Applied Mathematical Modelling 70 (2019) 677–695

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Low-rank tensor completion via smooth matrix factorization

Yu-Bang Zheng [a], Ting-Zhu Huang [a,*], Teng-Yu Ji [b], Xi-Le Zhao [a,*], Tai-Xiang Jiang [a], Tian-Hui Ma [c]

[a] School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, PR China
[b] School of Science, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, PR China
[c] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, PR China

A R T I C L E   I N F O

Article history:
Received 20 July 2018
Revised 26 January 2019
Accepted 6 February 2019
Available online 11 February 2019

Keywords:
Low-rank tensor completion
Smooth matrix factorization
Total variation
Framelet
Block successive upper-bound minimization

A B S T R A C T

Low-rank modeling has achieved great success in tensor completion. However, the low-rank prior is not sufficient for the recovery of the underlying tensor, especially when the sampling rate (SR) is extremely low. Fortunately, many real world data exhibit the piece-wise smoothness prior along both the spatial and the third modes (e.g., the temporal mode in video data and the spectral mode in hyperspectral data). Motivated by this observation, we propose a novel low-rank tensor completion model using smooth matrix factorization (SMF-LRTC), which exploits the piecewise smoothness prior along all modes of the underlying tensor by introducing smoothness constraints on the factor matrices. An efficient block successive upper-bound minimization (BSUM)-based algorithm is developed to solve the proposed model. The developed algorithm converges to the set of the coordinate-wise minimizers under some mild conditions. Extensive experimental results demonstrate the superiority of the proposed method over the compared ones.

1. Introduction

With the rocketing development of information technology, realistic data, such as magnetic resonance image (MRI), hyperspectral image, and video, tend to have high dimensions and complex structures. As a high-dimensional generalization of vector and matrix, tensor can better express the complex essential structures of higher-order data. Thus, higher-order tensors have extensive applications in many fields, such as MRI data recovery [1,2], hyperspectral image recovery [3–8], video rain streak removal [9,10], image/video inpainting [11–14], and signal reconstruction [15,16].

Tensor completion is to recover the higher-order tensor with missing entries. The key is to built up the relationship between the available and the missing entries [17]. As an extension of low-rank matrix completion (LRMC) [18,19], low-rank tensor completion (LRTC) utilizes the low-rank prior to characterize the relationship between the available and the missing entries. Mathematically, the LRTC problem can be written as

$$\begin{aligned} \min_{\mathcal{Y}} \quad & \text{rank}(\mathcal{Y}) \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{F}, \end{aligned} \tag{1}$$

* Corresponding authors.
E-mail addresses: zhengyubang@163.com (Y.-B. Zheng), tingzhuhuang@126.com, tzhuang@uestc.edu.cn (T.-Z. Huang), tengyu_j66@126.com (T.-Y. Ji), xlzhao122003@163.com (X.-L. Zhao), taixiangjiang@gmail.com (T.-X. Jiang), nkmth0307@126.com (T.-H. Ma).

https://doi.org/10.1016/j.apm.2019.02.001
0307-904X/

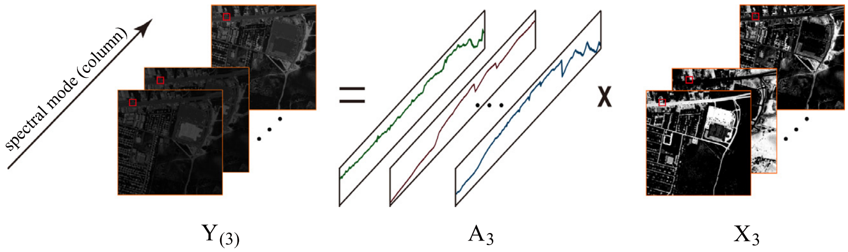

Fig. 1. Illustrations of the physical interpretations of the factors A_3 and X_3 in hyperspectral unmixing. Each row of $Y_{(3)}$ and X_3 are reshaped to image for visualization.

where $\mathcal{Y}$ is the underlying tensor, $\mathcal{F}$ is the observed tensor, Ω is the index set for available entries, and $\mathcal{P}_\Omega(\cdot)$ is the projection operator that keeps the entries of $\mathcal{Y}$ in Ω and zeros out others. Among all definitions for the rank of tensors, the Tucker rank (also named n-rank, see details in Section 2) is widely used to depict the low-rankness of the underlying tensor [17,20,21]. However, directly minimizing the Tucker rank is NP-hard [22,23].

In the past decade, the nuclear norm, as the tightest convex surrogate of the matrix rank, has been widely used for low-rank matrix approximation [24,25]. Inspired by this, Liu et al. [17] established the following definition of the nuclear norm for tensors:

$$\|\mathcal{Y}\|_* = \sum_{n=1}^{N} \alpha_n \|Y_{(n)}\|_*, \tag{2}$$

where $\alpha_n \geq 0$ $(n = 1, 2, \ldots, N)$, $\sum_{n=1}^{N} \alpha_n = 1$, and $Y_{(n)}$ denotes the mode-n unfolding of $\mathcal{Y}$ (see details in Section 2). With the definition in (2), their model can be written as

$$\begin{aligned} \min_{\mathcal{Y}} \quad & \sum_{n=1}^{N} \alpha_n \|Y_{(n)}\|_* \\ \text{s.t.} \quad & \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F}. \end{aligned} \tag{3}$$

To solve (3), Liu et al. [17] proposed three algorithms (SiLRTC, FaLR-TC, and HaLRTC); Gandy et al. [26] developed two algorithms (Douglas–Rachford splitting technique and ADMM). However, all these methods involve the singular value decomposition (SVD) of $Y_{(n)}$ with high computational complexity. To tackle this issue, Xu et al. [27] proposed a new model to recover a low-rank tensor by parallelly performing low-rank matrix factorizations to the unfoldings of $\mathcal{Y}$ along all modes, i.e.,

$$\begin{aligned} \min_{\mathcal{Y},X,A} \quad & \sum_{n=1}^{N} \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2 \\ \text{s.t.} \quad & \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F}, \end{aligned} \tag{4}$$

where $A = (A_1, A_2, \ldots, A_N)$, $X = (X_1, X_2, \ldots, X_N)$, $\alpha_n \geq 0$ $(n = 1, 2, \ldots, N)$, and $\sum_{n=1}^{N} \alpha_n = 1$. Their method, named low-rank tensor completion by parallel matrix factorization (TMac), has been shown higher time efficiency and better performance than FaLRTC.

Not limited to the low-rank prior, many real-world data, such as natural color image, video, and hyperspectral image, exhibit the spatial piecewise smoothness prior [28,29,38,39]. And in many real-world applications, the factors A_3 and X_3 in (4) have clear physical interpretations. For example, in hyperspectral unmixing [4,40,41], the linear mixing model of a hyperspectral image $\mathcal{Y} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ (d_3 is the number of spectral bands) can be written as

$$Y_{(3)} = A_3 X_3,$$

where $Y_{(3)}$ denotes the mode-3 unfolding of $\mathcal{Y}$, the ith $(i = 1, 2, \ldots, d_3)$ row of $Y_{(3)}$ is the vectorized image of the ith spectral band, $A_3 \in \mathbb{R}^{d_3 \times r_3}$ is a spectral library, each column of which denotes the spectral signatures of the corresponding endmembers, and $X_3 \in \mathbb{R}^{r_3 \times d_1 d_2}$ is the abundance matrix, each row of which denotes the abundances of the corresponding endmembers. Fig. 1 shows the ground truth (includes the spectral library A_3 and the abundance matrix X_3) of the hyperspectral image *Urban*[1]. As observed, each row of the factor X_3 mainly reflects the spatial structure of the original data and is piecewise smooth. Thus, many methods exploit the spatial piecewise smoothness prior of the underlying tensor by boosting the piecewise smoothness of the rows of X_3 [28,29,38,39]. Among them, Ji et al. [28] and Jiang et al. [29] introduced the total variation (TV) regularizer and framelet regularizer into the LRTC problem, respectively. Their model can be

[1] http://lesun.weebly.com/hyperspectral-data-set.html.

Table 1
A comparison of the related LRTC methods and their properties.

Method	Characterization for low-rankness	Characterization for spatial smoothness	Characterization for the third mode's smoothness	The solving algorithm
	Low-rank factorization or minimizing the rank	Constraints on factors or target tensors	Constraints on factors or target tensors	
HaLRTC [17]	minimizing the Tucker rank	–	–	an alternating direction method of multipliers based algorithm
TMac [27]	low-rank matrix factorization	–	–	an alternating minimization based algorithm
MF-TV [28]	low-rank matrix factorization	isotropic TV for factors	–	a block successive upper-bound minimization based algorithm
MF-Framelet [29]	low-rank matrix factorization	framelet for factors	–	a block successive upper-bound minimization based algorithm
TNN [30]	minimizing the tubal rank	–	–	an alternating direction method of multipliers based algorithm
TCTF [31]	low-tubal-rank tensor factorization	–	–	an alternating minimization based algorith
Tubal-Alt-Min [32]	low-tubal-rank tensor factorization	–	–	an alternating minimization based algorithm
GTV [33]	–	generalized TV for target tensors	generalized TV for target tensors	a novel Augmented Lagrange Multiplier based algorithm
SPC-TV/SPC-QV [34]	PARAFAC decomposition	unidirectional TV/QV for factors	unidirectional TV/QV for factors	a hierarchical alternating least squares based algorithm
LRTC-TV-I [35]	minimizing the Tucker rank	anisotropic TV for target tensors	anisotropic TV for target tensors	an alternating direction method of multipliers based algorithm
LRTC-TV-II [35]	Tucker decomposition	anisotropic TV for target tensors	anisotropic TV for target tensors	an alternating direction method of multipliers based algorithm
LRTV-PDS [36]	minimizing the Tucker rank	isotropic TV for target tensors	isotropic TV for target tensors	a primal-dual splitting based algorithm
MDT-LRTC [37]	Tucker decomposition with multi-way delay-embedding transform	–	–	an alternating least squares based algorithm
SMF-LRTC	low-rank matrix factorization	framelet for factors	unidirectional TV for factors	a block successive upper-bound minimization based algorithm

generally written as

$$\begin{aligned} \min_{\mathcal{Y},X,A} \quad & \sum_{n=1}^{3} \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2 + \lambda \mathrm{R}(X_3) \\ \text{s.t.} \quad & \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F}, \end{aligned} \tag{5}$$

where λ denotes the regularization parameter and $\mathrm{R}(X_3)$ is the regularization term. As exhibited in [28] and [29], regularizing the TV/framelet of rows of X_3 effectively enhances the spatial piecewise smoothness of the underlying tensor, leading to a significant improvement. Not limited to this studying route, the other related LRTC methods and their properties are summarized in Table 1.

1.1. Motivations and contributions

It should be noted that many real-world data exhibit piecewise smoothness along the third mode, e.g., the temporal mode in video data and the spectral mode in hyperspectral data, which were not considered in [28] and [29]. Furthermore, as shown in Fig. 1, in hyperspectral unmixing, the piecewise smoothness along the spectral mode of a hyperspectral image is related to the spectral library A_3, each column of which is piecewise smooth.

Generally and mathematically, assuming that the Tucker rank of a three-way tensor $\mathcal{Y} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ is (r_1, r_2, r_3), the mode-3 unfolding $Y_{(3)}$ can be factorized as $Y_{(3)} = A_3 X_3$, where $A_3 \in \mathbb{R}^{d_3 \times r_3}$ and $X_3 \in \mathbb{R}^{r_3 \times d_1 d_2}$ are factor matrices. It is easy to

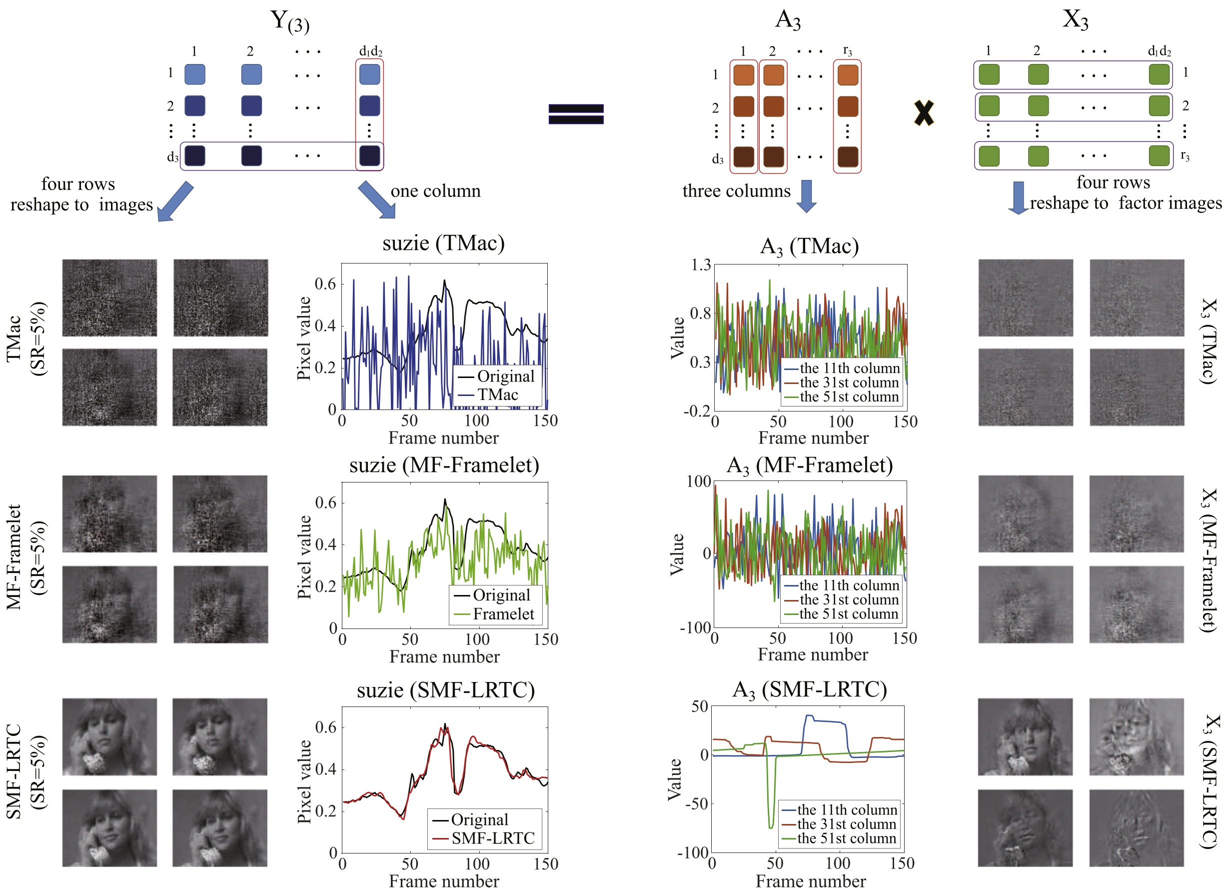

Fig. 2. Illustrations of the effectiveness of the proposed method. The first and second columns are four frames and the intensity of one mode-3 fiber of the reconstructed video data *suzie*, respectively. The third and fourth columns are the intensity of selected three columns of the factor matrix A_3 and four factor images reshaped by the corresponding rows of the factor matrix X_3, respectively. From the second to fourth row: the results obtained by TMac [27], MF-Framelet [29], and the proposed method, respectively.

see that each column of $Y_{(3)}$ is the linear combination of all columns of A_3 and each row of $Y_{(3)}$ is the linear combination of all rows of X_3, i.e., columns of A_3 (rows of X_3) can be viewed as a basis of the column (row) space of $Y_{(3)}$. As piecewise smooth bases tend to generate piecewise smooth data, we can enhance the smoothness of columns of $Y_{(3)}$ (the third mode of $\mathcal{Y}$) by boosting the piecewise smoothness of the columns of A_3 and enhance the smoothness of rows of $Y_{(3)}$ (spatial mode of $\mathcal{Y}$) by boosting the piecewise smoothness of the rows of X_3. Motivated by this, we propose a novel LRTC model simultaneously considering the low-rankness and the piecewise smoothness priors of the underlying tensor $\mathcal{Y}$. The proposed model, named low-rank tensor completion by smooth matrix factorization (SMF-LRTC), is formulated as

$$\begin{aligned}\min_{\mathcal{Y},X,A} \quad & \sum_{n=1}^{N} \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2 + \lambda_1 \|W X_3^T\|_{1,1} + \lambda_2 \|\nabla_y A_3\|_{1,1} \\ \text{s.t.} \quad & \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F},\end{aligned} \tag{6}$$

where λ_1 and λ_2 are regularization parameters, W denotes the framelet transformation matrix, ∇_y indicates the vertical derivative operator, and the $\ell_{1,1}$-norm of a matrix is defined as the sum of the absolute values of its all elements (see details in Section 3.1). The major difference between the proposed method and the method of Jiang et al. [29] lies on the consideration of the piecewise smoothness prior along the third mode of the underlying tensor. The distinctions and relations between the proposed method and the other related LRTC methods are summarized in Table 1.

To better comprehend our motivation, in Fig. 2, we provide an example conducted on the video data *suzie* with a low sampling rate (SR = 5%), it means that only 5% entries are known. The left two columns suggest that (1) the results by the proposed method have considerably higher visual quality than those by TMac [27] and the MF-Framelet [29]; (2) the mode-3 fibers by the proposed method are much smoother and closer to the original ones comparing with those by the other two methods. The right two columns indicate that (1) the columns of A_3 by the proposed method are much smoother than those by the compared methods; (2) owing to the smoothness constraint on the factor matrix A_3, the factor images (the rows of X_3) by the proposed method contain more details and geometrical features than those by the compared methods. The above observations from Fig. 2 are consistent with the fore discussion of our motivation regarding the piecewise smoothness, i.e., smoothness constraint on X_3 can lead the spatial mode of the recovered data to be smooth and smoothness constraint on A_3 can make the third mode of the recovered data to be smooth.

The contributions of this paper are mainly three folds: (1) we observe that the smoothness of the underlying tensor can be enhanced by boosting the piecewise smoothness of the factor matrices; (2) based on the above observation, we propose a LRTC model using smooth matrix factorization, which can simultaneously exploits the low-rankness and the piecewise smoothness priors of the underlying tensor; (3) we develop an efficient block successive upper-bound minimization (BSUM)-based algorithm to solve the proposed model, numerical experiments demonstrate that our method can significantly improve the quality of the results.

1.2. Organization of the paper

The outline of this paper is as follows. Section 2 reviews some preliminary knowledge about tensor, framelet, the proximal operator, and the BSUM algorithm. Section 3 gives model formulation and an efficient BSUM-based solver with convergence analysis. Section 4 evaluates the performance of the proposed method and compares the results with three competing methods. Section 5 concludes this paper.

2. Preliminary

In this paper, we denote vectors as lowercase letters (e.g., a), matrices as uppercase letters (e.g., A), and tensors as calligraphic letters (e.g., $\mathcal{A}$). Below, we review some preliminary knowledge that will be used in this paper.

2.1. Tensor basics

In this section, we partially adopt the nomenclatures of Kolda and Bader's [42] review on tensor.

A *fiber* of a tensor $\mathcal{A}$ is a vector generated by fixing every index but one. The *mode-n fibers* are all vectors $\mathcal{A}(i_1, \ldots, i_{n-1}, :, i_{n+1}, \ldots, i_N)$ for all $i_1, i_2, \ldots, i_N$.

A *slice* of a tensor $\mathcal{A}$ is a matrix generated by fixing every index but two. For a three-way tensor $\mathcal{A}$, *horizontal slices* are all matrices $\mathcal{A}(i_1, :, :)$ for all i_1, i_2, i_3, *lateral slices* are all matrices $\mathcal{A}(:, i_2, :)$ for all i_1, i_2, i_3, and *frontal slices* are all matrices $\mathcal{A}(:, :, i_3)$ for all i_1, i_2, i_3.

The *Frobenius norm* of an N-way tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$ is defined as

$$\|\mathcal{A}\|_F = \left(\sum_{i_1=1}^{d_1} \sum_{i_2=1}^{d_2} \cdots \sum_{i_N=1}^{d_N} |a_{i_1 i_2 \ldots i_N}|^2 \right)^{\frac{1}{2}},$$

where $a_{i_1 i_2 \ldots i_N}$ is the $(i_1, i_2, \ldots, i_N)$-th element of the tensor $\mathcal{A}$. Furthermore, $\|\mathcal{A}\|_F$ also can be written as $\sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$ with the following definition of *inner product* of two same-sized tensors $\mathcal{A}$ and $\mathcal{B}$:

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i_1, i_2, \ldots, i_N} a_{i_1 i_2 \ldots i_N} \cdot b_{i_1 i_2 \ldots i_N}.$$

The *mode-n unfolding* of a tensor $\mathcal{A}$ is denoted as $A_{(n)} \in \mathbb{R}^{d_n \times \prod_{i \neq n} d_i}$, whose (i_n, j)th element maps to the $(i_1, i_2, \ldots, i_N)$th element of $\mathcal{A}$, where

$$j = 1 + \sum_{k=1, k \neq n}^{N} (i_k - 1) J_k \quad \text{with} \quad J_k = \sum_{m=1, m \neq n}^{k-1} d_m.$$

The inverse operator of unfolding is denoted as "fold", i.e., $\mathcal{A} = \text{fold}_n(A_{(n)})$.

The *Tucker rank* (*n-rank*) of a tensor $\mathcal{A}$ is defined as the following array.

$$\text{rank}_t(\mathcal{A}) = \big(\text{rank}(A_{(1)}), \text{rank}(A_{(1)}), \ldots, \text{rank}(A_{(N)})\big).$$

Interested readers can refer to [42] for a more extensive overview.

2.2. Framelet

A system $X \subset L_2(\mathbb{R})$ is called a tight frame of $L_2(\mathbb{R})$ if

$$f = \sum_{g \in X} \langle f, g \rangle g, \quad \forall f \in L_2(\mathbb{R}),$$

where $\langle \cdot, \cdot \rangle$ is the inner product of $L_2(\mathbb{R})$. A wavelet (also called affine) system $X(\Psi)$ is defined by the following collection of dilations and shifts of a finite set $\Psi = \{\psi_1, \psi_2, \ldots, \psi_r\} \subset L^2(\mathbb{R})$.

$$X(\Psi) = \left\{2^{k/2} \psi_l (2^k \cdot -j) : \psi_l \in \Psi; 1 \leq l \leq r; k, j, l \in \mathcal{Z}\right\}.$$

Particularly, ψ_l $(l = 1, 2, \ldots, r)$ are called the (tight) framelets and $X(\Psi)$ is called a tight wavelet frame if $X(\Psi)$ is a tight frame for $L_2(\mathbb{R})$ simultaneously.

In the discrete setting, a discrete image f is considered as the coefficients $\{f_i = \langle f, \psi(\cdot\text{-}i)\rangle\}$ up to a dilation, where φ is a refinable function associated with the framelet system. Then the L-level discrete framelet decomposition of f is the coefficients $\{f = \langle f, 2^{\text{-}L/2}\psi_i(2^{\text{-}L}\cdot\text{-}j)\rangle\}$ at a prescribed coarsest level L, and the framelet coefficients are

$$\{f = \langle f, 2^{\text{-}L/2}\psi_i(2^{\text{-}L}\cdot\text{-}j)\rangle,\ 1 \le i \le r^2 - 1\} \quad \text{for } 0 \le l \le L.$$

This decomposition can be written into a linear operator applied to the discrete image (vector form) $f \in \mathbb{R}^{mn}$, i.e., Wf with $W \in \mathbb{R}^{k\times mn}$. By the unitary extension principle (UEP) of [43], $W^T W = I$, where W^T is the inverse framelet transform. Thus the row vectors of W form a tight frame system in $\mathbb{R}^{mn}$. Our implementations mainly use the piecewise linear B-spline framelets constructed by [43]. A detailed description about framelet can be found in [39,43,44].

2.3. Proximal operator

The proximal operator [45] of a given convex function $f(x)$ is defined as

$$\text{prox}_f(y) = \arg\min_x \left\{ f(x) + \frac{\rho}{2}\|x - y\|^2 \right\}, \tag{7}$$

where $\rho > 0$ is a constant. There are two attractive conclusions about the proximal operator. One is that $\min_x\{f(x)\}$ is equivalent to $\min_{x,y}\{f(x) + \rho/2\|x - y\|^2\}$, and another is that (7) is strongly convex with respect to x when $f(x)$ is convex. Thus, proximal algorithms minimize $\{f(x)\}$ by iteratively solving $\text{prox}_f(x^k)$, where x^k is the latest update of x.

2.4. Block successive upper-bound minimization algorithm

Assuming that the feasible set X is the Cartesian product of n closed convex sets: $X = X_1 \times X_2 \times \cdots \times X_n$ with $X_i \in \mathbb{R}^{m_i}$, the optimization variable $x \in X$ can be decomposed as $x = (x_1, x_2, \ldots, x_n)$ with $x_i \in X_i$ for $i = 1, 2, \ldots, n$. Considering the problem

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in X, \end{aligned} \tag{8}$$

the BSUM algorithm updates only the single block of variables in each iteration, i.e., (8) can be iteratively solved by

$$\begin{cases} \text{Step } 1: \ x_1^{k+1} = \arg\min\limits_{x_1} \quad \text{prox}_f(x_1^k), \\ \text{Step } 2: \ x_2^{k+1} = \arg\min\limits_{x_2} \quad \text{prox}_f(x_2^k), \\ \qquad\qquad \vdots \\ \text{Step } n: \ x_n^{k+1} = \arg\min\limits_{x_n} \quad \text{prox}_f(x_n^k). \end{cases} \tag{9}$$

Details about BSUM algorithm can be found in [46].

3. Proposed model and algorithm

3.1. Proposed model

Considering a three-way tensor $\mathcal{Y} \in \mathbb{R}^{d_1\times d_2\times d_3}$, the proposed model (6) is as follows:

$$\min f(X, A, \mathcal{Y}) = \sum_{n=1}^{3} \frac{\alpha_n}{2} \parallel Y_{(n)} - A_n X_n\|_F^2 + \lambda_1 \parallel W X_3^T\|_{1,1} + \lambda_2 \parallel \nabla_y A_3\|_{1,1} + \iota(\mathcal{Y}), \tag{10}$$

where λ_1 and λ_2 are regularization parameters, $Y_{(n)}$ denotes the mode-n unfolding of $\mathcal{Y}$, $A = (A_1, A_2, A_3)$, $X = (X_1, X_2, X_3)$, $\alpha_n \ge 0$ $(n = 1, 2, 3)$, $\sum_{n=1}^{3}\alpha_n = 1$, and $\iota(\cdot)$ is the following indicator function:

$$\iota(\mathcal{Y}) := \begin{cases} 0, & \text{if } \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F}, \\ \infty, & \text{otherwise}. \end{cases}$$

In the fidelity term $\sum_{n=1}^{3}\frac{\alpha_n}{2}\|Y_{(n)} - A_n X_n\|_F^2$, the Tucker rank of a tensor $\mathcal{Y}$ is (r_1, r_2, r_3), $A_n \in \mathbb{R}^{d_n\times r_n}$ and $X_n \in \mathbb{R}^{r_n\times s_n}$ with $s_n = \prod_{i\neq n} d_i$ are the factor matrices. This term is used to promote the low-rankness of the underlying tensor.

In the framelet-based regularization term $\|WX_3^T\|_{1,1}$, W indicates the framelet transformation matrix satisfying $W^T W = I$. As pointed out in [29,47], a smooth gray-level image have good sparse approximations in framelet domain. It implies that for a gray-level image (vector form) $f \in \mathbb{R}^{mn}$, $\|Wf\|_1$ is able to enhance its piecewise smoothness. Note that X_3^T can be rewritten as

$$X_3^T = \left((x_3^1)^T, (x_3^2)^T, \dots, (x_3^{r_3})^T\right),$$

where $x_3^i \in \mathbb{R}^{1\times d_1 d_2} (i = 1, 2, \dots, r_3)$ indicates the i-th row of X_3. In detail, each $(x_3^i)^T$ can be viewed as a vectorized factor image and the framelet decomposition operation $W(x_3^i)^T$ acts on each x_3^i independently, but it can be concisely calculated by WX_3^T with the good structure of X_3. Therefore, $\|WX_3^T\|_{1,1}$ can enhance smoothness of the rows of X_3. As analysis in Section 1.1 and [4,28,29], the piecewise smoothness along the spatial mode of the underlying tensor $\mathcal{Y}$ can be enhanced by this regularization term.

In the TV-based regularization term $\|\nabla_y A_3\|_{1,1}$, ∇_y indicates the vertical derivative operator, and $\nabla_y A_3$ can be calculated by DA_3, where D is the first-order difference matrix

$$D = \begin{pmatrix} -1 & 1 & \dots & 0 & 0 \\ 0 & -1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -1 & 1 \\ 1 & 0 & \dots & 0 & -1 \end{pmatrix}.$$

This term aims to enhance the piecewise smoothness along the third mode of the underlying tensor $\mathcal{Y}$ by boosting the piecewise smoothness of the columns of A_3.

In summary, these two smoothness constraints ensure the piecewise smoothness of the rows of X_3 and the columns of A_3. As a result, the piecewise smoothness along all modes of the underlying tensor $\mathcal{Y}$ can be enhanced.

3.2. Proposed algorithm

In this section, we develop an BSUM-based algorithm to solve (10).

It is clear that although the objective function of (10) is not jointly convex for $(X, A, \mathcal{Y})$, it is convex with respect to $X, A, \mathcal{Y}$ independently. Let $\mathcal{Z} = (X, A, \mathcal{Y})$, naturally $\mathcal{Z}^k = (X^k, A^k, \mathcal{Y}^k)$, with utilization of the proximal operator (7) and (10), (6) can be solved through the following problem:

$$\mathcal{Z}^{k+1} = \arg\min_{\mathcal{Z}} \ h(\mathcal{Z}, \mathcal{Z}^k) = \arg\min_{\mathcal{Z}} \ f(\mathcal{Z}) + \frac{\rho}{2}\|\mathcal{Z} - \mathcal{Z}^k\|_F^2, \tag{11}$$

where $\rho > 0$ is the proximal parameter. Let $\mathcal{Z}_1^k = (X^k, A^k, \mathcal{Y}^k)$, $\mathcal{Z}_2^k = (X^{k+1}, A^k, \mathcal{Y}^k)$, and $\mathcal{Z}_3^k = (X^{k+1}, A^{k+1}, \mathcal{Y}^k)$, with utilization of the BSUM algorithm, (11) can be rewritten as

$$\begin{cases} \text{Step 1}: \ X^{k+1} = \arg\min\limits_{X} \left\{h_1(X, \mathcal{Z}_1^k)\right\} = \arg\min\limits_{X} \left\{f(X, A^k, \mathcal{Y}^k) + \frac{\rho}{2}\|X - X^k\|_F^2\right\}, \\ \text{Step 2}: \ A^{k+1} = \arg\min\limits_{A} \left\{h_2(A, \mathcal{Z}_2^k)\right\} = \arg\min\limits_{A} \left\{f(X^{k+1}, A, \mathcal{Y}^k) + \frac{\rho}{2}\|A - A^k\|_F^2\right\}, \\ \text{Step 3}: \ \mathcal{Y}^{k+1} = \arg\min\limits_{\mathcal{Y}} \left\{h_3(\mathcal{Y}, \mathcal{Z}_3^k)\right\} = \arg\min\limits_{\mathcal{Y}} \left\{f(X^{k+1}, A^{k+1}, \mathcal{Y}) + \frac{\rho}{2}\|\mathcal{Y} - \mathcal{Y}^k\|_F^2\right\}. \end{cases} \tag{12}$$

It is easy to note that the X- and A- subproblems can be decomposed into three independent problems. Thus, it is clear that (12) has the following solutions.

Step 1 (X-subproblems):

$$X_n^{k+1} = \begin{cases} \left(\alpha_n (A_n^k)^T A_n^k + \rho I_1\right)^\dagger \left(\alpha_n (A_n^k)^T Y_{(n)}^k + \rho X_n^k\right), \quad n = 1, 2, \\ \arg\min\limits_{X_3} \left\{\frac{\alpha_3}{2}\|Y_{(3)}^k - A_3^k X_3\|_F^2 + \lambda_1\|WX_3^T\|_{1,1} + \frac{\rho}{2}\|X_3 - X_3^k\|_F^2\right\}, \quad n = 3, \end{cases} \tag{13}$$

Step 2 (A-subproblems):

$$A_n^{k+1} = \begin{cases} \left(\alpha_n Y_{(n)}^k (X_n^{k+1})^T + \rho A_n^k\right)\left(\alpha_n X_n^{k+1} (X_n^{k+1})^T + \rho I_2\right)^\dagger, \quad n = 1, 2, \\ \arg\min\limits_{A_3} \left\{\frac{\alpha_3}{2}\|Y_{(3)}^k - A_3 X_3^{k+1}\|_F^2 + \lambda_2\|DA_3\|_{1,1} + \frac{\rho}{2}\|A_3 - A_3^k\|_F^2\right\}, \quad n = 3, \end{cases} \tag{14}$$

Step 3 ($\mathcal{Y}$-subproblems):

$$\mathcal{Y}^{k+1} = \mathcal{P}_{\Omega^c}\left(\sum_{n=1}^{3}\alpha_n \text{fold}_n\left(\frac{\alpha_n A_n^{k+1}X_n^{k+1} + \rho Y_{(n)}^k}{\alpha_n + \rho}\right)\right) + \mathcal{F}, \tag{15}$$

where $\mathcal{F}$ denotes the observed data and $(\cdot)^\dagger$ indicates the Moore–Penrose pseudoinverse of $(\cdot)$. The complexity of computing X_n $(n = 1, 2)$ is $\mathcal{O}(r_n^2 d_n + r_n d_1 d_2 d_3)$, the complexity of computing A_n $(n = 1, 2)$ is $\mathcal{O}(r_n^2 d_n + r_n d_1 d_2 d_3)$, and the complexity of computing $\mathcal{Y}$ is $\mathcal{O}(\sum_{n=1}^3 r_n d_1 d_2 d_3)$.

Next, we give the details for solving the X_3- and A_3- subproblems. For the X_3- subproblem in Step 1, it is easy to find that the problem fits the framework of the alternating direction method of multipliers (ADMM) [48]. Thus, we rewrite the X_3 subproblem as the following equivalent constrained problem

$$\begin{aligned} \min_{X_3, V} \quad & \frac{\mu_1}{2}\|Y_{(3)}^k - A_3^k X_3\|_F^2 + \|V\|_{1,1} + \frac{\rho_x}{2}\|X_3 - X_3^k\|_F^2 \\ \text{s.t.} \quad & V = W X_3^T, \end{aligned} \tag{16}$$

where $\mu_1 = \alpha_3/\lambda_1$ and $\rho_x = \rho/\lambda_1$. The concise form of the augmented Lagrangian function of (16) can be expressed as

$$L_{\beta_1}(X_3, V, \Lambda) = \frac{\mu_1}{2}\|Y_{(3)}^k - A_3^k X_3\|_F^2 + \|V\|_{1,1} + \frac{\rho_x}{2}\|X_3 - X_3^k\|_F^2 + \frac{\beta_1}{2}\left\|W X_3^T - V + \frac{\Lambda}{\beta_1}\right\|_F^2 + C, \tag{17}$$

where Λ denotes the Lagrange multiplier and $\beta_1 > 0$ is the penalty parameter. Then, (17) can be updated through alternating direction as

$$\begin{cases} X_3^{k+1,p+1} = \left(\mu_1 (A_3^k)^T A_3^k + (\rho_x + \beta_1) I\right)^\dagger \left(\mu_1 (A_3^k)^T Y_{(3)}^k + \rho_x X_3^k + \beta_1\left[W^T\left(V^p - \dfrac{\Lambda^p}{\beta_1}\right)\right]^T\right), \\ V^{p+1} = S_{\frac{1}{\beta_1}}\left(W\left(X_3^{k+1,p+1}\right)^T + \dfrac{\Lambda^p}{\beta_1}\right), \\ \Lambda^{p+1} = \Lambda^p + \beta_1\left(W\left(X_3^{k+1,p+1}\right)^T - V^{p+1}\right), \end{cases} \tag{18}$$

where $S_\alpha(\cdot)$ denotes the component-wise soft thresholding operator with threshold α, i.e.,

$$[S_\alpha(x)]_{ij} = \text{sgn}(x_{ij}) \max\left\{(|x_{ij}| - \alpha), 0\right\}. \tag{19}$$

The complexity of computing X_3 is $\mathcal{O}(r_3^2 d_3 + r_3 d_1 d_2 d_3 + r_3 d_1^2 d_2^2)$.

For the A_3-subproblem in Step 2, similar to the X_3-subproblem, we rewrite the A_3-subproblem as

$$\begin{aligned} \min_{A_3, M} \quad & \frac{\mu_2}{2}\|Y_{(3)}^k - A_3 X_3^{k+1}\|_F^2 + \|M\|_{1,1} + \frac{\rho_a}{2}\|A_3 - A_3^k\|_F^2 \\ \text{s.t.} \quad & M = D A_3, \end{aligned} \tag{20}$$

where $\mu_2 = \alpha_3/\lambda_2$ and $\rho_a = \rho/\lambda_2$. The concise form of the augmented Lagrangian function of (20) can be written as

$$L_{\beta_2}(A_3, M, \Theta) = \frac{\mu_2}{2}\|Y_{(3)}^k - A_3 X_3^{k+1}\|_F^2 + \|M\|_{1,1} + \frac{\rho_a}{2}\|A_3 - A_3^k\|_F^2 + \frac{\beta_2}{2}\left\|D A_3 - M + \frac{\Theta}{\beta_2}\right\|_F^2 + C, \tag{21}$$

where Θ denotes Lagrange multiplier and $\beta_2 > 0$ is the penalty parameter. To minimize (21), we can update A, M, and Θ as

$$\begin{cases} A_3^{k+1,p+1} \in \arg\min\limits_{A_3} L_{\beta_2}(A_3, M^p, \Theta^p), \\ M^{p+1} = S_{\frac{1}{\beta_2}}\left(D A_3^{k+1,p+1} + \dfrac{\Theta^p}{\beta_2}\right), \\ \Theta^{p+1} = \Theta^p + \beta_2\left(D A_3^{k+1,p+1} - M^{p+1}\right). \end{cases} \tag{22}$$

For the A_3-subproblem in (22), we solve the following problem:

$$\arg\min_{A_3}\left\{\frac{\mu_2}{2}\|Y_{(3)}^k - A_3 X_3^{k+1}\|_F^2 + \frac{\rho_a}{2}\|A_3 - A_3^k\|_F^2 + \frac{\beta_2}{2}\left\|D A_3 - M^p + \frac{\Theta^p}{\beta_2}\right\|_F^2\right\}, \tag{23}$$

which can be solved via the classical Sylvester matrix equation

$$\mu_2 A_3 X_3^{k+1}(X_3^{k+1})^T + \rho_a A_3 + \beta_2 D^T D A_3 = \mu_2 Y_{(3)}^k (X_3^{k+1})^T + \rho_a A_3^k + \beta_2 D^T\left(M^p - \frac{\Theta^p}{\beta_2}\right). \tag{24}$$

To solve (24), we develop the following theorem.

Theorem 1. *Assuming that $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{n\times n}$, and $X, C \in \mathbb{R}^{m\times n}$. The following classical Sylvester matrix equation*

$$AX + XB = C \tag{25}$$

has a unique solution if only if $G = I_n \otimes A + B^T \otimes I_m$ is a invertible matrix, where $\otimes$ denotes the Kronecker product. Especially, if matrices A and B satisfy

$$A = U_1\Lambda_1 U_1^T, \qquad B = U_2\Lambda_2 U_2^T,$$

where Λ_1 and Λ_2 are diagonal matrices; U_1 and U_2 are unitary matrices. Then the unique solution is

$$X = U_1\big((1./T).*(U_1^T C U_2)\big)U_2^T,$$

where "." represents the component-wise multiplication, "./" represents the component-wise division, and $T = \big(\mathrm{diag}^{2}(\Lambda_1), \mathrm{diag}(\Lambda_1), \ldots, \mathrm{diag}(\Lambda_1)\big)^T + \big(\mathrm{diag}(\Lambda_2), \mathrm{diag}(\Lambda_2), \ldots, \mathrm{diag}(\Lambda_2)\big)$.*

Proof. Using the Kronecker product notations, (25) can be rewritten as

$$(I_n \otimes A + B^T \otimes I_m)\mathrm{vec}(X) = \mathrm{vec}(C), \tag{26}$$

where "vec(·)" refers to a vector by lexicographical ordering of the entries in a matrix. Since $A = U_1\Lambda_1 U_1^T$ and $B = U_2\Lambda_2 U_2^T$, we rewrite (26) as

$$\begin{aligned} &\big(I_n \otimes (U_1\Lambda_1 U_1^T) + (U_2\Lambda_2 U_2^T) \otimes I_m\big)\mathrm{vec}(X) = \mathrm{vec}(C) \\ \Longleftrightarrow\quad &\big((U_2 I_n U_2^T) \otimes (U_1\Lambda_1 U_1^T) + (U_2\Lambda_2 U_2^T) \otimes (U_1 I_m U_1^T)\big)\mathrm{vec}(X) = \mathrm{vec}(C) \\ \Longleftrightarrow\quad &\big((U_2 \otimes U_1)(I_n \otimes \Lambda_1 + \Lambda_2 \otimes I_m)(U_2^T \otimes U_1^T)\big)\mathrm{vec}(X) = \mathrm{vec}(C), \end{aligned} \tag{27}$$

then, vec(X) can be expressed as

$$\begin{aligned} \mathrm{vec}(X) &= \big((U_2 \otimes U_1)(I_n \otimes \Lambda_1 + \Lambda_2 \otimes I_m)^{-1}(U_2^T \otimes U_1^T)\big)\mathrm{vec}(C) \\ &= (U_2 \otimes U_1)\big(\mathrm{diag}(\mathrm{vec}(1./T))\big)(U_2^T \otimes U_1^T)\mathrm{vec}(C) \\ &= (U_2 \otimes U_1)\big(\mathrm{diag}(\mathrm{vec}(1./T))\big)\mathrm{vec}(U_1^T C U_2) \\ &= (U_2 \otimes U_1)\mathrm{vec}\big((1./T).*(U_1^T C U_2)\big). \end{aligned} \tag{28}$$

Thus, the unique solution of (25) is $X = U_1\big((1./T).*(U_1^T C U_2)\big)U_2^T$. □

In (24), the matrix $D^T D$ is a circulant matrix, which can diagonalized via one-dimensional Fourier transformation; meanwhile, the matrix X_3^{k+1} can be diagonalized by using the singular value decomposition. Letting

$$X_3^{k+1} = U\Sigma V^*, \qquad D^T D = F^*\Psi^2 F,$$

and

$$K = \mu_2 Y_{(3)}^k (X_3^{k+1})^T + \rho_a A_3^k + \beta_2 D^T\left(M^p - \frac{\Theta^p}{\beta_2}\right).$$

With Theorem 1, the solution of (24) can be expressed as

$$A_3^{k+1,p+1} = F^*\big(1./T.*(FKU)\big)U^*, \tag{29}$$

where $T = \mu_2\big(\mathrm{diag}(\Sigma^2), \mathrm{diag}(\Sigma^2), \ldots, \mathrm{diag}(\Sigma^2)\big)^T + \beta_2\big(\mathrm{diag}(\Psi^2), \mathrm{diag}(\Psi^2), \ldots, \mathrm{diag}(\Psi^2)\big) + \rho_a \mathrm{ones}(d_3, r_3)$[3].

The complexity of computing A_3 is $\mathcal{O}\big(r_3^2 d_3 + r_3 d_1 d_2 d_3 + r_3 d_3^2 + r_3 d_1 d_2 \min(r_3, d_1 d_2)\big)$. Thus, the complexity of computing all the variables at each iteration is $\mathcal{O}\big(\sum_{n=1}^3 (r_n^2 d_n + r_n d_1 d_2 d_3) + r_3(d_1^2 d_2^2 + d_3^2)\big)$.

As the Tucker rank r is an important parameter, we considered the following heuristic rank-increasing scheme to adjust it automatically.

Rank-increasing scheme. This scheme starts with an underestimated rank, i.e., $r^0 = (r_1, r_2, r_3) \le \mathrm{rank}_t(\mathcal{Y})$, where $\mathcal{Y} \in \mathbb{R}^{d_1\times d_2\times d_3}$ is the underlying tensor. Following [27,49], we increase r_n to $\min(r_n + \Delta r_n, r_n^{\max})$ at iteration $k+1$ if

$$\left|1 - \frac{\|\mathcal{P}_{\Omega^c}(A_n^{k+1}X_n^{k+1})\|_F}{\|\mathcal{P}_{\Omega^c}(A_n^k X_n^k)\|_F}\right| < 10^{-2}, \ \ n = 1, 2, 3,$$

which implies "slow" progress in the r_n dimensional space along the nth mode. Here, Δr_n is a positive integer and $r_n^{\max}$ is the maximal rank estimate. If the r_n is increased at iteration $k+1$, we update A_n^{k+1} to $[A_n^{k+1}, \mathrm{rand}(d_n, \Delta r_n)]$[4] and X_n^{k+1}

2 When X is a diagonal matrix, diag(X) is defined as a column vector whose elements are the diagonal elements of X. When $\mathbf{x}$ is a column vector, diag($\mathbf{x}$) is defined as a diagonal matrix whose diagonal elements are the elements of $\mathbf{x}$.

3 ones(m, n) is an $m \times n$ matrix whose elements are all 1.

4 rand(m, n) is an $m \times n$ random matrix whose elements are generated from the uniform distribution on the interval (0,1).

to $[X_n^{k+1}; \mathrm{rand}(\Delta r_n, s_n)]$, i.e., adding Δr_n randomly generated columns to A_n^{k+1} and Δr_n randomly generated rows to X_n^{k+1}, respectively.

Finally, we show the pseudocode of BSUM-based algorithm for the proposed model (6) in Algorithm 1.

Algorithm 1 BSUM-based optimization algorithm for proposed model (6).

Input: the observed tensor $\mathcal{F}$, the set of index of observed entries Ω,the initial Tucker rank $r^0 = (r_1^0, r_2^0, \ldots, r_n^0)$,$\Delta r = (\Delta r_1^0, \Delta r_2^0, \ldots, \Delta r_n^0)$,$r^{\max} = (r_1^{\max}, r_2^{\max}, \ldots, r_n^{\max})$,parameters $\mu_1,\ \beta_1,\ \mu_2,\ \beta_2,$ and ρ.
Output: The completed tensor $\mathcal{Y}$.
1: Initialization: $A_n^0 = \mathrm{rand}(d_n \times r_n)$, $X_n^0 = \mathrm{rand}(r_n \times \prod_{i\neq n} d_i)$ with $n = 1, 2, 3$, $\mathcal{Y} = \mathcal{P}_\Omega(\mathcal{F})$, and $N_{\max}$.
2: **while** not converged and $k < N_{\max}$ **do**
3: Update $X_n (n = 1, 2)$ via (13) and update X_3 via (18).
4: Update $A_n (n = 1, 2)$ via (14) and update A_3 via (22).
5: Update $\mathcal{Y}$ via (15).
6: **end while**
7: **return** $\mathcal{Y}$.

3.3. Convergence analysis

In this section, we discuss the convergence of the proposed algorithm. We recall the convergence result of BSUM [46], i.e., the core scheme of the proposed algorithm.

Lemma 1. *Suppose $\mathcal{X}$ is the feasible set, given the problem $\min f(x)$ and subject to $x \in \mathcal{X}$, and assume that $u(x, x^{k-1})$ is an approximation of $f(x)$ at the $(k-1)$th iteration, which satisfies the following conditions:*

$$
\begin{aligned}
&u_i(y_i, y) = f(y),\ \ \forall y \in \mathcal{X}, \forall i;\\
&u_i(x_i, y) \geq f(y_1, \ldots, y_{i-1}, x_i, y_{i+1}, \ldots, y_n), \forall x_i \in \mathcal{X}_i, \forall y \in \mathcal{X}, \forall i;\\
&u_i'(x_i, y; d_i)\ |_{x_i=y_i} = f'(y; d), \forall d = (0, \ldots, d_i, \ldots, 0) \text{ s.t. } y_i + d_i \in \mathcal{X}_i,\ \ \forall i;\\
&u_i(x_i, y) \text{ is continuous in } (x_i, y),\ \ \forall i,
\end{aligned}
$$

where $u_i(x_i, y)$ is the subproblem with respect to the i-th block and $f'(y; d)$ is the direction derivative of f at the point y in direction d. Suppose $u_i(x_i, y)$ is quasi-convex in x_i for $i = 1, \ldots, n$. Furthermore, assume that each subproblem $\operatorname{argmin} u_i(x_i, x^{k-1})$, s.t. $x \in \mathcal{X}_i$ has a unique solution for any point $x^{k-1} \in \mathcal{X}$. Then, the iterates generated by the BSUM algorithm converge to the set of coordinatewise minimum of f. In addition, if $f(\cdot)$ is regular at z, then z is a stationary point.

Next, we illustrate the convergence of the proposed algorithm for the model (6).

Theorem 2. *The sequence generated by (12) converges to the set of the coordinate-wise minimizers.*

Proof. It is easy to verify that $h(\mathcal{Z}, \mathcal{Z}^k)$ is an approximation and a global upper bound of $f(\mathcal{Z})$ at the k-th iteration, which satisfies the following conditions:

$$
\begin{aligned}
&h_i(\mathcal{Z}_i, \mathcal{Z}) = f(\mathcal{Z}),\ \ \forall \mathcal{Z}, i = 1, 2, 3,\\
&h_i(\bar{\mathcal{Z}}_i, \mathcal{Z}) \geq f(\mathcal{Z}_1, \ldots, \bar{\mathcal{Z}}_i, \ldots, \mathcal{Z}_3),\ \ \forall \bar{\mathcal{Z}}_i, \forall \mathcal{Z}, i = 1, 2, 3,\\
&h_1'(\bar{\mathcal{Z}}_1, \mathcal{Z}; \mathcal{D}_1)\ |_{\bar{\mathcal{Z}}_1=\mathcal{Z}_1} = f'(\mathcal{Z}; \mathcal{D}^1),\ \ \forall \mathcal{D}^1 = (\mathcal{D}_1, 0, 0),\\
&h_2'(\bar{\mathcal{Z}}_2, \mathcal{Z}; \mathcal{D}_2)\ |_{\bar{\mathcal{Z}}_2=\mathcal{Z}_2} = f'(\mathcal{Z}; \mathcal{D}^2),\ \ \forall \mathcal{D}^2 = (0, \mathcal{D}_2, 0),\\
&h_3'(\bar{\mathcal{Z}}_3, \mathcal{Z}; \mathcal{D}_3)\ |_{\bar{\mathcal{Z}}_3=\mathcal{Z}_3} = f'(\mathcal{Z}; \mathcal{D}^3),\ \ \forall \mathcal{D}^3 = (0, 0, \mathcal{D}_3),\\
&h_i(\bar{\mathcal{Z}}_i, \mathcal{Z}) \text{ is continuous in } (\bar{\mathcal{Z}}_i, \mathcal{Z})\ i = 1, 2, 3,
\end{aligned}
$$

where $\mathcal{Z} = (X, A, \mathcal{Y})$ and $\mathcal{Z}_i$ equals to $X, A, \mathcal{Y}$ for $i = 1, 2, 3,$ respectively. In addition, the subproblem h_i, $(i = 1, 2, 3)$ is strictly convex with respect to X, A, and $\mathcal{Y}$ respectively and thus each subproblem has a unique solution. Therefore, all assumptions in Lemma 1 are satisfied. □

4. Numerical experiments

In this section, we evaluate the performance of the proposed method[5] on completing three kinds of three-way tensors: video, hyperspectral image, and MRI. The peak signal to noise rate (PSNR) and the structural similarity index (SSIM) [50] are

[5] The code of SMF-LRTC is available at https://github.com/uestctensorgroup/code_SMFLRTC.

adopted to measure the quality of the reconstructed results. The compared LRTC methods include: TMac[6] [27], MF-TV[7] [28], and MF-Framelet[8] [29], representing state-of-the-arts for matrix factorization based method; SPC-QV[9] [34], representing state-of-the-arts for PARAFAC decomposition based method; LRTC-TV-I[10] [35], representing state-of-the-arts for Tucker decomposition based method.

The stopping criterion of all methods is the relative change (RelCha) of two successive reconstructed tensors, which can be expressed as RelCha $= \frac{\|\mathcal{Y}^{k+1}-\mathcal{Y}^{k}\|_F}{\|\mathcal{Y}^{k}\|_F} < \varepsilon$, where ε is a tolerance.

In all experiments, TMac [27], MF-TV [28], and MF-Framelet [29] are implemented using the parameters reported in [29]. SPC-QV [34] and LRTC-TV-I [35] are implemented using the parameters reported in their paper. For the proposed method, the parameters are set as: the proximal parameter $\rho = 0.01$, the first regularization parameter $\mu_1 = 10$, the second regularization parameter $\mu_2 = 100$, the first penalty parameter $\beta_1 = 1000$, the second penalty parameter $\beta_2 = 1$, the tolerance $\varepsilon = 2 \times 10^{-4}$, the weights $\alpha_n = 1/3$ $(n = 1, 2, 3)$, the initial Tucker rank $r^0 = (10, 10, 10)$, and $\Delta r = (5, 5, 5)$. All tests are implemented on the platform of Windows 7 and MATLAB (R2017b) with an Intel Core i5-4590 3.30GHz and 16GB RAM.

4.1. Video data

In this section, we test eight videos, including *coastguard, news, salesman, foreman, suzie, hall, highway*, and *container*[11]. All videos are in the YUV format. In our tests, we only used the first 150 frames of Y channel. All testing videos are of size $144 \times 176 \times 150$. The maximum Tucker rank is set to be $r^{\max} = (85, 95, 65)$. The SRs are set to be 5%, 10%, 20%, 30%, 40%, and 50%, respectively.

Table 2 summarizes the PSNR, SSIM, and average CPU time (in minutes) of all testing videos reconstructed by six utilized LRTC methods for different SRs. It shows that except that when SR = 5%, the proposed method consistently outperforms the compared methods in terms of both PSNR and SSIM values. Fig. 3 shows one frame of all videos reconstructed by six utilized LRTC methods for SR = 10%. We observe that the visual effect of the reconstructed videos by the proposed method is superior to those by the compared methods. Specifically, the proposed method is capable of better completing the missing entries while finely preserving the structure of the underlying videos, while the results obtained by TMac, MF-TV, and MF-Framelet remain large amount of missing entries. SPC-QV and LRTC-TV-I can perform comparatively better in missing entries completing, but their results contain evident blurry area, leading to some details missing.

For comprehensive comparisons of the performance of six LRTC methods, we select two reconstructed videos *news* and *hall* as representations. The PSNR values of each frame reconstructed by six compared methods are shown in Fig. 4. As observed, the proposed method has an overall better performance in all frames than the compared methods in term of PSNR values. The piecewise smoothness along the temporal mode of the reconstructed videos can be seen from Fig. 5, which shows the pixel values of one mode-3 fiber of these two videos reconstructed by six compared methods for different SRs. It can be seen that although the curves estimated by three compared methods are excessive fluctuation and deviated from the original, the proposed method has a nice touch of the original and enhance the piecewise smoothness along the temporal mode of the reconstructed videos.

4.2. Hyperspectral data

In this section, the *Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Cuprite* data[12]. and the *Washington DC Mall* data[12] are used to test the performance of different methods. We only select a part of them (of size $150 \times 150 \times 130$) as the testing hyperspectral images. The maximum Tucker rank is set to be $r^{\max} = (85, 85, 10)$. The SR is set to be 5%. Fig. 6 shows a band of the test hyperspectral images reconstructed by the proposed methods and three compared methods. It is observed that the proposed method is able to produce visually superior results than the compared methods. The PSNR and SSIM values of each band of the reconstructed hyperspectral images are shown in Fig. 7. We can see that the PSNR and SSIM values in all bands obtained by the proposed method are higher than those obtained by the compared methods. We display the intensity of two mode-3 fibers of the reconstructed hyperspectral images in Fig. 8. As observed, the curves output by the proposed methods are much smoother and more closer to the original than those obtained by the compared methods.

4.3. MRI data

This test uses the MRI data[13] of size $181 \times 217 \times 181$ as the testing data. The maximum Tucker rank is set to be $r^{\max} = (70, 70, 70)$. The SR is set to be 10%. Since the slices of all directions can be treated as images, in Fig. 9, we display three

[6] The code of TMac is available at https://xu-yangyang.github.io/TMac/.
[7] The code of MF-TV is available at https://github.com/uestctensorgroup/MF_TV.
[8] The code of MF-Framelet is available at https://github.com/uestctensorgroup/code_MF_Framelet.
[9] The code of SPC-QV is available at https://sites.google.com/site/yokotatsuya/home/software.
[10] The code of LRTC-TV-I is available at https://xutaoli.weebly.com/.
[11] http://trace.eas.asu.edu/yuv/.
[12] http://lesun.weebly.com/hyperspectral-data-set.html.
[13] http://brainweb.bic.mni.mcgill.ca/brainweb/selection_normal.html.–

Table 2
The PSNR, SSIM, and average CPU time (in minutes) obtained by six utilized LRTC methods for videos.

video	SR	5%		10%		20%		30%		40%		50%		Time
	Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
	TMac	6.9472	0.0175	7.7145	0.0319	9.6940	0.0733	13.124	0.1941	22.827	0.6680	29.466	0.8675	**8.699**
	MF-TV	7.5460	0.0372	8.6137	0.0625	11.503	0.1308	17.482	0.3462	26.335	0.7586	31.146	0.9001	121.2
coastguard	MF-Framelet	10.401	0.1131	13.025	0.1857	18.164	0.3167	21.969	0.5471	28.594	0.8350	32.079	0.9211	291.4
	SPC-QV	**22.878**	0.5626	24.534	0.6732	27.036	0.7982	28.366	0.8480	29.564	0.8829	30.756	0.9105	58.37
	LRTC-TV-I	19.789	0.4320	21.497	0.5334	24.127	0.6942	26.254	0.7973	29.113	0.8637	29.959	0.9099	16.73
	SMF-LRTC	22.596	**0.5709**	**24.965**	**0.6999**	**27.190**	**0.8047**	**29.107**	**0.8651**	**30.772**	**0.9030**	**32.252**	**0.9301**	24.76
	TMac	9.7905	0.0849	11.216	0.1822	14.851	0.4176	25.940	0.7527	33.399	0.9027	37.011	0.9563	**14.54**
	MF-TV	10.611	0.1158	12.658	0.2177	19.132	0.5282	30.475	0.8404	34.987	0.9321	37.589	0.9644	112.4
news	MF-Framelet	14.351	0.3850	17.286	0.4912	22.458	0.6663	33.332	0.9209	36.221	0.9571	38.286	0.9743	217.3
	SPC-QV	26.639	0.8536	29.308	0.8841	31.906	0.9415	33.279	0.9546	34.407	0.9630	35.470	0.9701	56.31
	LRTC-TV-I	19.187	0.6574	21.107	0.7426	24.463	0.8667	27.481	0.9269	29.994	0.9575	32.546	0.9751	20.13
	SMF-LRTC	**26.711**	**0.8541**	**30.773**	**0.9208**	**33.744**	**0.9522**	**35.748**	**0.9685**	**37.222**	**0.9762**	**38.721**	**0.9824**	23.62
	TMac	12.048	0.0712	14.759	0.2662	21.910	0.7306	33.191	0.9143	37.839	0.9659	40.102	0.9807	**13.78**
	MF-TV	13.155	0.0956	17.568	0.3450	27.852	0.8322	35.271	0.9425	38.176	0.9695	40.161	0.9813	74.24
salesman	MF-Framelet	17.646	0.4481	20.227	0.6196	30.629	0.8875	36.043	0.9556	38.625	0.9744	40.465	0.9834	198.1
	SPC-QV	30.304	0.8732	32.410	0.9192	34.322	0.9463	35.457	0.9580	36.449	0.9662	37.436	0.9729	48.64
	LRTC-TV-I	22.401	0.5115	25.412	0.6659	28.977	0.8301	31.351	0.9010	33.517	0.9400	35.785	0.9642	18.37
	SMF-LRTC	**30.931**	**0.8892**	**34.229**	**0.9412**	**36.253**	**0.9607**	**37.721**	**0.9710**	**38.989**	**0.9779**	**40.962**	**0.9856**	22.81
	TMac	6.3165	0.0145	11.101	0.1701	26.850	0.7970	32.718	0.9146	34.417	0.9408	35.897	0.9576	**11.51**
	MF-TV	7.0164	0.0248	13.733	0.2642	29.707	0.8457	32.779	0.9156	34.452	0.9413	35.898	0.9584	61.54
foreman	MF-Framelet	7.2518	0.0291	14.345	0.3142	30.255	0.8590	32.918	0.9203	34.526	0.9439	35.911	0.9599	154.3
	SPC-QV	**25.597**	**0.7651**	27.037	0.7813	28.321	0.8461	29.264	0.8684	30.139	0.8867	31.050	0.9041	49.92
	LRTC-TV-I	19.186	0.5741	21.869	0.7080	26.792	0.8301	29.594	0.9141	31.665	0.9440	33.643	0.9631	18.64
	SMF-LRTC	24.329	0.6613	**28.482**	**0.8395**	**32.041**	**0.9159**	**34.115**	**0.9422**	**35.726**	**0.9585**	**37.221**	**0.9696**	15.82
	TMac	11.653	0.0488	17.886	0.5077	27.635	0.8242	34.509	0.9223	36.596	0.9479	38.041	0.9621	**11.68**
	MF-TV	13.775	0.0963	22.281	0.6050	31.888	0.8711	35.089	0.9279	36.723	0.9493	38.106	0.9628	58.17
suize	MF-Framelet	17.049	0.2912	24.838	0.6979	32.502	0.8851	35.263	0.9311	36.811	0.9505	38.167	0.9635	161.1
	SPC-QV	**29.589**	**0.8249**	30.863	0.8527	32.009	0.8779	32.905	0.8959	33.729	0.9109	34.654	0.9258	41.16
	LRTC-TV-I	23.516	0.6936	27.722	0.8034	31.406	0.8873	33.747	0.9259	35.622	0.9486	37.419	0.9644	16.68
	SMF-LRTC	27.667	0.7932	**31.755**	**0.8823**	**34.254**	**0.9228**	**36.057**	**0.9456**	**37.387**	**0.9580**	**38.825**	**0.9697**	16.19
	TMac	12.349	0.4085	21.492	0.7890	33.106	0.9399	34.995	0.9579	36.306	0.9682	37.520	0.9759	**6.243**
	MF-TV	13.781	0.4300	25.171	0.8335	33.213	0.9415	35.034	0.9592	36.255	0.9687	37.481	0.9762	20.74
hall	MF-Framelet	13.101	0.4855	24.591	0.8424	33.515	0.9476	35.217	0.9630	36.449	0.9715	37.664	0.9782	64.26
	SPC-QV	**27.977**	**0.9007**	29.109	0.9178	30.108	0.9306	30.878	0.9398	31.665	0.9472	32.469	0.9552	24.11
	LRTC-TV-I	19.815	0.6451	22.493	0.7923	26.983	0.9090	29.869	0.9466	32.227	0.9674	34.737	0.9804	17.51
	SMF-LRTC	26.004	0.8686	**32.578**	**0.9536**	**36.106**	**0.9730**	**37.856**	**0.9793**	**39.256**	**0.9840**	**40.551**	**0.9873**	16.01
	TMac	27.904	0.7644	31.689	0.8768	33.420	0.9119	34.430	0.9299	35.437	0.9438	36.475	0.9660	**2.613**
	MF-TV	27.035	0.7825	31.673	0.8786	33.390	0.9129	34.401	0.9305	35.393	0.9440	36.427	0.9660	6.551
highway	MF-Framelet	28.966	0.8103	31.764	0.8866	33.480	0.9158	34.467	0.9321	35.445	0.9452	36.493	0.9668	16.54
	SPC-QV	28.171	0.8183	28.483	0.8288	29.024	0.8484	29.618	0.8601	30.323	0.8765	31.107	0.8945	9.149
	LRTC-TV-I	26.975	0.8105	28.096	0.8549	30.019	0.9008	31.959	0.9295	33.813	0.9494	35.549	0.9636	15.21
	SMF-LRTC	**29.142**	**0.8220**	**32.257**	**0.9101**	**34.857**	**0.9389**	**36.181**	**0.9512**	**37.417**	**0.9610**	**38.582**	**0.9693**	12.38
	TMac	15.393	0.6089	28.492	0.8882	34.271	0.9455	35.934	0.9597	37.258	0.9695	38.211	0.9773	**4.210**
	MF-TV	18.131	0.6494	29.410	0.8980	34.286	0.9464	35.841	0.9598	37.177	0.9697	38.384	0.9774	8.254
container	MF-Framelet	16.581	0.6458	29.562	0.9045	34.344	0.9479	35.901	0.9603	37.246	0.9696	38.429	0.9769	21.57
	SPC-QV	**27.238**	**0.8654**	28.589	0.8906	29.804	0.9112	30.608	0.9249	31.378	0.9368	32.234	0.9481	28.59
	LRTC-TV-I	20.198	0.6722	22.305	0.7512	25.721	0.8561	28.279	0.9114	30.563	0.9443	33.123	0.9671	15.94
	SMF-LRTC	25.623	0.8317	**31.894**	**0.9418**	**36.261**	**0.9664**	**38.278**	**0.9753**	**39.749**	**0.9809**	**41.087**	**0.9854**	14.28

representative slices of the reconstructed MRI data which are observed from three different directions, respectively. For all directions, we see that the visual quality of the reconstructed MRI by the proposed method is superior to the compared methods. In Fig. 10, we show the PSNR and SSIM values of each slice of the reconstructed MRI data observed from three different directions, respectively. It can be seen that no matter which direction they are from, the proposed method achieves the best PSNR and SSIM values among six LRTC methods. Fig. 11 presents the intensity of nine fibers (three mode-1 fibers, mode-2 fibers, and mode-3 fibers) of the reconstructed MRI data. It is obvious that the proposed method can obtain a smooth approximating curve, whereas the curves estimated by the compared methods are fluctuated and deflected.

4.4. Discussions

An extreme case (*SR*=1%): we test five videos and one hyperspectral data for SR = 1%. As this test is an extreme case, we fix the Tucker rank and use the numbers of the singular values which are larger than 0.5% of the largest one to approximate it. The tolerance ε is set as 10^{-5}. We display one frame (band) of the testing data reconstructed by the proposed method in Fig. 12. We observe that our method can recognize the shape of the original data for the testing data without obvious

Fig. 3. One frame of the testing videos reconstructed by six utilized LRTC methods with SR = 10%. From left to right: the original data, the observed data, the reconstructed results by TMac [27], MF-TV [28], MF-Framelet [29], SPC-QV [34], LRTC-TV-I [35], and the proposed method, respectively.

Table 3
The PSNR, SSIM, and CPU time (in minutes) with respect to different values of maximum Tucker rank.

Maximum rank ($r^{\max}$)	(55,55,55)	(60,60,60)	(65,65,65)	(70,70,70)	(75,75,75)	(80,80,80)	(85,85,85)	(90,90,90)	(95,95,95)	(100,100,100)
PSNR	34.254	34.288	34.334	34.299	34.216	34.172	34.164	34.162	34.141	34.109
SSIM	0.9168	0.9201	0.9214	0.9222	0.9220	0.9219	0.9185	0.9210	0.9208	0.9205
Time	9.4225	10.078	10.470	11.779	12.118	13.141	14.933	15.045	15.948	16.802

low-rankness (such as *suzie, salesman*, and *news*) and obtain promising visual results for the testing data with obvious low-rankness (such as *hall, highway*, and *Cuprite*).

Parameter analysis: we analyze the robustness of the proposed method with respect to different parameters using the video data *suzie* with SR = 20% in this test. The parameter analysis is presented in Fig. 13. As observed, (1) different values of the proximal parameter ρ lead to nearly the same PSNR value, i.e., the proximal parameter mainly affects the computational efficiency rather than the performance; (2) the proposed method is slightly sensitive to the regularization parameters μ_1 and μ_2, which are set to be 10 and 100 in all experiments, respectively; (3) the values of the penalty parameters β_1 and β_2 have an impact on the performance of the proposed method, although the convergence of the proposed method is theoretically guaranteed regardless of the penalty parameter as long as it is a positive number. The penalty parameters β_1 and β_2 are set to be 1000 and 1 in all experiments, respectively.

Maximum rank analysis: we analyze the robustness of the proposed method with respect to different maximum rank $r^{\max} = (r_1^{\max}, r_2^{\max}, r_3^{\max})$. The testing data is video *suzie* and the SR is set to be 20%. Table 3 lists the PSNR, SSIM, and CPU time (in minutes) output by the proposed method for different values of maximum Tucker rank. We observe that within

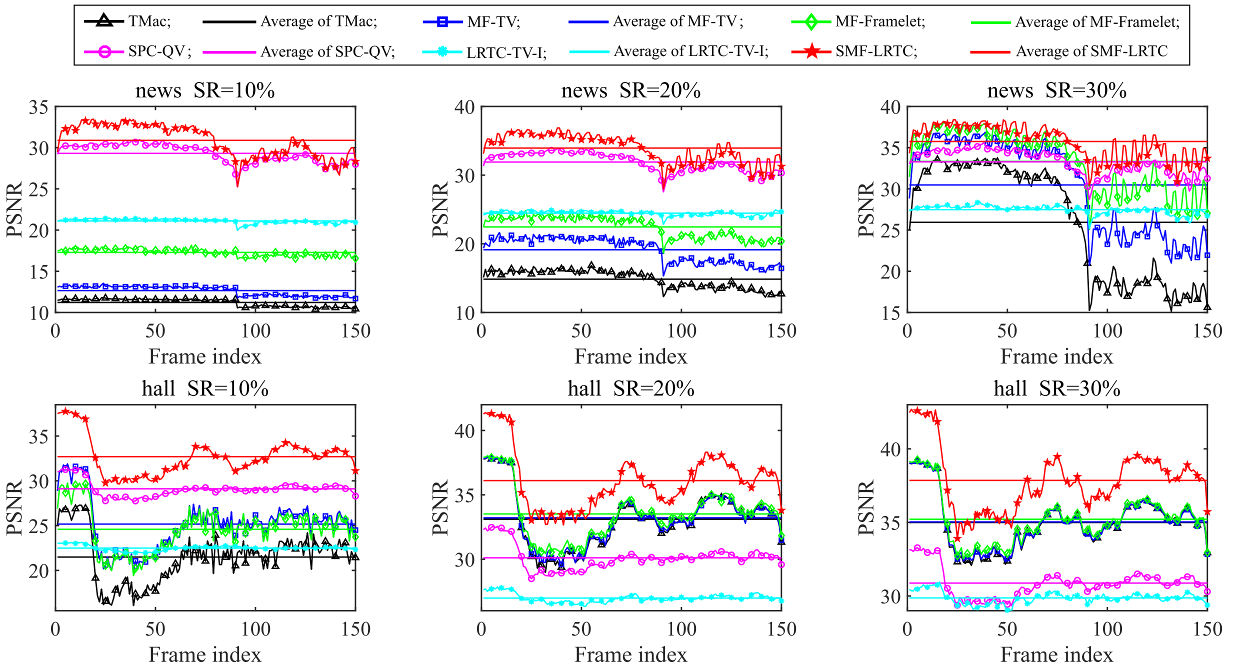

Fig. 4. The PSNR values of all frames of the reconstructed videos *news* and *hall* obtained by six utilized LRTC methods. The top and bottom rows are the results of videos *news* and *hall*, respectively. From left to right: the SR are set to be 10%, 20%, and 30%, respectively.

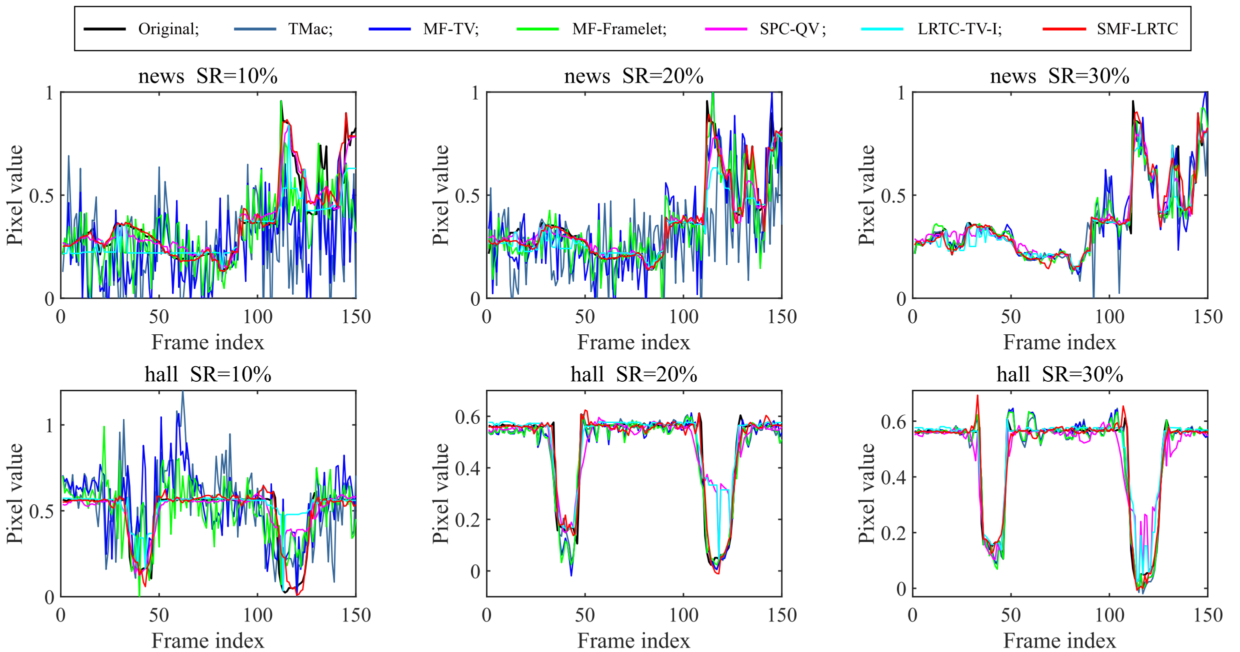

Fig. 5. The pixel values of one mode-3 fiber (the same location of each frame) of the reconstructed videos *news* and *hall* obtained by six utilized LRTC methods. The top and bottom rows are the results of videos *news* and *hall*, respectively. From left to right: the SR are set to be 10%, 20%, and 30%, respectively.

limits, different values of $r^{\max}$ lead to nearly the same PSNR value, i.e., it mainly affects the computational efficiency rather than the performance. For video data $\mathcal{Y} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, we empirically recommend users to select $r^{\max}$ as $(\lfloor \frac{3}{5} d_1 \rfloor \pm 10, \lfloor \frac{3}{5} d_2 \rfloor \pm 10, \lfloor \frac{2}{5} d_3 \rfloor \pm 10)$. For hyperspectral images, the correlation along their spectral mode should be much stronger than those along their spatial modes, thus we empirically recommend users to select $r^{\max}$ as $(\lfloor \frac{3}{5} d_1 \rfloor \pm 10, \lfloor \frac{3}{5} d_2 \rfloor \pm 10, \lfloor \frac{1}{10} d_3 \rfloor \pm 10)$.

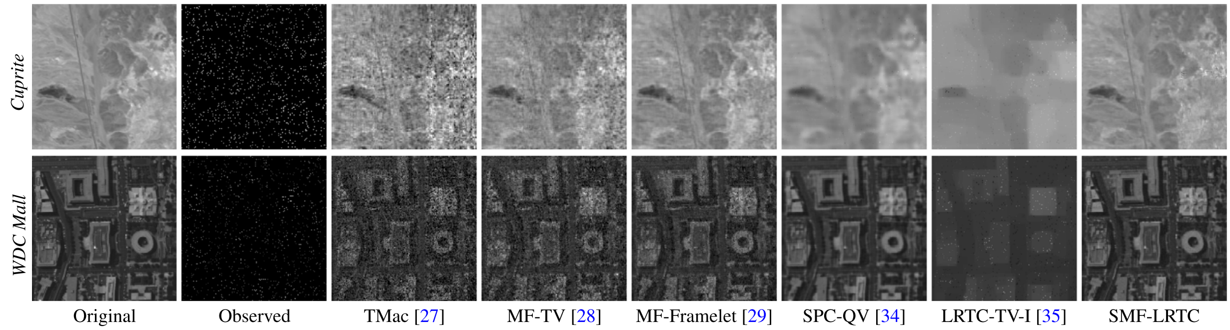

Fig. 6. One band of the testing hyperspectral images *Cuprite* and *Washington DC Mall* reconstructed by six utilized LRTC methods with SR = 5%. From left to right: the original data, the observed data, the reconstructed results by TMac [27], MF-TV [28], MF-Framelet [29], SPC-QV [34], LRTC-TV-I [35], and the proposed method, respectively.

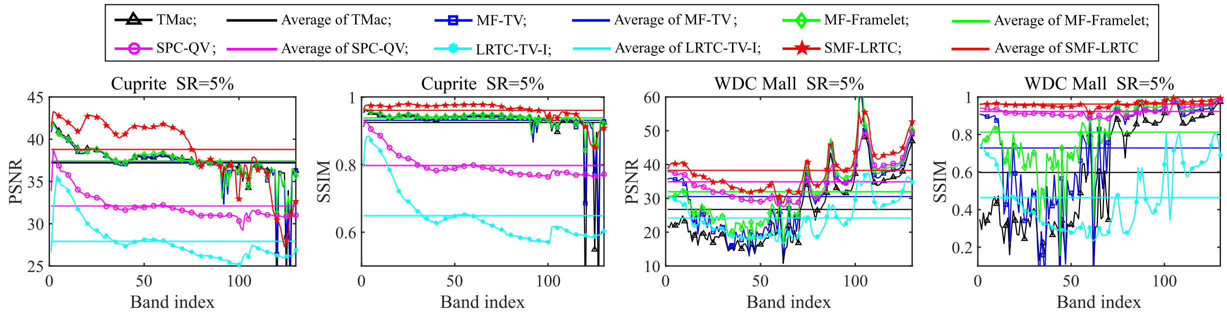

Fig. 7. The PSNR and SSIM values of all bands of the reconstructed hyperspectral images *Cuprite* and *Washington DC Mall* obtained by six utilized LRTC methods.

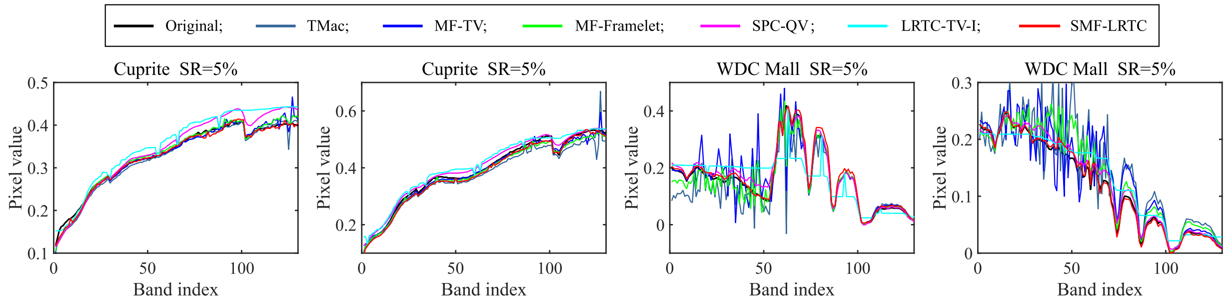

Fig. 8. The pixel values of two mode-3 fibers of the reconstructed hyperspectral images *Cuprite* and *Washington DC Mall* obtained by six utilized LRTC methods.

Table 4
The PSNR, SSIM, and CPU time (in minutes) with respect to different iterations for computing $\mathcal{X}_3$.

Inner iterations	2	5	8	10	12	15	18	20
PSNR	33.846	34.375	34.386	34.331	34.233	34.241	34.204	34.211
SSIM	0.9097	0.9242	0.9247	0.9249	0.9235	0.9241	0.9235	0.9236
Time	6.3490	10.996	13.914	18.713	24.212	26.835	34.671	38.296

For MRI data, since the slices of all directions of them can be treated as images, we empirically recommend users to select $r^{\max}$ as $(\lfloor \frac{2}{5} d_1 \rfloor \pm 10, \lfloor \frac{2}{5} d_2 \rfloor \pm 10, \lfloor \frac{2}{5} d_3 \rfloor \pm 10)$.

Inner iteration analysis: we analyze the sensitivity of the iterations for computing $\mathcal{X}_3$ and A_3 on video *suzie* with SR = 20%. In Table 4 and Table 5, we report the PSNR, SSIM, and CPU time with respect to different iterations for computing $\mathcal{X}_3$ and A_3, respectively. We observe that the iterations greater than 5 lead to nearly the same PSNR and SSIM values, but the CPU time increased along with inner iteration increasing. Following the guidance of Tables 4 and 5, in all tests, the inner iterations for computing both $\mathcal{X}_3$ and A_3 are set to be 5. Thus, although the ADMM is performed repeatedly as the inner loop, the proposed method is still efficient.

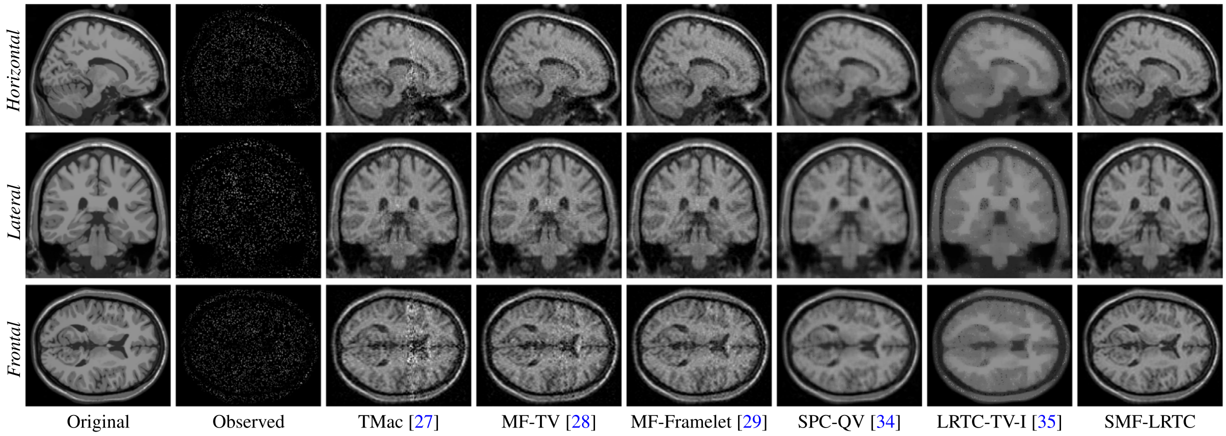

Fig. 9. Three slices observed from three different directions of the MRI data reconstructed by six utilized LRTC methods with SR = 10%. From top to bottom: horizontal slices, lateral slices, and frontal slices, respectively. From left to right: the original data, the observed data, the reconstructed results by TMac [27], MF-TV [28], MF-Framelet [29], SPC-QV [34], LRTC-TV-I [35], and the proposed method, respectively.

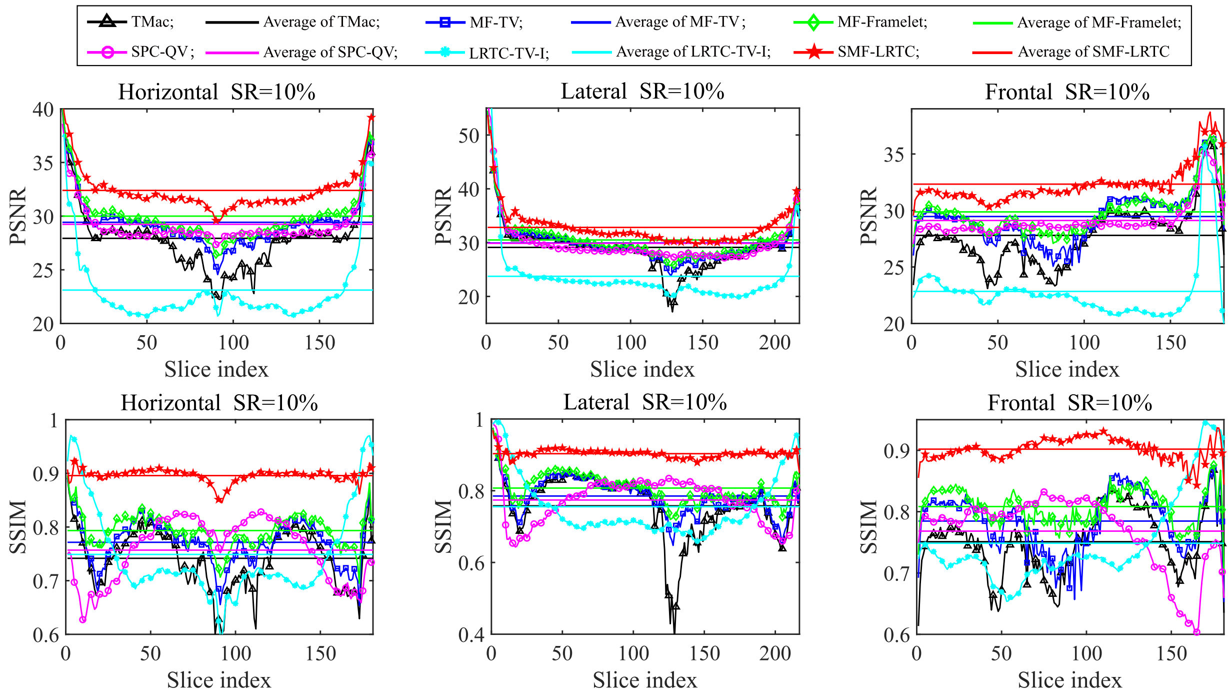

Fig. 10. The PSNR and SSIM values of all slices observed from different directions of the reconstructed MRI data obtained by six utilized LRTC methods. From left to right: horizontal slices, lateral slices, and frontal slices, respectively.

Table 5
The PSNR, SSIM, and CPU time (in minutes) with respect to different iterations for computing A_3.

Inner iterations	2	5	8	10	12	15	18	20
PSNR	34.350	34.375	34.336	34.328	34.329	34.320	34.234	34.326
SSIM	0.9239	0.9242	0.9235	0.9238	0.9231	0.9234	0.9221	0.9235
Time	10.308	10.996	11.113	11.655	11.839	12.464	12.523	12.842

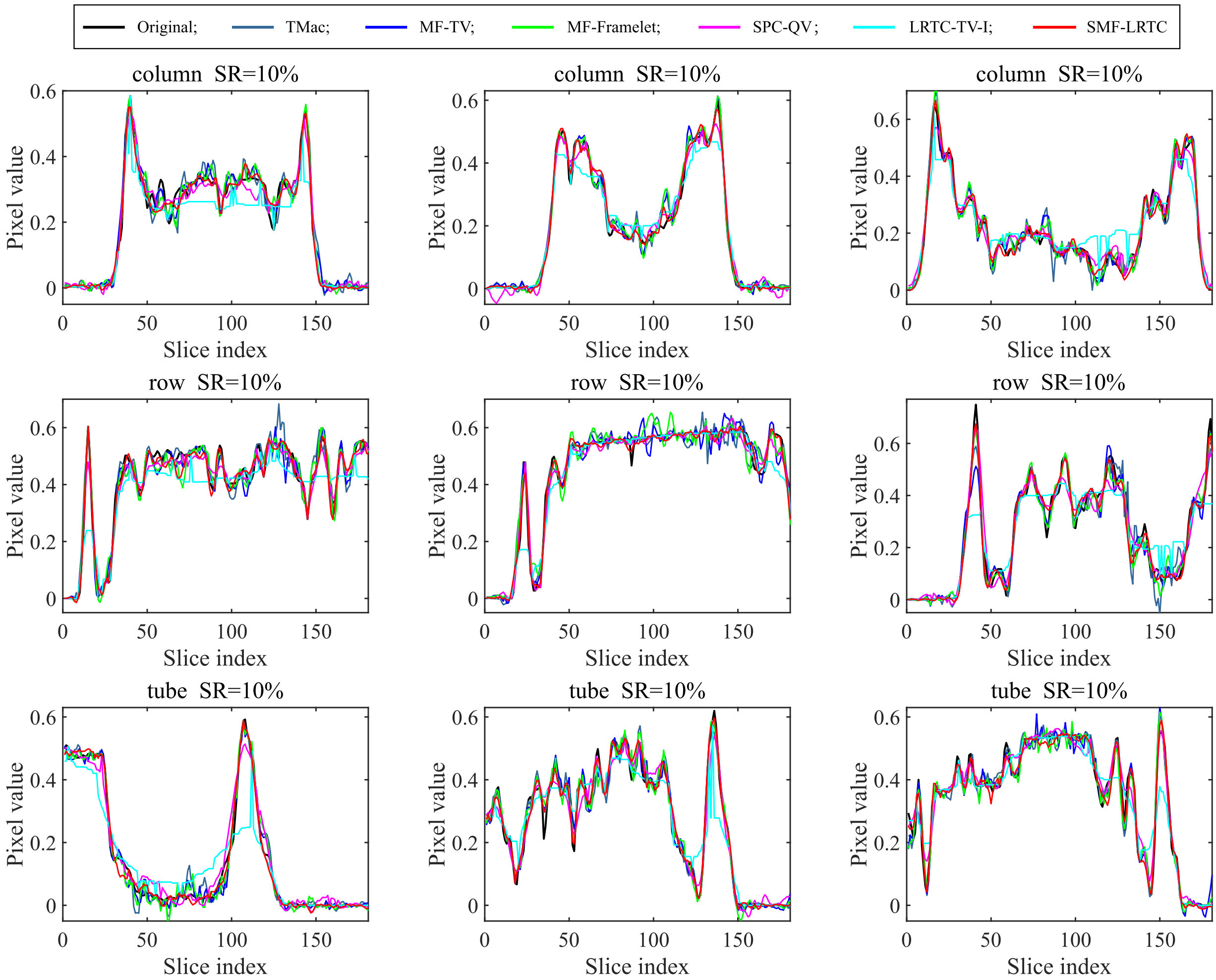

Fig. 11. The pixel values of nine fibers of the reconstructed MRI data obtained by six utilized LRTC methods. From top to bottom: mode-1 fibers (columns), mode-2 fibers (rows), and mode-3 fibers (tubes), respectively.

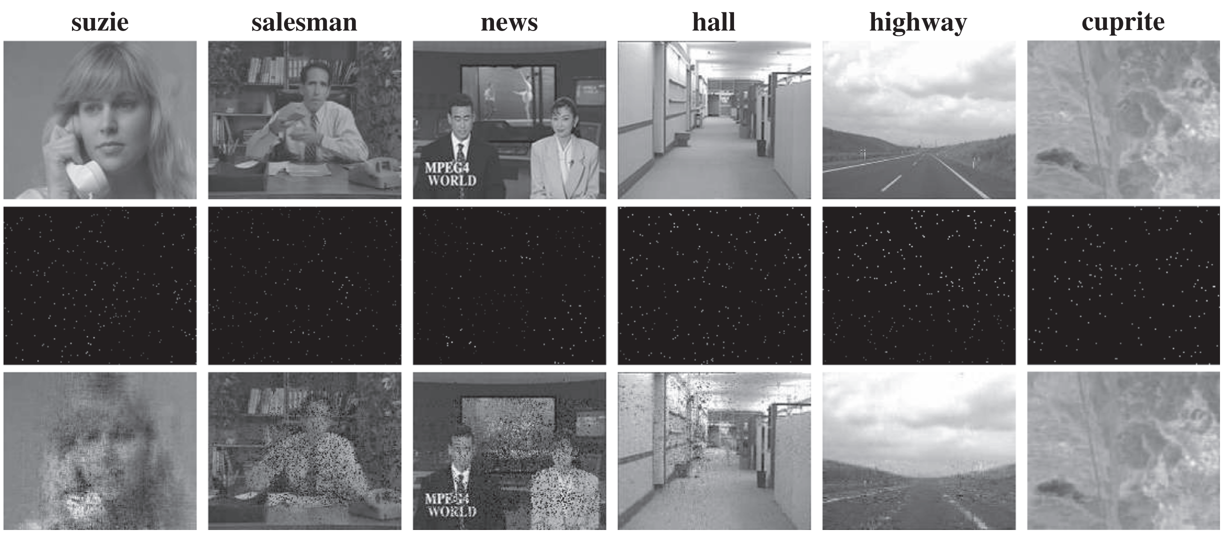

Fig. 12. One frame (band) of five videos and one hyperspectral image reconstructed by the proposed method. From top to bottom: the original data, the observed data, and the reconstructed data, respectively.

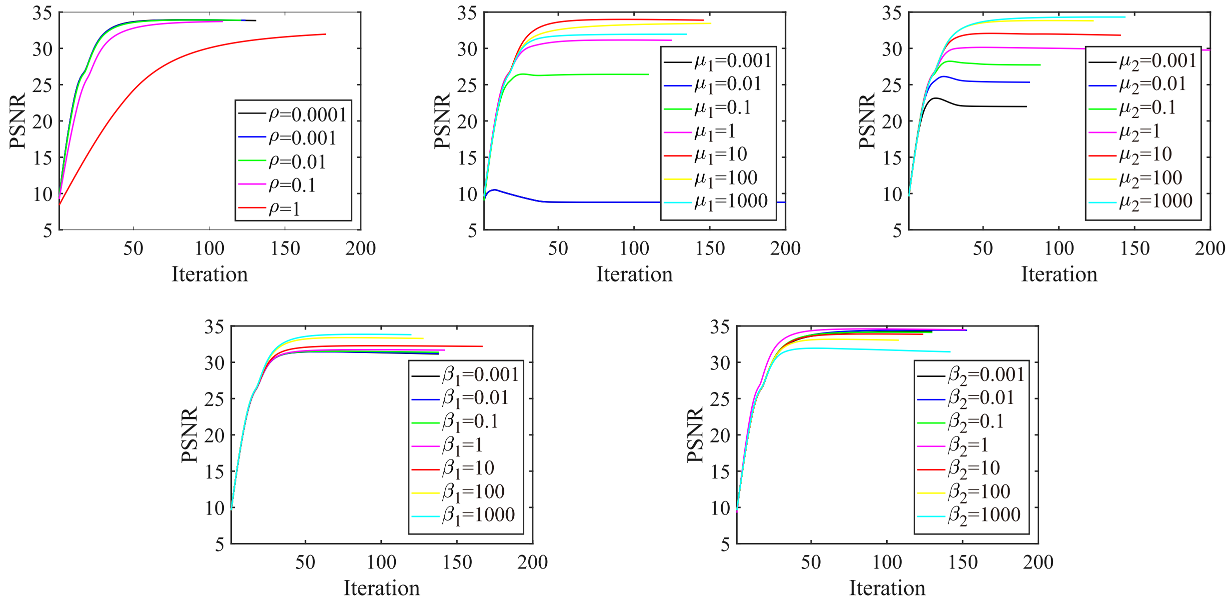

Fig. 13. The PSNR values with respect to the iteration for different values of parameters: $\rho, \mu_1, \mu_2, \beta_1$, and β_2.

5. Conclusions

In this paper, we proposed a model for low-rank tensor completion by combining low-rank matrix factorization, framelet, and total variation. Meanwhile, an efficient BSUM-based algorithm was developed to solve the proposed model with guaranteed convergence. Numerical results demonstrated some superiorities of the proposed method: (1) qualitatively, the proposed method produced the best results both in recovering visual effects and in enhancing the piecewise smoothness; (2) quantitatively, the proposed method had an overall better performance than the compared methods in terms of both PSNR and SSIM values.

Acknowledgments

The authors would like to express their sincere thanks to the reviewers for their much helpful suggestions for revising this paper. The authors would like to express their sincere thanks to Dr. Y. Xu, Dr. T. Yokota, and Dr. X. Li for sharing the codes of the TMac [27], SPC-QV [34], and LRTC-TV-I [35], respectively. The research is supported by the National Natural Science Foundation of China (61772003, 61876203), the Fundamental Research Funds for the Central Universities (31020180QD126), the National Postdoctoral Program for Innovative Talents (BX20180252), and the Project funded by China Postdoctoral Science Foundation (2018M643611).

References

[1] V.N. Varghees, M.S. Manikandan, R. Gini, Adaptive MRI image denoising using total-variation and local noise estimation, in: Proceedings of the International Conference on Advances in Engineering, Science and Management (ICAESM), 2012, pp. 506–511.
[2] T.-Y. Ji, T.-Z. Huang, X.-L. Zhao, T.-H. Ma, L.-J. Deng, A non-convex tensor rank approximation for tensor completion, Appl. Math. Modell. 48 (2017) 410–422.
[3] F. Li, M.K. Ng, R.J. Plemmons, Coupled segmentation and denoising/deblurring models for hyperspectral material identification, Numer. Linear Algebr. Appl. 19 (1) (2012) 153–173.
[4] 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.
[5] Y. Chang, L. Yan, S. Zhong, Hyperspectral image denoising via spectral and spatial low-rank approximation, in: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2017, pp. 4193–4196.
[6] J.-H. Yang, X.-L. Zhao, J.-J. Mei, S. Wang, T.-H. Ma, T.-Z. Huang, Total variation and high-order total variation adaptive model for restoring blurred images with cauchy noise, Comput. Math. Appl. 77 (5) (2019) 1255–1272, doi:10.1016/j.camwa.2018.11.003.
[7] L. Zhuang, J.M. Bioucas-Dias, Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 11 (3) (2018) 730–742.
[8] Y. Chen, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, Hyperspectral image restoration using framelet-regularized low-rank nonnegative matrix factorization, Appl. Math. Modell. 63 (2018) 128–147.
[9] H. Zhang, V.M. Patel, Convolutional sparse and low-rank coding-based rain streak removal, in: Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV), 2017, pp. 1259–1267.
[10] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, Y. Wang, Fastderain: A novel video rain streak removal method using directional gradient priors, IEEE Trans. Image Process. 28 (4) (2019) 2089–2102.
[11] Q. Zhao, L. Zhang, A. Cichocki, Bayesian CP factorization of incomplete tensors with automatic rank determination, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1751–1763.

[12] T. Korah, C. Rasmussen, Spatiotemporal inpainting for recovering texture maps of occluded building facades, IEEE Trans. Image Process. 16 (9) (2007) 2262–2271.
[13] T. Yokota, N. Lee, A. Cichocki, Robust multilinear tensor rank estimation using higher order singular value decomposition and information criteria, IEEE Trans. Signal Process. 65 (5) (2017) 1196–1206.
[14] J.-J. Mei, Y. Dong, T.-Z. Huang, W. Yin, Cauchy noise removal by nonconvex ADMM with convergence guarantees, J. Sci. Comput. 74 (2) (2018) 743–766.
[15] M. Ding, T.-Z. Huang, S. Wang, J.-J. Mei, X.-L. Zhao, Total variation with overlapping group sparsity for deblurring images under cauchy noise, Appl. Math. Comput. 341 (2019) 128–147.
[16] N.D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E.E. Papalexakis, C. Faloutsos, Tensor decomposition for signal processing and machine learning, IEEE Trans. Signal Process. 65 (13) 3551–3582.
[17] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 208–220.
[18] E.J. Candès, B. Recht, Exact matrix completion via convex optimization, Found. Comput. Math. 9 (6) (2009) 717–772.
[19] E.J. Candès, Y. Plan, Matrix completion with noise, Proc. IEEE 98 (6) (2010) 925–936.
[20] H. Tan, B. Cheng, W. Wang, Y.-J. Zhang, B. Ran, Tensor completion via a multi-linear low-n-rank factorization model, Neurocomputing 133 (2014) 161–169.
[21] M. Filipović, A. Jukić, Tucker factorization with missing data with application to low-n-rank tensor completion, Multidimens. Syst. Signal Process. 26 (3) (2015) 677–692.
[22] C.J. Hillar, L.-H. Lim, Most tensor problems are NP-hard, J. ACM 60 (6) (2013). 45:1–45:39.
[23] T.-H. Ma, Y. Lou, T.-Z. Huang, Truncated ℓ_{1--2} models for sparse recovery and rank minimization, SIAM J. Imaging Sci. 10 (3) (2017) 1346–1380.
[24] B. Recht, M. Fazel, P.A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, SIAM Rev. 52 (3) (2010) 471–501.
[25] K.-C. Toh, S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems, Pac. J. Optim. 6 (3) (2010) 615–640.
[26] S. Gandy, B. Recht, I. Yamada, Tensor completion and low-n-rank tensor recovery via convex optimization, Inverse Probl. 27 (2) (2011). 025010.
[27] Y. Xu, R. Hao, W. Yin, Z. Su, Parallel matrix factorization for low-rank tensor completion, Inverse Probl. Imaging 9 (2) (2015) 601–624.
[28] T.-Y. Ji, T.-Z. Huang, X.-L. Zhao, T.-H. Ma, G. Liu, Tensor completion using total variation and low-rank matrix factorization, Inf. Sci. 326 (2016) 243–257.
[29] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, T.-Y. Ji, L.-J. Deng, Matrix factorization for low-rank tensor completion using framelet prior, Inf. Sci. 436 (2018) 403–417.
[30] Z. Zhang, S. Aeron, Exact tensor completion using t-SVD, IEEE Trans. Signal Process. 65 (6) (2017) 1511–1526.
[31] P. Zhou, C. Lu, Z. Lin, C. Zhang, Tensor factorization for low-rank tensor completion, IEEE Trans. Image Process. 27 (3) (2018) 1152–1163.
[32] X.-Y. Liu, S. Aeron, V. Aggarwal, X. Wang, in: Low-tubal-rank Tensor Completion Using Alternating Minimization, 2016. ArXiv: 1610.01690.
[33] X. Guo, Y. Ma, Generalized tensor total variation minimization for visual data recovery? in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3603–3611.
[34] T. Yokota, Q. Zhao, A. Cichocki, Smooth parafac decomposition for tensor completion, IEEE Trans. Signal Process. 64 (20) (2016) 5423–5436.
[35] X. Li, Y. Ye, X. Xu, Low-rank tensor completion with total variation for visual data inpainting, in: Proceedings of the AAAI, 2017, pp. 2210–2216.
[36] T. Yokota, H. Hontani, Simultaneous visual data completion and denoising based on tensor rank and total variation minimization and its primal-dual splitting algorithm, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3843–3851.
[37] T. Yokota, B. Erem, S. Guler, S.K. Warfield, H. Hontani, Missing slice recovery for tensors using a low-rank model in embedded space, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 8251–8259.
[38] 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.
[39] J.-F. Cai, R.H. Chan, Z. Shen, A framelet-based image inpainting algorithm, Appl. Comput. Harmon. Anal. 24 (2) (2008) 131–149.
[40] W. He, H. Zhang, L. Zhang, Total variation regularized reweighted sparse nonnegative matrix factorization for hyperspectral unmixing, IEEE Trans. Geosci. Remote Sens. 55 (7) (2017) 3909–3921.
[41] X. Fu, W.K. Ma, J.M. Bioucas-Dias, T.H. Chan, Semiblind hyperspectral unmixing in the presence of spectral library mismatches, IEEE Trans. Geosci. Remote Sens. 54 (9) (2016) 5171–5184.
[42] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (3) (2009) 455–500.
[43] A. Ron, Z. Shen, Affine systems in $L_2(\mathbb{R}^d)$: the analysis of the analysis operator, J. Funct. Anal. 148 (2) (1997) 408–447.
[44] A. Chai, Z. Shen, Deconvolution: a wavelet frame approach, Numerische Mathematik 106 (4) (2007) 529–587.
[45] N. Parikh, S. Boyd, Proximal algorithms, Found. Trends Optim. 1 (3) (2014) 127–239.
[46] M. Razaviyayn, M. Hong, Z.-Q. Luo, A unified convergence analysis of block successive minimization methods for nonsmooth optimization, SIAM J. Optim. 23 (2) (2013) 1126–1153.
[47] G. Zhang, Y. Xu, F. Fang, Framelet-based sparse unmixing of hyperspectral images, IEEE Trans. Image Process. 25 (4) (2016) 1516–1529.
[48] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2011) 1–122.
[49] Z. Wen, W. Yin, Y. Zhang, Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm, Math. Program. Comput. 4 (4) (2012) 333–361.
[50] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity 13 (4) (2004) 600–612.