



# Adaptive total variation and second-order total variation-based model for low-rank tensor completion

Xin Li<sup>1</sup> · Ting-Zhu Huang<sup>1</sup> · Xi-Le Zhao<sup>1</sup>  · Teng-Yu Ji<sup>2</sup> · Yu-Bang Zheng<sup>1</sup> · Liang-Jian Deng<sup>1</sup>

Received: 26 July 2019 / Accepted: 3 January 2020 / Published online: 29 April 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

Recently, low-rank regularization has achieved great success in tensor completion. However, only considering the global low-rankness is not sufficient, especially for a low sampling rate (SR). Total variation (TV) is introduced into low-rank tensor completion (LRTC) problem to promote the local smoothness by incorporating the first-order derivatives information. However, TV usually leads to undesirable staircase effects. To alleviate these staircase effects, we suggest a first- and second-order TV-based parallel matrix factorization model for LRTC problem, which integrates the local smoothness and global low-rankness by simultaneously exploiting the first- and second-order derivatives information. To solve the proposed model, an efficient

---

✉ Ting-Zhu Huang  
tingzhuhuang@126.com

✉ Xi-Le Zhao  
xlzhao122003@163.com

Xin Li  
15280995987@163.com

Teng-Yu Ji  
tengyu.j66@126.com

Yu-Bang Zheng  
zhengyubang@163.com

Liang-Jian Deng  
liangjian1987112@126.com

<sup>1</sup> School of Mathematical Sciences/Research Center for Image and Vision Computing, University of Electronic Science and Technology of China, Chengdu, 611731, Sichuan, People's Republic of China

<sup>2</sup> School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an, 710072, Shaanxi, People's Republic of China

proximal alternating optimization (PAO)-based algorithm is developed with theoretical guarantee. Moreover, we suggest a regularization parameter selection strategy to automatically update two regularization parameters, which is able to take advantage of the best properties of each of the two regularization terms. Extensive experiments on different tensor data show the superiority of the proposed method over other methods, particularly for extremely low SRs.

**Keywords** Low-rank tensor completion · Total variation · Staircase effects · Second-order total variation · Proximal alternating optimization

## 1 Introduction

Tensors are high-order generalizations of vectors and matrices, which can represent the intrinsic structure of more complex higher-order data. In fact, tensors are the natural form of high-dimensional and multi-directional real-world data which have high dimensional structure, for example, hyperspectral images (HSI) and multispectral images (MSI). As a fundamental problem in tensor processing and analysis, tensor completion is to predict the missing data by inferring the physical factors from partially observed entries. Due to the fact that the problem of data missing is inevitable, tensor completion problem holds significant practical importance for subsequent applications, such as HSI unmixing [1] and image segmentation [2]. Recent studies have shown that high-dimensional tensor, like color image and video, are generally low-rank or approximate low-rank in nature [3]. Therefore, most previous methods focused on the low-rank property to recover incomplete tensors, which was called low-rank tensor completion (LRTC).

Obviously, the second-order tensor completion can be regarded as matrix completion [4–6]. Thus, LRTC is often associated with low-rank matrix completion (LRMC). Let us briefly review the LRMC problem which can be described as the affine rank minimization:

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

where  $Y \in \mathbb{R}^{m \times n}$  represents the original matrix,  $F \in \mathbb{R}^{m \times n}$  represents the observed matrix,  $\Omega$  is the set of matrix for available entries, and  $\mathcal{P}_{\Omega(\cdot)}$  is the projection operator that keeps the entries of  $Y$  in  $\Omega$ .

The global solution to (1) may not be obtained, because of the non-convexity of matrix rank [4]. Then, [5] proved that under certain incoherence conditions, (1) can be transformed to minimizing the matrix nuclear norm [6]:

$$\begin{aligned} \min_Y \quad & \|Y\|_* \\ \text{s.t.} \quad & \mathcal{P}_\Omega(Y) = F. \end{aligned} \quad (2)$$

Many optimization algorithms can be used to solve (2), e.g., FPCA [7], APGL [8], and LMaFit [4].

As an extension of LRMC, LRTC methods can effectively capture the global information of tensor by minimizing tensor rank. Thus, LRTC model can be homoplastically formulated as:

$$\begin{aligned} \min_{\mathcal{Y}} \quad & \text{rank}(\mathcal{Y}) \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{F}, \end{aligned} \quad (3)$$

where  $\mathcal{Y} \in \mathbb{R}^{d_1 \times \dots \times d_N}$  denotes the underlying tensor,  $\mathcal{F}$  denotes the observed tensor, the meaning of  $\Omega$  and  $\mathcal{P}_{\Omega}(\cdot)$  is similar to the LRMC problem (see details in Section 2). However, the definition of the rank of tensors is not unique [3, 9, 10] and directly minimizing tensor rank is always NP-hard [11]. Many convex or non-convex alternatives are designed to settle this problem. For example, Zhang et al. suggested using nonconvex functions to approximate the rank of the tensor [12]. Among them, the most widely used one is to transfer (3) to minimize the nuclear norm:

$$\begin{aligned} \min_{\mathcal{Y}} \quad & \|\mathcal{Y}\|_* \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{F}. \end{aligned} \quad (4)$$

Obviously, the recovery quality obtained by directly unfolding the tensor  $\mathcal{Y}$  into a matrix to solve problem (4) (that is equal to solve problem (2)) is limited, because the problem (2) only utilizes low-rankness of one mode of the tensor [13, 14]. Considering that, Liu et al. proposed a new method that minimized the nuclear norm of  $N$  unfolding matrices [13]:

$$\begin{aligned} \min_{\mathcal{Y}} \quad & \sum_{n=1}^N \alpha_n \|Y_{(n)}\|_* \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{F}. \end{aligned} \quad (5)$$

where  $\alpha_n > 0$ ,  $\sum_{n=1}^N \alpha_n = 1$ , and  $Y_{(n)}$  represents the mode- $n$  unfolding of  $\mathcal{Y}$ . To efficiently solve problem (5), some optimization algorithms can be used, such as HaLRTC [13], Douglas-Rachford splitting technique, and the alternating direction multiplier method (ADMM) [15–17]. However, one limitation of this method is its high computational complexity of singular value decomposition (SVD), especially when dealing with large-scale problems. To address this problem, [14] suggested performing low-rank parallel matrix factorization for LRTC problem (called TMac) instead of minimizing nuclear norm:

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

where  $\|\cdot\|_F$  denotes the Frobenius norm, and  $A = (A_1, A_2, \dots, A_N)$  and  $X = (X_1, X_2, \dots, X_N)$  represent the low-rank factor matrices. In fact, there are accurate physical interpretations for  $A$  and  $X$  in many real-world applications. For example,  $A_n$  is the spectral library and  $X_n$  is the fractional abundance in hyperspectral images unmixing [1]. Compared with SVD-based methods, TMac costs less computation

and improves performance by applying low-rank matrix decomposition to each mode unfolding of tensor.

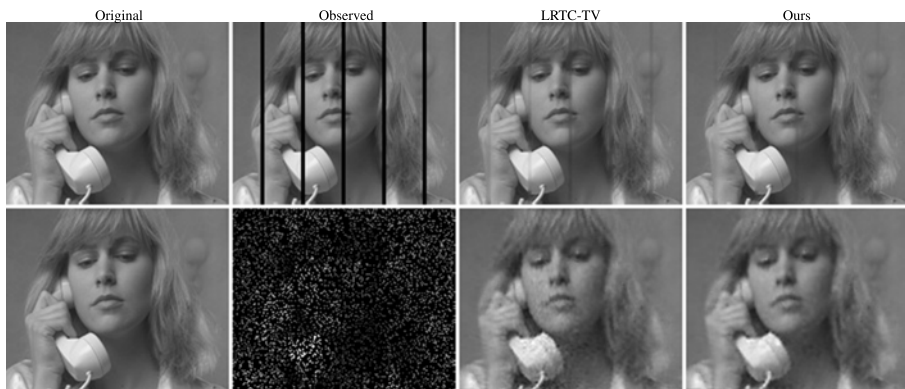
Although (5) and (6) can catch the global information of tensor, they are limited to exploit the structure of the tensor [18, 19]. In fact, many tensors in real-world reflect the characteristic of piecewise smoothness which is worthy to be taken into consideration, e.g., natural images, videos, HSI, and MSI [20–23]. Since the TV-based regularization can utilize local smooth prior, it [24, 25] has performed well in image segmentation [2, 26, 27], image restoration [28–30], MRI denoising [31], rain streak removal [32], and image destriping [33]. Particularly, Zhao et al. [1] successfully presented the TV-based model for sparse hyperspectral unmixing to promote the local similarity and they found that the TV regularizer of factor matrix  $X_3$  enhances the local similarity of the image  $\mathcal{Y}$  more essentially and economically. Motivated by that, Ji et al. [20] applied the TV into the LRTC problem (in this paper, we called it LRTC-TV):

$$\begin{aligned} \min_{\mathcal{Y}, X, A} \quad & \sum_{n=1}^N \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2 + \mu TV(X_3) \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{F}, \end{aligned} \quad (7)$$

which made a significant improvement for LRTC problem by simultaneously considering the global low-rankness and local smoothness. However, since TV converts smooth areas into piecewise constant areas, it usually leads to undesirable staircase effects [34, 35], which not only visually reduces the image quality, but also increases their inappropriateness for subsequent programs, such as image unmixing and face recognition. To tackle this problem, several high-order methods utilizing high-order information to reduce the staircase effects for additive noise [34–36] are proposed. However, these methods lead the recovered images to over smooth and cannot preserve edges and details well. To overcome the above drawbacks, many combined models have been proposed [37–39]. For example, [37] proposed combining a TV filter with a fourth-order partial differential equations filter to remove noise, and [39] considered a novel total variation and high-order total variation adaptive model for restoring blurred images with Cauchy noise. These methods have the advantages of better preserving edges and reducing staircase effects.

Motivated by the properties of high-order TV holding high-order smoothness and first-order TV preserving local discontinuity well, we propose a LRTC model which contains two terms: parallel matrix factorization as the data fidelity term to guarantee global low rankness, the first-order TV and second-order TV as regularization terms to simultaneously exploit the first- and second-order derivatives information, which effectively integrates the global and local property of the underlying tensor.

To present our motivation in detail, two examples performed on video data *suize* are given: one is structure missing with three pixels width, which forms black stripes; the other is random missing with a 20% sampling rate. In Fig. 1, the first example shows that when the stripes are wide, considering the first-order information is not enough. The proposed method excludes more stripes, since it can utilize second-order information of tensor. The second example shows that the proposed method alleviates staircase effects, since it can hold second-order smoothness of tensor. Two



**Fig. 1** The first row: the recovered results with structure missing. The second row: the recovered results with random missing

examples demonstrate that incorporating the first- and second-order derivatives information achieves higher visual quality than only considering first-order derivatives information.

**Contributions.** We summarize the contributions of our work into three folds: (1) We observe that the high-order TV can make use of high-order derivatives information to reduce the staircase effect in denoising problems, and first-order TV can use first-order derivatives information to preserve edges well; however, both have their advantages and disadvantages; (2) Inspired by the observation above, we suggest a LRTC model integrating first- and second-order TV into parallel matrix factorization (TMac)-based method, which can exploit first- and second-order derivatives information via an adaptive balancing parameter and simultaneously utilize the global and local property of tensor; (3) An efficient proximal alternating optimization (PAO)-based algorithm is developed and its convergence theory is guaranteed. The effectiveness and efficiency of the proposed algorithms have been validated by carrying out extensive experiments on three types of data, including color images, grayscale videos, and MRIs.

**Organization.** The remainder of this paper is organized as follows. In Section 2, we introduce some tensor basics and PAO algorithm. In Section 3, we propose our model and develop an efficient PAO-based scheme to solve the proposed model. In Section 4, numerical results are reported to show the effectiveness of the proposed method. Related discussions are given in Section 5 and conclusions are in Section 6.

## 2 Preliminary

In this paper, we use low-case letters (such as  $a$ ) for vectors, use upper-case letters (such as  $A$ ) for matrices, and calligraphic letters (such as  $\mathcal{A}$ ) for tensors [3]. In the following subsection, we will introduce some preliminary knowledge of tensor.

## 2.1 Tensor basic

$\mathcal{Y} \in \mathbb{R}^{d_1 \times \dots \times d_N}$  denotes an  $N$ -mode tensor, its Frobenius norm is defined as

$$\|\mathcal{Y}\|_F = \sqrt{\sum_{e_1}^{d_1} \sum_{e_2}^{d_2} \dots \sum_{e_N}^{d_N} y_{e_1 e_2 \dots e_N}^2}. \quad (8)$$

where  $y_{e_1, \dots, e_N}$  is its  $(e_1, \dots, e_N)$ th component.  $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{e_1, e_2, \dots, e_N} x_{e_1 e_2 \dots e_N} y_{e_1 e_2 \dots e_N}$  represents the inner product of  $\mathcal{X}$  and  $\mathcal{Y}$ .

$Y_{(n)} \in \mathbb{R}^{d_n \times \prod_{i \neq n} d_i}$  is the mode- $n$  unfolding of  $\mathcal{Y}$ , where the tensor element  $(e_1, e_2, \dots, e_N)$  maps to the matrix element  $(e_n, j)$  satisfying

$$j = 1 + \sum_{k=1, k \neq n}^N (e_k - 1) J_k, \quad J_k = \prod_{m=1, m \neq n}^{k-1} e_m. \quad (9)$$

The corresponding folding operation is expressed as  $\mathcal{Y} = \text{fold}_n(Y_{(n)})$ . We adopt the  $n$ -rank (Tucker rank) definition for the tensor  $\mathcal{Y}$  in our work:

$$n\text{-rank}(\mathcal{Y}) = (\text{rank}(Y_{(1)}), \dots, \text{rank}(Y_{(N)})). \quad (10)$$

When  $Y_{(n)}$  is low-rank for all  $n$ , the tensor  $\mathcal{Y}$  is low-rank.

Let  $P_\Omega(\mathcal{Y})$  denote the related entries from tensor  $\mathcal{Y}$  in  $\Omega$  which is an index set for available entries, and the remaining entries be zeros. Mathematically, it can be expressed as

$$(\mathcal{P}_\Omega(\mathcal{Y}))_{d_1, d_2, \dots, d_N} = \begin{cases} Y_{d_1, d_2, \dots, d_N}, & (d_1, d_2, \dots, d_N) \in \Omega(i, j) \\ 0, & \text{otherwise} \end{cases}. \quad (11)$$

For more information on the basics of tensors, please refer to [3].

## 2.2 Proximal operator

A formula is given to characterize the proximal operator operation of a convex function  $f(x)$  in [40]:

$$\text{prox}_f(y) = \arg \min_x f(x) + \frac{\rho}{2} \|x - y\|^2, \quad (12)$$

where  $\rho > 0$  is proximal parameter. Through (12), we can obtain two important conclusions: one is that  $\min_x f(x)$  is equivalent to  $\min_{x, y} f(x) + \frac{\rho}{2} \|x - y\|^2$ , and another is that (12) 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  $x$ .

## 3 Proposed model and algorithm

In this section, we will give our LRTC model and its optimization.

### 3.1 Proposed model

TV can preserve discontinuity well but suffer from staircase effects, while second-order TV can alleviate staircase effects but not preserve discontinuity well. An adaptive model can take advantage of the best characteristics of two regularization terms and simultaneously manage to overcome shortcomings of each other. We formally integrate the first-order TV and second-order TV into parallel matrix factorization model for LRTC problem:

$$\begin{aligned} \min_{\mathcal{Y}, X, A} \quad & \sum_{n=1}^N \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2 + \eta \sum_{i=1}^{r_3} \sum_{j=1}^{d_1 d_2} \|(DX_3)_{i,j}\|_2 \\ & + (1 - \eta) \sum_{i=1}^{r_3} \sum_{j=1}^{d_1 d_2} \|(D^2 X_3)_{i,j}\|_2 \\ \text{s.t.} \quad & \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F}, \end{aligned} \quad (13)$$

where  $A_n \in \mathbb{R}^{d_n \times r_n}$ ,  $X_n \in \mathbb{R}^{r_n \times s_n}$  ( $s_n = \prod_{i=1, i \neq n}^N d_i$ );  $\sum_{n=1}^N \alpha_n = 1$ ; the rank of  $Y_{(n)}$  is given as a prior, denoted by  $r_n$ ;  $\eta$  is a adaptive regularization parameter to balance two regularization term.

The first term in (13) is the data fidelity term, which guarantees the global low-rankness. The second and third terms are the regularization terms, which exert gradient domain sparse constraints along horizontal and vertical directions in spatial domain to promote the smoothness of the underlying tensor  $\mathcal{Y}$  [1, 20]. The two regularization terms in model (13) are as follows:

$$\|(DX_3)_{i,j}\|_2 = \sqrt{((D_x^+ X_3)_{i,j})^2 + ((D_y^+ X_3)_{i,j})^2} \quad (14)$$

and

$$\|(D^2 X_3)_{i,j}\|_2 = \sqrt{((D_{xx}^+ X_3)_{i,j})^2 + ((D_{xy}^{++} X_3)_{i,j})^2 + ((D_{yx}^{++} X_3)_{i,j})^2 + ((D_{yy}^+ X_3)_{i,j})^2}. \quad (15)$$

where  $D_x^\pm$  and  $D_y^\pm$  are the discrete gradient operators in the horizontal and vertical directions of the  $j$ th data point of  $X_3^{(i)}$ ;  $D_{xx}^{++}$ ,  $D_{xy}^{++}$ ,  $D_{yx}^{++}$ ,  $D_{yy}^{++}$  represent the second-order difference operator (the details in Table 1).

The methods to choose the value of regularization parameters  $\eta$  are not unique [37, 38, 41]. Refer to [37], we scheme that:

$$\eta = \frac{\gamma_0 + \gamma_1 \|DX_3\|_2^2}{1 + \gamma_0 + \gamma_1 \|DX_3\|_2^2}, \quad (16)$$

where  $\gamma_0$  is a very small positive number and  $\gamma_1$  is a contrast factor. These two parameters can be selected empirically as fixed values [41]. Large  $\|DX_3\|_2$  corresponds to non-smooth area where it is more likely to exist some sharp edges. Thus, the corresponding  $\eta$  is large according to (16), which leads to that the first-order TV term

**Table 1** Discretization used in the implementations

Difference operators	The details of operators
$(D_x^\pm u)_{i,j}$	$\pm[u_{i\pm 1,j} - u_{i,j}]$
$(D_y^\pm u)_{i,j}$	$\pm[u_{i,j\pm 1} - u_{i,j}]$
$(D_{xx}^+ u)_{i,j}$	$[(D_x^+ u)_{i,j} - (D_x^+ u)_{i-1,j}]$
$(D_{xy}^{++} u)_{i,j}$	$[(D_x^+ u)_{i,j+1} - (D_x^+ u)_{i,j}]$
$(D_{yx}^{++} u)_{i,j}$	$[(D_y^+ u)_{i+1,j} - (D_y^+ u)_{i,j}]$
$(D_{yy}^+ u)_{i,j}$	$[(D_y^+ u)_{i,j} - (D_y^+ u)_{i,j-1}]$

dominates and preserves discontinuity well. Small  $\|DX_3\|_2$  corresponds to smooth area where it is more likely to generate staircase effects. Thus, the corresponding  $\eta$  is small according to (16), which leads to that the second-order TV term dominates and alleviates staircase effects.

### 3.2 Proposed algorithm

In this subsection, we develop the optimization algorithm. It is obvious that (13) is not a jointly convex problem for  $(X, A, Y)$ , which makes it difficult for us to obtain a global optimal solution directly. However, the local optimal solution of (13) is satisfactory likewise since the convexity for each variables  $X$ ,  $A$ , and  $Y$ , while keeping with others fixed [14, 20, 42].

#### 3.2.1 Proximal alternating optimization scheme

Firstly, we introduce the proximal operator into objective function  $f(X, A, \mathcal{Y})$  in (13). Specifically,  $X$ ,  $A$ , and  $Y$  can be iteratively updated as follows:

$$\begin{aligned}
 X^{k+1} &= \arg \min_X f(X, A^k, Y^k) + \frac{\rho}{2} \|X - X^k\|_F^2, \\
 A^{k+1} &= \arg \min_A f(X^{k+1}, A, Y^k) + \frac{\rho}{2} \|A - A^k\|_F^2, \\
 Y^{k+1} &= \arg \min_Y f(X^{k+1}, A^{k+1}, Y) + \frac{\rho}{2} \|Y - Y^k\|_F^2,
 \end{aligned} \tag{17}$$

where  $\rho > 0$ , and the superscript  $k$  denotes the iteration indices. Obviously,  $X_n$  ( $n = 1, 2, 4, \dots, N$ ),  $A_n$  ( $n = 1, 2, 3, \dots, N$ ), and  $Y_{(n)}$  ( $n = 1, 2, 3, \dots, N$ ) have closed-form solution, but  $X_3$  has not. Next, we present the optimizations in detail.

#### 3.2.2 Optimization with respect to $X$ , $A$ , and $\mathcal{Y}$

Since the  $X$ - and  $A$ -subproblems can be decomposed into  $N$  independent problems, we solve them separately. Obviously, all subproblems but one ( $X_3$ -subproblem) have



close-form solutions, then (17) can be updated as:

$$\begin{aligned} X_n^{k+1} &= \left( (A_n^k)^T A_n^k + \rho I_1 \right)^{\dagger} \left( (A_n^k)^T Y_n^k + \rho X_n^k \right), n = 1, 2, 4, \dots, N, \\ A_n^{k+1} &= \left( Y_n^{(k)} (X_n^{k+1})^T + \rho A_n^k \right) \left( X_n^{k+1} (X_n^{k+1})^T + \rho I_2 \right)^{\dagger}, n = 1, 2, 3, \dots, N, \\ \mathcal{Y}^{k+1} &= \mathcal{P}_{\Omega} \left( \sum_{n=1}^N \alpha_n \text{fold}_n \left( \frac{A_n^{k+1} X_n^{k+1} + \rho Y_{(n)}^k}{1 + \rho} \right) \right) + \mathcal{F}, n = 1, 2, 3, \dots, N, \end{aligned} \quad (18)$$

where  $I_1$  and  $I_2$  are identity matrices of size  $r_n \times r_n$ . For  $X_n$  ( $n \neq 3$ ) subproblems, the complexity is  $O(d_n r_n^2 + d_n r_n s_n + r_n^2 s_n)$ . For  $A_n$  and  $\mathcal{Y}$  subproblems, the complexities are  $O(d_n r_n^2 + d_n r_n s_n + r_n^2 s_n)$  and  $O\left(\sum_{n=1}^N r_n d_n s_n\right)$  respectively.

### 3.2.3 Optimization with respect to $X_3$

The  $X_3$ -subproblem is as follows:

$$\begin{aligned} \hat{X}_3^{k+1} &= \arg \min_{\hat{X}_3} \frac{1}{2} \left\| \hat{Y}_{(3)}^k - \hat{X}_3 \hat{A}_3^k \right\|_F^2 + \frac{\rho}{2} \left\| \hat{X}_3 - \hat{X}_3^k \right\|_F^2 \\ &\quad + \eta \sum_{i=1}^{r_3} \sum_{j=1}^{d_1 d_2} \left\| (D \hat{X}_3)_{i,j} \right\|_2 + (1 - \eta) \sum_{i=1}^{r_3} \sum_{j=1}^{d_1 d_2} \left\| (D^2 \hat{X}_3)_{i,j} \right\|_2, \end{aligned} \quad (19)$$

where  $\hat{X}_3$  denotes the transpose of  $X_3$ . We iteratively solve it by using the ADMM [1, 15, 16] which is a powerful tool for convex optimization problems. To make ADMM applicable, we transform the  $X_3$ -subproblem to the following problem via introducing auxiliary variables  $B$  and  $R$ :

$$\begin{aligned} \arg \min_{X_3} \quad & \frac{1}{2} \left\| \hat{Y}_{(3)}^k - \hat{X}_3 \hat{A}_3^k \right\|_F^2 + \frac{\rho}{2} \left\| \hat{X}_3 - \hat{X}_3^k \right\|_F^2 + \eta \sum_{i=1}^{r_3} \sum_{j=1}^{d_1 d_2} \left\| (B)_{i,j} \right\|_2 \\ & + (1 - \eta) \sum_{i=1}^{r_3} \sum_{j=1}^{d_1 d_2} \left\| (R)_{i,j} \right\|_2 \\ \text{s.t.} \quad & B = D \hat{X}_3, R = D^2 \hat{X}_3, \end{aligned} \quad (20)$$

where  $B = (B_1, B_2)$ ,  $B_i \in \mathbb{R}^{r_3 \times d_3}$  ( $i = 1, 2$ );  $R = (R_1, R_2, R_3, R_4)$ ,  $R_i \in \mathbb{R}^{r_3 \times d_3}$  ( $i = 1, 2, 3, 4$ ). We separate the variables in (20) into two groups:  $\hat{X}_3$  and  $(B, R)$ , such that (20) fits the framework of ADMM:

$$\begin{aligned} \hat{g}_1(\hat{X}_3) &= \frac{1}{2} \left\| \hat{Y}_{(3)}^k - \hat{X}_3 \hat{A}_3^k \right\|_F^2 + \frac{\rho}{2} \left\| \hat{X}_3 - \hat{X}_3^k \right\|_F^2, \\ \hat{g}_2([B, R]) &= \eta \sum_{i=1}^{r_3} \sum_{j=1}^{d_1 d_2} \left\| B_{i,j} \right\|_2 + (1 - \eta) \sum_{i=1}^{r_3} \sum_{j=1}^{d_1 d_2} \left\| R_{i,j} \right\|_2, \end{aligned} \quad (21)$$

where the superscript  $t$  denotes the inner iteration indices under the default  $k$ th outer iteration. And the constraints are rewritten as follows:

$$\begin{aligned} D\hat{X}_3 - I_3B &= 0, \\ D^2\hat{X}_3 - I_4R &= 0, \end{aligned} \quad (22)$$

where  $I_3$  and  $I_4$  are identity matrices of size  $r_3 \times s_3$ .

The augmented Lagrangian function of (20) is

$$\begin{aligned} \mathcal{L}(\hat{X}_3, B, R, \Lambda_1, \Lambda_2) &= \hat{g}_1(\hat{X}_3) + \hat{g}_2([B, R]) + \left\langle \Lambda_1, D\hat{X}_3 - I_3B \right\rangle + \left\langle \Lambda_2, D^2\hat{X}_3 - I_4R \right\rangle \\ &\quad + \frac{\beta}{2} \|D\hat{X}_3 - I_3B\|_F^2 + \frac{\beta}{2} \|D^2\hat{X}_3 - I_4R\|_F^2, \end{aligned} \quad (23)$$

where  $\Lambda_i$  is Lagrangian multiplier and  $\beta > 0$  is the penalty parameter to control the speed of convergence. Because the joint minimization problem can be decomposed into two easier and smaller subproblems:  $X_3$  and  $(B, R)$ , such that the minimization of two blocks of variables can be solved by an alternating order, so we solve problem (23) with following scheme:

$$\begin{cases} \hat{X}_3^{t+1} = \arg \min \mathcal{L}(\hat{X}_3, B, R, \Lambda_1, \Lambda_2), \\ (B^{t+1}, R^{t+1}) = \arg \min \mathcal{L}(\hat{X}_3^{t+1}, B, R, \Lambda_1, \Lambda_2), \\ \Lambda_1^{t+1} = \Lambda_1^t + \beta(D\hat{X}_3^{t+1} - I_3B^{t+1}), \\ \Lambda_2^{t+1} = \Lambda_2^t + \beta(D^2\hat{X}_3^{t+1} - I_4R^{t+1}). \end{cases} \quad (24)$$

Since  $\mathcal{L}$  is decoupled about  $B$  and  $R$ , we remark that the calculations of two variables are independent. Following, we give the details on each step of the ADMM.

For the  $X_3$ -subproblem:

$$\begin{aligned} \hat{X}_3^{t+1} &= \arg \min_{\hat{X}_3} \frac{1}{2} \|\hat{Y}_{(3)} - \hat{X}_3\hat{A}_3\|_F^2 + \frac{\rho}{2} \|\hat{X}_3 - \hat{X}_3^t\|_F^2 + \frac{\beta}{2} \|B^t - D\hat{X}_3 - \frac{\Lambda_1^t}{\beta}\|_F^2 \\ &\quad + \frac{\beta}{2} \|R^t - D^2\hat{X}_3 - \frac{\Lambda_2^t}{\beta}\|_F^2, \end{aligned} \quad (25)$$

then it can be expressed as follows through the classical Sylvester matrix equation,

$$\begin{aligned} &\hat{X}_3(\hat{A}_3(\hat{A}_3)^T) + \beta D^T D\hat{X}_3 + \beta(D^2)^T D^2\hat{X}_3 + \rho\hat{X}_3 \\ &= \rho\hat{X}_3^t + \hat{Y}_{(3)}(\hat{A}_3)^T - D^T\Lambda_1^t + \beta D^T I_3B^t - (D^2)^T\Lambda_2^t + \beta(D^2)^T I_4R^t. \end{aligned} \quad (26)$$

To solve (26), we refer [43]:

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

$$AX + XB = C, \quad (27)$$

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

$$A = U_1 \Lambda_1 U_1^T, \quad B = U_2 \Lambda_2 U_2^T, \quad (28)$$

where  $\Lambda_1$  and  $\Lambda_2$  are diagonal matrices;  $U_1$  and  $U_2$  are unitary matrices. Then the unique solution is

$$X = U_1 \left( (1./T) .* \left( U_1^T C U_2 \right) \right) U_2^T, \quad (29)$$

where “.” represents the component-wise multiplication, “./” represents the component-wise division, and  $T = (\text{diag}(\Lambda_1), \text{diag}(\Lambda_1), \dots, \text{diag}(\Lambda_1))^T + (\text{diag}(\Lambda_2), \text{diag}(\Lambda_2), \dots, \text{diag}(\Lambda_2))$ .

Because  $D^T D$  and  $(D^2)^T D^2$  are circulant matrix, we can use the Fourier transformation to diagonalize them; at the same time, we can use the SVD to diagonalize  $\hat{A}_3(\hat{A}_3)^T$ . Letting

$$D^T D = F^* \Psi_1^2 F, \quad (D^2)^T (D^2) = F^* \Psi_2^2 F, \quad \hat{A}_3(\hat{A}_3)^T = U \Sigma^2 U^*. \quad (30)$$

Then, by using the Kronecker product, (26) can be reformulated as:

$$\begin{aligned} \text{vec}(\hat{X}_3) &= (U \otimes F^*) (\Sigma^2 \otimes I + \beta I \otimes \Psi_1^2 + \beta I \otimes \Psi_2^2 + \rho I \otimes I)^{-1} (U^* \otimes F) \\ &\quad \cdot \text{vec}(\rho \hat{X}_3 + \hat{Y}_{(3)}(\hat{A}_3)^T - D^T (\Lambda_1^t + \beta B^t) - (D^2)^T (\Lambda_1^t + \beta R^t)), \end{aligned} \quad (31)$$

where  $\text{vec}(\cdot)$  is a vector obtained by lexicographical ordering.

With **Theorem** (1), the solution to (25) can be expressed as

$$\hat{X}_3^{t+1} = F^* (1./T .* (F K U)) U^*, \quad (32)$$

where

$$K = \rho \hat{X}_3 + \hat{Y}_{(3)}(\hat{A}_3)^T - D^T (\Lambda_1^t + \beta B^t) - (D^2)^T (\Lambda_1^t + \beta R^t) \quad (33)$$

and

$$\begin{aligned} T &= (\text{diag}(\Sigma^2), \text{diag}(\Sigma^2), \dots, \text{diag}(\Sigma^2)) + \beta I (\text{diag}(\Psi_1^2), \text{diag}(\Psi_1^2), \dots, \text{diag}(\Psi_1^2)) \\ &\quad + \beta I (\text{diag}(\Psi_2^2), \text{diag}(\Psi_2^2), \dots, \text{diag}(\Psi_2^2)) + \rho .* \text{ones}(d_3, r_3). \end{aligned} \quad (34)$$

Here, the complexity of computing  $\hat{X}_3$  is  $O(2r_3^2 s_3 + r_3 s_3 \log s_3)$ , while the complexity of computing  $X_3$  is  $O(d_3 r_3^2 + d_3 r_3 s_3 + r_3^2 s_3)$  in the case of first- and second-order TV of  $Y_3$ . And the complexity of computing  $Y_3$  is  $O(d_3 r_3 s_3)$  in the both cases.

For the  $B$ -subproblem:

$$B^{t+1} = \arg \min_B \eta \sum_{i=1}^{r_3} \sum_{j=1}^{d_1 d_2} \|B_{i,j}\|_2 + \frac{\beta}{2} \left\| B^t - D \hat{X}_3^{t+1} - \frac{\Lambda_1^t}{\beta} \right\|_F^2. \quad (35)$$

It has the following closed-form solution by the soft shrinkage operator:

$$B_{i,j}^{t+1} = \text{shrink} \left( \left( D X_3^{t+1} + \frac{\Lambda_1^t}{\eta} \right)_{i,j}, \frac{\eta}{\beta} \right), \quad (36)$$

and the shrinkage operator is defined by

$$\text{shrink}(p, q) = \begin{cases} 0 & , p = 0 \\ (\|p\|_2 - q) \frac{p}{\|p\|_2} & , p \neq 0 \end{cases}. \quad (37)$$

Here, the complexity of computing  $B$  is  $O(s_3 r_3)$ , while the complexity is  $O(s_3 d_3)$  in the case of TV of  $Y_3$ . For the  $R$ -subproblem:

$$R^{t+1} = \arg \min_R (1 - \eta) \sum_{i=1}^{r_3} \sum_{j=1}^{d_1 d_2} \|R_{i,j}\|_2 + \frac{\beta}{2} \left\| R^t - D^2 \hat{X}_3 - \frac{\Lambda_2^{(t)}}{\beta} \right\|_F^2. \quad (38)$$

Similarly, it has the closed-form solution as follows:

$$R_{i,j}^{t+1} = \text{shrink} \left( \left( D^2 X_3^{t+1} + \frac{\Lambda_2^t}{\eta} \right)_{i,j}, \frac{1 - \eta}{\beta} \right). \quad (39)$$

Here, the complexity of computing  $R$  is  $O(s_3 r_3)$ , while the complexity is  $O(s_3 d_3)$  in the case of second-order TV of  $Y_3$ .

Finally, we update Lagrangian multiplier  $\Lambda$ :

$$\begin{aligned} \Lambda_1^{t+1} &= \Lambda_1^t + \beta \left( D \hat{X}_3^{t+1} - I_3 B^{t+1} \right), \\ \Lambda_2^{t+1} &= \Lambda_2^t + \beta \left( D^2 \hat{X}_3^{t+1} - I_4 R^{t+1} \right). \end{aligned} \quad (40)$$

Obviously,  $d_3$  is greater than  $r_3$ , so the above computational complexity can exactly demonstrate the economy of the first- and second-order TV operating on the factor matrix  $X_3$ , not on the mode-3 unfolding of tensor  $Y_3$ .

The proposed algorithm can be summarized as **Algorithm 1**.

---

**Algorithm 1** The proposed algorithm for solving (13).

---

**Input:** The observed tensor  $\mathcal{Y}_{observed} \in \mathbb{R}^{d_1 \times \dots \times d_N}$ , index set  $\Omega$ .

**Output:** The estimated tensor  $\mathcal{Y}$ .

```

1: Parameters:  $\alpha_i, \rho, \beta, k = 1, \dots, k_{max}, l = 1, \dots, l_{max}, \eta;$ 
2: Initialize:  $X^0, A^0, Y^0, \varepsilon = 10^{-4}$ , outer iteration  $k_{max} = 2000$ , inner iteration
    $l_{max} = 10;$ 
3: Out loop: While not converged and  $k < k_{max}$ 
4:     Update  $X_n^k (n = 1, 2, 4, \dots, N)$  by solving (18);
5:     Update  $A_n^k, \mathcal{Y}^k$ , by solving (18);
6:     Inner loop: While not converged and  $l < l_{max}$  do
7:         Update  $\eta$  by (16);
8:         Update  $X_3$  by solving (31);
9:         Update  $B$  by solving (36);
10:        Update  $R$  by solving (39);
11:        Update  $\Lambda_1, \Lambda_2$  via (40).
12:    End While
13: End While
14: return  $\mathcal{Y}$ 

```

---

### 3.3 Convergence of the proposed algorithm

In this section, we analyze the convergence of the proposed algorithm. For convenience, the function (13) is rewritten as

$$f(X, A, \mathcal{Y}) = Q(X, A, \mathcal{Y}) + f_{X_3}(X_3), \quad (41)$$

where

$$Q(X, A, \mathcal{Y}) = \sum_{n=1}^N \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2 \quad (42)$$

and

$$f_{X_3}(X_3) = \eta \sum_{i=1}^{r_3} \sum_{j=1}^{d_1 d_2} \|(DX_3)_{i,j}\|_2 + (1 - \eta) \sum_{i=1}^{r_3} \sum_{j=1}^{d_1 d_2} \|(D^2 X_3)_{i,j}\|_2. \quad (43)$$

Before proceeding with our analysis, let us briefly review the Kurdyka-Lojasiewicz (KL) property, which is central to the convergence analysis.

**Definition 1** (KL property, [44])

- (a) The function  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$  is said to have the KL property at  $x \in \text{dom} \partial f$  if there exist  $\eta \in (0, +\infty)$ , a neighborhood  $U$  of  $\tilde{x}$  and a continuous concave function  $\phi : [0, \eta) \rightarrow [0, +\infty)$  such that:
  - (i)  $\phi(0) = 0$ ,
  - (ii)  $\phi$  is  $C^1$  on  $(0, \eta)$ ,
  - (iii) for all  $s \in (0, \eta)$ ,  $\phi(s) > 0$ ,
  - (iv) for all  $x$  in  $U \cap [f(\tilde{x}) < f < f(\tilde{x}) + \eta]$ , the KL inequality holds  $\phi'(f(x) - f(\tilde{x})) \text{dist}(0, \partial f(x)) \geq 1$ ;
- (b) Proper lower semi-continuous functions which satisfies the KL inequality at each point of  $\text{dom} \partial f$  are called KL functions, where the norm involved in  $\text{dist}(\cdot, \cdot)$  is  $\|\cdot\|_2$  and the convention  $\text{dist}(0, 0) := +\infty$ .

**Lemma 1** [44] Let  $f : \mathbb{R}^n \rightarrow \mathbb{R} \cup +\infty$  be a proper lower semi-continuous function. The sequences  $(x^k)_k \in N \subset \mathbb{R}^n$ , which satisfy the following conditions:

- H1. (Sufficient decrease condition). For each  $k \in N$ ,  $f(x^{k+1}) + a \|x^{k+1} - x^k\|_2^2 \leq f(x^k)$  holds for a constant  $a \in (0, +\infty)$ ;
- H2. (Relative error condition). For each  $k \in N$ , there exists  $w^{k+1} \in \partial f(x^{k+1})$  such that  $\|w^{k+1}\|_2 \leq b \|x^{k+1} - x^k\|_2$  holds for a constant  $b \in (0, +\infty)$ ;
- H3. (Continuity condition). There exists a subsequence  $(x^{k_j})_{j \in \mathbb{N}}$  and  $\tilde{x} \in \mathbb{R}^n$  such that  $x^{k_j} \rightarrow \tilde{x}$  and  $f(x^{k_j}) \rightarrow f(\tilde{x})$ , as  $j \rightarrow \infty$ . if  $f$  has the KL property at  $\tilde{x}$  appearing in H3, then
  - (i)  $x^k \rightarrow \tilde{x}$ ;
  - (ii)  $\tilde{x}$  is a critical point of  $f$ , i.e.,  $0 \in \partial f(\tilde{x})$ ;
  - (iii) the sequence  $x_k^k \in \mathbb{N}$  has a finite length, i.e.,  $\sum_{k=0}^{+\infty} \|x^{k+1} - x^k\|_2^2 < +\infty$ .

Next, we show that the objective function  $f$  and the iterative sequence  $(X^k, A^k, \mathcal{Y}^k)$  generated by (18) satisfy the assumptions imposed in **Lemma 1**, by which we establish the convergence of the proposed algorithm.

**Theorem 2** Assume that the sequence  $(X^k, A^k, \mathcal{Y}^k)$  generated by Algorithm 1 is bounded, then it converges to a critical point of  $f$ .

*Proof* Firstly, it is obvious that function  $Q$  is  $C^1$  with Lipschitz continuous gradient (consequence of the boundedness of  $(X^k, A^k, \mathcal{Y}^k)_{k \in \mathbb{N}}$ ), and  $f_{X_3}$  is proper and lower semi-continuous. Thus,  $f$  is a proper lower semi-continuous function. And another is the structural. Secondly, noting that the **Algorithm 1** is an instance of algorithm (61)-(62)-(63) shown in [44] with  $B_i = \rho I$  ( $\rho > 0$ ) (see [44], Remark 6.1), then the proof is a direct application of ([44], Th. 6.2). Thus, the iterative sequences  $(X^k, A^k, \mathcal{Y}^k)$  generated by (18) and (32) satisfy the condition  $H1$ ,  $H2$ , and  $H3$  in Lemma 1.

Thirdly, the reshape operator is linear mapping between finite-dimension spaces; thus,  $Q$  is semi-algebraic functions [44]. Since function  $f_{X_3}$  is a finite linear combinations of compositions of absolute value function and linear polynomials, all of which are semi-algebraic functions,  $f_{X_3}$  is a semi-algebraic function. Then, finite sums and finite products of semi-algebraic are semi-algebraic [44], so  $f$  is semi-algebraic. Therefore, we can obtain that  $f$  satisfies the KL property at each  $(X^k, A^k, \mathcal{Y}^k) \in \text{dom}(f)$ , since  $f$  is semi-algebraic on  $\text{dom}(f)$ .

According to Lemma 1, the bounded sequences generated by **Algorithm 1** converges to a critical point of  $f$ . The proof is completed.  $\square$

## 4 Experiments

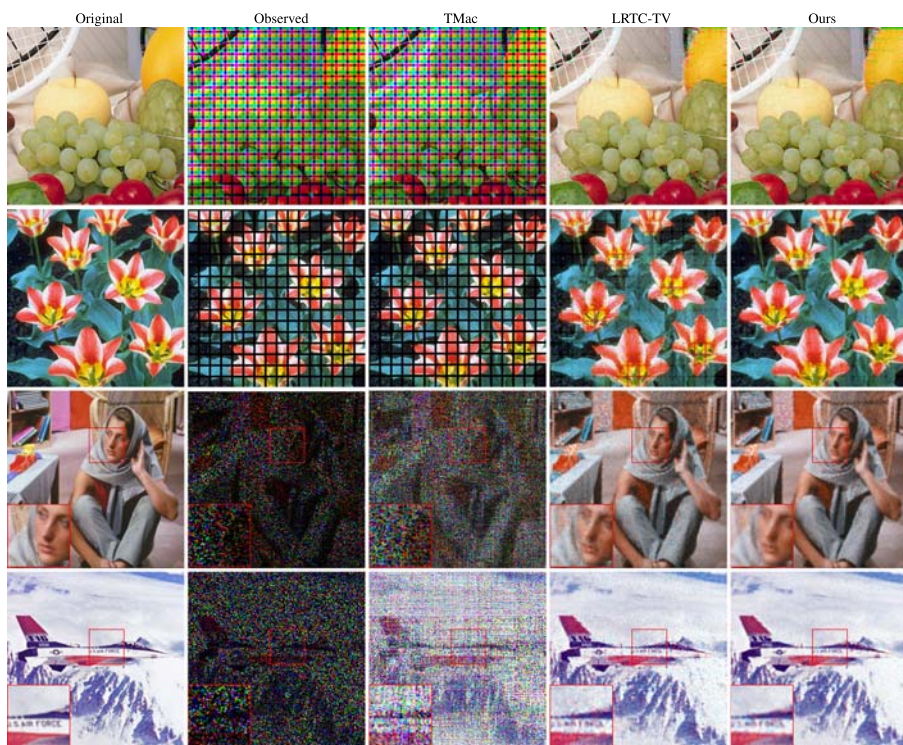
To demonstrate the effectiveness of our algorithm, we compare the proposed method with TMac proposed in [14], and LRTC-TV proposed in [20]. Three kinds of tensors, including color images, grayscale videos, and MRIs are chosen as test objects. We adopt two indicators to measure the quality of the recovered matrix. One of them is the peak signal-to-noise ratio (PSNR), which is defined by

$$\text{PSNR} = 10 \log_{10} \frac{n^2 \bar{Y}_{\text{true}}^2}{\|Y - Y_{\text{true}}\|_F^2}. \quad (44)$$

And another is the structural similarity (SSIM) [45], which measures the structural detail similarity between the estimated matrix and the original matrix. It is defined by

$$\text{SSIM} = \frac{(2\mu_Y \mu_{Y_{\text{true}}})(2\sigma_{Y Y_{\text{true}}} + I_2)}{(\mu_Y^2 + \mu_{Y_{\text{true}}}^2 + I_1)(\sigma_Y^2 + \sigma_{Y_{\text{true}}}^2 + I_2)}, \quad (45)$$

where  $Y_{\text{true}}$ ,  $\bar{Y}_{\text{true}}$ , and  $Y$  are the original matrix, the maximum pixel value of the original matrix, and the estimated matrix respectively.  $\mu_Y$  and  $\mu_{Y_{\text{true}}}$  are the mean of matrix  $Y$  and  $Y_{\text{true}}$  respectively.  $\sigma_Y$  and  $\sigma_{Y_{\text{true}}}$  denote their standard deviations respectively,  $\sigma_{Y Y_{\text{true}}}$  is the covariance of  $Y$  and  $Y_{\text{true}}$ , and  $I_1, I_2$  are constants. There, we calculate the PSNR and SSIM of each frame (band) of the tensor and take their average values as the result.



**Fig. 2** The recovered results for four color images: *fruits*, *tulps*, *barbara*, and *airplane*. From left to right: the original images, the observed images, the recovered results by TMac, LRTC-TV, and our method, respectively

When the relative change of the two successive estimated tensors (Rel) is less than or equal to the preset tolerance value, the algorithm stops. Mathematically, the termination condition for algorithm can be expressed as

$$\text{Rel} = \frac{\|\mathcal{Y}^{(k+1)} - \mathcal{Y}^{(k)}\|_F}{\|\mathcal{Y}^{(k)}\|_F} < \varepsilon, \quad (46)$$

**Parameter settings** For TMac, LRTC-TV, and the proposed method, the set of proximal parameter is  $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$ . For LRTC-TV and the proposed method, the set of penalty parameter is  $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3\}$ . We adjust the parameters in the above sets of all experiments to obtain the best PSNR as the completion result for three methods. For the proposed method,  $\gamma_0$  and  $\gamma_1$  in (16) are empirically set to be 0.0001 and 0.01 [41]. In the following experiments, the maximum iteration number is 2000, the tolerance is set to be  $10^{-4}$ , and the weights of each mode unfolding of tensor  $\alpha_n = 1/N$  ( $n = 1, 2, \dots, N$ ).

For simplicity, we use the numbers of the largest 0.5% singular values to approximate the rank of each mode instead of rank-adjusting schedule. We perform all

**Table 2** The PSNR and SSIM values of eight color images obtained by three LRTC methods

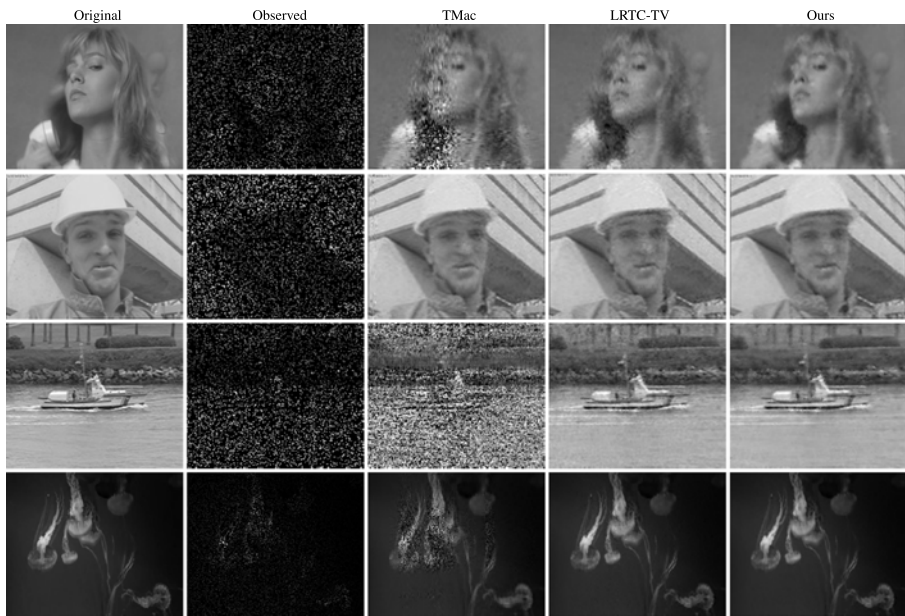
Color Images	SR(%)	PSNR(dB)			SSIM		
		TMac	LRTC-TV	Ours	TMac	LRTC-TV	Ours
Barbara	10	7.54	22.00	<b>24.08</b>	0.0136	0.4841	<b>0.5835</b>
	20	9.20	25.47	<b>26.76</b>	0.0437	0.6298	<b>0.7028</b>
	30	11.68	27.56	<b>28.25</b>	0.1077	0.7198	<b>0.7547</b>
Airplane	10	4.14	18.45	<b>22.62</b>	0.0080	0.2328	<b>0.3962</b>
	20	9.08	22.42	<b>24.39</b>	0.0388	0.4072	<b>0.5334</b>
	30	17.09	26.92	<b>28.86</b>	0.2709	0.5690	<b>0.6414</b>
House	10	6.67	23.43	<b>25.96</b>	0.0105	0.2777	<b>0.3680</b>
	20	11.18	27.52	<b>28.26</b>	0.0848	0.4669	<b>0.5031</b>
	30	19.50	28.85	<b>29.55</b>	0.3325	0.5674	<b>0.5887</b>
Watch	10	11.23	25.10	<b>25.94</b>	0.0140	0.4827	<b>0.5707</b>
	20	12.13	27.77	<b>28.76</b>	0.0399	0.6305	<b>0.7080</b>
	30	13.23	30.49	<b>30.73</b>	0.0762	0.7199	<b>0.7408</b>
Tulips	10	6.48	17.69	<b>20.07</b>	0.0868	0.7275	<b>0.7746</b>
	20	7.58	20.41	<b>25.02</b>	0.0447	0.5481	<b>0.7273</b>
	30	9.03	26.05	<b>26.81</b>	0.0868	0.7275	<b>0.7746</b>
Fruits	10	5.66	17.69	<b>20.65</b>	0.0069	0.2001	<b>0.3509</b>
	20	7.90	19.62	<b>23.13</b>	0.0240	0.2572	<b>0.4643</b>
	30	12.44	24.75	<b>27.02</b>	0.1017	0.4398	<b>0.5514</b>
Sails	10	6.98	19.76	<b>22.37</b>	0.0106	0.4324	<b>0.5313</b>
	20	8.13	23.97	<b>25.31</b>	0.0301	0.6519	<b>0.7069</b>
	30	9.62	25.89	<b>27.02</b>	0.0597	0.7549	<b>0.7996</b>
Average	10	8.25	20.58	<b>23.09</b>	0.0215	0.4053	<b>0.5107</b>
	20	9.32	23.88	<b>25.94</b>	0.0442	0.5130	<b>0.6209</b>
	30	13.23	27.21	<b>28.32</b>	0.1480	0.6427	<b>0.6930</b>

experiments in MATLAB R2014b on a PC with Windows 7, a 1.70GHz Intel Pentium CPU and 4GB of memory.

#### 4.1 Color images

In this subsection, we test seven color images of size  $256 \times 256 \times 3$  with TMac, LRTC-TV, and our proposed method, respectively. We randomly remove 90%, 80%, and 70% data of eight images. Figure 2 displays the recovered images obtained by different methods with  $SR = 0.2$ . Obviously, since LRTC-TV method is piecewise smooth, there are some staircase effects in its recovered images. And TMac method only considers low-rankness, then its recovered images have a poorer visual effect than that recovered by other two TV-based methods. From the zoom-in regions of recovered images, we observe that there are less staircase effects and more smoothness in the





**Fig. 3** The recovered results for four videos: *suize*, *foreman*, *coastguard*, and *sea*. From left to right: the original images, the observed images, the recovered results by TMac, LRTC-TV, and our method, respectively

recovered images by the proposed method. Therefore, the proposed method is visually better than other methods. Table 2 summarizes the PSNR and SSIM values of all test color images recovered by three LRTC methods with different SRs, and the best results are highlighted in bold. Clearly, the proposed method is always superior to two other methods in terms of PSNR and SSIM values, particularly for extremely low SRs.

Moreover, Fig. 2 also shows the recovered results for two images (named *fruits* and *tuilps*) with structure missing. For *fruits*, the fixed gridlines with three-pixels width are removed out of the same position in different channels, which forms color mask. For *tuilps*, the fixed gridlines with three-pixels width are removed out of the different position in different channels, which forms black mask. It can be observed that some residual gridlines are still existing in images via TMac and LRTC-TV, while the most noticeable gridlines are fewer by the proposed method. This is mainly because our method can exploit the second-order derivatives information on image in addition to first-order information.

Overall, the completion quality of our method for all test images under the two sampling methods is consistently better than that of the two compared methods, verifying the superiority of our model.

**Table 3** The PSNR and SSIM values of four videos obtained by three LRTC methods

Gray videos	SR(%)	PSNR(dB)			SSIM		
		TMac	LRTC-TV	Ours	TMac	LRTC-TV	Ours
Suize	5	11.68	27.76	<b>28.48</b>	0.0074	0.4552	<b>0.4845</b>
	10	17.81	30.13	<b>30.56</b>	0.2367	0.5679	<b>0.5745</b>
	20	27.75	32.76	<b>33.51</b>	0.5977	0.6679	<b>0.7031</b>
Coastguard	5	6.62	20.42	<b>22.00</b>	0.0103	0.2894	<b>0.3403</b>
	10	7.80	23.41	<b>24.17</b>	0.0382	0.4527	<b>0.4889</b>
	20	10.94	26.32	<b>26.59</b>	0.1110	0.6295	<b>0.6593</b>
Foreman	5	7.61	22.35	<b>24.18</b>	0.0141	0.3487	<b>0.4551</b>
	10	13.97	26.09	<b>26.71</b>	0.1914	0.5180	<b>0.5887</b>
	20	30.15	30.41	<b>30.89</b>	0.6852	0.6804	<b>0.7167</b>
Sea	5	16.73	33.98	<b>34.10</b>	0.0151	0.4644	<b>0.4673</b>
	10	18.89	36.38	<b>37.29</b>	0.1357	0.5380	<b>0.5704</b>
	20	22.92	39.92	<b>40.52</b>	0.3454	0.5892	<b>0.6588</b>
Average	5	10.66	26.13	<b>27.19</b>	0.0117	0.3894	<b>0.4368</b>
	10	14.62	29.00	<b>29.68</b>	0.1505	0.5192	<b>0.5556</b>
	20	26.55	32.35	<b>32.80</b>	0.4348	0.6418	<b>0.6845</b>

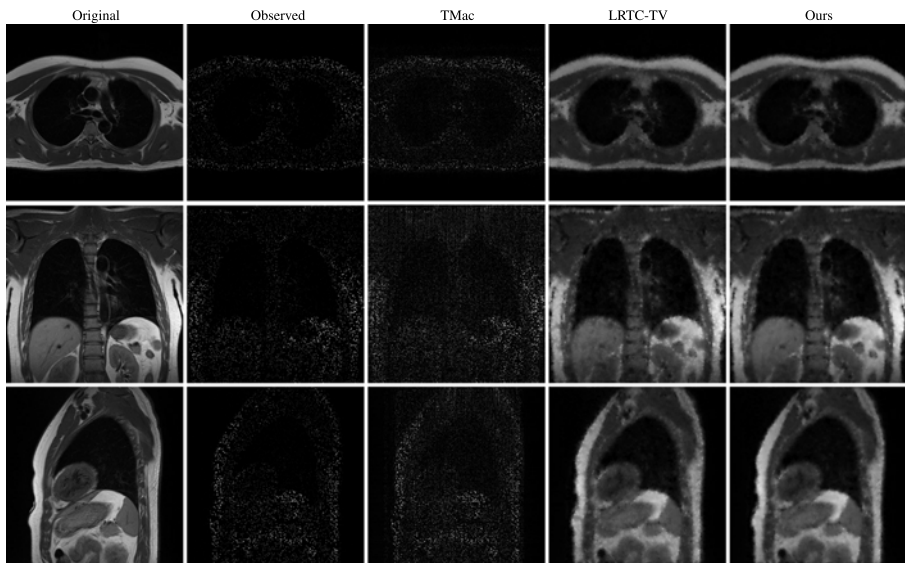
## 4.2 Video

In this subsection, we test four grayscale videos, three of them are of size  $144 \times 176 \times 150$  (named *suize*, *foreman*, and *coastguard*), one is of size  $240 \times 360 \times 118$  (named *sea*), with TMac, LRTC-TV, and the proposed method. We randomly remove 95%, 90%, and 80% data of four videos. Figure 3 shows the recovered results of one frame by different methods with  $SR = 0.2$ . Visually, our method outperforms TMac and LRTC-TV in alleviating staircase effects and keeping second-order smooth of underlying videos. Table 3 summarizes the PSNR and SSIM values of all testing videos obtained by three LRTC methods with different SRs. Numerically, the proposed method has higher improvement than TMac and LRTC-TV.

## 4.3 MRI

In this subsection, we test three MRIs collected in the standard image plane of axial, coronal, and sagittal plane.<sup>1</sup> All of them are cardiac MRI data of size  $256 \times 256 \times 14$ ,  $256 \times 256 \times 19$ , and  $256 \times 256 \times 27$ , respectively. We randomly remove 90%, 80%, and 70% data of three MRIs. Figure 4 displays the recovered results of cardiac data with  $SR = 0.1$ . Table 4 shows the PSNR and SSIM values of all testing MRIs obtained by different methods with different SRs. Clearly, compared with TMac and LRTC-TV, our method has obvious improvement in visual effect and numerical value.

<sup>1</sup><http://www.cardiacatlas.org/studies/amrg-cardiac-atlas/>



**Fig. 4** The recovered results for real MRI data: axial, coronal, and sagittal plane. From left to right: the original images, the observed images, the recovered results by TMac, LRTC-TV, and our method, respectively

## 5 Discussions

In this section, we study the convergences of the proposed method, the effects of the parameters, and the sensitivity to the number of inner iterations of  $X_3$ -subproblem. We choose *suize* data as the test object with  $SR = 0.2$ .

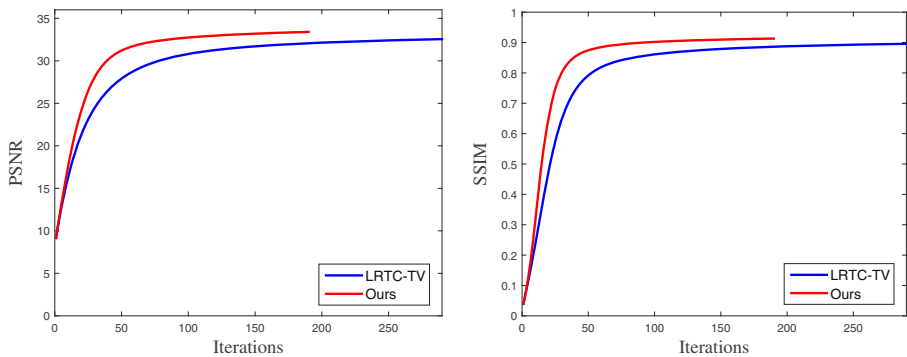
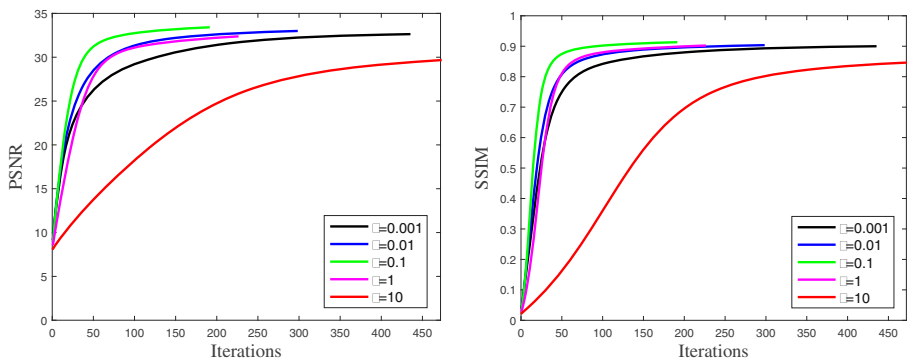
**Convergence.** Figure 5 displays the convergence curves of the proposed algorithm with respect to PSNR and SSIM. Obviously, our method converges, which exactly validates our theory in Section 3.

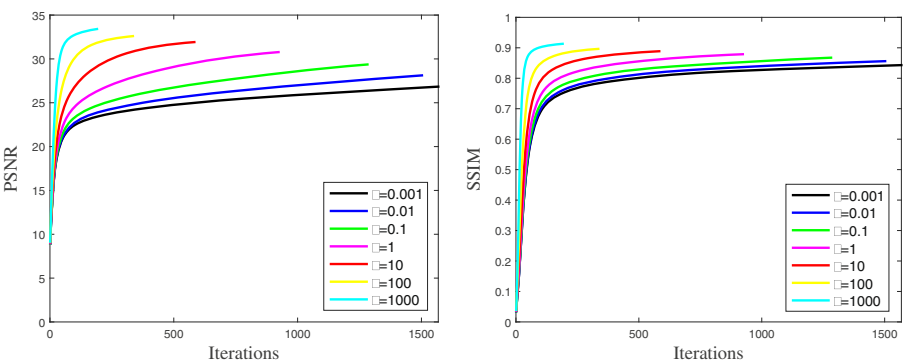
**Parameter analysis.** Firstly, we study the effects of the proximal parameter  $\rho$  and the penalty parameter  $\beta$  on experiments. Figure 6 displays the PSNR and SSIM values with different proximal parameters  $\rho$  during iterations. The plots suggest that for different  $\rho$ , the method can up to approximate PSNR and SSIM values. This is because  $\rho$  mainly affects computing efficiency, not performance. Figure 7 displays the PSNR and SSIM values with different penalty parameters  $\beta$  during iterations. The plots suggest that no matter what  $\beta$  is, the proposed method is effective with theoretically guaranteed. In addition,  $\beta$  can affect the performance of the algorithm.

Secondly, we study the different balancing parameter  $\eta$ . Figure 8 shows the visual effects with the  $\eta = [0, 0.5, 1]$  and when  $\eta$  is adaptive. The figures obviously show that the results are more smooth with the weight of seconder-order TV terms increasing. However, since high-order TV can not preserve details very well, the recovered

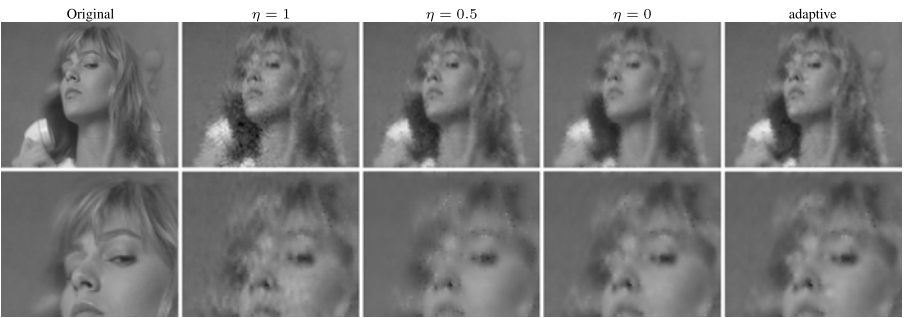
**Table 4** The PSNR and SSIM values of three MRIs obtained by three LRTC methods

MRI	SR(%)	PSNR(dB)			SSIM		
		TMac	LRTC-TV	Ours	TMac	LRTC-TV	Ours
Axial plane	10	14.98	27.85	<b>28.28</b>	0.0406	0.3676	<b>0.3842</b>
	20	15.74	30.21	<b>30.51</b>	0.1013	0.4922	<b>0.5127</b>
	30	16.66	31.78	<b>32.39</b>	0.1676	0.5742	<b>0.6053</b>
Coronal plane	10	12.20	26.83	<b>27.59</b>	0.0208	0.4434	<b>0.4684</b>
	20	13.22	29.48	<b>30.11</b>	0.0579	0.5528	<b>0.5718</b>
	30	14.53	31.58	<b>32.13</b>	0.1158	0.6355	<b>0.6488</b>
Sagittal plane	10	13.24	26.62	<b>27.52</b>	0.0511	0.4059	<b>0.4205</b>
	20	14.07	29.36	<b>29.66</b>	0.1057	0.5296	<b>0.5462</b>
	30	15.06	31.07	<b>31.48</b>	0.1690	0.6165	<b>0.6318</b>
Average	10	13.47	27.21	<b>27.80</b>	0.0375	0.4056	<b>0.4244</b>
	20	14.34	29.68	<b>30.10</b>	0.0883	0.5249	<b>0.5436</b>
	30	15.42	31.48	<b>32.00</b>	0.1508	0.6086	<b>0.6280</b>


**Fig. 5** The convergence curves of the proposed algorithm with respect to PSNR and SSIM

**Fig. 6** The PSNR values and SSIM values obtained by the proposed method for different  $\rho$



**Fig. 7** The PSNR values and SSIM values obtained by the proposed method for different  $\beta$



**Fig. 8** The recovered results of video *suize* obtained by the proposed method for different  $\eta$

**Table 5** The PSNR, SSIM, and CPU time obtained by the proposed method for different iteration number for computing  $X_3$

Inner iteration	2	4	6	8	10	12	14	16
PSNR	33.51	33.51	33.51	33.51	33.52	33.52	33.52	33.52
SSIM	0.7000	0.7022	0.7027	0.7031	0.7036	0.7036	0.7036	0.7036
Time	17.325	19.176	20.805	31.363	45.257	58.430	80.891	100.522

results perform not well in the visual effect. When  $\eta$  is adaptive, the result is visually suitable compared with other results.

Finally, we study the sensitivity of the number of inner iterations for solving the  $X_3$ -subproblem. In order to get the exact solution of the (25), we need to know a certain number of iterations. However, the estimation error may not be accurate enough; thus, employing the exact solutions is not necessary in practical applications. In order to improve the computational efficiency, we have implemented an imprecise version of (25), which only performs several iterations. Table 5 shows that with the increasing of numbers, PSNR and SSIM values almost remain unchanged, but the run time is much longer. Moreover, we find that setting the number of inner iterations to be 4 is sufficient for solving the  $X_3$ -subproblem, which exactly verifies the validity and feasibility of the imprecise version of (25).

## 6 Conclusion

In this paper, we presented an adaptive first- and second-order TV-based model for LRTC problem. Since the weights of first- and second-order TV of factor matrix  $X_3$  are automatically updated, the proposed model can preserve discontinuity well while hold second-order smoothness of tensor and alleviate staircase effects. Then, we proposed an efficient PAO-based algorithm with theoretical guarantee to solve the proposed model. Three kinds of tensor were extensively tested. Compared with TMac and LRTC-TV, our method has great improvement in visual effect and numerical value, particularly for extremely low SRs.

**Funding information** This work is financially supported by the National Natural Science Foundation of China (61772003, 61876203) and the Fundamental Research Funds for the Central Universities (31020180QD126).

## References

1. Zhao, X.-L., Wang, F., Huang, T.-Z., Ng, M.K., Plemmons, R.J.: Deblurring and sparse unmixing for hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **51**(7), 4045–4058 (2013)
2. Wang, B., Gao, X.-B., Tao, D.-C., Li, X.-L.: A unified tensor level set for image segmentation. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **40**(3), 857–867 (2010)
3. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. *SIAM Rev.* **51**(3), 455–500 (2009)
4. Wen, Z.: Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Math. Program. Comput.* **4**(4), 333–361 (2012)
5. Candès, E.J., Plan, Y.: Matrix completion with noise. *Proc. IEEE* **98**(6), 925–936 (2010)
6. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. *Found. Comput. Math.* **9**(6), 717 (2009)
7. Ma, S., Goldfarb, D., Chen, L.: Fixed point and bregman iterative methods for matrix rank minimization. *Math. Program.* **128**(1–2), 321–353 (2011)
8. Toh, K.-C., Yun, S.: An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. *Pac. J. Optim.* **6**(3), 615–640 (2010)
9. Tan, H.-C., Cheng, B., Wang, W.-H., Zhang, Y.-J., Ran, B.: Tensor completion via a multi-linear low-n-rank factorization model. *Neurocomputing* **133**, 161–169 (2014)
10. Yang, J.-H., Zhao, X.-L., Ji, T.-Y., Ma, T.-H., Huang, T.-Z.: Low-rank tensor train for tensor robust principal component analysis. *Appl. Math. Comput.* **367**, 124783 (2020)

11. Hillar, C.J., Lim, L.-H.: Most tensor problems are np-hard. *J. ACM (JACM)* **60**(6), 45 (2013)
12. Zhang, X.-J.: A nonconvex relaxation approach to low-rank tensor completion. *IEEE Trans. Neural Netw. Learn. Syst.* **30**, 1659–1671 (2019)
13. Liu, J., Musialski, P., Wonka, P., Ye, J.: Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(1), 208–220 (2013)
14. Xu, Y., Hao, R., Yin, W., Su, Z.: Parallel matrix factorization for low-rank tensor completion. *Inverse Probl. Imag.* **9**(2), 601–624 (2017)
15. Sauve, A.C., Hero, A., Rogers, W.L., Wilderman, S., Clinthorne, N.: 3d image reconstruction for a compton spect camera model. *IEEE Trans. Nucl. Sci.* **46**(6), 2075–2084 (1999)
16. Xu, Y.-Y., Yin, W.-T., Wen, Z.-W., Zhang, Y.: An alternating direction algorithm for matrix completion with nonnegative factors. *Front. Math. China* **7**(2), 365–384 (2012)
17. Gandy, S., Recht, B., Yamada, I.: Tensor completion and low-n-rank tensor recovery via convex optimization, vol. 27 (2011)
18. Li, X.-T., Ye, Y.-M., Xu, X.-F.: Low-rank tensor completion with total variation for visual data inpainting. In: 31st AAAI conference on artificial intelligence (2017)
19. Li, X.-T., Zhao, X.-L., Jiang, T.-X., Zheng, Y.-B., Ji, T.-Y., Huang, T.-Z.: Low-rank tensor completion via combined non-local self-similarity and low-rank regularization. *Neurocomputing* **367**, 1–12 (2019)
20. Ji, T.-Y., Huang, T.-Z., Zhao, X.-L., Ma, T.H., Liu, G.: Tensor completion using total variation and low-rank matrix factorization. *Inform. Sci.* **326**(C), 243–257 (2016)
21. Jiang, T.-X., Huang, T.-Z., Zhao, X.-L., Ji, T.-Y., Deng, L.-J.: Matrix factorization for low-rank tensor completion using framelet prior. *Inform. Sci.* **436**, 403–417 (2018)
22. Cai, J.-F., Chan, R.H., Shen, Z.: A framelet-based image inpainting algorithm. *Appl. Comput. Harmon. Anal.* **24**(2), 131–149 (2008)
23. Zheng, Y.-B., Huang, T.-Z., Zhao, X.-L., Jiang, T.-X., Huang, J.: Hyperspectral image denoising via convex low-fibered-rank regularization, *IEEE Transactions on Geoscience and Remote Sensing*, to be published. <https://doi.org/10.1109/TGRS.2019.2940534>
24. Rubin, L.: Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena* **60**, 259–265 (1992)
25. Yang, J.-H., Zhao, X.-L., Ji, T.-Y., Ma, T.-H., Huang, T.-Z.: Low-rank tensor train for tensor robust principal component analysis. *Appl. Math. Comput.* **367**, 124783 (2020)
26. Chan, T.F., Vese, L.A.: Active contours without edges. *IEEE Trans. Image Process.* **10**(2), 266–277 (2001)
27. Chen, C., Li, X.-T., Ng, M.K., Yuan, X.-M.: Total variation based tensor decomposition for multi-dimensional data with time dimension. *Numer. Linear Algebra Appl.* **22**(6), 999–1019 (2015)
28. Li, F., Ng, M.K., Plemmons, R.J.: Coupled segmentation and denoising/deblurring models for hyperspectral material identification. *Numer. Linear Algebra Appl.* **19**(1), 153–173 (2012)
29. Zhao, X.-L., Wang, W., Zeng, T.-Y., Huang, T.-Z., Ng, M.K.: Total variation structured total least squares method for image restoration. *SIAM J. Sci. Comput.* **35**(6), B1304–B1320 (2013)
30. Yue, Z.-S., Meng, D.-Y., Sun, Y.-Q., Zhao, Q.: Hyperspectral image restoration under complex multi-band noises. *Remote Sens.* **10**(10), 1631 (2018)
31. Varghees, V.N., Manikandan, M.S., Gini, R.: Adaptive mri image denoising using total-variation and local noise estimation. In: *IEEE-international conference on advances in engineering, science and management (ICAESM-2012)*. IEEE, pp. 506–511 (2012)
32. Jiang, T.-X., Huang, T.-Z., Zhao, X.-L., Deng, L.-J., Wang, Y.: Fastderain: A novel video rain streak removal method using directional gradient priors. *IEEE Trans. Image Process.* **28**(4), 2089–2102 (2018)
33. Yang, J.-H., Zhao, X.-L., Ma, T.-H., Chen, Y., Huang, T.-Z., Ding, M.: Remote sensing images destriping using unidirectional hybrid total variation and nonconvex low-rank regularization. *J. Comput. Appl. Math.* **363**, 124–144 (2020)
34. Hu, Y., Jacob, M.: Higher degree total variation (hdtv) regularization for image recovery. *IEEE Trans. Image Process.* **21**(5), 2559–2571 (2012)
35. Zhu, J., Li, K., Hao, B.: Restoration of remote sensing images based on nonconvex constrained high-order total variation regularization. *J. Appl. Remote Sens.* **13**(2), 022006 (2019)
36. Adam, T., Paramesran, R.: Image denoising using combined higher order non-convex total variation with overlapping group sparsity. *Multidim. Syst. Sign. Process.* **30**(1), 503–527 (2019)

37. Li, F., Shen, C.-M., Fan, J.-S., Shen, C.-L.: Image restoration combining a total variational filter and a fourth-order filter. *J. Vis. Commun. Image Represent.* **18**(4), 322–330 (2007)
38. Lysaker, M., Tai, X.-C.: Iterative image restoration combining total variation minimization and a second-order functional. *Int. J. Comput. Vis.* **66**(1), 5–18 (2006)
39. Yang, J.-H., Zhao, X.-L., Mei, J.-J., Wang, S., Ma, T.-H., Huang, T.-Z.: Total variation and high-order total variation adaptive model for restoring blurred images with cauchy noise. *Comput. Math. Appl.* **77**(5), 1255–1272 (2019)
40. Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the kurdyka-?ojasiewicz inequality. *Math. Oper. Res.* **35**(2), 438–457 (2010)
41. Wang, S., Huang, T.-Z., Zhao, X.-L., Mei, J.-J., Huang, J.: Speckle noise removal in ultrasound images by first-and second-order total variation. *Numer. Algorithms* **78**(2), 513–533 (2018)
42. Li, S.-T., Dian, R., Fang, L., Bioucas-Dias, J.M.: Fusing hyperspectral and multispectral images via coupled sparse tensor factorization. *IEEE Trans. Image Process.* **27**(8), 4118–4130 (2018)
43. Zheng, Y.-B., Huang, T.-Z., Ji, T.-Y., Zhao, X.-L., Jiang, T.-X., Ma, T.-H.: Low-rank tensor completion via smooth matrix factorization. *Appl. Math. Model.* **70**, 677–695 (2019)
44. Attouch, H., Bolte, J., Svaiter, B.F.: 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), 91–129 (2013)
45. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., et al.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.