

Tensor train rank minimization with hybrid smoothness regularization for visual data recovery



Jing-Hua Yang^{a,b}, Xi-Le Zhao^{b,*}, Tian-Hui Ma^c, Meng Ding^b, Ting-Zhu Huang^b

^a Department of Arts and Sciences, Chengdu College of University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, PR China

^b School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, PR China

^c School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, PR China

ARTICLE INFO

Article history:

Received 17 April 2019

Revised 31 December 2019

Accepted 14 January 2020

Available online 21 January 2020

Keywords:

Low-rank tensor completion

Total variation

Framelet

Tensor train

Proximal alternating minimization

ABSTRACT

Recently, the tensor train (TT) rank has received much attention for tensor completion, due to its ability to explore the global low-rankness of tensors. However, existing methods still leave room for improvement, since the low-rankness itself is generally not sufficient to recover the underlying data. Inspired by this, we consider a novel tensor completion model by simultaneously exploiting the global low-rankness and local smoothness of visual data. In particular, we use low-rank matrix factorization to characterize the global TT low-rankness, and framelet and total variation regularization to enhance the local smoothness. We develop an efficient proximal alternating minimization algorithm to solve the proposed new model with guaranteed convergence. Extensive experiments on various data demonstrated that the proposed method outperforms compared methods in terms of visual and quantitative measures.

© 2020 Elsevier Inc. All rights reserved.

1. Introduction

As the high-dimensional generalization of vectors and matrices, tensors play an increasingly significant role in computer vision. However, during the acquisition process, observed tensor data often contain missing entries, which severely degrades the data quality and limits the processing tasks. Therefore, tensor completion is a fundamental and important problem in high-dimensional data processing, such as color image and video completion [1–4], hyperspectral images recovery [5,6], and seismic data reconstruction [7].

The low-rank regularization is a powerful tool for tensor completion, namely low-rank tensor completion (LRTC). Mathematically, the LRTC issue is modeled as

$$\begin{aligned} \min_{\mathcal{M}} \quad & \text{rank}(\mathcal{M}), \\ \text{s.t.} \quad & P_{\Omega}(\mathcal{M}) = P_{\Omega}(\mathcal{T}), \end{aligned} \tag{1}$$

where $\mathcal{M} \in R^{n_1 \times \dots \times n_l}$ denotes the underlying tensor, $\mathcal{T} \in R^{n_1 \times \dots \times n_l}$ denotes the observed tensor, and $P_{\Omega}(\cdot)$ is the projection operator on Ω , which is the index of observed entries. A central issue in LRTC is the characterization of the low-rankness

* Corresponding author.

E-mail addresses: yangjinghua110@126.com (J.-H. Yang), xlzhao122003@163.com (X.-L. Zhao), nkmth0307@126.com (T.-H. Ma), dingmeng56@163.com (M. Ding), tingzhuhuang@126.com (T.-Z. Huang).

of tensors. Below we briefly review several representative works on this topic, including CANDECOMP/PARAFA (CP) rank [8], Tucker rank [9], tubal rank [10], and tensor train (TT) rank [11].

1.1. The Related Work

The CP rank [8] denotes the smallest number of rank-one tensors, which generate the target tensor as their sum. Several heuristic methods have been proposed for CP rank minimization [12–14]. However, the computation of the CP-rank is NP-hard [15], which limits the application to practical problems.

The Tucker rank [16] denotes a vector composed of unfolding matrices ranks of the target tensor. For an l th-order tensor $\mathcal{M} \in R^{n_1 \times n_2 \times \dots \times n_l}$, its Tucker rank [8] is given by

$$\text{rank}_{\text{tc}}(\mathcal{M}) := (\text{rank}(M_{(1)}), \text{rank}(M_{(2)}), \dots, \text{rank}(M_{(l)})),$$

where $M_{(i)} \in R^{n_i \times \prod_{k \neq i} n_k}$ is the mode- i unfolding of \mathcal{M} . There exist several methods for tucker rank minimization. Liu et al. [17] proposed to replace the nonconvex rank function with its convex surrogate the sum of nuclear norms (SNN) and formulated tensor completion as a convex optimization problem. Because of calculating the singular value decomposition of unfolding matrices in every iteration, minimizing the SNN suffers from high computational cost for some large-scale problems. To reduce the computational complexity, Xu et al. [18] proposed a LRTC model by performing low-rank matrix factorization to each unfolding matrix of the underlying tensor. However, the Tucker rank only explores the correlation between one mode and rest modes of the tensor [19], owing to the unbalanced unfolding scheme.

The tensor tubal rank is the number of nonzero singular tubes of the target tensor [10]. More precisely, a tensor $\mathcal{M} \in R^{n_1 \times n_2 \times n_3}$ is decomposed

$$\mathcal{M} = \mathcal{U} * \mathcal{S} * \mathcal{V}, \quad (2)$$

where $\mathcal{S} \in R^{n_1 \times n_2 \times n_3}$ is an f -diagonal¹ tensor containing the singular tubes of \mathcal{M} , $\mathcal{U} \in R^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in R^{n_2 \times n_2 \times n_3}$ are orthogonal tensors. Zhang et al. [20] applied the tensor tubal rank to LRTC by minimizing its convex surrogate tensor nuclear norm (TNN). However, the tensor tubal rank is not much effective in characterizing the correlation within higher-order tensors.

Recently, the TT rank [11] has shown a powerful capacity for characterizing the correlations between different modes in higher-order tensors. For a higher-order tensor $\mathcal{M} \in R^{n_1 \times n_2 \times \dots \times n_l}$, the TT rank [21] is

$$\text{rank}_{\text{tt}}(\mathcal{M}) := (\text{rank}(M_{[1]}), \text{rank}(M_{[2]}), \dots, \text{rank}(M_{[l-1]})),$$

where $M_{[i]} \in R^{\prod_{k=1}^i n_k \times \prod_{k=i+1}^l n_k}$ denotes the mode-(1, 2, …, k) canonical matricization of \mathcal{M} . TT rank is built on a well balanced matricization scheme and can complement the correlations between a few modes and the rest modes compared to Tucker rank. Due to its desired performance, Bengua et al. [19] proposed two TT rank-based optimization methods to handle the LRTC problem. The first method minimizes the convex nuclear norm surrogate of TT rank (SiLRTC-TT), i.e.,

$$\begin{aligned} \min_{\mathcal{M}} \quad & \sum_{i=1}^{l-1} \alpha_i \|M_{[i]}\|_*, \\ \text{s.t.} \quad & P_{\Omega}(\mathcal{M}) = P_{\Omega}(\mathcal{T}), \end{aligned} \quad (3)$$

where α_i are positive satisfying $\sum_{i=1}^{l-1} \alpha_i = 1$. The second method factorizes each $M_{[i]}$ into the product of two smaller matrices (TMac-TT), i.e.,

$$\begin{aligned} \min_{\mathcal{M}, \mathcal{X}_i, \mathcal{Y}_i} \quad & \sum_{i=1}^{l-1} \frac{\alpha_i}{2} \|X_i Y_i - M_{[i]}\|_F^2, \\ \text{s.t.} \quad & P_{\Omega}(\mathcal{M}) = P_{\Omega}(\mathcal{T}), \end{aligned} \quad (4)$$

where $X_i \in R^{(\prod_{k=1}^i n_k) \times r_i}$ and $Y_i \in R^{r_i \times (\prod_{k=i+1}^l n_k)}$ are factor matrices, and r_i denotes the matrix rank of $M_{[i]}$. The results restored by SiLRTC-TT and TMac-TT suffer from serious block-artifacts, due to these two methods borrowed *ket augmentation* (KA) (see Section 2.3 for more details) to transform a lower-order tensor to a higher-order tensor. In order to tackle this issue, Ding et al. [22] considered to reduce the block-artifacts by introducing the total variation (TV) regularizer, which often suffers from artifacts.

1.2. Motivations and contributions

The above methods still leave room for further improvement, since the low-rankness itself is generally not sufficient to recover the underlying data. As another significant prior, smoothness appears widely in real-world high-dimensional data. For example, a video exhibits smoothness in spatial dimensions and temporal smoothness due to the little change between

¹ A tensor is defined f -diagonal if all frontal slices are diagonal matrixes.

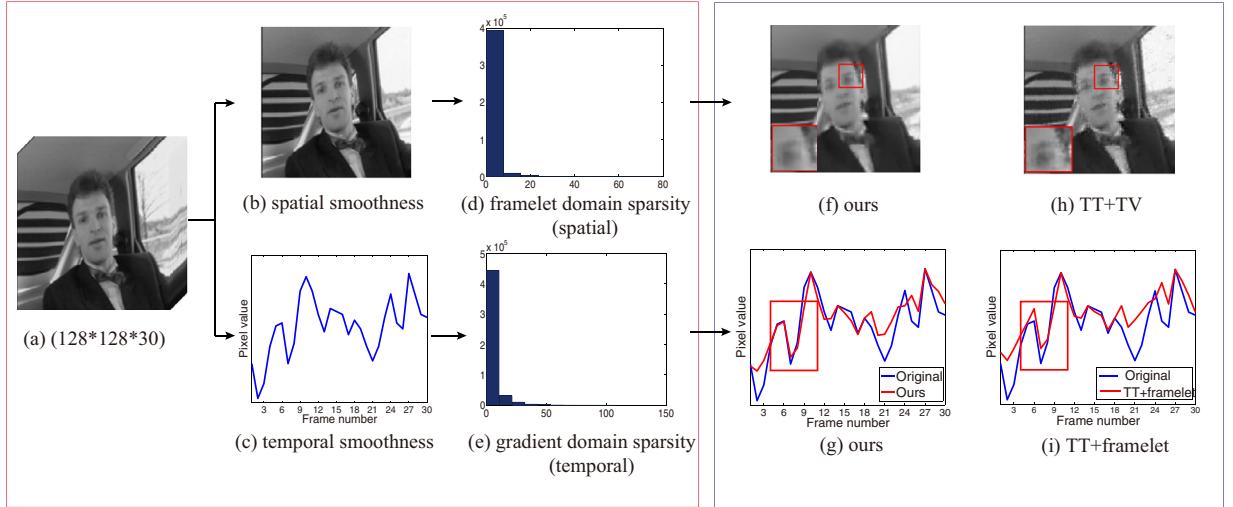


Fig. 1. Motivations and effectiveness of the proposed method. (a) The original tensor *carphone*, (b) and (c) the smoothness in the spatial and temporal dimensions, (d) and (e) the sparsity in the framelet domain and the gradient domain, (f) and (h) frames of the reconstructed results with SR = 0.1 by the proposed method and TT+TV (without framelet regularization in the proposed model), (g) and (i) the intensity of a tube of the restored video by the proposed method and TT+framelet (without TV regularization in the proposed model).

adjacent frames. Inspired by this nature, we build a low-rank tensor completion model to simultaneously exploit the global low-rankness and the local spatial-temporal smoothness. We propose the following model:

$$\begin{aligned} \min_{X, Y, \mathcal{M}} \quad & \sum_{i=1}^{l-1} \frac{\alpha_i}{2} \|\mathcal{K}(\mathcal{M})_{[i]} - X_i Y_i\|_F^2 + \lambda_1 \|WM_{(3)}^T\|_{1,1} + \lambda_2 \|D_s M_{(3)}\|_{1,1}, \\ \text{s.t.} \quad & P_\Omega(\mathcal{M}) = P_\Omega(\mathcal{T}), \end{aligned} \quad (5)$$

where \mathcal{M} is the underlying tensor, \mathcal{K} is the KA operator, $X = (X_1, X_2, \dots, X_{l-1})$ and $Y = (Y_1, Y_2, \dots, Y_{l-1})$ are factor matrices. In the model (5), the first term is the low-rank matrix factorizations for characterizing the global TT low-rankness of the tensor. The second term is the framelet regularization to capture the abundant details. And the third term is to enhance the temporal smoothness of the tensor by the TV regularization (see Section 3.1 for details). Compared with the model in [22], our model takes into account not only the superior performance of the framelet regular to TV, but also the consideration of the temporal smoothness.

To illustrate our motivation, we test the video *carphone* with the sampling rate (SR) = 0.1 in Fig. 1 as an example. We obtain three insights from Fig. 1. First, the visual data show the smoothness prior in the spatial and temporal dimensions; see the sparse approximation shown in Fig. 1(d) and (e). Second, the framelet prior can preserve details and promote spatial smoothness of the underlying tensor; see Fig. 1(f) and (h). Third, TV prior is necessary to keep the smoothness of the tensor in the temporal dimension; see Fig. 1(g) and (i), where the intensity of a tube of the recovered result in the proposed method is smoother and nearer to the original one.

The contribution of our work is summarized into three parts. First, a novel tensor completion model is proposed by simultaneously exploiting the global low-rankness and local smoothness of visual data. Second, we develop a proximal alternating minimization (PAM) algorithm for solving the proposed model with global convergence analysis. Third, extensive numerical experiments demonstrate the outperformance of the proposed method over the other compared methods.

There are five sections in this paper. Section 2 presents some preliminary knowledge. Section 3 shows the proposed model and the PAM algorithm with guaranteed convergence. Section 4 provides experimental results. Section 5 concludes this paper.

2. Preliminaries

2.1. Tensor basics

We use capital letters to denote matrices (e.g., M) and calligraphic letters to denote tensors (e.g., \mathcal{M}). A tensor is formed from a high-dimensional array and the number of tensor dimensions denotes its mode. Given an l th-order tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_l}$, some notations are given below.

The Frobenius norm of \mathcal{M} is $\|\mathcal{M}\|_F := \sqrt{\sum_{n_1} \sum_{n_2} \dots \sum_{n_l} x_{n_1 n_2 \dots n_l}^2}$, where $x_{n_1 n_2 \dots n_l}$ is the (n_1, n_2, \dots, n_l) -th element of $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times \dots \times n_l}$.

The mode- i unfolding of \mathcal{M} denotes the process to unfold the tensor into a matrix $M_{(i)} \in R^{n_i \times \prod_{k \neq i} n_k}$ [8]. The Tucker rank of \mathcal{M} is $\mathbf{r} = (M_{(1)}, M_{(2)}, \dots, M_{(l)})$.

The mode- $(1, 2, \dots, k)$ canonical matricization of \mathcal{M} is denoted as $M_{[i]} \in R^{\prod_{k=1}^i n_k \times \prod_{k=i+1}^l n_k}$ [11,21]. The matrix $M_{[i]}$ could be formed by the following reshape function in MATLAB, i.e.,

$$M_{[i]} = \text{reshape}(\mathcal{M}, \prod_{k=1}^i n_k, \prod_{k=i+1}^l n_k). \quad (6)$$

We denote “unreshape”, i.e., $\text{unreshape}_{[i]}(M_{[i]}) := \mathcal{M}$, as the inverse operator of reshape. The TT rank of \mathcal{M} is $\mathbf{r} = (M_{[1]}, M_{[2]}, \dots, M_{[l-1]})$. The detailed introduction of TT can be seen in [11].

2.2. Framelet

In this subsection, we introduce briefly the framelet system. We mainly show the univariate framelets, because of the bivariate framelets can be computed from univariate framelets [23]. If any $f \in L_2(R)$ has

$$f = \sum_{g \in X} \langle f, g \rangle g,$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product of $L_2(R)$, and the system $X \subset L_2(R)$ is named a tight frame of $L_2(R)$. For a finite set $\Psi = \{\psi_1, \psi_2, \dots, \psi_r\} \subset L_2(R)$, a wavelet (also called affine) system $X(\Psi)$ is defined as

$$X(\Psi) = \{2^{k/2} \psi_t(2^k \cdot -j) : \psi_t \in \Psi; 1 \leq t \leq r; k, j, t \in Z\},$$

where Z represents an integer set. Particularly, each function ψ_t ($t = 1, 2, \dots, r$) are called a (tight) framelets if $X(\Psi)$ is a (tight) frame for $L_2(R)$, and the all system $X(\Psi)$ is named a (tight) wavelet frame system.

We give the discrete form of framelet. The L -level framelet decomposition of a discrete image f is $\{f = \langle f, 2^{-L/2} \psi_t(2^{-L} \cdot -j) \rangle\}$. If the discrete image $f \in R^{mn}$ is a vector, the decomposition coefficients could be considered as a linear operator Wf with $W \in R^{k \times mn}$ and $W^T W = I$, where W^T denotes the inverse framelet operator. Thus, a tight frame system in R^{mn} can be formed from the row vectors of W . For the tensor cases, we test in the bivariate case. We can obtain the corresponding transform matrix by the Kronecker product (see [24] for details). In the following, we still employ W to denote the bivariate framelet transform.

2.3. KA

KA [19] is a tensor augmentation tool. The augmented tensor by using KA explores the low-rank structure more obviously than the original one. As a result, KA is a helpful pretreatment step for TT rank-based optimization.

The original KA is designed for casting a grey scale image to a higher-order tensor by a block structured way [25]. Recently, KA has been extended to the lower-order tensors, such as third-order tensors. For instance, KA reshapes a color image $\mathcal{M} \in R^{m_1 \times m_2 \times 3}$ ($m_1 \times m_2 = 2^n \times 2^n$ is the pixel number in the image) into an l th-order tensor $\tilde{\mathcal{M}} \in R^{n_1 \times n_2 \times \dots \times n_l}$, where $l \geq 3$, $3m_1m_2 = \prod_{d=1}^l n_d$, and n_d is a block structured processing.

The details of the structured block addressing procedure of KA is introduced as follows. First, we consider the initial smallest block (labeled as i_1) with size 2×2 related to the color channel j and represent this block as $\tilde{\mathcal{M}} = \sum_{i_1=1}^4 c_{i_1 j} \mathbf{e}_{i_1}$, where $c_{i_1 j}$ is the pixel value related to the color channel j and \mathbf{e}_{i_1} is the orthonormal base with $\mathbf{e}_1 = (1, 0, 0, 0)$, $\mathbf{e}_2 = (0, 1, 0, 0)$, $\mathbf{e}_3 = (0, 0, 1, 0)$, and $\mathbf{e}_4 = (0, 0, 0, 1)$. Then, the block with size $2 \times 2 \times 3$ corresponding to three color channels can be written as $\tilde{\mathcal{M}} = \sum_{i_1=1}^4 \sum_{j=1}^3 c_{i_1 j} \mathbf{e}_{i_1} \otimes \mathbf{u}_j$, where \mathbf{u}_j is also an orthonormal base with $\mathbf{u}_1 = (1, 0, 0)$, $\mathbf{u}_2 = (0, 1, 0)$, and $\mathbf{u}_3 = (0, 0, 1)$ and \otimes denotes the Kronecker product, see Fig. 2(a). Next, we consider a larger block (labeled as i_2) with size $4 \times 4 \times 3$ as shown in Fig. 2(b). Similarly, we have $\tilde{\mathcal{M}} = \sum_{i_2=1}^4 \sum_{i_1=1}^4 \sum_{j=1}^3 c_{i_2 i_1 j} \mathbf{e}_{i_2} \otimes \mathbf{e}_{i_1} \otimes \mathbf{u}_j$. This block addressing procedure is performed step by step until it can present all the pixels in the image. Finally, the general form of $\tilde{\mathcal{M}}$ is given by

$$\tilde{\mathcal{M}} = \sum_{i_1, \dots, i_n=1}^4 \sum_{j=1}^3 c_{i_n, \dots, i_1 j} \mathbf{e}_{i_n} \otimes \dots \otimes \mathbf{e}_{i_1} \otimes \mathbf{u}_j. \quad (7)$$

An extensive overview can be found in [19].

The processing of transforming a k th order tensor $\mathcal{M} \in R^{m_1 \times m_2 \times \dots \times m_k}$ into an l th order tensor $\tilde{\mathcal{M}} \in R^{n_1 \times n_2 \times \dots \times n_l}$ using KA can be expressed as a linear operator $\mathcal{K} : R^{m_1 \times m_2 \times \dots \times m_k} \rightarrow R^{n_1 \times n_2 \times \dots \times n_l}$, where $l \geq k$ and $\prod_{i=1}^k m_i = \prod_{j=1}^l n_j$. It is worth noting that KA just rearranges the tensor elements without changing their values, so that $\|\mathcal{M}\|_F = \|\tilde{\mathcal{M}}\|_F$.

3. Tensor completion combining TT rank minimization and smoothness

This section has two parts. Section 3.1 presents the proposed model with the discussions. Section 3.2 gives an efficient algorithm and establishes its convergence.

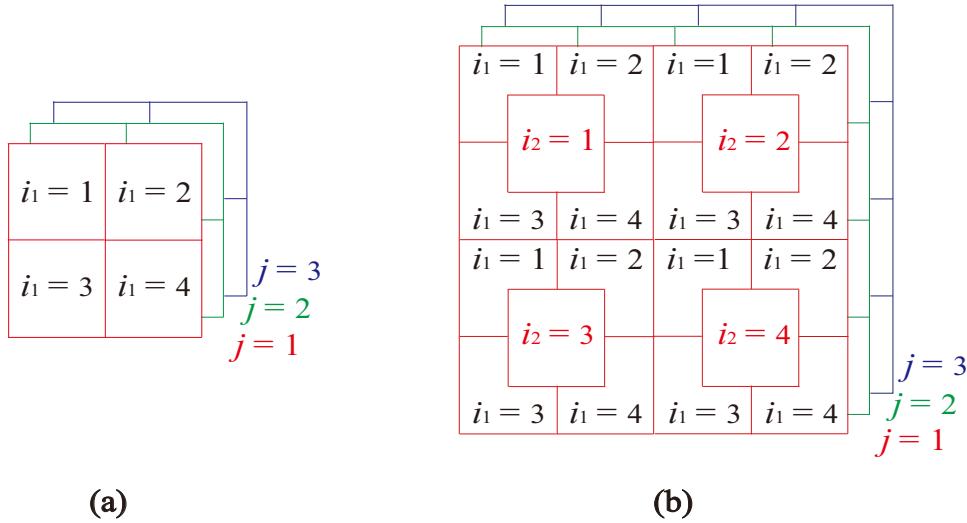


Fig. 2. Illustration of KA to cast a color image to a higher-order tensor. (a) A block with size $2 \times 2 \times 3$. (b) A block with size $2^2 \times 2^2 \times 3$.

3.1. The proposed model

We propose a tensor completion model by combining TT rank minimization with hybrid smoothness regularization as follows:

$$\begin{aligned} \min_{X, Y, \mathcal{M}} \quad & \sum_{i=1}^{l-1} \frac{\alpha_i}{2} \|\mathcal{K}(\mathcal{M})_{[i]} - X_i Y_i\|_F^2 + \lambda_1 \|WM_{(3)}^T\|_{1,1} + \lambda_2 \|D_s M_{(3)}\|_{1,1}, \\ \text{s.t. } \quad & P_\Omega(\mathcal{M}) = P_\Omega(\mathcal{T}), \end{aligned} \quad (8)$$

where $\mathcal{K}(\mathcal{M}) = \tilde{\mathcal{M}} \in R^{n_1 \times n_2 \times \dots \times n_l}$ is the augmented tensor by KA, $X = (X_1, X_2, \dots, X_{l-1})$ and $Y = (Y_1, Y_2, \dots, Y_{l-1})$ are factor matrices, α_i are positive weight parameters satisfying $\sum_{i=1}^{l-1} \alpha_i = 1$, λ_1 and λ_2 are regularization parameters. This model focuses on third-order tensors and can be generalized to high-order tensors. Below we present the details of each regularization term and discuss their functions.

The first term $\|\mathcal{K}(\mathcal{M})_{[i]} - X_i Y_i\|_F^2$ is the TT low-rank regularization term for the augmented tensor $\tilde{\mathcal{M}}$. For convenience, we rewrite it as $\|\tilde{M}_{[i]} - X_i Y_i\|_F^2$ in the following, where $\tilde{M}_{[i]}$ is the mode- i canonical matricization of $\tilde{\mathcal{M}}$, $X_i \in R^{(\prod_{k=1}^i n_k) \times r_i}$, $Y_i \in R^{r_i \times (\prod_{k=i+1}^l n_k)}$, and r_i denotes the rank of the matrix $M_{[i]}$. We adopt KA for transforming the third-order tensor into the higher-order one for enhancing the TT low-rank structure included in the original one. The term can capture the global low-rankness of the higher-order tensor well by exploring the correction of different modes.

The second term $\|WM_{(3)}^T\|_{1,1}$ is the framelet regularization term, where W denotes the framelet transformation matrix with $W^T W = I$ and $M_{(3)}$ is the mode-3 unfolding of \mathcal{M} . For $\mathcal{M} \in R^{n_1 \times n_2 \times n_3}$, $M_{(3)}^T$ can be rewritten as

$$M_{(3)}^T = ((\mathbf{m}_{(3)}^1)^T, (\mathbf{m}_{(3)}^2)^T, \dots, (\mathbf{m}_{(3)}^{n_3})^T),$$

where $\mathbf{m}_{(3)}^i \in R^{1 \times n_1 n_2}$ ($i = 1, 2, \dots, n_3$) indicates the vectorization of each frame. Since the multi-level structure of tight frame systems and the diversity of filters [26], we use the framelet regularizer for spatial smoothness since it is good at preserving abundant details of the underlying tensor.

The third term $\|D_s M_{(3)}\|_{1,1}$ is the TV regularization term, where D_s indicates the vertical derivative operator and $\|D_s M_{(3)}\|_{1,1}$ is the isotropic TV of $M_{(3)} \in R^{n_3 \times n_1 n_2}$.

$$(\mathcal{D}_f(\mathcal{M}_{(3)}))_{i,j} = \begin{cases} (M_{(3)})_{k+1,i} - (M_{(3)})_{k,i}, & 1 \leq k \leq n_3 - 1, \\ (M_{(3)})_{1,i} - (M_{(3)})_{n,i}, & k = n_3, \end{cases}$$

and

$$\|D_s M_{(3)}\|_{1,1} = \sum_{1 \leq k \leq n_3, 1 \leq i \leq n_1 n_2} |D_s M_{(3)}|_{k,i}.$$

The term has the ability to effectively keep edge information and smoothness [27–29]. Thus, this term characterizes the smoothness along the temporal dimension.

We remark that most of work consider the theoretical recoverability of matrix/tensor completion under convex setting [20,30–32]. Their results cannot be directly extended to our model (8), because of the simultaneous low-rank and sparsity structure characterization [33], and the objective function is non-convex.

3.2. The PAM solver with convergence

We employ the proximal alternating minimization (PAM) [34] to resolve the nonconvex problem (8) into three convex subproblems. Specifically, X , Y , and \mathcal{M} are iteratively computed as follows:

$$\begin{cases} X^{t+1} = \arg \min_X f(X, Y^t, \mathcal{M}^t) + \frac{\rho}{2} \|X - X^t\|_F^2, \\ Y^{t+1} = \arg \min_Y f(X^{t+1}, Y, \mathcal{M}^t) + \frac{\rho}{2} \|Y - Y^t\|_F^2, \\ \mathcal{M}^{t+1} = \arg \min_{\mathcal{M}} f(X^{t+1}, Y^{t+1}, \mathcal{M}) + \frac{\rho}{2} \|\mathcal{M} - \mathcal{M}^t\|_F^2, \end{cases} \quad (9)$$

where $f(X, Y, \mathcal{M})$ is the objective function in (8), ρ is a positive parameter, and the superscripts t denote the iteration indices. Next, we discuss more details for solving the X , Y , and \mathcal{M} subproblems.

1. (X -subproblem) Since the minimization X -subproblem with reference to each X_i is decoupled, we decomposed it into $l-1$ independent problems as follows:

$$X_i^{t+1} = \arg \min_{X_i} \frac{\alpha_i}{2} \|X_i Y_i^t - \tilde{M}_{[i]}^t\|_F^2 + \frac{\rho}{2} \|X_i - X_i^t\|_F^2. \quad (10)$$

Its closed-form solution is

$$X_i^{t+1} = (\alpha_i \tilde{M}_{[i]}^t (Y_i^t)^T + \rho I) (\alpha_i Y_i^t (Y_i^t)^T + \rho I)^{-1}, \quad (11)$$

where $I \in R^{r_i \times r_i}$ is an identity matrix. Its computational complexity is X_i is $O(r_i \prod_{k=1}^l n_k)$.

2. (Y -subproblem) Similarly, we decomposed the Y -subproblem into $l-1$ independent problems as follows:

$$Y_i^{t+1} = \arg \min_{Y_i} \frac{\alpha_i}{2} \|X_i^{t+1} Y_i - \tilde{M}_{[i]}^t\|_F^2 + \frac{\rho}{2} \|Y_i - Y_i^t\|_F^2. \quad (12)$$

Similarly, its closed-form solution is

$$Y_i^{t+1} = (\alpha_i (X_i^{t+1})^T X_i^{t+1} + \rho I)^{-1} ((\alpha_i X_i^{t+1})^T \tilde{M}_{[i]}^t + \rho Y_i^t). \quad (13)$$

The cost of computing Y_i is $O(r_i \prod_{k=1}^l n_k)$.

3. (\mathcal{M} -subproblem) The \mathcal{M} -subproblem is

$$\mathcal{M}^{t+1} = \arg \min_{P_\Omega(\mathcal{M})=P_\Omega(\mathcal{T})} \sum_{i=1}^{l-1} \frac{\alpha_i}{2} \|X_i^{t+1} Y_i^{t+1} - \tilde{M}_{[i]}\|_F^2 + \lambda_1 \|WM_{(3)}^T\|_{1,1} + \lambda_2 \|D_s M_{(3)}\|_1 + \frac{\rho}{2} \|\mathcal{M} - \mathcal{M}^t\|_F^2. \quad (14)$$

We iteratively solve the convex optimization problem (14) by the alternating direction method of multipliers (ADMM) [35]. We introduce auxiliary variables $\{\mathcal{A}_i\}_{i=1}^{l-1}$, Z , and P to transform the \mathcal{M} -subproblem into the following problem:

$$\begin{aligned} & \arg \min_{\mathcal{M}} \sum_{i=1}^{l-1} \frac{\alpha_i}{2} \|X_i^{t+1} Y_i^{t+1} - \mathcal{A}_{i[i]}\|_F^2 + \lambda_1 \|Z\|_{1,1} + \lambda_2 \|P\|_{1,1} + \frac{\rho}{2} \|\mathcal{M} - \mathcal{M}^t\|_F^2, \\ & \text{s.t. } P_\Omega(\mathcal{A}_i) = P_\Omega(\mathcal{T}), \mathcal{A}_i = \tilde{M}, Z = WM_{(3)}^T, P = D_s M_{(3)}. \end{aligned} \quad (15)$$

Due to the variables in (15) can be regarded as two groups \mathcal{M} and $(\{\mathcal{A}_i\}_{i=1}^{l-1}, Z, P)$, (15) can directly apply the framework of ADMM [36]. First, we consider the corresponding augmented Lagrangian function of (15)

$$\begin{aligned} \mathcal{L}(\mathcal{M}, \mathcal{A}_i, Z, P, C_i, E, F) = & \sum_{i=1}^{l-1} \frac{\alpha_i}{2} \|X_i^{t+1} Y_i^{t+1} - \mathcal{A}_{i[i]}\|_F^2 + \sum_{i=1}^{l-1} \frac{\beta_i}{2} \left\| \mathcal{A}_i - \tilde{M} + \frac{C_i}{\beta_i} \right\|_F^2 \\ & + \lambda_1 \|Z\|_{1,1} + \frac{\gamma_1}{2} \left\| Z - WM_{(3)}^T + \frac{E}{\gamma_1} \right\|_F^2 + \lambda_2 \|P\|_{1,1} \\ & + \frac{\gamma_2}{2} \left\| P - D_s M_{(3)} + \frac{F}{\gamma_2} \right\|_F^2 + \frac{\rho}{2} \|\mathcal{M} - \mathcal{M}^t\|_F^2, \end{aligned} \quad (16)$$

where $\{\mathcal{C}_i\}_{i=1}^{l-1}$, E , and F denote Lagrangian multipliers, and $\{\beta_i\}_{i=1}^{l-1}$, γ_1 and γ_2 are penalty parameters. Then, ADMM use the following scheme to solve the problem (15):

$$\begin{cases} \mathcal{M}^{t+1,k+1} = \arg \min_{\mathcal{M}} \mathcal{L}(\mathcal{M}, \mathcal{A}_i^k, Z^k, P^k, \mathcal{C}_i^k, E^k, F^k), \\ (\mathcal{A}_i^{k+1}, Z^{k+1}, P^{k+1}) = \arg \min_{(\mathcal{A}_i, Z, P)} \mathcal{L}(\mathcal{M}^{t+1,k+1}, \mathcal{A}_i, Z, P, \mathcal{C}_i^k, E^k, F^k), \\ \mathcal{C}_i^{k+1} = \mathcal{C}_i^k + \beta_i(\mathcal{A}_i^{k+1} - \mathcal{M}^{t+1,k+1}), \\ E^{k+1} = E^k + \gamma_1(Z^{k+1} - W(M_{(3)}^{t+1,k+1})^T), \\ F^{k+1} = F^k + \gamma_2(P^{k+1} - D_s M_{(3)}^{t+1,k+1}). \end{cases} \quad (17)$$

We remark that the calculations of $\{\mathcal{A}_i\}_{i=1}^{l-1}$, Z , P are independent since \mathcal{L} is decoupled about them. Following, we give some details on each step of the ADMM.

- **Calculation of \mathcal{M} .** Given $\{\mathcal{A}_i^k\}_{i=1}^{l-1}$, Z^k , P^k , $\{\mathcal{C}_i^k\}_{i=1}^{l-1}$, E^k , and F^k , $\mathcal{M}^{t+1,k+1}$ is to solve the following a least squares problem:

$$\begin{aligned} \mathcal{M}^{t+1,k+1} &= \arg \min_{\mathcal{M}} \sum_{i=1}^{l-1} \frac{\beta_i}{2} \|\mathcal{A}_i^k - \tilde{\mathcal{M}} + \mathcal{C}_i^k / \beta_i\|_F^2 + \frac{\gamma_1}{2} \|Z^k - WM_{(3)}^T + E^k / \gamma_1\|_F^2 \\ &\quad + \frac{\gamma_2}{2} \|P^k - D_s M_{(3)} + F^k / \gamma_2\|_F^2 + \frac{\rho}{2} \|\mathcal{M} - \mathcal{M}^t\|_F^2. \end{aligned} \quad (18)$$

Assuming periodic boundaries, $\mathcal{M}^{t+1,k+1}$ has the following solution in the Fourier domain

$$M_{(3)}^{t+1,k+1} = \mathcal{F}^{-1} \left(\frac{(\mathcal{F}(\sum_{i=1}^{l-1} \beta_i((\mathcal{A}_i^k)_{(3)} + (\mathcal{C}_i^k)_{(3)}) + \gamma_1(W^T(Z^k + E^k / \gamma_1))^T + \rho M_{(3)}^t + \gamma_2 D_s^T (P^k + F^k / \gamma_2)))}{\mathcal{F}(\sum_{i=1}^{l-1} \beta_i + \gamma_1 + \gamma_2 D_s^T D_s + \rho)} \right), \quad (19)$$

where \mathcal{F} is the discrete Fourier transform and \mathcal{F}^{-1} denotes the inverse Fourier transform. Therefore,

$$\mathcal{M}^{t+1,k+1} = \text{fold}_{(3)}(M_{(3)}^{t+1,k+1}). \quad (20)$$

The complexities of calculating \mathcal{M} is mainly in calculating the fast Fourier transforms on a matrix with size $n_3 \times \prod_{k \neq 3} n_k$, and its time complexities is $O(n_3^3 + (\sum_{k=1}^l \log n_k + n_3 + lm^2) \prod_{k=1}^l n_k)$, where m is the number of filters and l denotes the level of the framelet system.

- **Calculation of \mathcal{A}_i .** Given $\mathcal{M}^{t+1,k+1}$, Z^k , P^k , \mathcal{C}_i^k , E^k , and F^k , we compute \mathcal{A}_i^{k+1} as follows:

$$\begin{aligned} \{\mathcal{A}_i^{k+1}\} &= \arg \min_{P_\Omega(\mathcal{A}_i)=P_\Omega(\mathcal{T})} \sum_{i=1}^{l-1} \frac{\alpha_i}{2} \|X_i^{t+1} Y_i^{t+1} - \mathcal{A}_i\|_F^2 + \sum_{i=1}^{l-1} \frac{\beta_i}{2} \|\mathcal{A}_i - \tilde{\mathcal{M}}^{t+1,k+1} + \mathcal{C}_i^k / \beta_i\|_F^2 \\ &= \arg \min_{\{\mathcal{A}_i\}} \sum_{i=1}^{l-1} \left(\frac{\alpha_i}{2} \|\text{unreshape}_{[i]}(X_i^{t+1} Y_i^{t+1}) - \mathcal{A}_i\|_F^2 + \frac{\beta_i}{2} \|\mathcal{A}_i - \tilde{\mathcal{M}}^{t+1,k+1} + \mathcal{C}_i^k / \beta_i\|_F^2 \right). \end{aligned} \quad (21)$$

The optimal function of computing \mathcal{A}_i is quadratic, thus, \mathcal{A}_i can be solved separately

$$\mathcal{A}_i^{k+1} = P_{\Omega^c} \left(\frac{\alpha_i \text{unreshape}_{[i]}(X_i^{t+1} Y_i^{t+1}) + \beta_i \tilde{\mathcal{M}}^{t+1,k+1} - \mathcal{C}_i^k}{\alpha_i + \beta_i} \right) + P_\Omega(\mathcal{T}). \quad (22)$$

The calculation of \mathcal{A}_i involves the product of X_i and Y_i with sizes $(\prod_{k=1}^i n_k) \times r_i$ and $r_i \times (\prod_{k=i+1}^l n_k)$, whose time complexities is $O((r_i + 1) \prod_{k=1}^l n_k)$.

- **Calculation of Z .** Given $\mathcal{M}^{t+1,k+1}$, \mathcal{A}_i^{k+1} , P^k , \mathcal{C}_i^k , E^k , and F^k , Z^{k+1} can be computed by

$$Z^{k+1} = \arg \min_Z \lambda_1 \|Z\|_{1,1} + \frac{\gamma_1}{2} \|Z - W(M_{(3)}^{t+1,k+1})^T + E^k / \gamma_1\|_F^2. \quad (23)$$

It can be solved by the soft shrinkage operator

$$Z^{k+1} = \max\{|W(M_{(3)}^{t+1,k+1})^T - \lambda_1 / \gamma_1|, 0\} \frac{W(M_{(3)}^{t+1,k+1})^T - E^k / \gamma_1}{|W(M_{(3)}^{t+1,k+1})^T - E^k / \gamma_1|}, \quad (24)$$

where the division is operated component-wise and we assume $0 \cdot (0/0) = 0$. The cost of computing Z is $O(lm^2 \prod_{k=1}^l n_k)$.

- **Calculation of P .** Given $\mathcal{M}^{t+1,k+1}$, \mathcal{A}_i^{k+1} , Z^{k+1} , \mathcal{C}_i^k , E^k , and F^k , the P can be computed by

$$P^{k+1} = \arg \min_Y \lambda_2 \|P\|_{1,1} + \frac{\gamma_2}{2} \|P - D_s(M_{(3)}^{t+1,k+1})^T + F^k / \gamma_2\|_F^2. \quad (25)$$

Similarly, the closed-form solution is

$$P^{k+1} = \max \left\{ |D_s(M_{(3)}^{t+1,k+1})^T - F^k / \gamma_2| - \frac{\lambda_2}{\gamma_2}, 0 \right\} \frac{D_s(M_{(3)}^{t+1,k+1})^T - F^k / \gamma_2}{|D_s(M_{(3)}^{t+1,k+1})^T - F^k / \gamma_2|}, \quad (26)$$

which complexities is $O(n_3 \prod_{k=1}^l n_k)$.

It is clear that every subproblem in ADMM for solving the (15) can be solved accurately, thus, the convergence of ADMM is guaranteed [37]. **Algorithm 1** presents the PAM algorithm for solving (8). And its total complexity is

Algorithm 1 The PAM algorithm to solve (8).

Input: The observed data $\mathcal{T} \in R^{n_1 \times \dots \times n_l}$, index set Ω .

1: **Parameters:** $\alpha_i, \beta_i, i = 1, \dots, l-1, \lambda_1, \lambda_2, f, \gamma_1, \gamma_2, \rho$.

2: **Initialize:** $X^0, Y^0, \mathcal{M}^0, \epsilon = 1e-5$, outer iteration $t_{out} = 100$, inner iteration $k_{inner} = 15$.

3: **Outer iteration:** **While** $t \leq t_{out}$ and not converged

4: **for** $i = 1$ to $l-1$ **do**

5: solve X_i by (11);

6: solve Y_i by (13);

7: **Inner iteration:** **While** $k \leq k_{inner}$ and not converged **do**

8: Update \mathcal{M} by solving (20);

9: Update A_i by solving (22);

10: Update Z by solving (24);

11: Update P by solving (26);

12: Update C_k, E, F via (17);

13: **End**

14: **End**

Output: The restored data \mathcal{M} .

$$O\left(\left(\sum_{i=1}^{l-1} r_i + \sum_{k=1}^l \log n_k + n_3 + lm^2\right) \prod_{k=1}^l n_k + n_3^3\right).$$

Now, we show the convergence of the **Algorithm 1**. For convenience, we rewrite the objective function (8) as

$$f(X, Y, \mathcal{M}) = f_1(X, Y, \mathcal{M}) + f_2(\mathcal{M}), \quad (27)$$

where $f_1(X, Y, \mathcal{M}) = \sum_{i=1}^{l-1} \frac{\alpha_i}{2} \|K(\mathcal{M})_{[i]} - X_i Y_i\|_F^2$ and $f_2(\mathcal{M}) = \lambda_1 \|W M_{(3)}^T\|_{1,1} + \lambda_2 \|D_s M_{(3)}\|_{1,1}$.

The following convergence theory will be used to illustrate the convergence of PAM algorithm.

Lemma 1 [34]. Let $f : R^n \rightarrow R \cup +\infty$ be a proper lower semi-continuous function. Let $x^t, t \in N \subset R^n$ be a sequence so that

H1. $\forall t \in N$, there is $f(x^{t+1}) + a\|x^{t+1} - x^t\|_2^2 \leq f(x^t)$ with a constant $a \in (0, +\infty)$;

H2. $\forall t \in N$, $\exists \omega^{t+1} \in \partial f(x^{t+1})$, and $\|\omega^{t+1}\|_2 \leq b\|x^{t+1} - x^t\|_2$ holds with a constant $b \in (0, +\infty)$;

H3. $\exists \{x^{t_j}\}_{j \in N}$ and $\bar{x} \in R^n$ so that

$$x^{t_j} \rightarrow \bar{x} \quad \text{and} \quad f(x^{t_j}) \rightarrow f(\bar{x}), \quad \text{as } j \rightarrow \infty.$$

If f satisfies the KL property [38] at \bar{x} , then

(i) $x^t \rightarrow \bar{x}$;

(ii) $0 \in \partial f(\bar{x})$, where \bar{x} is a critical point of f .

Next, we show that the function f (27) and the sequence $(X^t, Y^t, \mathcal{M}^t)$ generated by PAM algorithm satisfy the conditions of Lemma 1. Therefore, we establish the following convergence theorem:

Theorem 1. Assume that **Algorithm 1** generate the bounded sequence $(X^t, Y^t, \mathcal{M}^t)$. Then, it can converge to a critical point of f .

Proof. We prove it in three steps, according to the assumptions in Lemma 1.

First, we show that f is a proper lower semi-continuous function. Because of f_1 is a C^1 Lipschitz continuous gradient function and f_2 is lower semicontinuous and proper, the f is a proper lower semi-continuous function.

Second, the iterative sequence $(X^t, Y^t, \mathcal{M}^t)$ generated by PAM algorithm satisfies the conditions H1, H2, H3 in Lemma 1. It is worth noting that **Algorithm 1** is an example of algorithm (61)-(62)-(63) displayed in [34] with $\mathbf{B}_i = \rho \mathbf{I}$ (more details see [34]). Thus, the sequence $(X^t, Y^t, \mathcal{M}^t)$ satisfies the condition H1, H2, and H3.

Third, we show that f satisfies the KL property at each $(X^t, Y^t, \mathcal{M}^t) \in \text{dom}(f)$ by proving that f is semi-algebraic on $\text{dom}(f)$. Since finite sums and finite products of semi-algebraic functions are semi-algebraic [34], it keeps showing that $f_1(X, Y, \mathcal{M})$ and $f_2(\mathcal{M})$ are semi-algebraic. In fact, $f_1(X, Y, \mathcal{M})$ is a polynomial of (X, Y, \mathcal{M}) , since the reshape operator is a linear mapping between finite-dimension spaces. Then f_1 is semi-algebraic, since polynomials are semi-algebraic functions

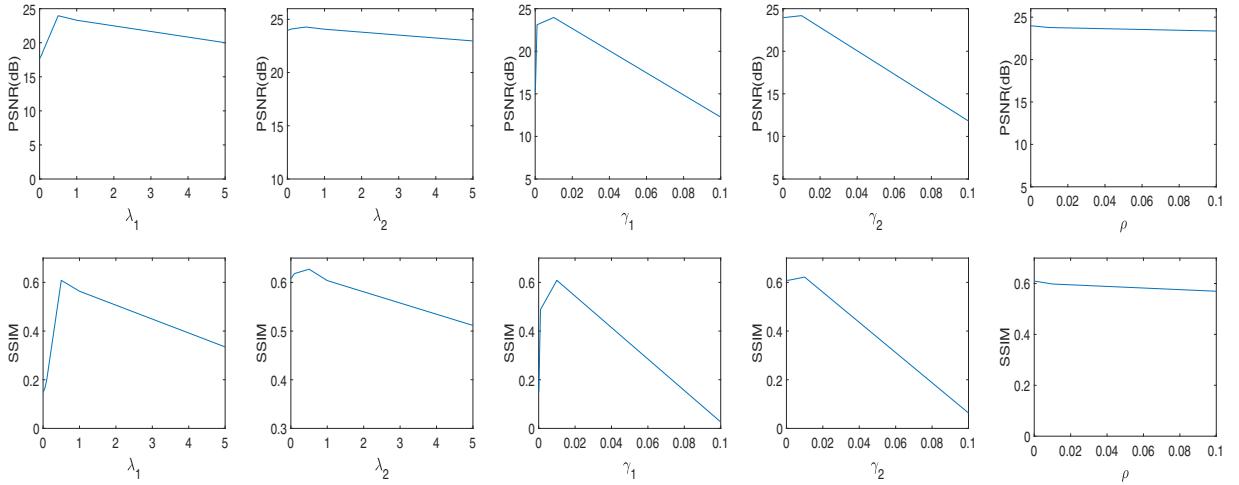


Fig. 3. The numerical results corresponding to parameters: λ_1 , λ_2 , γ_1 , γ_2 , and ρ .

[34]. The function $f_2(\mathcal{M})$ is a finite linear combination of the absolute value function and linear polynomials, which are semi-algebraic. Therefore, f_2 is a semi-algebraic function. Then, f is semi-algebraic.

According to Lemma 1, the bounded sequences generated by [Algorithm 1](#) converge to a critical point of f . Therefore, the proof is complete. \square

4. Numerical experiments

To test the effectiveness and superiority of the proposed LRTC method, we employ a set of well-known methods for comparison, namely, HaLRTC [17], t-SVD [20], SiLRTC-TT [19], and TMac-TT [19]. We perform all methods on two testing data including color images and videos. All test tensors are scaled into the interval $[0, 255]$. All numerical experiments are tested in MATLAB R2012a with an Intel Core i7-8700M 3.70 GHz and 8 GB.

We utilize the peak signal-to-noise ratio (PSNR) [39] and the structural similarity index (SSIM) [40] to measure the quantitative indexes of recovered results. PSNR (dB) satisfies the human subjective sensation and SSIM matches the quality perception of the human visual system. The higher the PSNR and SSIM value, the better the recovery result. The PSNR and SSIM values of the higher-order tensor can be obtained by calculating the mean of all frames.

We stop our algorithm when the outer iteration $t > t_{out}$ or the relative error of the tensor \mathcal{M} satisfies

$$\frac{\|\mathcal{M}^{k+1} - \mathcal{M}^k\|_F}{\|\mathcal{M}^k\|_F} \leq 10^{-5}. \quad (28)$$

Effect of tuning parameters. Our method involves the following parameters: α_i in (8) controlling the low-rankness of the underlying tensor, regularization parameters λ_1 and λ_2 , penalty parameters β_i , γ_1 , and γ_2 , and proximal parameter ρ .

For α_i , we set

$$\alpha_i = \frac{\delta_i}{\sum_{i=1}^{l-1} \delta_i} \quad \text{with} \quad \delta_i = \min(\prod_{k=1}^i n_k, \prod_{k=i+1}^l n_k), \quad (29)$$

where $i = 1, \dots, l-1$. For β_i , we set $\beta_i = f\alpha_i$ and empirically choose f from one value in $\{0.005, 0.01\}$. Next, we test the effect of regularization parameters λ_1 and λ_2 , penalty parameters γ_1 and γ_2 , and proximal parameter ρ . [Fig. 3](#) shows the PSNR and SSIM curves corresponding to the parameters involved in the proposed algorithm on the video data *mobile* with $SR = 0.1$. We get two findings: (1) the proposed method is sensitive to the regularization parameters λ_1 , λ_2 , γ_1 , and γ_2 , and the PSNR and SSIM curves are concave with unique optimal parameter settings which can be obtained by the hand-tuning strategy; (2) the proposed method is robust with respect to ρ , with little effect on the completion performance. Based on these observations, we empirically search the optimal parameter settings for λ_1 and λ_2 in the range $[0.1, 1]$ with the increment 0.1, γ_1 and γ_2 in the candidate set: $\{0.001, 0.01, 0.1\}$, and fix $\rho = 0.001$. More specifically, we set $\gamma_1 = 0.01$, $\gamma_2 = 0.001$, and $\rho = 0.001$ for all experiments. For color images, we set parameters $\lambda_1 = 0.5$ and $\lambda_2 = 0$ because of the low correlation between the three color channels. For videos, we set $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$. We try our best to tune the parameters involved in the competing methods as suggested in the reference papers to obtain the highest PSNR value.

Effect of the framelet system. We test the effect of the framelet system, which mainly determined by two factors: the framelet function ψ and the decomposition level. In our experiment, we consider three framelet functions: the Haar wavelet, the piecewise linear framelet, and the piecewise cubic framelet (denoted as function=Haar, Linear, and Cubic, respectively), with three levels 1, 2, and 4. [Table 1](#) lists the restore performance of the proposed method under different framelet systems

Table 1Numerical performance of the proposed method on image *house* corresponding to different framelet functions and levels.

Framelet systems	Function	Haar			Linear			Cubic		
		Level	1	2	4	1	2	4	1	2
Metrics	PSNR	26.32	26.79	26.88	27.49	27.46	27.41	27.95	27.68	27.37
	SSIM	0.7643	0.7721	0.7740	0.7843	0.7841	0.7827	0.7907	0.7866	0.7831

Table 2

Numerical performance of the five algorithms with random missing entries on color images.

Image	SR	0.1			0.2			0.3			0.4		
		Method	PSNR	SSIM	TIME	PSNR	SSIM	TIME	PSNR	SSIM	TIME	PSNR	SSIM
peppers	HaLRTC	16.24	0.3325	18	20.31	0.5357	19	23.37	0.6851	19	26.06	0.7899	19
	t-SVD	16.61	0.2514	188	20.30	0.4496	188	23.40	0.6067	204	26.02	0.7310	189
	SiLRTC-TT	18.86	0.5244	118	22.13	0.6827	77	24.38	0.7785	69	26.44	0.8438	50
	TMac-TT	19.51	0.4135	320	25.00	0.7269	353	27.48	0.8220	161	29.21	0.8697	73
	Ours	25.27	0.8155	5202	27.73	0.8842	4612	29.48	0.9129	5460	31.00	0.9351	5332
barbara	HaLRTC	18.00	0.3924	16	21.72	0.5959	21	24.53	0.7307	21	26.93	0.8202	25
	t-SVD	18.31	0.3600	168	22.01	0.5736	218	24.99	0.7261	196	27.82	0.8294	199
	SiLRTC-TT	20.04	0.5307	110	22.86	0.6861	74	25.10	0.7872	55	27.16	0.8550	39
	TMac-TT	19.28	0.4012	406	25.05	0.7463	256	27.70	0.8363	152	29.41	0.8816	62
	Ours	24.57	0.7732	13115	27.85	0.8542	12981	29.38	0.8914	12049	30.47	0.9179	13147
baboon	HaLRTC	17.36	0.2854	18	19.78	0.4432	17	21.43	0.5729	17	22.87	0.6771	16
	t-SVD	17.34	0.2532	219	19.61	0.4171	196	21.29	0.5571	204	22.99	0.6715	190
	SiLRTC-TT	18.19	0.3224	126	20.34	0.4775	85	21.88	0.6067	66	23.27	0.7056	47
	TMac-TT	19.30	0.3828	339	21.55	0.5398	125	22.73	0.6356	81	23.51	0.7097	37
	Ours	20.64	0.4805	12770	22.29	0.6266	13199	23.57	0.7199	12655	24.91	0.7946	12825
house	HaLRTC	20.02	0.4733	21	23.69	0.6575	24	26.72	0.7749	22	29.27	0.8486	22
	t-SVD	20.53	0.3879	191	24.50	0.6047	217	27.53	0.7443	195	30.26	0.8293	198
	SiLRTC-TT	21.38	0.5949	104	24.32	0.7210	68	26.92	0.8064	53	29.04	0.8632	41
	TMac-TT	24.14	0.6429	327	28.32	0.7854	110	29.85	0.8303	46	31.06	0.8640	45
	Ours	27.95	0.7907	4768	30.02	0.8410	4039	31.57	0.8747	2928	32.71	0.9009	1509
Lena	HaLRTC	18.56	0.4106	19	22.27	0.6025	21	25.02	0.7335	22	27.45	0.8235	20
	t-SVD	18.87	0.3469	191	22.57	0.5562	194	25.33	0.7015	205	28.09	0.8104	195
	SiLRTC-TT	20.96	0.5924	92	24.00	0.7384	61	26.26	0.8228	48	28.33	0.8811	39
	TMac-TT	23.36	0.6553	287	26.97	0.8130	99	28.68	0.8610	61	29.66	0.8916	59
	Ours	26.39	0.8168	13064	28.04	0.8702	12491	29.63	0.9048	11956	30.93	0.9265	10958

on color image *house* with SR = 0.1. We observe that both the framelet function and the level have an influence on the complete performance. In our work, we choose the framelet system (function=Cubic and level=1) in all experiments for the highest PSNR values.

4.1. Color image completion

We test the performance of the proposed method on color images completion, including random missing and structural missing. In particular, structural missing is more complex than random missing, so structural missing is more challenging.

Random Missing. We randomly sample entries for color images with size $256 \times 256 \times 3$. The sampling rates (SRs) are set as 0.1, 0.2, 0.3, and 0.4, respectively. For the low-rank matrix factorizations term in (8), the color image is transformed into a ninth-order $\tilde{\mathcal{M}} \in R^{4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 3}$ by KA.

Fig. 4 compares the visual quality of color images for random sampling entries with SR = 0.1. Obviously, there are many artifacts in the recovered results by HaLRTC and t-SVD, which lead to a poor visual effect. The recovered images of SiLRTC-TT and TMac-TT contain serious block-artifacts. As a comparison, the proposed method keeps the smoothness and provides clear details, which can be seen from the enlarged regions.

Table 2 summarizes the numerical performance (PSNR and SSIM) and execution time (in seconds) of restored results using five utilized LRTC approaches on color images with different SRs. It is clear that the best results are denoted in bold. From Table 2, the proposed method achieves higher numerical results at the cost of more running time.

Structural Missing. Fig. 5 shows the recovered performance on color images for the structural missing case, including black stripes and texts. The proposed method yields a much better visual effect in keeping details and structures. However, the results obtained by HaLRTC and t-SVD remain a large number of missing entries, especially for black stripes. SiLRTC-TT can perform comparatively completing results, but it can still clearly see the outlines of text on the recovered image. In addition, the recovered images by TMac-TT suffer from serious block-artifacts.

Table 3 presents the quality indexes by different approaches with structural missing entries and marks the best results in bold for clarity. It is clear that for different images and structural missing cases, the proposed method consistently performs better on PSNR and SSIM values.

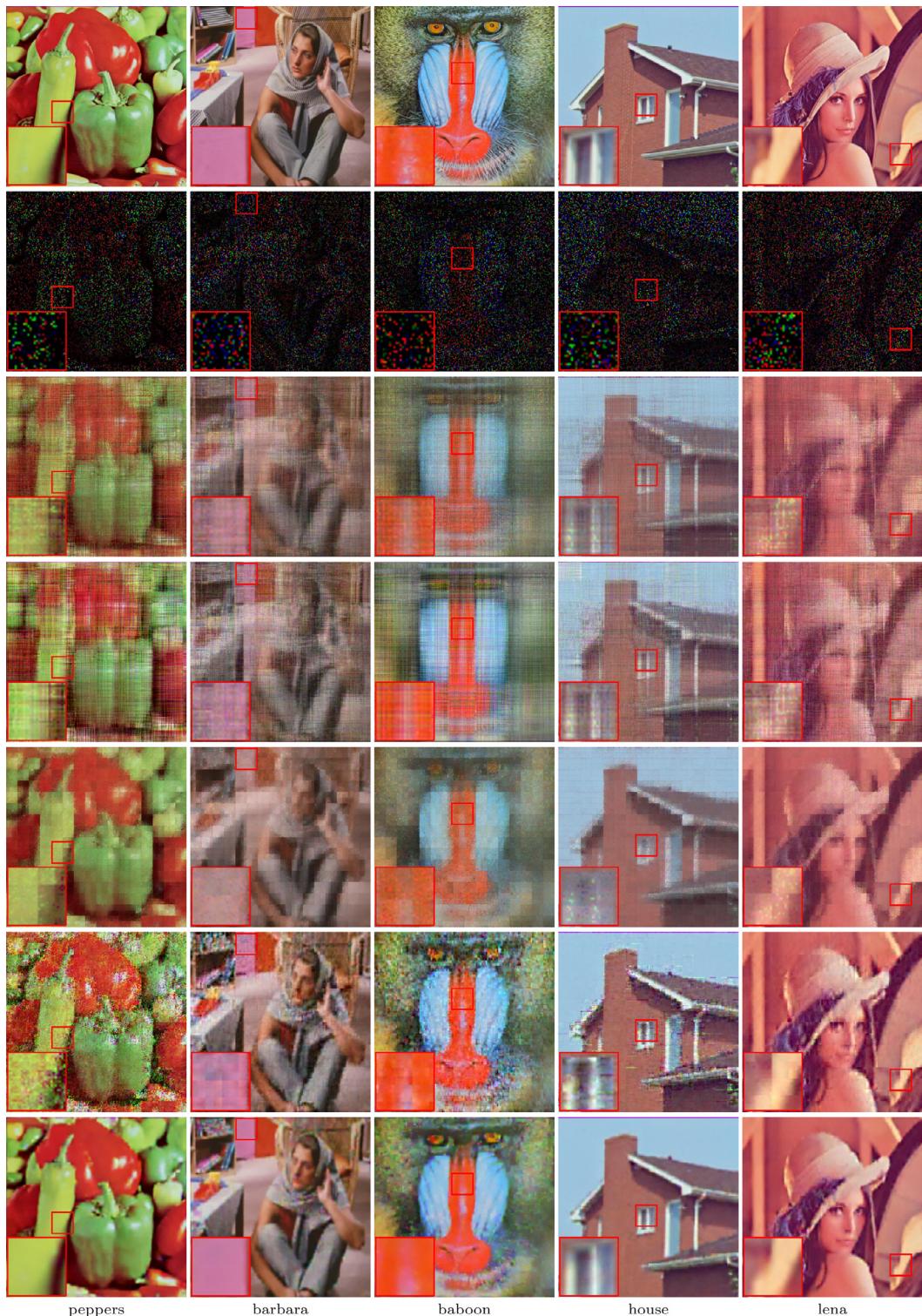


Fig. 4. Recovered color images for random missing entries with $SR = 0.1$. From top to bottom: the original data, the observed data, the recovered results by HaLRTC, t-SVD, SiLRTC-TT, TMac-TT, and Our method, respectively. From left to right: peppers, barbara, baboon, house, and lena.

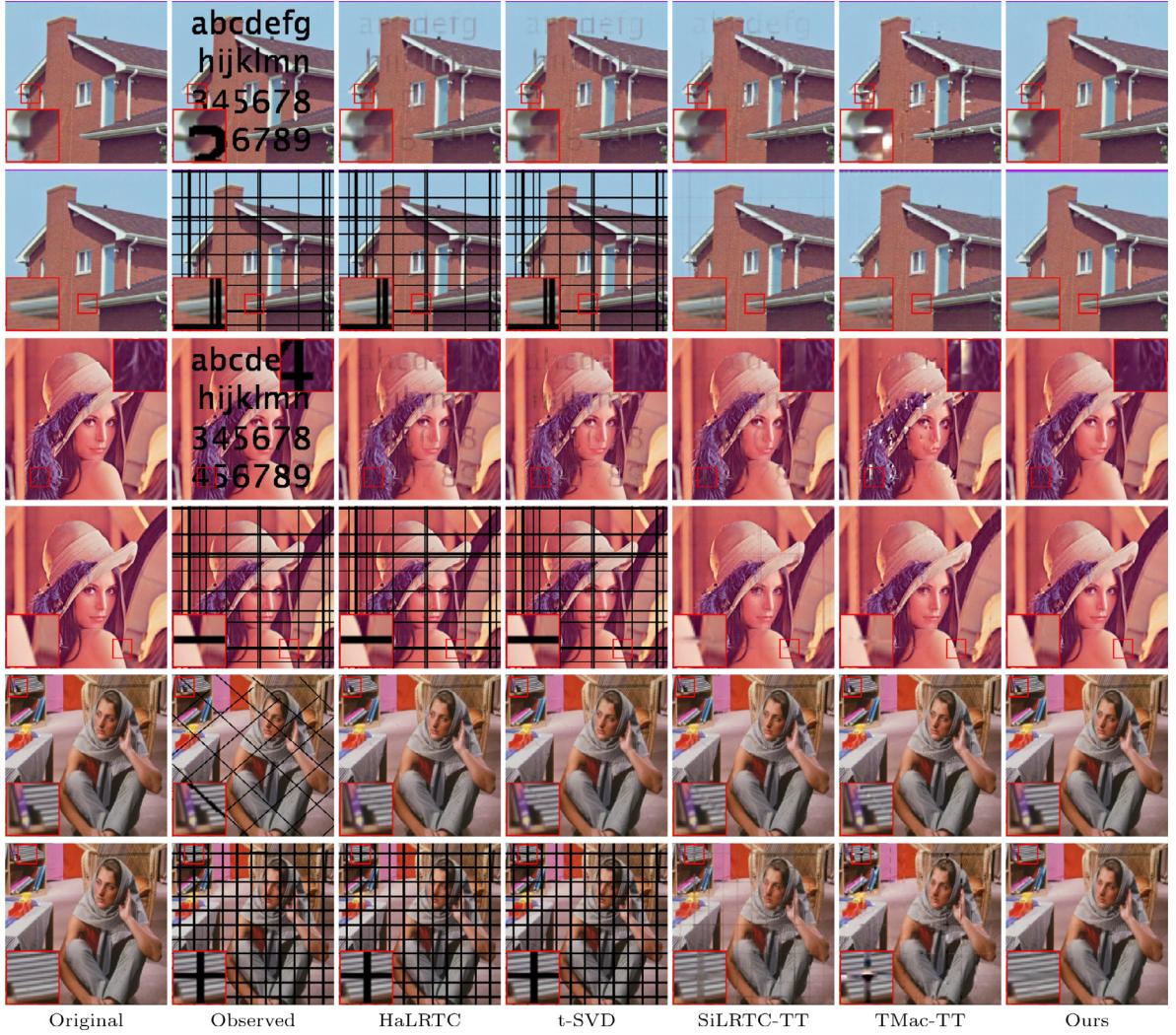


Fig. 5. Recovered color images with fixed missing entries. From left to right: the original image, the observed image, the recovered results by HaLRTC, t-SVD, SiLRTC-TT, TMac-TT, and Our method, respectively. From top to bottom: house-1, house-2, lena-1, lena-2, barbara-1, and barbara-2.

Table 3

Numerical performance of the five algorithms with structural missing entries on color images.

Method	HaLRTC			t-SVD			SiLRTC-TT			TMac-TT			Ours		
	PSNR	SSIM	TIME	PSNR	SSIM	TIME	PSNR	SSIM	TIME	PSNR	SSIM	TIME	PSNR	SSIM	TIME
Image															
house-1	27.98	0.8976	8	28.09	0.8959	142	30.30	0.9327	38	30.65	0.9234	21	34.30	0.9578	2768
house-2	10.91	0.4741	7	10.91	0.4741	126	28.49	0.9254	25	26.80	0.9149	8	31.06	0.9577	2336
lena-1	26.24	0.8742	10	26.17	0.8687	145	28.37	0.9165	37	27.74	0.9042	55	30.56	0.9493	4519
lena-2	11.79	0.5008	7	11.79	0.5008	147	30.66	0.9279	26	31.54	0.9383	9	34.34	0.9682	2027
barbara-1	36.83	0.9779	3	36.47	0.9749	168	36.61	0.9802	13	36.45	0.9764	17	39.59	0.9894	1187
barbara-2	11.65	0.4298	4	11.65	0.4298	236	26.89	0.8710	88	29.16	0.8763	116	30.25	0.9383	2627

The convergence analysis. In Theorem 1, we theoretically prove that under bounded assumptions, the sequences generated by Algorithm 1 converges to a critical point. To demonstrates the convergence of the PAM algorithm in numerical, Fig. 6 displays the relative error between two successive restored tensor \mathcal{M}^k and \mathcal{M}^{k+1} , i.e., $\|\mathcal{M}^{k+1} - \mathcal{M}^k\|_F / \|\mathcal{M}^k\|_F$ on the color image *house* and *Lena* with SR = 0.1. We can see that the relative error curves keep declining over the iterations.

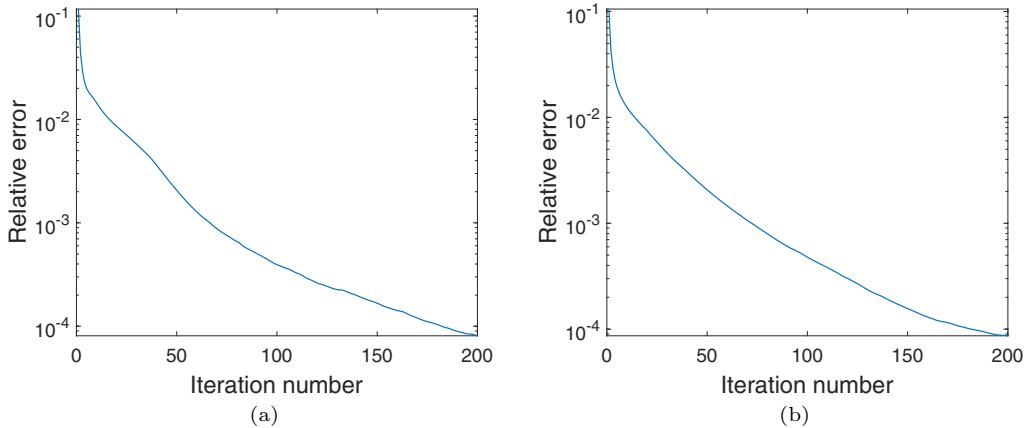


Fig. 6. Curves of relative error values versus iterations. (a) *house* and (b) *lena*.

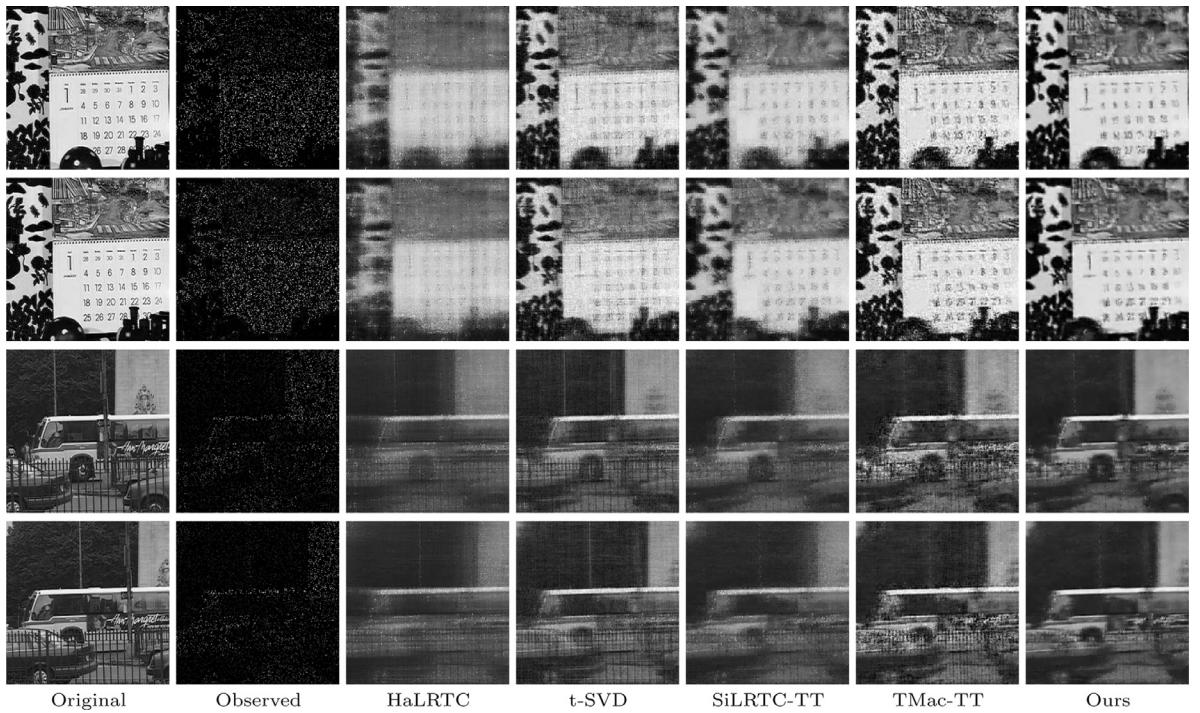


Fig. 7. Recovered gray videos for random missing entries with $SR = 0.1$. The recovered results by different methods on videos with $SR = 0.1$. From left to right: the original image, the observed image, the recovered results by HalRTC, t-SVD, SiLRTC-TT, TMac-TT, and Our method, respectively.

4.2. Video completion

The algorithms are also tested on gray videos, including *mobile* and *bus*.² The size of all test videos is $256 \times 256 \times 30$. To fully explore the TT low-rankness, we cast the third-order video tensor into a fifth-order tensor with size of $16 \times 16 \times 16 \times 16 \times 30$ by KA.

Fig. 7 compares one frame of restored videos *mobile* and *bus* with $SR = 0.1$. From the visual comparison, our method outperforms HalRTC, t-SVD, SiLRTC-TT, and TMac-TT in remaining details and structures. Fig. 8 presents the random pixel values along with the temporal mode of these restored videos with different methods in $SR = 0.1$, which shows their smoothness in the temporal mode. It is clear that the proposed method preserves the smoothness in the temporal mode of the restored videos, and the figure has a better touch of the original than other methods. This phenomenon illustrates the power of promoting the smoothness of the proposed method in the temporal direction.

² <http://trace.eas.asu.edu/yuv/>.

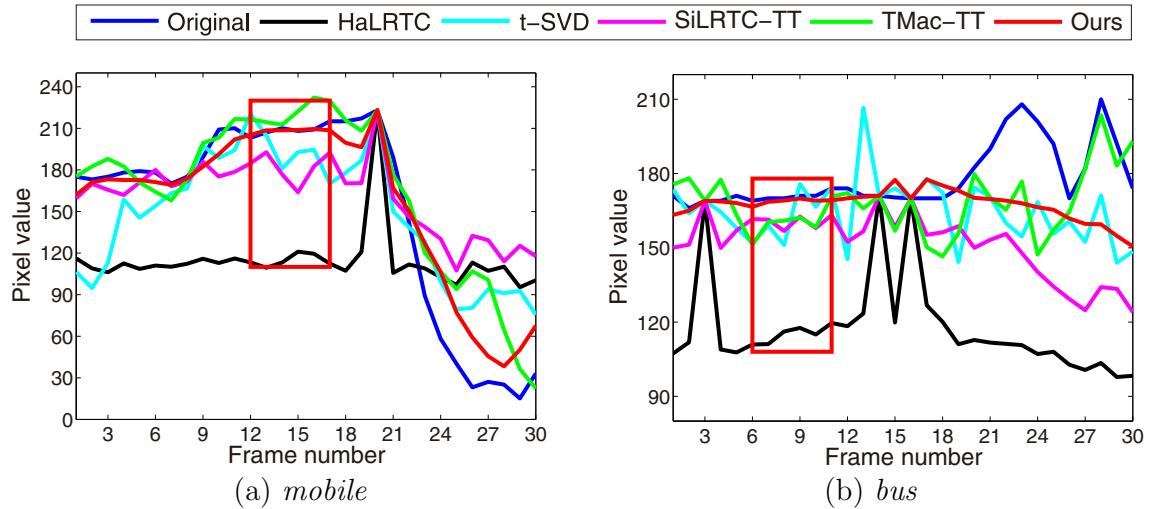


Fig. 8. The pixel values of the recovered videos *mobile* and *bus*($SR = 0.1$) by HaLRTC, t-SVD, SiLRTC-TT, TMac-TT, and Our method, respectively.

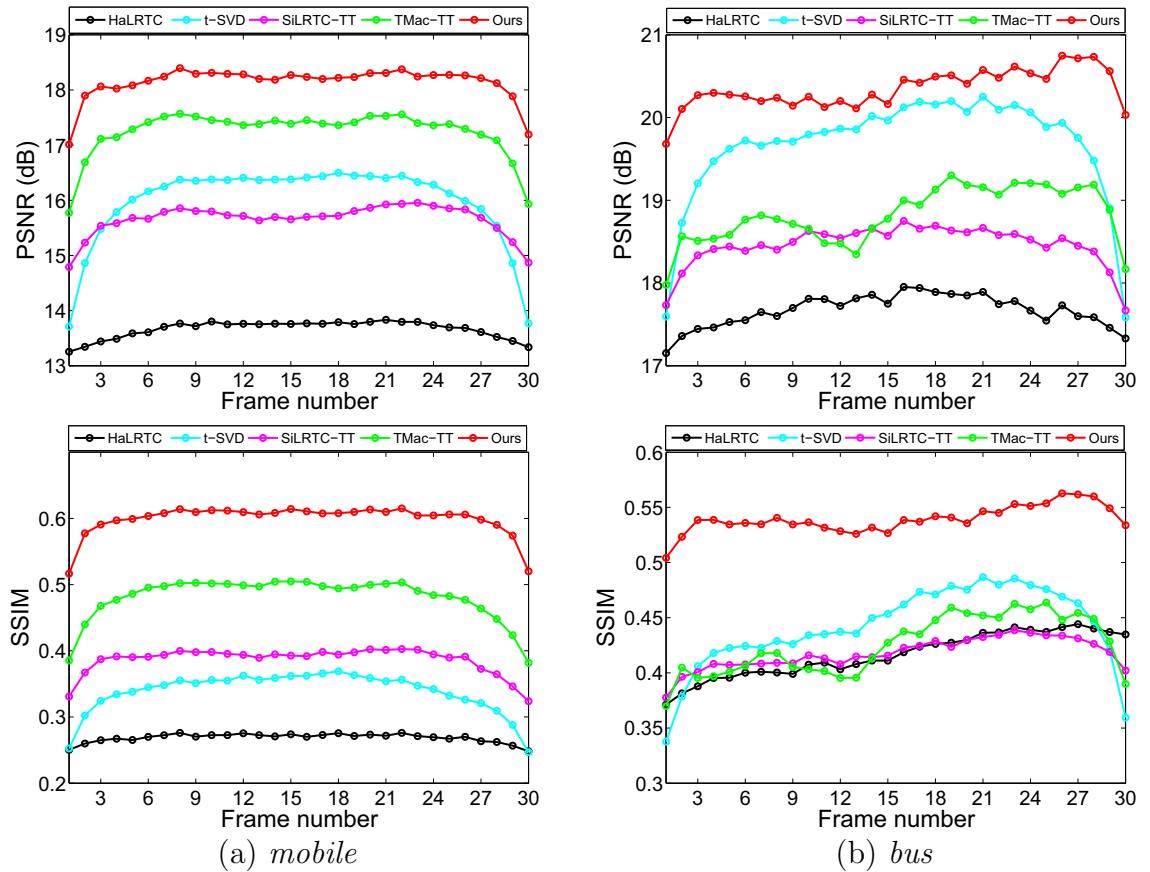


Fig. 9. The quality indexes for each frame of videos *mobile* and *bus* ($SR = 0.1$).

Fig. 9 plots the numerical (PSNR and SSIM) curves for each frame of the videos, respectively. Clearly, for each frame, our method obtains the highest values in terms of PSNR and SSIM. **Table 4** shows the PSNR and SSIM values of completed videos with different SRs. We find from it that the proposed method obtains higher values in most cases, which is agree with the visual results shown in **Fig. 7**.

Table 4
Numerical performance of the five algorithms on gray videos.

Video	SR	0.05			0.1			0.2		
		Method	PSNR	SSIM	TIME	PSNR	SSIM	TIME	PSNR	SSIM
mobile	HaLRTC	12.12	0.1885	17	13.66	0.2685	15	15.79	0.4213	12
	t-SVD	14.40	0.2176	2669	15.97	0.3381	2652	18.28	0.5112	3845
	SiLRTC-TT	14.16	0.2646	993	15.66	0.3862	650	17.88	0.5772	590
	TMac-TT	15.97	0.3525	279	17.23	0.4803	283	19.09	0.6112	115
	Ours	16.85	0.4769	17835	18.14	0.5985	17769	21.12	0.7927	17737
	HaLRTC	15.74	0.3296	102	17.63	0.4165	75	19.84	0.5570	44
	t-SVD	18.20	0.3270	3982	19.65	0.4413	5019	21.83	0.6011	4029
bus	SiLRTC-TT	17.00	0.3344	910	18.46	0.4174	755	20.36	0.5601	531
	TMac-TT	16.73	0.2489	356	18.82	0.4247	373	20.86	0.5698	232
	Ours	19.11	0.4513	17256	20.34	0.5392	16706	21.53	0.6393	17377

5. Conclusion

In this paper, we propose a new tensor completion model using low-TT-rank constraint and the hybrid smoothness regularization. Specifically, we use low-TT-rank matrix factorization to exploit the global low-rankness and utilize the framelet and TV regularization to enhance the local smoothness of the expected tensor. An effective PAM-based algorithm is designed for solving the proposed model. Meanwhile, the sequences generated by the proposed algorithm converges to a critical point, thus we theoretically prove the convergence. Numerical experiments using various tensor data show the effectiveness of the proposed method in keeping the local smoothness in the spatial and temporal dimensions.

In the future, we will research the theoretical results of the proposed model, investigate a faster numerical algorithms with convergence, and extend the proposed method to other image processing tasks.

Acknowledgments

The authors would like to thank the anonymous referees and editor for their valuable remarks, questions, and comments that enabled the authors to improve this paper. The research is supported by NSFC (61876203, 61772003, 11901450), Project funded by China Postdoctoral Science Foundation (2018M643611), and National Postdoctoral Program for Innovative Talents (BX20180252).

References

- [1] S.Q. Gao, Q.B. Fan, A mixture of nuclear norm and matrix factorization for tensor completion, *SIAM J. Sci. Comput.* 75 (11) (2018) 43–64.
- [2] L. Zhen, Y.P. Liu, L.X. Chen, C. Zhu, Low rank tensor completion for multiway visual data, *Signal Process.* 155 (2019) 301–316.
- [3] T. Yokota, Q. Zhao, A. Cichocki, Smooth PARAFAC decomposition for tensor completion, *IEEE Trans. Signal Process.* 64 (20) (2016) 5423–5436.
- [4] J.H. Yang, X.L. Zhao, T.Y. Ji, T.H. Ma, T.Z. Huang, Low-rank tensor train for tensor robust principal component analysis, *Appl. Math. Comput.* 367 (2020), doi:10.1016/j.amc.2019.124783.
- [5] Z.M. Xing, M.Y. Zhou, A. Castrodad, G. Sapiro, L. Carin, Dictionary learning for noisy and incomplete hyperspectral images, *SIAM J. Imag. Sci.* 5 (1) (2012) 33–56.
- [6] J.H. Yang, X.L. Zhao, T.H. Ma, Y. Chen, T.Z. Huang, Remote sensing image desriping using unidirectional high-order total variation and non-convex low-rank regularization, *J. Comput. Appl. Math.* 363 (2020) 124–144.
- [7] Y. Fu, W.S. Dong, 3D magnetic resonance image denoising using low-rank tensor approximation, *Neurocomputing* 195 (2016) 30–39.
- [8] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, *SIAM Rev.* 51 (3) (2009) 455–500.
- [9] T.G. Kolda, J.M. Sun, Scalable tensor decompositions for multi-aspect data mining, *Proceedings of the ICDM* (2008) 363–372.
- [10] C.Y. Lu, J.S. Feng, Y.D. Chen, Z.C. Lin, S.C. Yan, Tensor robust principal component analysis with a new tensor nuclear norm, *IEEE Trans. Pattern Anal.* (2018).
- [11] I.V. Oseledets, Tensor-train decomposition, *SIAM J. Sci. Comput.* 33 (5) (2011) 2295–2317.
- [12] E. Acar, D.M. Dunlavy, T.G. Kolda, M. Morettini, Scalable tensor factorizations for incomplete data, *Chemometr. Intell. Lab.* 106 (1) (2011) 41–56.
- [13] Q. Shi, H. Lu, Y.M. Cheung, Tensor rank estimation and completion via CP-based nuclear norm, In *CIKM* (2017) 949–958.
- [14] Y.P. Liu, L. Zhen, H.Y. Huang, C. Zhu, Low CP rank and Tucker rank tensor completion for estimating missing components in image data, *IEEE Trans. Circ. Syst. Vid.* (2019), doi:10.1109/TCSVT.2019.2901311.
- [15] Y. Wang, D.Y. Meng, M. Yuan, Sparse recovery: from vectors to tensors, *Natl. Sci. Rev.* 5 (5) (2018) 756–767.
- [16] M. Filipović, A. Jukić, Tucker factorization with missing data with application to low-n-rank tensor completion, *Multidim. Syst. Sign. Process.* 26 (3) (2015) 677–692.
- [17] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, *IEEE Trans. Pattern Anal.* 35 (1) (2013) 208–220.
- [18] Y.Y. Xu, R.R. Hao, W.T. Yin, Z.X. Su, Parallel matrix factorization for low-rank tensor completion, *Inverse Probl. Imaging* 9 (2) (2017) 601–624.
- [19] J.A. Bengua, H.N. Phien, H.D. Tuan, M.N. Do, Efficient tensor completion for color image and video recovery: low-rank tensor train, *IEEE Trans. Image Process.* 26 (5) (2017) 2466–2479.
- [20] Z.M. Zhang, S.C. Aeron, Exact tensor completion using t-SVD, *IEEE Trans. Signal Process.* 65 (6) (2017) 1511–1526.
- [21] J.A. Bengua, H.N. Phien, H.D. Tuan, Optimal feature extraction and classification of tensors via matrix product state decomposition, *Proceedings of the IEEE International Congress on Big Data* (2015) 669–672.
- [22] M. Ding, T.Z. Huang, T.Y. Ji, X.L. Zhao, J.H. Yang, Low-rank tensor completion using matrix factorization based on tensor train rank and total variation, *J. Sci. Comput.* 81 (2019) 941–964.
- [23] F.M. Fang, G.X. Zhang, F. Li, C.M. Shen, Framelet based pan-sharpening via a variational method, *Neurocomputing* 129 (2014) 362–377.
- [24] A. Chai, Z. Shen, Deconvolution: a wavelet frame approach, *Numer. Math.*, 106 (4) (2007) 529–587.
- [25] J.I. Latorre, Image compression and entanglement, *Comput. Sci.* (2005). Available: <https://arxiv.org/abs/quant-ph/0510031>.

- [26] J.F. Cai, R.H. Chan, Z. Shen, A framelet-based image inpainting algorithm, *Appl. Comput. Harmon. A.* 24 (2) (2008) 131–149.
- [27] X.L. Zhao, W. Wang, T.Y. Zeng, T.Z. Huang, M.K. Ng, Total variation structured total least squares method for image restoration, *SIAM J. Sci. Comput.* 35 (6) (2013) 1304–1320.
- [28] Y. Wang, J.J. Peng, Q. Zhao, Y. Leung, X.L. Zhao, D.Y. Meng, Hyperspectral image restoration via total variation regularized low-rank tensor decomposition, *IEEE J-STARS* 11 (4) (2017) 1227–1243.
- [29] Y.P. Liu, L. Zhen, C. Zhu, Image completion using low tensor tree rank and total variation minimization, *IEEE Trans. Multimedia* 21 (2) (2019) 338–350.
- [30] E. Candès, B. Recht, Exact matrix completion via convex optimization, *Commun. ACM* 55 (6) (2012) 111–119.
- [31] R. Tomioka, T. Suzuki, Convex tensor decomposition via structured Schatten norm regularization, *Proceedings of the NIPS* (2013).
- [32] M. Imaizumi, T. Maehara, K. Hayashi, On tensor train rank minimization: statistical efficiency and scalable algorithm, *Proceedings of the NIPS* (2017).
- [33] S. Oymak, A. Jalali, M. Fazel, Y.C. Eldar, B. Hassibi, Simultaneously structured models with application to sparse and low-rank matrices, *IEEE Trans. Inform. Theory* 61 (5) (2015) 2886–2908.
- [34] H. Attouch, J. Bolte, B.F. Svaiter, Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods, *Math. Program.* 137 (1-2) (2013) 91–129.
- [35] T.H. Ma, Y. Lou, T.Z. Huang, Truncated l_{1-2} models for sparse recovery and rank minimization, *SIAM J. Imag. Sci.* 10 (3) (2017) 1346–1380.
- [36] X.L. Zhao, F. Wang, M.K. Ng, A new convex optimization model for multiplicative noise and blur removal, *SIAM J. Imaging Sci.* 7 (1) (2014) 456–475.
- [37] J. Eckstein, D.P. Bertsekas, On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators, *Math. Program.* 55 (1) (1992) 293–318.
- [38] J. Bolte, A. Daniilidis, A. Lewis, M. Shiota, Clarke subgradients of stratifiable functions, *SIAM J. Optim.* 18 (2) (2007) 556–572.
- [39] Y.B. Zheng, T.Z. Huang, X.L. Zhao, T.X. Jiang, T.H. Ma, T.Y. Ji, Mixed noise removal in hyperspectral image via low-fibered-rank regularization, *IEEE Trans. Geosci. Remote* (2019), doi:10.1109/TGRS.2019.2940534.
- [40] X.L. Zhao, F. Wang, T.Z. Huang, M.K. Ng, R.J. Plemmons, Deblurring and sparse unmixing for hyperspectral images, *IEEE Trans. Geosci. Remote Sens.* 51 (7) (2013) 4045–4058.