constraints becomes more prominent. While ensuring the performance, LTC achieves about 80% running time reduction compared to PDS and SPC, which implies that introducing matrix factorization indeed improves the efficiency of tensor completion.

In order to analyze the effect of parameters λ_1 and λ_2 , we evaluate the performance of the recovered color image

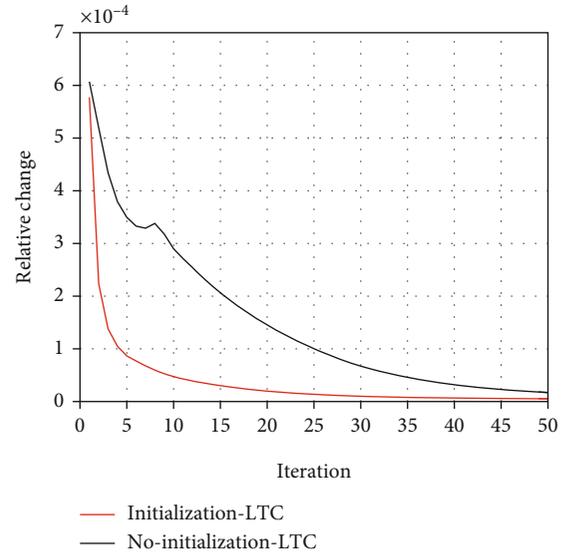


FIGURE 6: The convergence behavior of LTC with or without tensor initialization to show the effect of smoothness constraints in tensor completion.

“Lena” by LTC with MR = 70%. Figure 5 shows the change of the PSNR values for different values of λ_1 and λ_2 . It is observed that both λ_1 and λ_2 evidently effect the tensor completed results of our proposed LTC scheme. Additionally, the LTC achieves higher PSNR values when $\lambda_1 = 10$, $\lambda_2 = 10$.

To compare the convergence behavior of LTC with or without tensor initialization, we display the relative change (RelCha) values of the recovered color image “Lena” with MR = 90% in Figure 6. It is observed that the RelCha value of LTC with tensor initialization decreases faster than LTC without tensor initialization, which proves the effectiveness of our proposed tensor initialization method to accelerate the convergence speed.

4.3. Video Completion. Here, two videos: “suzie” and “hall” are tested, and only the first 30 frames of each video are used

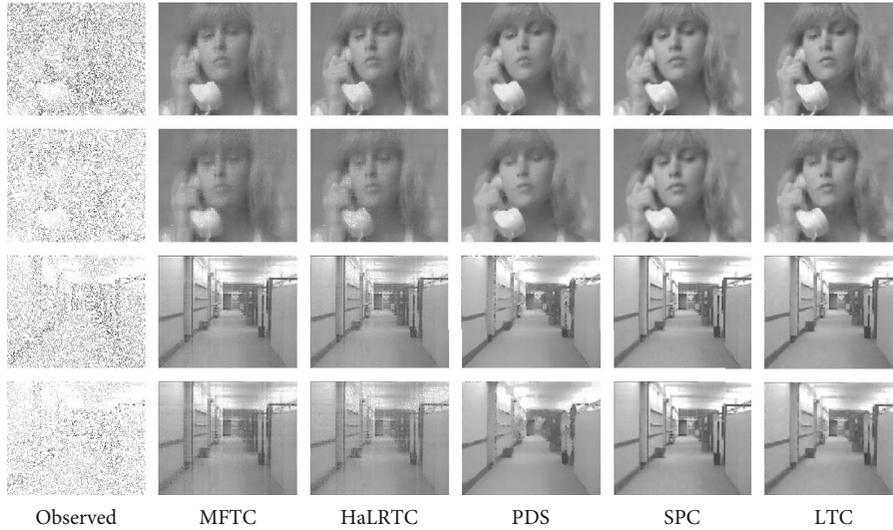


FIGURE 7: The single frame of recovered videos by five tensor completion methods with MR = 70%, 80%.

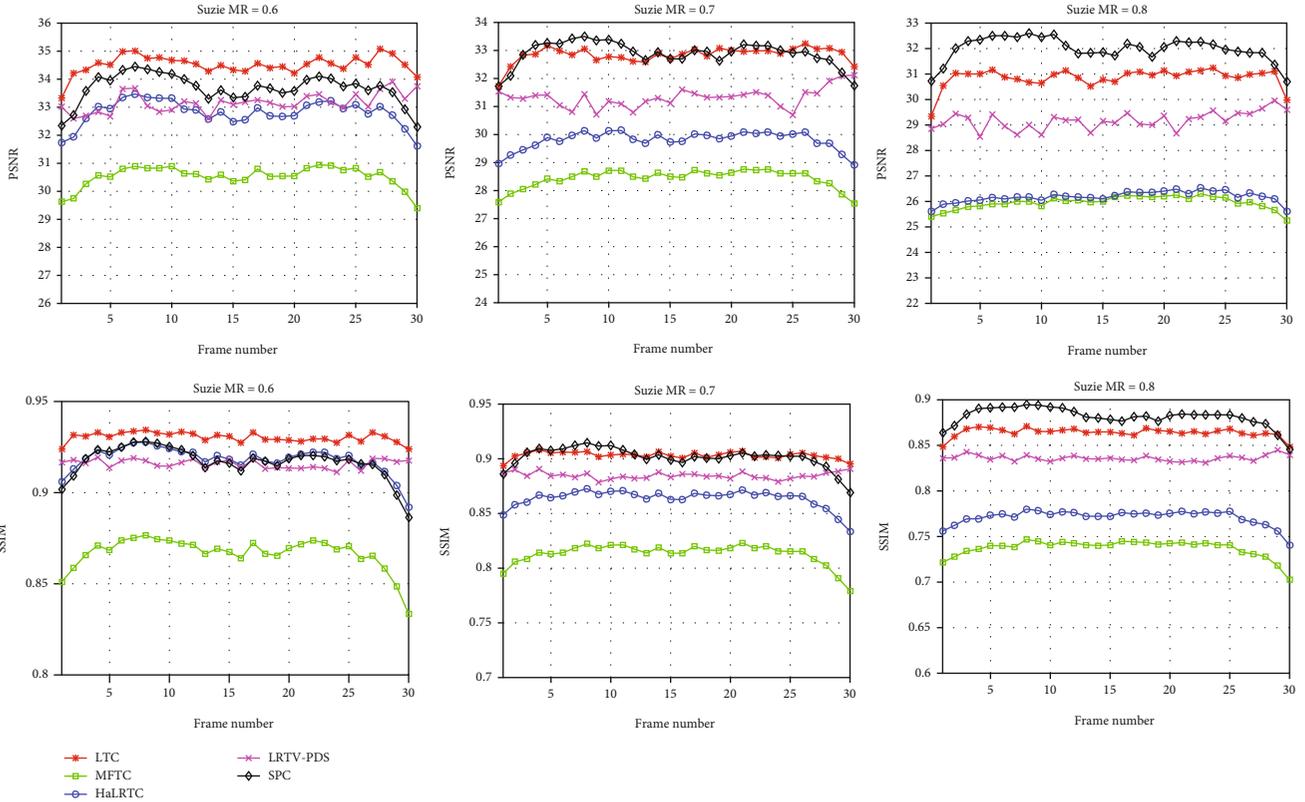


FIGURE 8: The PSNR and SSIM values by five tensor completion methods with MR = 60%, 70%, 80%.

for tensor completion. The size of videos for LRTC is $144 \times 176 \times 30$.

Figure 7 shows the 10th frame in two videos recovered by five tensor completion methods with MRs = 70% and 80%, which presents the visual results of recovered videos. Obviously, the videos recovered by PDS, SPC, and LTC are visually superior to MFTC and HaLRTC. Definitely, SPC and LTC has a better performance to complete the video for pre-

serving details well, while some parts of recovered video by PDS are still blurry.

Figure 8 presents the PSNR and SSIM values of each frame of completed video “suzie” by five algorithms with MRs = 60%, 70% and 80%, which shows the performance comparison more straightforward. From curves in Figure 8, we can conclude that SPC and LTC are better than other algorithms in both stability and effectiveness.

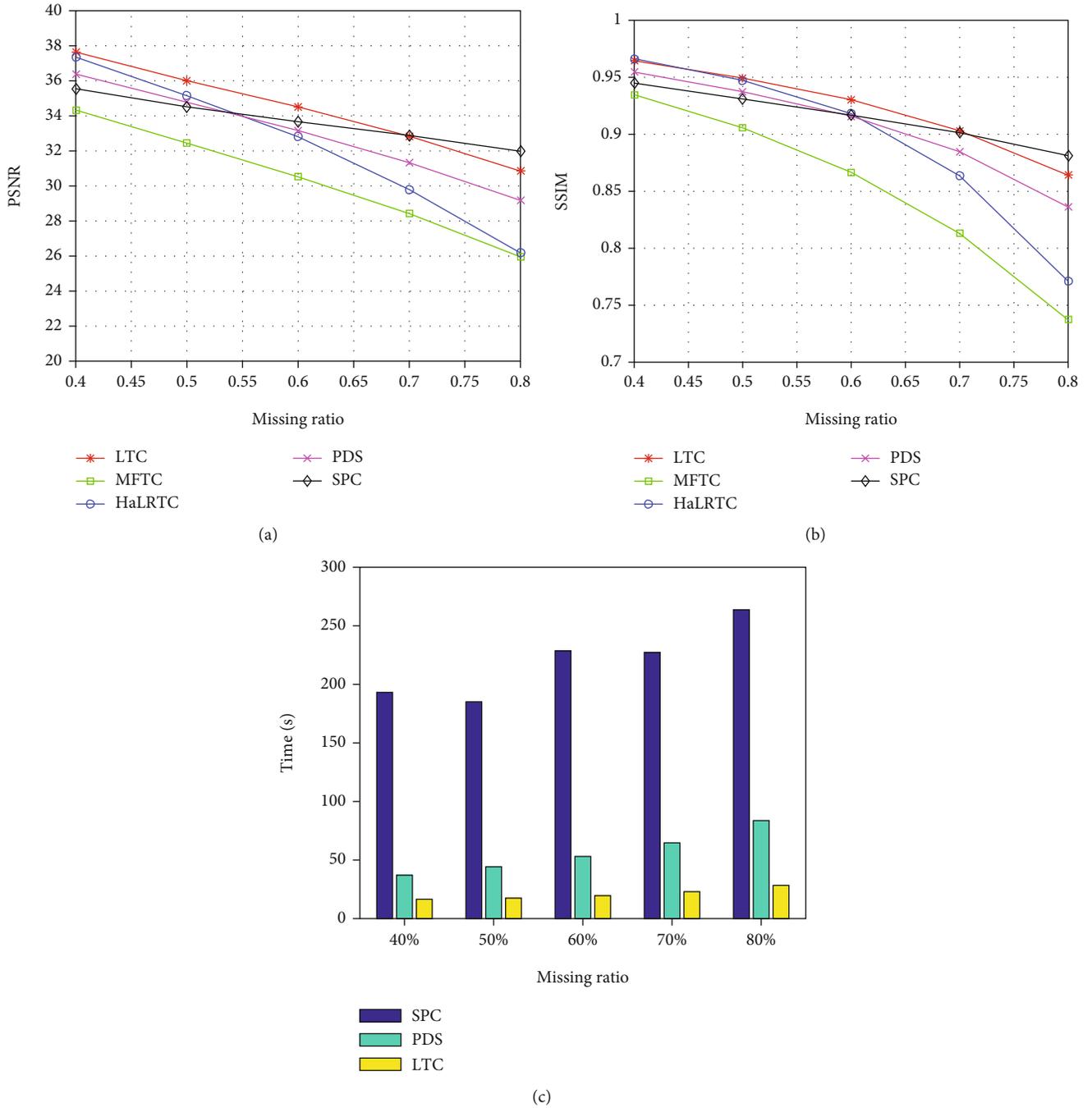


FIGURE 9: Comparison of performances by five LRTC methods on the video. (a) Comparison of PSNR, (b) comparison of SSIM, and (c) comparison of running time.

Figure 9 shows performance comparison of tensor completion for video “suzie” by our proposed LTC and the comparing four algorithms in PSNR, SSIM, and running time. The MRs are set as 40%, 50%, 60%, 70%, and 80%, respectively. Apparently, PDS, SPC, and LTC perform better than MFTC and HaLRTC when the MR is high. While ensuring the performance, LTC achieves about 90% and 50% running time reduction compared to SPC and PDS, respectively. It means that introducing matrix factorization can effectively reduce the running time of recovering incomplete videos.

5. Conclusion

In this paper, we propose a low-complexity tensor completion scheme. Our model takes advantage of SNN to exploit the low-rankness, TV and framelet to recover details and characterize the smoothness, and matrix factorization to improve the efficiency. Besides, a novel tensor initialization method is proposed to accelerate convergence speed. Moreover, an efficient ADMM-based algorithm is developed to solve the SMF model. The numerical experiments on

synthetic and real-world data demonstrate both the efficiency and effectiveness of proposed LTC scheme for tensor completion. Smoothing constraints for higher-dimensional data will be the research direction of tensor completion in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the China NSF Grants (61971217, 61971218, and 61631020), Jiangsu NSF Grant (BK20200444), the fund of Sonar Technology Key Laboratory (Research on the theory and algorithm of signal processing for two-dimensional underwater acoustics coprime array) and the fund of Sonar Technology Key Laboratory (Range estimation and location technology of passive target via multiple array combination), and Jiangsu Key Research and Development Project (BE2020101).

References

- [1] N. D. Sidiropoulos, L. de Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.
- [2] J. He, L. Li, and T. Shu, "Sparse nested arrays with spatially spread orthogonal dipoles: high accuracy passive direction finding with less mutual coupling," *IEEE Transactions on Aerospace and Electronic Systems*, 2021.
- [3] X. Wang, M. Huang, and L. Wan, "Joint 2D-DOD and 2D-DOA estimation for coprime EMVS–MIMO radar," *Circuits, Systems, and Signal Processing*, pp. 1–17, 2021.
- [4] X. Wang, L. T. Yang, D. Meng, M. Dong, K. Ota, and H. Wang, "Multi-UAV cooperative localization for marine targets based on weighted subspace fitting in SAGIN environment," *IEEE Internet of Things Journal*, 2021.
- [5] F. Wen, J. Shi, and Z. Zhang, "Closed-form estimation algorithm for EMVS-MIMO radar with arbitrary sensor geometry," *Signal Processing*, vol. 186, article 108117, 2021.
- [6] Y. Wang, J. Peng, Q. Zhao, Y. Leung, X.-L. Zhao, and D. Meng, "Hyperspectral image restoration via total variation regularized low-rank tensor decomposition," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 4, pp. 1227–1243, 2018.
- [7] T. Yokota, N. Lee, and A. Cichocki, "Robust multilinear tensor rank estimation using higher order singular value decomposition and information criteria," *IEEE Transactions on Signal Processing*, vol. 65, no. 5, pp. 1196–1206, 2017.
- [8] M. Zhou, Y. Liu, Z. Long, L. Chen, and C. Zhu, "Tensor rank learning in CP decomposition via convolutional neural network," *Signal Processing: Image Communication*, vol. 73, pp. 12–21, 2019.
- [9] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, article 025010, 2011.
- [10] J. Douglas Carroll, S. Pruzansky, and J. B. Kruskal, "Candelinc: a general approach to multidimensional analysis of many-way arrays with linear constraints on parameters," *Psychometrika*, vol. 45, no. 1, pp. 3–24, 1980.
- [11] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
- [12] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013.
- [13] Y. Liu and F. Shang, "An efficient matrix factorization method for tensor completion," *IEEE Signal Processing Letters*, vol. 20, no. 4, pp. 307–310, 2013.
- [14] Z. Wang and C. Ling, "Lattice Gaussian sampling by Markov chain Monte Carlo: bounded distance decoding and trapdoor sampling," *IEEE Transactions on Information Theory*, vol. 65, no. 6, pp. 3630–3645, 2019.
- [15] Z. Wang and C. Ling, "On the geometric ergodicity of Metropolis-Hastings algorithms for lattice Gaussian sampling," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 738–751, 2018.
- [16] Q. Yuan, L. Zhang, and H. Shen, "Hyperspectral image denoising employing a spectral-spatial adaptive total variation model," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 10, pp. 3660–3677, 2012.
- [17] Y. Liu, S. Wu, X. Huang, B. Chen, and C. Zhu, "Hybrid CS-DMRI: periodic time-variant subsampling and omnidirectional total variation based reconstruction," *IEEE Transactions on Medical Imaging*, vol. 36, no. 10, pp. 2148–2159, 2017.
- [18] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.
- [19] X. Li, Y. Ye, and X. Xu, "Low-rank tensor completion with total variation for visual data inpainting," in *AAAI'17: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, 2017.
- [20] Y. Chen, S. Wang, and Y. Zhou, "Tensor nuclear norm-based low-rank approximation with total variation regularization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1364–1377, 2018.
- [21] T. Yokota and H. Hontani, "Simultaneous tensor completion and denoising by noise inequality constrained convex optimization," *IEEE Access*, vol. 7, pp. 15669–15682, 2019.
- [22] T. Yokota, Q. Zhao, and A. Cichocki, "Smooth PARAFAC decomposition for tensor completion," *IEEE Transactions on Signal Processing*, vol. 64, no. 20, pp. 5423–5436, 2016.
- [23] D. C. Dobson and F. Santosa, "Recovery of blocky images from noisy and blurred data," *SIAM Journal on Applied Mathematics*, vol. 56, no. 4, pp. 1181–1198, 1996.
- [24] J. F. Cai, R. H. Chan, and Z. Shen, "A framelet-based image inpainting algorithm," *Applied and Computational Harmonic Analysis*, vol. 24, no. 2, pp. 131–149, 2008.
- [25] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
- [26] A. Ron and Z. Shen, "A ne systems in L2 (IRd): the analysis of the analysis operator," *J. Funct. Anal.*, 1995.

- [27] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Mathematical Programming Computation*, vol. 4, no. 4, pp. 333–361, 2012.
- [28] Y. Shen, Z. Wen, and Y. Zhang, "Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization," *Optimization Methods and Software*, vol. 29, no. 2, pp. 239–263, 2014.