

Low-rank tensor completion via combined non-local self-similarity and low-rank regularization

Xiao-Tong Li^a, Xi-Le Zhao^{a,*}, Tai-Xiang Jiang^a, Yu-Bang Zheng^a, Teng-Yu Ji^b, Ting-Zhu Huang^a

^aSchool of Mathematical Sciences, Research Center for Image and Vision Computing, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, PR China

^bSchool of Science, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, PR China

ARTICLE INFO

Article history:

Received 7 April 2019

Revised 24 June 2019

Accepted 29 July 2019

Available online 6 August 2019

Communicated by Dr. Xin Luo

Keywords:

Low-rank tensor completion

Parallel low-rank matrix factorization

Non-local self-similarity

Plug and Play

Block successive upper-bound minimization

ABSTRACT

Global low-rank methods have achieved great successes in tensor completion. However, these methods neglected the abundant non-local self-similarities, which exist in a wide range of multi-dimensional imaging data. To integrate the global and non-local property of the underlying tensor, we propose a novel low-rank tensor completion model via combined non-local self-similarity and low-rank regularization, which is named as NLS-LR. We adopt the parallel low-rank matrix factorization to guarantee the global low-rankness while plugging in non-local based denoisers to promote the non-local self-similarity instead of tailoring regularizers. To tackle the proposed model, we develop an efficient block successive upper-bound minimization (BSUM) based algorithm. Numerical experiment results demonstrate that the proposed method outperforms many state-of-the-art tensor completion methods in terms of quality metrics and visual effects.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Nowadays information has been explosively increasing in our society, real-world data such as magnetic resonance image (MRI), hyperspectral/multispectral image (HSI/MSI), color image, and video usually have high dimensional structure. As an extension of vectors and matrices, tensors play a significant role in representing complex multidimensional data. Owing to information missing or unacceptable cost to acquire complete data, tensors in the real world may be incomplete. The problem of estimating the missing data from the observed incompleting tensor is called tensor completion. Higher-order tensor completion has a wide range of realistic applications, such as image inpainting [1,2], magnetic resonance imaging data recovery [3,4], rain streak removal [5,6], and hyperspectral image recovery [7,8].

To tackle the tensor completion problem, we need to exploit the latent relationship between the observed and the missing values. Actually, real-world data usually have a strong inherent correlation, which is described as low-rank property. There are a great many studies, which utilize the low-rank property to characterize the relationship between the observed and the missing

values [4,9–27], producing good performances on tensor completion problem. Mathematically, the low-rank tensor completion (LRTC) problem can be written as:

$$\begin{aligned} \min_{\mathcal{Y}} \quad & \text{rank}(\mathcal{Y}) \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{F}, \end{aligned} \quad (1.1)$$

where \mathcal{Y} is the underlying tensor, \mathcal{F} is the observed data, Ω is the index set corresponding to the observed entries, and $\mathcal{P}_{\Omega}(\cdot)$ is the projection function that keeps the entries of \mathcal{Y} in Ω while making others be zeros. Particularly, the low-rank matrix completion (LRMC) problem can be viewed as a second-order tensor completion problem [28].

Different from the matrix, there is not a unique definition for tensor rank. Among those definitions, there are two popular ways to formulate tensor rank: the CANDECOMP/PARAFAC (CP) rank and the Tucker rank (n -rank) [29]. Given a N -way tensor $\mathcal{Y} \in \mathbb{R}^{d_1 \times \dots \times d_N}$, the CP-rank of \mathcal{Y} is defined as the smallest number of rank-one tensors that generate \mathcal{Y} as their sum. There is another more common definition called Tucker rank (n -rank). The Tucker rank of \mathcal{Y} is defined as $(\text{rank}(Y_{(1)}), \text{rank}(Y_{(2)}), \dots, \text{rank}(Y_{(N)}))$, where $Y_{(N)}$ is the mode- n unfolding of tensor \mathcal{Y} (see details in Section 2.2).

However, directly minimizing the CP-rank or Tucker rank is NP-hard [30]. In the past decade, the nuclear norm is found to be the tightest convex surrogate approximation of a matrix's rank and has

* Corresponding author.

E-mail address: xlzhao122003@163.com (X.-L. Zhao).

been widely used to tackle rank minimization problem [31,32]. Utilizing the nuclear norm, a new model is first introduced by Liu et al. in 2009 [33], which considers the low-rankness to all mode of the tensor, and the low-rank tensor completion model can be rewritten in this form:

$$\begin{aligned} \min_{\mathcal{Y}} \quad & \sum_{n=1}^N \alpha_n \|Y_{(n)}\|_* \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{F}, \end{aligned} \quad (1.2)$$

where $\alpha_n \geq 0$ ($n = 1, 2, \dots, N$), $\sum_{n=1}^N \alpha_n = 1$, and $Y_{(n)}$ is the mode- n unfolding of \mathcal{Y} . The optimization problem (1.2) can be solved by high accuracy low-rank tensor completion (HaLRTC) [33] and the Douglas–Rachford splitting method [34]. Optimizing the problem (1.2) involves calculating the singular value decomposition (SVD) of each $Y_{(n)}$, which is computationally expensive at each iteration. To tackle this problem, Xu et al. proposed a new model which is called low-rank tensor completion by parallel matrix factorization (TMac) [35], i.e.,

$$\begin{aligned} \min_{Y, X, A} \quad & \sum_{n=1}^N \alpha_n \|Y_{(n)} - A_n X_n\|_F^2 \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{F}, \end{aligned} \quad (1.3)$$

where $A = (A_1, A_2, \dots, A_N)$ and $X = (X_1, X_2, \dots, X_N)$ represent the low-rank factor matrices, respectively. TMac applies low-rank matrix factorization to each mode unfolding matrices and updates the factorization matrices alternatively, which costs less time and gains better performance than HaLRTC.

Low-rank property can catch the global information of the underlying tensor. However, it is not sufficient enough to exploit the structure of the tensor. Fortunately, real-world data often exhibit smooth prior in the spatial domain. Inspired by this point, many recent studies investigate smoothness constraints for their work [3,4,36–43]. Among those smoothness constraints, the total variation (TV) regularizer is widely used and has shown reasonable performances on preserving edges in image restoration. Ji et al. introduced the TV regularization term into their model [37] and the model can be written as:

$$\begin{aligned} \min_{Y, X, A} \quad & \sum_{n=1}^N \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2 + \lambda \text{TV}(X_3) \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{F}, \end{aligned} \quad (1.4)$$

where $\text{TV}(X_3)$ is the TV regularizer of factor matrix X_3 using piecewise smooth prior and λ denotes the parameter which controls the power of the TV regularizer. Due to the smoothness of the spatial domain, the model gains a great improvement compared to the based model TMac. Instead of introducing regularizer on factor matrices, Li et al. adopted TV regularizer on each mode unfolding matrices of tensor \mathcal{Y} , their model [42] is written as:

$$\begin{aligned} \min_{\mathcal{Y}, \mathcal{G}, \{U^{(n)}\}_{n=1}^N} \quad & \lambda_1 \sum_{n=1}^N \|F_n Y_{(n)}\|_1 + \frac{1}{N} \sum_{n=1}^N \|U^{(n)}\|_* + \lambda_2 \|\mathcal{G}\|_F^2 \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{F}, \mathcal{Y} = \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \dots \times_N U^{(N)}, \end{aligned} \quad (1.5)$$

where \mathcal{G} denotes the core tensor and $\{U^{(n)}\}_{n=1}^N$ are the Tucker decomposition factors. F_n is the TV matrix where $F_n(i, i) = 1$, $F_n(i, i+1) = -1$, and other elements are zeros. Not limited to these methods, the other related LRTC methods with their properties are shown in Table 1.

1.1. Motivations and contributions

Although the local smoothness methods have achieved promising performances, they neglect the redundant non-local self-similarities, which can be observed in most multi-dimensional

imaging data. Non-local methods can utilize not only the neighborhood pixels but also the far away pixels in similar patches. Meanwhile, the non-local methods surpass the TV based methods when dealing with many imaging inverse problems [46–48].

Therefore, we propose a low-rank tensor completion model using non-local prior to enhance the self-similarities of the underlying tensor, which would be helpful for preserving the abundant details. Our tensor completion model is formulated as:

$$\begin{aligned} \min_{Y, X, A} \quad & \sum_{n=1}^3 \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2 + \lambda \Phi(\mathcal{Y}) \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{F}, \end{aligned} \quad (1.6)$$

where $\sum_{n=1}^3 \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2$ is to guarantee the global low-dimensionality along each mode and λ is a regularization parameter. Instead of investing efforts in tailoring non-local regularizers, we develop an implicit regularizer $\Phi(\mathcal{Y})$ using Plug and Play framework (see details in Section 2.3). The implicit regularizer $\Phi(\mathcal{Y})$ is introduced by plugging in the non-local denoiser engine, which is convinced to express the non-local self-similarities of the underlying tensor. By integrating both the global low-rankness and non-local self-similarities of the underlying tensor, the proposed model can effectively maintain the general structure as well as capture the details of the underlying tensor.

1.2. Organization of this paper

The structure of this paper is as follows: Section 2 introduces some basic tensor knowledge, operators, and details about Plug and Play. Section 3 gives the formulation of the proposed model as well as the solver algorithm. Section 4 evaluates the performances of NLS-LR and the compared methods in numerical experiments. Section 5 gives some discussions. Section 6 summarizes this paper.

2. Notations and preliminaries

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. We will introduce some preliminary knowledge in the following subsection.

2.1. Tensor basics

We define a N -way tensor as $\mathcal{X} \in \mathbb{R}^{d_1 \times \dots \times d_N}$, whose (i_1, i_2, \dots, i_N) th component is denoted as x_{i_1, i_2, \dots, i_N} . A fiber of a tensor is defined by fixing every index but one. The mode- n fibers are vectors $\mathcal{X}(i_1, \dots, i_{n-1}, :, i_{n+1}, \dots, i_N)$ respectively for all i_1, i_2, \dots, i_N .

The mode- n unfolding of a tensor \mathcal{X} is denoted as $X_{(n)} \in \mathbb{R}^{d_n \times \prod_{d \neq n} d_d}$, which is a matrix with columns being the mode- n fibers of \mathcal{X} . The element (i_1, i_2, \dots, i_N) of the tensor \mathcal{X} maps to the (i_n, j) th element of the matrix $X_{(n)}$ in the lexicographical order, where

$$j = 1 + \sum_{k=1, k \neq n}^N (i_k - 1) J_k \quad \text{with} \quad J_k = \prod_{m=1, m \neq n}^{k-1} d_m, \quad (2.1)$$

the inverse operator of unfolding is denoted as “fold”, i.e., $\mathcal{X} = \text{fold}_n(X_{(n)})$.

The Tucker rank (n -rank) of \mathcal{X} is defined as the following array.

$$\text{rank}(\mathcal{X}) = (\text{rank}(X_{(1)}), \text{rank}(X_{(2)}), \dots, \text{rank}(X_{(N)})). \quad (2.2)$$

The inner product of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_N}$ is defined as

$$\langle \mathcal{X}, \mathcal{Y} \rangle := \sum_{i_1, i_2, \dots, i_N} x_{i_1, i_2, \dots, i_N} y_{i_1, i_2, \dots, i_N}. \quad (2.3)$$

Table 1

An introduction of the related LRTC methods with their properties.

	Low-rankness Low-rankness or minimizing the rank	Spatial smoothness Constrains on factors/tensors	The solving algorithm
HaLRTC [33]	The Tucker rank	–	An alternating direction method of multipliers based algorithm
LRTC-TV-I [42]	The Tucker rank	Isotropic TV for underlying tensors	An alternating direction method of multipliers based algorithm
LRTC-TV-II [42]	The Tucker decomposition	Isotropic TV for underlying tensors	An alternating direction method of multipliers based algorithm
TMac [35]	Low-rank matrix factorization	–	An alternating direction method of multipliers based algorithm
MF-TV [37]	Low-rank matrix factorization	Isotropic TV for factors	A block successive upper-bound minimization based algorithm
MF-Framelet [4]	Low-rank matrix factorization	Framelet for factors	A block successive upper-bound minimization based algorithm
SMF-LRTC [44]	Low-rank matrix factorization	Unidirectional TV for underlying tensors and framelet for factors	A block successive upper-bound minimization based algorithm
SPC-TV/QV [36]	The PARAFAC/polyadic decomposition rank	Unidirectional TV/QV for underlying tensors	A hierarchical alternating least squares based algorithm
TNN [45]	The tubal rank	–	An alternating direction method of multipliers based algorithm

The Frobenius norm is defined as:

$$\|\mathcal{X}\|_F := \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}. \quad (2.4)$$

The tensor inner product follows the exchange law and combination law, which can be proven as follows.

$$\begin{aligned} \langle \mathcal{X}, \mathcal{Y} \rangle &= \sum_{i_1, i_2, \dots, i_N} x_{i_1, i_2, \dots, i_N} y_{i_1, i_2, \dots, i_N} \\ &= \sum_{i_1, i_2, \dots, i_N} y_{i_1, i_2, \dots, i_N} x_{i_1, i_2, \dots, i_N} \\ &= \langle \mathcal{Y}, \mathcal{X} \rangle, \end{aligned} \quad (2.5)$$

$$\begin{aligned} \langle \mathcal{X}, \mathcal{Y} \rangle + \langle \mathcal{X}, \mathcal{Z} \rangle &= \sum_{i_1, i_2, \dots, i_N} x_{i_1, i_2, \dots, i_N} y_{i_1, i_2, \dots, i_N} + \sum_{i_1, i_2, \dots, i_N} x_{i_1, i_2, \dots, i_N} z_{i_1, i_2, \dots, i_N} \\ &= \sum_{i_1, i_2, \dots, i_N} x_{i_1, i_2, \dots, i_N} (y_{i_1, i_2, \dots, i_N} + z_{i_1, i_2, \dots, i_N}) \\ &= \langle \mathcal{X}, \mathcal{Y} + \mathcal{Z} \rangle, \end{aligned} \quad (2.6)$$

the exchange law and combination law will be used in the derivation of (3.8).

2.2. Projected and proximal operators

The projected operator $\mathcal{P}_\Omega(\mathcal{Y})$ is a function that keeps the entries of \mathcal{Y} in Ω while making others be zeros. i.e.,

$$(\mathcal{P}_\Omega(\mathcal{Y}))_{i_1, i_2, \dots, i_N} := \begin{cases} y_{i_1, i_2, \dots, i_N}, & (i_1, i_2, \dots, i_N) \in \Omega, \\ 0, & \text{otherwise.} \end{cases} \quad (2.7)$$

The proximal operator of a convex function $f(x)$ is defined as:

$$\text{prox}_f(y) = \arg \min_x \left\{ f(x) + \frac{\rho}{2} \|x - y\|^2 \right\}, \quad (2.8)$$

2.3. Plug and Play (PnP)

In the field of image reconstruction, numerous efforts have been made on matching effective regularizers with advanced optimization algorithms [49–52]. Factly, natural image priors such as spatial sparsity, piecewise smooth are widely used, while their corresponding regularizers $\|x\|_1$ and $\|x\|_{TV}$ are not differentiable. To tackle the non-differentiability of many regularizers, some proximal algorithms came to being in the past two decades, such as the variants of iterative shrinkage/thresholding algorithm (ISTA) [49] and the alternating direction method of multipliers algorithm

(ADMM) [50]. These algorithms make itself equivalent to solve the regularized image denoising problem.

Recently, the Plug and Play framework has become a hot topic, which is first proposed by Venkatakrishnan et al. [53]. Extensive experiments have demonstrated its effectiveness [54–58]. The PnP framework can allow state-of-the-art denoisers into some inverse problems, such as image deblurring [55] and super-resolution [54]. We usually have the following sub-problems in the iterative algorithm for solving inverse problems in image processing:

$$x^{(k+1)} = \arg \min_x \frac{\rho}{2} \|x - z\|_2^2 + \lambda \Phi(x), \quad (2.9)$$

where $\Phi(x)$ is the regularizer and λ is a regularization parameter that trades off the fidelity term and the regularizer. If we define $\sigma = \sqrt{\frac{\lambda}{\rho}}$, the problem can be rewritten as:

$$x^{(k+1)} = \arg \min_x \frac{1}{2\sigma^2} \|x - z\|_2^2 + \Phi(x). \quad (2.10)$$

Treating z as the noise image, the above problem can be regarded as a denoising problem. Given the regularizer $\Phi(x)$, we have a corresponding denoiser to tackle with the denoising problem. For example, if the regularizer $\Phi(x)$ is the TV regularizer, we can use the corresponding TV-based denoisers to solve the denoising problem. In PnP framework, $\Phi(x)$ is an implicit regularizer by plugging in off-the-shelf denoisers to express the prior we want, which is the main idea of the PnP framework. There are many state-of-the-art denoisers available to solve the problem, such as BM3D [46], BM4D [59], TNRD [60], and NLM [48]. Therefore, the problem can be solved as:

$$x^{(k+1)} = \mathcal{D}(z, \sigma), \quad (2.11)$$

where \mathcal{D} is the denoiser engine and σ is denoted as the denoiser parameter. This provides the basis for us to apply non-local engines for the LRTC problem.

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 objective function of the proposed model (1.6) is:

$$f(X, A, \mathcal{Y}) = \sum_{n=1}^3 \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2 + \lambda \Phi(\mathcal{Y}) + \iota(\mathcal{Y}), \quad (3.1)$$

where α_n are all positive weights $\sum_{n=1}^3 \alpha_n = 1$. $A = (A_1, A_2, A_3)$ and $X = (X_1, X_2, X_3)$ represent the low-rank factor matrices along each

mode, $Y_{(n)}$ denotes the mode- n unfolding of tensor \mathcal{Y} , λ is a regularization parameter, $\Phi(\mathcal{Y})$ is a non-local denoiser regularizer, and $\iota(\mathcal{Y})$ is the indicator function, i.e.,

$$\iota(\mathcal{Y}) := \begin{cases} 0, & \text{if } \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F}, \\ \infty, & \text{otherwise.} \end{cases} \quad (3.2)$$

The proposed model has two parts, one is the low-rank regularizer term $\sum_{n=1}^3 \frac{\alpha_n}{2} \|Y_{(n)} - A_n X_n\|_F^2$. We assume the Tucker rank of tensor $\mathcal{Y} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ is (r_1, r_2, r_3) , which is given as a low-rank prior. $A_n \in \mathbb{R}^{d_n \times r_n}$, $X_n \in \mathbb{R}^{r_n \times s_n}$ are the low-rank factor matrices. This term is modeled to enhance the low-rank property in each mode, thus can better capture the global information of the tensor \mathcal{Y} .

Another part is the regularizer $\Phi(\mathcal{Y})$, which is used to promote the self-similar property. Instead of tailoring non-local regularizers, the implicit regularizer $\Phi(\mathcal{Y})$ is introduced by plugging in the non-local denoiser engine, which is convinced to express the non-local self-similarities of the underlying tensor.

3.2. Proposed algorithm

In this section, we develop a BSUM-based algorithm to solve the proposed model.

Let $\mathcal{Z} = (X, A, \mathcal{Y})$, $\mathcal{Z}^k = (X^k, A^k, \mathcal{Y}^k)$, and $h(\mathcal{Z}, \mathcal{Z}^k) = f(\mathcal{Z}) + \frac{\rho}{2} \|\mathcal{Z} - \mathcal{Z}^k\|_F^2$. By introducing the proximal operator, we can solve the optimization problem (3.1) as follows:

$$\mathcal{Z}^{k+1} = \arg \min_{\mathcal{Z}} h(\mathcal{Z}, \mathcal{Z}^k) = \arg \min_{\mathcal{Z}} f(\mathcal{Z}) + \frac{\rho}{2} \|\mathcal{Z} - \mathcal{Z}^k\|_F^2, \quad (3.3)$$

where $\mathcal{Z} = (X, A, \mathcal{Y})$, $\mathcal{Z}^k = (X^k, A^k, \mathcal{Y}^k)$, and ρ is the proximal parameter. With utilization of the BSUM algorithm [61], the optimization variables of the objective function can be decomposed into multi-blocks. Then Eq. (3.3) can be iteratively solved by:

$$\begin{cases} X^{k+1} = \arg \min_X \left\{ h_1(X, \mathcal{Z}_1^k) = f(X, A^k, \mathcal{Y}^k) + \frac{\rho}{2} \|X - X^k\|_F^2 \right\}, \\ A^{k+1} = \arg \min_A \left\{ h_2(A, \mathcal{Z}_2^k) = f(X^{k+1}, A, \mathcal{Y}^k) + \frac{\rho}{2} \|A - A^k\|_F^2 \right\}, \\ \mathcal{Y}^{k+1} = \arg \min_{\mathcal{Y}} \left\{ h_3(\mathcal{Y}, \mathcal{Z}_3^k) = f(X^{k+1}, A^{k+1}, \mathcal{Y}) + \frac{\rho}{2} \|\mathcal{Y} - \mathcal{Y}^k\|_F^2 \right\}. \end{cases} \quad (3.4)$$

The X_n -subproblem and A_n -subproblem can be written as:

$$\begin{cases} X_n^{k+1} = \arg \min_{X_n} \left\{ \frac{1}{2} \|Y_{(n)}^k - A_n^{k+1} X_n\|_F^2 + \frac{\rho}{2} \|X_n - X_n^k\|_F^2 \right\}, \\ A_n^{k+1} = \arg \min_{A_n} \left\{ \frac{1}{2} \|Y_{(n)}^k - A_n X_n^{k+1}\|_F^2 + \frac{\rho}{2} \|A_n - A_n^k\|_F^2 \right\}. \end{cases} \quad (3.5)$$

Note that the minimization problem of the X -subproblem and A -subproblem can be solved easily due to its differentiability. Observing that the subproblems except \mathcal{Y} -subproblem have closed-form solutions, the core problem turns out to be how to solve the \mathcal{Y} -subproblem, which can be written as:

$$\mathcal{Y}^{k+1} = \arg \min_{\mathcal{Y}} \sum_{n=1}^3 \frac{\alpha_n}{2} \|Y_{(n)} - A_n^k X_n^k\|_F^2 + \frac{\rho}{2} \|\mathcal{Y} - \mathcal{Y}^k\|_F^2 + \lambda \Phi(\mathcal{Y}) + \iota(\mathcal{Y}). \quad (3.6)$$

The matrix Frobenius norm $\|Y_{(n)} - A_n^k X_n^k\|_F^2$ is to calculate the square root of the sum of the squares of all elements. If we define \mathcal{M}_n^k as the fold- n tensor of $A_n^k X_n^k$, it's not hard to find that the matrix Frobenius norm $\|Y_{(n)} - A_n^k X_n^k\|_F^2$ is equal to tensor Frobenius norm $\|\mathcal{Y} - \mathcal{M}_n^k\|_F^2$, which is also to calculate the square root

of the sum of the squares of all elements. Thus we can obtain the \mathcal{Y} -subproblem as follows:

$$\mathcal{Y}^{k+1} = \arg \min_{\mathcal{Y}} \sum_{n=1}^3 \frac{\alpha_n}{2} \|\mathcal{Y} - \mathcal{M}_n^k\|_F^2 + \frac{\rho}{2} \|\mathcal{Y} - \mathcal{Y}^k\|_F^2 + \lambda \Phi(\mathcal{Y}) + \iota(\mathcal{Y}), \quad (3.7)$$

further note that the tensor Frobenius norm can be also written as the tensor inner product form (2.1), we can merge the optimization into a more standard form as follows:

$$\begin{aligned} \mathcal{Y}^{k+1} &= \arg \min_{\mathcal{Y}} \sum_{n=1}^3 \frac{\alpha_n}{2} \|\mathcal{Y} - \mathcal{M}_n^k\|_F^2 + \frac{\rho}{2} \|\mathcal{Y} - \mathcal{Y}^k\|_F^2 + \lambda \Phi(\mathcal{Y}) \\ &\quad + \iota(\mathcal{Y}) \\ &= \arg \min_{\mathcal{Y}} \frac{(1+\rho)}{2} \langle \mathcal{Y}, \mathcal{Y} \rangle - \left\langle \mathcal{Y}, \sum_{n=1}^3 \alpha_n \mathcal{M}_n^k + \rho \mathcal{Y}^k \right\rangle \\ &\quad + \lambda \Phi(\mathcal{Y}) + \iota(\mathcal{Y}) \\ &= \arg \min_{\mathcal{Y}} \frac{(1+\rho)}{2} \left\langle \mathcal{Y} - \frac{\sum_{n=1}^3 \alpha_n \mathcal{M}_n^k + \rho \mathcal{Y}^k}{(1+\rho)}, \mathcal{Y} \right\rangle \\ &\quad - \frac{\sum_{n=1}^3 \alpha_n \mathcal{M}_n^k + \rho \mathcal{Y}^k}{(1+\rho)} \Bigg\rangle + \lambda \Phi(\mathcal{Y}) + \iota(\mathcal{Y}) \\ &= \arg \min_{\mathcal{Y}} \frac{(1+\rho)}{2\lambda} \left\| \mathcal{Y} - \frac{\sum_{n=1}^3 \alpha_n \mathcal{M}_n^k + \rho \mathcal{Y}^k}{(1+\rho)} \right\|_F^2 + \Phi(\mathcal{Y}) \\ &\quad + \iota(\mathcal{Y}), \end{aligned} \quad (3.8)$$

where $\mathcal{M}_n^k = \text{fold}_n(A_n^k X_n^k)$ is denoted as the fold- n tensor of $A_n^k X_n^k$. The exchange law and combination law of tensor inner product used in the derivation are proven previously in Section 2.1. Let $\tilde{\mathcal{Y}}^{(k)} = \frac{\sum_{n=1}^3 \alpha_n \mathcal{M}_n^k + \rho \mathcal{Y}^k}{(1+\rho)}$ and $\sigma = \sqrt{\frac{\lambda}{1+\rho}}$, the \mathcal{Y} -subproblem will be rewritten as:

$$\min_{\mathcal{Y}} \frac{1}{2\sigma^2} \|\mathcal{Y} - \tilde{\mathcal{Y}}^{(k)}\|_F^2 + \Phi(\mathcal{Y}) \quad (3.9)$$

$$\text{s.t. } \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{F},$$

then the question is formalized as a standard constrained optimization. Factly, $\Phi(\mathcal{Y})$ is the denoiser regularizer, which can be solved in Plug and Play method. By using the project operator to meet the constrain, the \mathcal{Y} -subquestion can be solved as:

$$\mathcal{Y}^{k+1} = \mathcal{P}_{\Omega^c}(\mathcal{D}(\tilde{\mathcal{Y}}^{(k)}, \sigma)) + \mathcal{F}, \quad (3.10)$$

where \mathcal{P}_{Ω^c} is the project function to keep constraint condition, \mathcal{D} is the denoiser engine, and σ is denoted as the denoiser parameter to control the strength of denoising. Note that the denoiser parameter σ is linked to the noise level in i.i.d. Gaussian denoising, but in our model the σ is linked to the general system error between $\tilde{\mathcal{Y}}^{(k)}$ and the ground truth. Thus, in our model σ is treated as a tunable parameter to obtain an appropriate effect. Finally, the proposed algorithm is summarized in Algorithm 1.

3.3. Rank-increasing scheme

In this subsection, we will talk about the rank-increasing scheme [35,44,62]. In the proposed model. This strategy starts with rank estimation $r^0 = (r_1^0, r_2^0, r_3^0)$, when the relative error becomes less than the tolerance we set, i.e.,

$$\left| 1 - \frac{\|P_{\Omega^c}(A_n^{k+1} X_n^{k+1})\|_F}{\|P_{\Omega^c}(A_n^k X_n^k)\|_F} \right| < \gamma, \quad n = 1, 2, 3 \quad (3.11)$$

then increase the corresponding rank r_n to $\min\{r_n + \Delta r_n, r_n^{\max}\}$ at iteration $k+1$, where Δr_n is a positive integer and r_n^{\max} is the



Fig. 1. The color images used in the experiments.

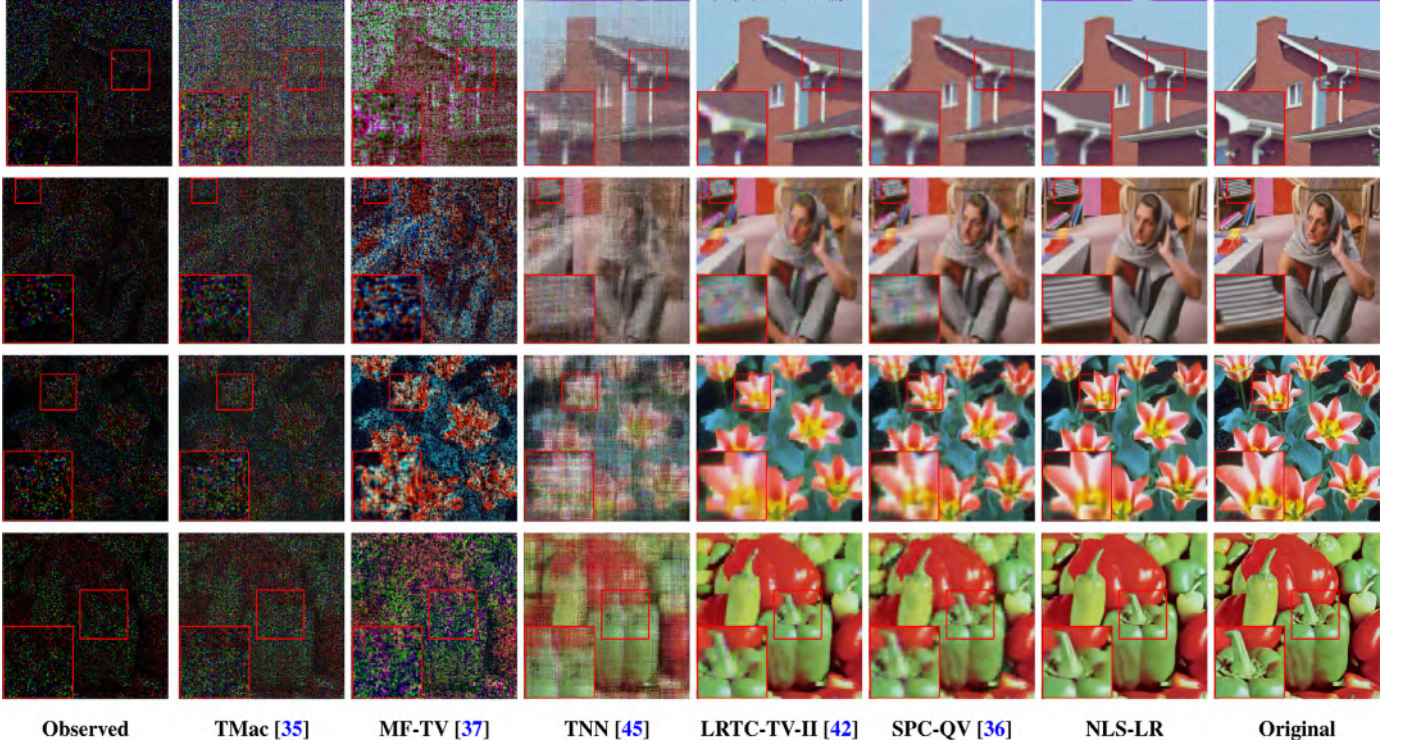


Fig. 2. Reconstructed results for color image data with SR=10%. From top to bottom : house, barbara, tulips, and pepper. From left to right: the observed data, the recovered results by TMac, MF-TV, TNN, SPC-QV, LRTC-TV-II, NLS-LR, and the original data.

Algorithm 1 BSUM based optimization algorithm for NLS-LR.

Input: the observed tensor \mathcal{F} , the set index Ω of the observed entries, the denoiser parameter σ , the proximal parameter ρ , the initial Tucker rank $r^0 = (r_1^0, r_2^0, r_3^0)$, $\Delta r = (\Delta r_1, \Delta r_2, \Delta r_3)$, and $r^{\max} = (r_1^{\max}, r_2^{\max}, r_3^{\max})$.

Output: the reconstructed tensor \mathcal{Y} .

1. Initialization: $A_n^0 = \text{rand}(d_n \times r_n)$, $X_n^0 = \text{rand}(r_n^0 \times \prod_{m \neq n} d_m)$, $\mathcal{Y} = \mathcal{P}_\Omega(\mathcal{F})$, and $k = 1$.
2. **while** not converged **do**:
3. $X_n^{k+1} \leftarrow ((A_n^k)^T A_n^k + \rho I_1)^\dagger ((A_n^k)^T Y_{(n)}^k + \rho X_n^k)$, $n = 1, 2, 3$.
4. $A_n^{k+1} \leftarrow (Y_{(n)}^k (X_n^{k+1})^T + \rho A_n^k) (X_n^{k+1} (X_n^{k+1})^T + \rho I_2)^\dagger$, $n = 1, 2, 3$.
5. $\mathcal{Y}^{k+1} \leftarrow \mathcal{P}_{\Omega^c} \left(\mathcal{D} \left(\frac{\sum_{n=1}^3 \alpha_n \text{fold}_n(A_n^k X_n^k + \rho Y_{(n)}^k)}{(1+\rho)}, \sigma \right) \right) + \mathcal{F}$.
4. $k \leftarrow k + 1$.
5. **end while**
6. **return** \mathcal{Y} .

maximum Tucker rank. More specifically, when the r_n increased at iteration $k + 1$, the A_n^{k+1} will be updated to $[A_n^k, \text{rand}(d_n, \Delta r_n)]$ and X_n^{k+1} will be updated to $[X_n^k, \text{rand}(\Delta r_n, s_n)]$ with $s_n = \prod_{m \neq n} d_m$, i.e., adding randomly generated columns Δr_n to A_n^k and randomly generated rows Δr_n to X_n^k . In the beginning, we can efficiently obtain

the structure of the underlying tensor by adopting low-rank estimation, while we can get more details recovered with the rank increasing. In this paper, we set the tolerance $\gamma = 10^{-3}$ for color image data and $\gamma = 10^{-2}$ for video and MSI data.

4. Experiments results

In this section, we evaluate the performance of the proposed model on three kinds of tensors: color image, video, and MSI. To validate the effectiveness of the proposed method, we adopt five compared methods: TMac [35], MF-TV [37], TNN [45], LRTC-TV-II [42], and SPC-QV [36]. The introduction of those models are provided in Table 1.

The peak signal to noise rate (PSNR) and structural similarity index (SSIM) [63] are adopted to measure the quality of each method. The stopping criterion of all the methods above relies on the relative change (RelCha) of the two successive reconstructed tensors, i.e., $\text{RelCha} = \frac{\|\mathcal{Y}^{k+1} - \mathcal{Y}^k\|_F}{\|\mathcal{Y}^k\|_F} < \varepsilon$, where ε is a tolerance. In our

experiments, the parameters are set as following: the parameter of proximal operator $\rho = 0.1$, the weights $\alpha_n = 1/3$ ($n = 1, 2, 3$), the tolerance $\varepsilon = 3 \times 10^{-4}$, $\Delta r = (5, 5, 5)$, and the denoiser parameter σ is selected from the set $\{5, 10, 15, 20\}$. By adopting the Plug and Play framework, we can flexibly choose the denoiser \mathcal{D} with the corresponding data. In all experiments, TMac [35] and MF-TV [37] are implemented using the default setting in [37]. TNN [45],

Table 2

The PSNR, SSIM, and average CPU time (seconds) of the results reconstructed by TMac [35], MF-TV [37], TNN [45], LRTC-TV-II [42], SPC-QV [36], and NLS-LR for color image completion on different sampling rates and images.

Color image	SR	5%		10%		20%		Time
		Method	PSNR	SSIM	PSNR	SSIM	PSNR	
House	TMac	5.624	0.013	6.932	0.022	11.378	0.145	10.3437
	MF-TV	6.344	0.022	9.243	0.050	18.959	0.364	647.459
	TNN	17.887	0.256	20.683	0.387	24.820	0.607	3.694
	LRTC-TV-II	23.723	0.716	25.921	0.778	28.476	0.840	42.441
	SPC-QV	23.983	0.695	25.686	0.752	27.248	0.808	24.086
	NLS-LR	26.875	0.784	30.353	0.826	33.189	0.866	148.385
Barbara	TMac	5.756	0.015	7.536	0.027	9.141	0.059	8.463
	MF-TV	8.042	0.046	9.427	0.087	12.916	0.226	619.314
	TNN	16.474	0.226	19.041	0.362	22.621	0.573	3.362
	LRTC-TV-II	22.785	0.649	24.328	0.724	26.366	0.803	43.498
	SPC-QV	22.783	0.634	24.908	0.729	27.213	0.816	28.323
	NLS-LR	24.776	0.743	27.736	0.821	30.791	0.891	136.138
Lena	TMac	5.756	0.015	6.517	0.023	8.516	0.057	9.672
	MF-TV	6.456	0.030	7.819	0.044	12.276	0.276	611.858
	TNN	16.968	0.220	19.642	0.350	23.251	0.557	3.441
	LRTC-TV-II	23.558	0.701	26.044	0.790	28.887	0.873	43.783
	SPC-QV	23.644	0.682	25.476	0.753	27.603	0.827	26.910
	NLS-LR	25.527	0.767	27.991	0.835	30.977	0.899	138.996
Tulips	TMac	6.232	0.017	6.679	0.027	7.751	0.058	10.021
	MF-TV	7.298	0.041	9.735	0.124	14.643	0.320	618.446
	TNN	12.656	0.118	14.429	0.205	17.627	0.388	3.248
	LRTC-TV-II	18.634	0.522	20.521	0.630	23.339	0.765	42.355
	SPC-QV	19.137	0.533	21.892	0.677	24.629	0.792	41.269
	NLS-LR	20.720	0.646	24.123	0.780	27.517	0.869	134.893
Sails	TMac	6.643	0.013	7.115	0.022	8.260	0.045	9.802
	MF-TV	7.440	0.046	8.663	0.087	11.512	0.211	661.248
	TNN	16.839	0.196	18.400	0.296	20.909	0.485	3.343
	LRTC-TV-II	21.156	0.468	22.105	0.528	23.973	0.662	42.012
	SPC-QV	21.221	0.463	22.958	0.586	25.203	0.733	34.994
	NLS-LR	21.907	0.499	24.120	0.643	26.890	0.785	135.590
Sailboat	TMac	5.611	0.018	6.146	0.031	7.423	0.066	10.006
	MF-TVc	6.626	0.037	8.602	0.094	12.683	0.266	597.596
	TNN	15.925	0.238	17.827	0.340	20.705	0.517	3.321
	LRTC-TV-II	20.142	0.624	21.065	0.676	23.116	0.771	42.388
	SPC-QV	20.627	0.593	22.502	0.689	24.843	0.787	32.774
	NLS-LR	21.017	0.662	23.241	0.762	25.756	0.846	132.719
Airplane	TMac	2.879	0.008	4.147	0.014	9.193	0.051	9.573
	MF-TV	3.561	0.015	5.606	0.027	13.288	0.219	562.817
	TNN	16.946	0.286	18.883	0.399	22.046	0.587	3.715
	LRTC-TV-II	20.885	0.691	22.561	0.763	26.265	0.878	41.958
	SPC-QV	21.294	0.659	23.183	0.735	25.054	0.802	24.444
	NLS-LR	22.713	0.756	25.373	0.833	28.423	0.896	53.327
Pepper	TMac	6.522	0.020	7.121	0.029	8.626	0.059	10.184
	MF-TV	7.118	0.037	8.355	0.056	13.669	0.250	670.287
	TNN	14.305	0.140	16.940	0.245	20.394	0.447	3.294
	LRTC-TV-II	22.445	0.710	24.497	0.793	26.914	0.867	42.451
	SPC-QV	21.661	0.646	23.861	0.736	26.455	0.818	32.527
	NLS-LR	24.079	0.766	26.864	0.839	29.775	0.900	120.097

LRTC-TV-II [42], and SPC-QV [36] are implemented using the parameters specified optimally by following the corresponding papers and models. All experiments were performed on the platform of Windows 10 and MATLAB (R2018a) with an Intel Core i7-8700K 3.70 GHz and 32GB RAM.

4.1. Color image

In this subsection, we test the proposed model on eight color images, named *barbara*, *lena*, *house*, *sailboat*, *tulips*, *sails*, *airplane*, and *pepper*. The color images are shown in Fig. 1 and all images are of size $256 \times 256 \times 3$. The incompleting tensors are generated by randomly sampling elements and the sampling rates (SRs) are set to be 5%, 10%, and 20%, respectively. The initial Tucker rank $r^0 = (10, 10, 3)$ and the maximum Tucker rank $r^{\max} = (125, 125, 3)$. For color image data, we adopt an off-the-shell CBM3D to be the denoiser and set the denoiser parameter to be $\sigma = 10$.

Table 2 presents the PSNR, SSIM, and average CPU time (seconds) of the reconstructed results obtained by NLS-LR and compared methods. Experimental results prove that NLS-LR consistently gains the best performances in terms of PSNR and SSIM. To further compare the results, we select four images *house*, *barbara*, *tulips*, and *pepper* to be displayed in Fig. 2. As observed, NLS-LR produces the best visual effect, especially for the images which have abundant details, such as *barbara* and *house*. Compared with TMac, NLS-LR gains a huge improvement, which reveals the effectiveness of our regularizer. TMac and MF-TV only consider the global low-rankness and neglect the relationship of the third dimension, so they are hard to reconstruct the color images. SPC-QV and LRTC-TV-II with the smoothness prior get commendable performance. However, the methods with smoothness priors usually have a staircase effect, leading to indistinct details. This is where the non-local method matters, we see that our non-local regularizer gets better performances on preserving the abundant details.

Table 3

The PSNR,SSIM, and average CPU time (seconds) of the results reconstructed by TMac [35], MF-TV [37], TNN [45], LRTC-TV-II [42], SPC-QV [36], and NLS-LR for text masked image, gridlines damaged image, and scratched image.

Type	Text masked image			Gridlines damaged image			Scratched image		
	Method	PSNR	SSIM	Time	PSNR	SSIM	Time	PSNR	SSIM
TMac	13.042	0.522	27.313	13.725	0.611	25.462	11.884	0.538	30.598
MF-TV	20.359	0.717	660.750	15.947	0.660	604.815	12.597	0.575	630.061
TNN	25.753	0.820	4.420	26.259	0.831	4.389	23.727	0.777	4.602
LRTC-TV-II	28.933	0.918	40.315	29.453	0.925	40.643	26.512	0.849	40.791
SPC-QV	28.095	0.889	18.721	27.208	0.866	14.629	25.228	0.818	25.287
NLS-LR	32.202	0.956	46.533	31.167	0.946	68.442	27.447	0.877	132.737

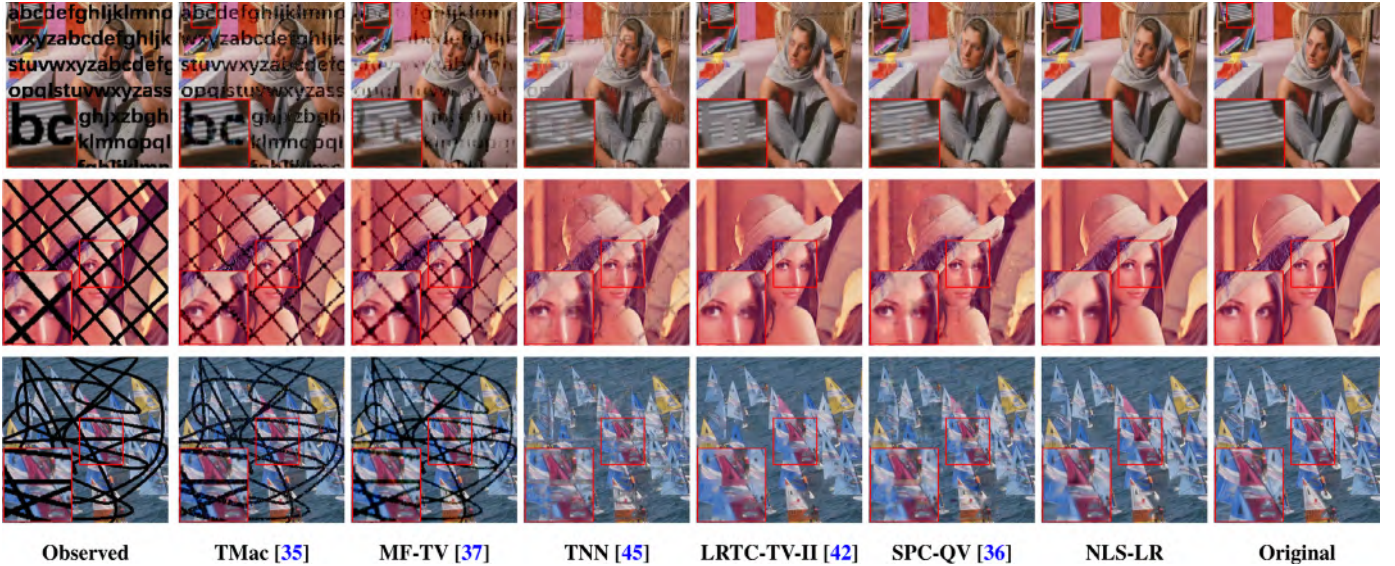


Fig. 3. Reconstructed results for various incomplete color image data. From top to bottom : text masked image, gridlines damaged image, and scratched image. From left to right: the observed data, the recovered results by TMac, MF-TV, TNN, LRTC-TV-II, SPC-QV, NLS-LR, and the original data.

The reconstructed results of three kinds of incomplete color image data are displayed in Fig. 3. Table 3 summarizes the PSNR, SSIM, and CPU time (seconds) of them. The test data include a masked image of *barbara*, a gridlines damaged image of *lena*, and a scratched image of *sails*. We can observe that the proposed method produces the best visual effect. It's visible that there is little difference between our results and the original images, while the results obtained by other methods are still not clear.

4.2. Video

In this section, we test five videos of size $144 \times 176 \times 150$, including *suzie*, *news*, *foreman*, and *carphone*. The maximum Tucker rank is set to be $r^{\max} = (105, 115, 75)$. In the experiment, we use an off-the-shell video denoiser VBM3D [64] to be the denoiser and the denoiser parameter is set to be $\sigma = 5$. The observed tensors are randomly sampled by pixel and the SRs are set to be 5%, 10%, and 20%, respectively. The PSNR, SSIM, and average CPU time (minutes) of the test videos reconstructed by different methods are shown in Table 4. As observed, the proposed method is superior to the compared methods in terms of both PSNR and SSIM values. Meanwhile, the time cost of NLS-LR is acceptable. Fig. 4 shows one frame of five videos which are reconstructed by different methods. TMac and MF-TV neglect the correlation of the third direction, thus their results remain a fair result on high sampling rates, while at a low sampling rate, their results contain evident blurry areas, leading to some details missing. It is obvious that NLS-LR achieves the

best reconstructed visual results. Especially for the moving objects, NLS-LR obtains the best recovering performances.

4.3. Multispectral image

In this section, Columbia multispectral image database¹ is used to test the performances of different methods. The MSI data are size of $256 \times 256 \times 31$. The maximum Tucker rank is set to be $r^{\max} = (185, 185, 5)$. The SRs are set to be 5%, 10%, and 20%, respectively. For MSI data, which is consisted of multiple bands, we input the n -th band and its two adjacent bands to CBM3D and the denoiser parameter is set to be $\sigma = 5$. Table 5 summarizes the PSNR, SSIM, and average CPU time (minutes) obtained by the compared methods and the proposed method. It can be seen that the results obtained by NLS-LR are still superior to the compared methods. Fig. 5 provides the reconstructed results of MSI data. We observe that the proposed method obtains a commendable performance.

5. Discussion

5.1. Parameter analysis

In this subsection, we analyze the effect of the existing parameters of the algorithm. In the proposed model, ρ represents the

¹ <http://www.cs.columbia.edu/CAVE/databases/multispectral/>.

Table 4

The PSNR, SSIM, and average CPU time (minutes) of the results reconstructed by TMac [35], MF-TV [37], TNN [45], LRTC-TV-II [42], SPC-QV [36], and NLS-LR for video completion on different sampling rates and videos.

Video	SR	5%		10%		20%		Time
	Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
Hall	TMac	14.681	0.480	24.916	0.832	34.001	0.948	2.886
	MF-TV	16.706	0.523	28.905	0.880	34.384	0.947	49.732
	TNN	29.720	0.917	32.434	0.946	35.229	0.965	1.191
	LRTC-TV-II	20.288	0.603	21.931	0.705	24.151	0.823	8.751
	SPC-QV	27.492	0.877	29.024	0.905	30.272	0.924	36.961
	NLS-LR	31.219	0.937	34.817	0.965	38.314	0.980	11.934
Coastguard	TMac	7.331	0.018	8.366	0.031	11.215	0.102	2.275
	MF-TV	7.962	0.035	9.338	0.060	13.913	0.171	180.996
	TNN	23.541	0.551	25.180	0.648	27.237	0.754	0.979
	LRTC-TV-II	20.743	0.466	21.935	0.549	25.535	0.723	9.116
	SPC-QV	23.855	0.586	25.193	0.666	26.812	0.755	22.272
	NLS-LR	25.775	0.675	28.230	0.769	31.378	0.866	23.198
News	TMac	10.323	0.088	12.447	0.225	19.066	0.570	3.661
	MF-TV	11.228	0.117	14.336	0.263	26.157	0.728	191.925
	TNN	27.394	0.816	29.610	0.873	32.314	0.921	1.535
	LRTC-TV-II	20.878	0.688	22.392	0.764	25.083	0.867	9.998
	SPC-QV	27.243	0.845	29.215	0.890	31.576	0.928	22.621
	NLS-LR	29.978	0.904	33.347	0.946	36.803	0.971	16.926
Foreman	TMac	7.586	0.018	13.634	0.270	28.835	0.831	3.979
	MF-TV	8.560	0.030	17.228	0.400	31.364	0.872	171.686
	TNN	22.738	0.515	25.419	0.650	28.164	0.770	0.979
	LRTC-TV-II	20.473	0.631	22.652	0.740	27.069	0.874	9.975
	SPC-QV	26.097	0.755	27.702	0.808	29.132	0.848	21.297
	NLS-LR	27.734	0.828	30.873	0.895	34.529	0.943	15.802
Suzie	TMac	11.663	0.047	17.706	0.488	27.219	0.816	3.903
	MF-TV	13.712	0.092	22.308	0.609	32.015	0.873	172.522
	TNN	25.801	0.666	28.116	0.748	30.694	0.831	0.926
	LRTC-TV-II	24.900	0.725	28.253	0.814	31.679	0.891	9.932
	SPC-QV	29.127	0.812	30.786	0.851	32.012	0.877	20.537
	NLS-LR	30.030	0.840	32.689	0.886	35.651	0.928	15.627

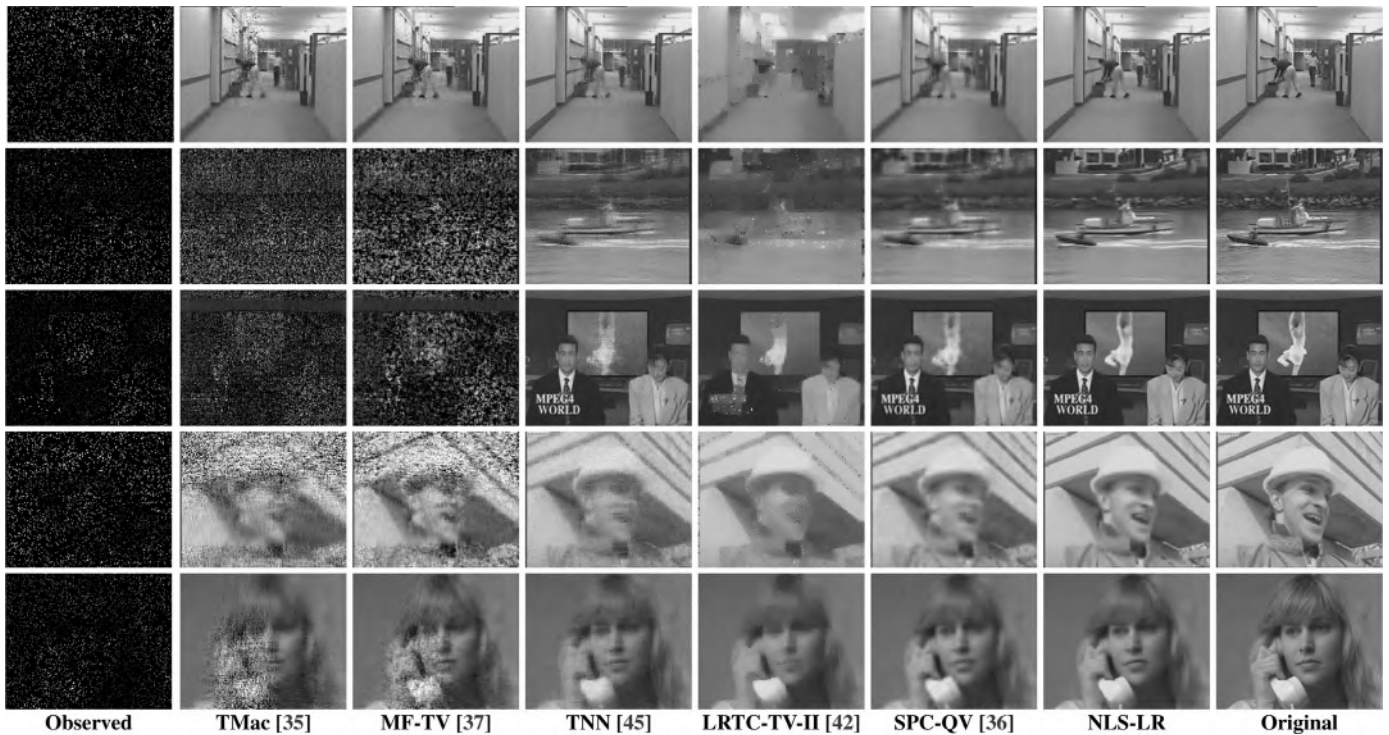


Fig. 4. One frame of five reconstructed videos *coastguard*, *suzie*, *news*, *foreman*, and *hall* with SR = 10%. From left to right: the observed data, the recovered results by TMac, MF-TV, TNN, LRTC-TV-II, SPC-QV, NLS-LR, and the original data.

Table 5

The PSNR,SSIM, and average CPU time (minutes) of the results reconstructed by TMac [35], MF-TV [37], TNN [45], LRTC-TV-II [42], SPC-QV [36], and NLS-LR for MSI completion on different sampling rates and different MSI data.

MSI	SR	5%		10%		20%		Time
	Method	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
Cloth	TMac	11.164	0.109	12.196	0.237	29.346	0.829	1.566
	MF-TV	14.483	0.209	19.102	0.445	29.346	0.829	107.226
	TNN	21.972	0.532	25.266	0.736	29.835	0.882	0.979
	LRTC-TV-II	19.605	0.350	21.402	0.502	24.081	0.692	9.975
	SPC-QV	23.639	0.627	25.431	0.748	27.901	0.853	21.297
	NLS-LR	25.082	0.709	28.223	0.843	32.640	0.935	15.802
Beads	TMac	13.942	0.143	14.516	0.195	16.207	0.383	1.175
	MF-TV	16.345	0.234	18.490	0.363	22.679	0.587	94.557
	TNN	20.202	0.449	23.516	0.660	28.154	0.842	0.690
	LRTC-TV-II	17.897	0.387	20.271	0.588	24.373	0.792	4.958
	SPC-QV	24.668	0.745	27.174	0.840	29.970	0.906	12.469
	NLS-LR	25.859	0.801	28.915	0.894	32.780	0.951	37.881
Toy	TMac	12.130	0.461	14.745	0.660	24.682	0.899	2.641
	MF-TV	13.838	0.519	18.929	0.742	38.159	0.971	157.783
	TNN	28.883	0.843	32.599	0.917	37.552	0.967	0.726
	LRTC-TV-II	23.433	0.774	26.983	0.893	30.489	0.935	4.925
	SPC-QV	29.821	0.895	32.875	0.940	35.156	0.960	8.707
	NLS-LR	31.957	0.936	35.439	0.965	39.533	0.984	11.774

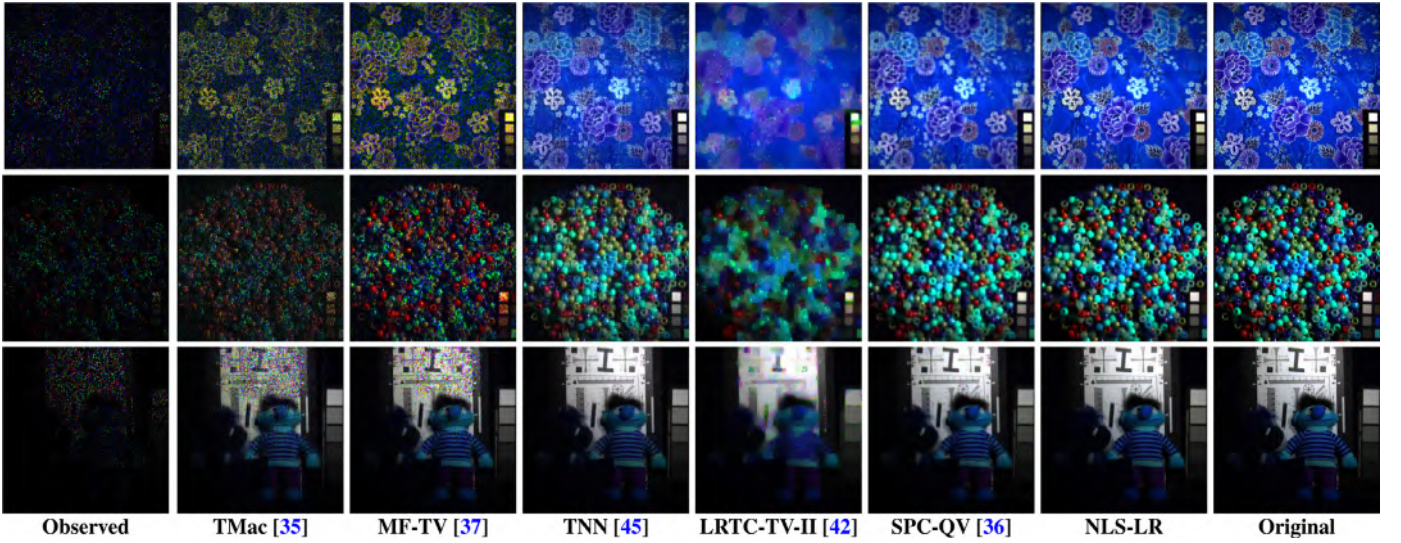


Fig. 5. The pseudo-color images (R-10 G-20 B-30) of three reconstructed MSI data *cloth*, *beads*, and *toy* with SR = 10%. From left to right: the observed data, the recovered results by TMac, MF-TV, TNN, LRTC-TV-II, SPC-QV, NLS-LR, and the original data. For better visualization, the intensity of the pixels has been adjusted.

proximal operator, λ denotes the regularization parameter, and the denoiser parameter σ is denoted as $\sigma = \sqrt{\frac{\lambda}{1+\rho}}$. The value of λ can be calculated by $\lambda = \sigma^2(1+\rho)$ given the values of ρ and σ . In Fig. 6, we demonstrate the PSNR values of the results by NLS-LR with respect to ρ and σ , on color image *barbara* with SR=10%. It is observed that the proximal parameter causes a slight difference in the final performance. Fig. 6 illustrates that the denoiser parameter σ is effective to its performance. Because the outcome will perform over-smoothing if σ is too large, while it is hard to recover the underlying tensor if σ is set to be small. Thus, we should carefully choose the value of denoiser parameter σ to get an appropriate outcome.

5.2. Complexity analysis

Given a three-way tensor $\mathcal{Y} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, the complexity of X -subproblem and A -subproblem at each iteration is $\mathcal{O}(\sum_{n=1}^3 (d_n r_n s_n + r_n^2 s_n + d_n r_n^2 + r_n^3))$, where $s_n = \sum_{m=1, m \neq n}^3 d_m$

and (r_1, r_2, r_3) is the Tucker rank of \mathcal{Y} . For the \mathcal{Y} -subproblem, the complexity at each iteration is $\mathcal{O}(\sum_{n=1}^3 (d_n r_n s_n + R))$, where R represents the computational cost of the denoiser engine, e.g. CBM3D [47] and VBM3D [64]. Therefore, the total computational complexity at each iteration is $\mathcal{O}(\sum_{n=1}^3 (d_n r_n s_n + r_n^2 s_n + d_n r_n^2 + r_n^3 + R))$. For example, if the denoiser engine is CBM3D, the computational complexity is $\mathcal{O}(\sum_{n=1}^3 (d_n r_n s_n + r_n^2 s_n + d_n r_n^2 + r_n^3 + C d_1 d_2 d_3))$, where C is a constant related to the parameters set in CBM3D.

5.3. Convergence

Although the Plug and Play framework has been widely proven to be effective, it still remains an open question whether the PnP framework can be written as a convex model that has a good property of convergence [65,66]. In Fig. 7, we display the RelCha curve of *barbara* with SR = 10%. Factly, it can be observed that the proposed algorithm has an obvious convergence.

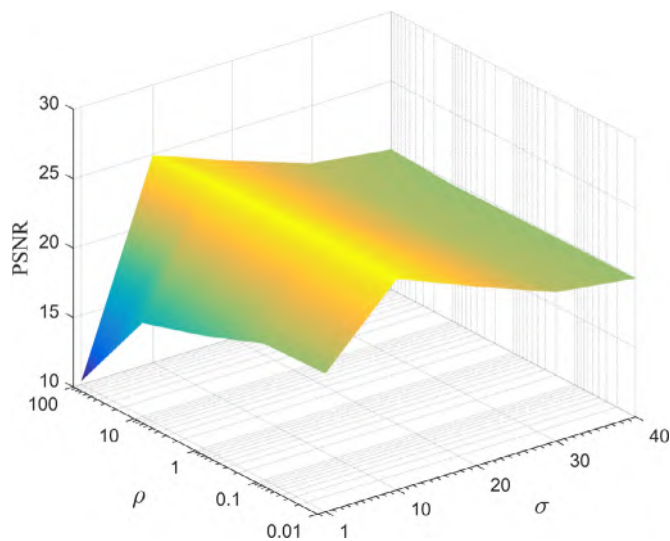


Fig. 6. The PSNR values with respect to iterations for different proximal parameter ρ and denoiser parameter σ .

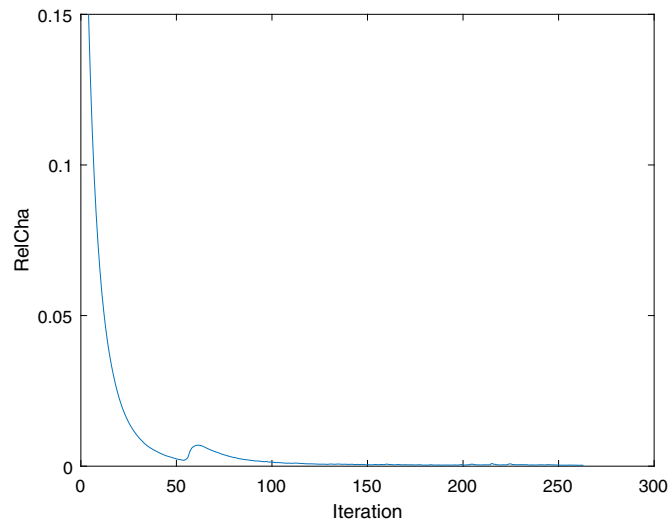


Fig. 7. The convergence curve with respect to iterations.

6. Concluding remarks

In this paper, we proposed a novel low-rank tensor completion model, which integrated low-rank constraints with non-local self-similar regularizer using Plug and Play framework. We adopted the low-rank matrix factorization to guarantee the global low-rankness of the underlying tensor and enhanced the self-similarity of the tensor by employing off-the-shell denoisers (i.e., CBM3D and VBM3D). We adopted the BSUM algorithm to solve the minimizing problem. The numerical experiments showed the effectiveness of the proposed method in preserving the abundant details, which demonstrated its superiorities to many state-of-the-art methods. The tensor reconstructed by the proposed method produced better visual effects and gained the best quality metrics.

Declaration of Competing Interest

None.

CRediT authorship contribution statement

Xiao-Tong Li: Data curation, Formal analysis, Investigation, Software, Writing - original draft. **Xi-Le Zhao:** Conceptualization, Investigation, Methodology, Project administration, Supervision, Writing - review & editing. **Tai-Xiang Jiang:** Formal analysis, Investigation, Writing - review & editing. **Yu-Bang Zheng:** Formal analysis, Investigation, Visualization, Writing - review & editing. **Teng-Yu Ji:** Formal analysis, Investigation, Writing - review & editing. **Ting-Zhu Huang:** Writing - review & editing.

Acknowledgments

The authors would like to express their great thankfulness to Dr. Y. Xu, Dr. T. Yokata, Dr. X. Li, and Dr. Z. Zhang for sharing the codes of the TMac [35], SPC-QV [36], LRTC-TV-II [42], and TNN [45] algorithm. This research is supported by NSFC (61876203, 61772003), the Fundamental Research Funds for the Central Universities (31020180QD126), and Science Strength Promotion Programme of UESTC.

References

- [1] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, in: SIGGRAPH '00, 2000, pp. 417–424.
- [2] H. Tan, B. Cheng, W. Wang, Y.-J. Zhang, B. Ran, Tensor completion via a multi-linear low-rank factorization model, *Neurocomputing* 133 (8) (2014) 161–169.
- [3] 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.
- [4] 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.* 436–437 (2018) 403–417.
- [5] 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.
- [6] Y.-T. Wang, X.-L. Zhao, T.-X. Jiang, L.-J. Deng, T.-H. Ma, Y.-T. Zhang, T.-Z. Huang, A total variation and group sparsity based tensor optimization model for video rain streak removal, *Signal Process. Image Commun.* 73 (2019) 96–108.
- [7] H. Zhang, W. He, L. Zhang, H. Shen, Q. Yuan, Hyperspectral image restoration using low-rank matrix recovery, *IEEE Trans. Geosci. Remote Sens.* 52 (8) (2014) 4729–4743.
- [8] 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.
- [9] Y. Wang, D. Meng, M. Yuan, Sparse recovery: from vectors to tensors, *Natl. Sci. Rev.* 5 (5) (2017) 756–767.
- [10] Z. Long, Y. Liu, L. Chen, C. Zhu, Low rank tensor completion for multiway visual data, *Signal Process.* 155 (2019) 301–316.
- [11] X. Luo, H. Wu, H. Yuan, M. Zhou, Temporal pattern-aware QoS prediction via biased non-negative latent factorization of tensors, *IEEE Trans. Cybern.* (2019), doi:10.1109/TCYB.2019.2903736.
- [12] W. Cao, Y. Wang, C. Yang, X. Chang, Z. Han, Z. Xu, Folded-concave penalization approaches to tensor completion, *Neurocomputing* 152 (2015) 261–273.
- [13] X. Luo, M. Zhou, Y. Xia, Q. Zhu, A.C. Ammari, A. Alabdulwahab, Generating highly accurate predictions for missing QoS data via aggregating nonnegative latent factor models, *IEEE Trans. Neural Netw. Learn. Syst.* 27 (3) (2016) 524–537.
- [14] M. Baburaj, S.N. George, DCT based weighted adaptive multi-linear data completion and denoising, *Neurocomputing* 318 (2018) 120–136.
- [15] X. Luo, M. Zhou, S. Li, Y. Xia, Z. You, Q. Zhu, H. Leung, Incorporation of efficient second-order solvers into latent factor models for accurate prediction of missing QoS data, *IEEE Trans. Cybern.* 48 (4) (2018) 1216–1228.
- [16] M. Che, Y. Wei, Randomized algorithms for the approximations of Tucker and the tensor train decompositions, *Adv. Comput. Math.* 45 (1) (2019) 395–428.
- [17] H. Lu, S. Li, Q. Liu, M. Zhang, MF-LRTC: multi-filters guided low-rank tensor coding for image restoration, *Neurocomputing* 303 (2018) 88–102.
- [18] V. Koltchinskii, K. Lounici, A.B. Tsybakov, Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion, *Ann. Stat.* 39 (5) (2011) 2302–2329.
- [19] Q. Xie, Q. Zhao, D. Meng, Z. Xu, Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery, *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (8) (2018) 1888–1902.
- [20] W. Lun, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, M. Yi, Robust photometric stereo via low-rank matrix completion and recovery, in: Proceedings of the Asian Conference on Computer Vision (ACCV), 2010, pp. 703–717.

- [21] X. Zhang, A nonconvex relaxation approach to low-rank tensor completion, *IEEE Trans. Neural Netw. Learn. Syst.* 30 (6) (2019) 1659–1671.
- [22] X. Luo, M. Zhou, S. Li, M. Shang, An inherently nonnegative latent factor model for high-dimensional and sparse matrices from industrial applications, *IEEE Trans. Ind. Inform.* 14 (5) (2018) 2011–2022.
- [23] J. Yang, X. Zhao, J. Mei, S. Wang, T. Ma, T. 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.
- [24] Y. Chang, L. Yan, S. Zhong, Hyper-Laplacian regularized unidirectional low-rank tensor recovery for multispectral image denoising, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5901–5909.
- [25] S. Gao, Q. Fan, Robust balancing scheme-based approach for tensor completion, *Neurocomputing* 330 (2019) 328–336.
- [26] M. Che, A. Cichocki, Y. Wei, Neural networks for computing best rank-one approximations of tensors and its applications, *Neurocomputing* 267 (2017) 114–133.
- [27] M.K. Ng, Q. Yuan, L. Yan, J. Sun, An adaptive weighted tensor completion method for the recovery of remote sensing images with missing data, *IEEE Trans. Geosci. Remote Sens.* 55 (6) (2017) 3367–3381.
- [28] E.J. Cands, B. Recht, Exact matrix completion via convex optimization, *Found. Comput. Math.* 9 (6) (2009) 717.
- [29] T. Kolda, B. Bader, Tensor decompositions and applications, *SIAM Rev.* 51 (3) (2009) 455–500.
- [30] C.J. Hillar, L.-H. Lim, Most tensor problems are NP-hard, *J. ACM* 60 (6) (2013) 45:1–45:39.
- [31] J.-F. Cai, Cand, E.J. S. Z. Shen, A singular value thresholding algorithm for matrix completion, *SIAM J. Optimiz.* 20 (4) (2010) 1956–1982.
- [32] B. Recht, M. Fazel, P. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, *SIAM Rev.* 52 (3) (2010) 471–501.
- [33] 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.
- [34] S. Gandy, B. Recht, I. Yamada, Tensor completion and low-n-rank tensor recovery via convex optimization, *Inverse Probl.* 27 (2) (2011) 025010.
- [35] Y. Xu, R. Hao, W. Yin, Z. Su, Parallel matrix factorization for low-rank tensor completion, *Inverse Probl. Imaging* 9 (2) (2015) 601–624.
- [36] T. Yokota, Q. Zhao, A. Cichocki, Smooth parafac decomposition for tensor completion, *IEEE Trans. Signal Process.* 64 (20) (2016) 5423–5436.
- [37] 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.
- [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] X.-L. Zhao, W. Wang, T.-Y. Zeng, T.-Z. Huang, M.K. Ng, Total variation structured total least squares method for image restoration, *SIAM J. Sci. Comput.* 35 (6) (2013) B1304–B1320.
- [40] J.-H. Yang, X.-L. Zhao, T.-H. Ma, Y. Chen, T.-Z. Huang, M. Ding, Remote sensing images destriping using unidirectional hybrid total variation and nonconvex low-rank regularization, *J. Comput. Appl. Math.* 363 (2020) 124–144.
- [41] X. Zhang, M.K. Zhang, A corrected tensor nuclear norm minimization method for noisy low-rank tensor completion, *SIAM J. Imaging Sci.* (2019), doi:10.1137/18M1202311.
- [42] X. Li, Y. Ye, X. Xu, Low-rank tensor completion with total variation for visual data inpainting, in: *Proceedings of the Conference on Artificial Intelligence (AAAI)*, 2017, pp. 2210–2216.
- [43] H.K. Aggarwal, A. Majumdar, Hyperspectral image denoising using spatio-spectral total variation, *IEEE Geosci. Remote Sens. Lett.* 13 (3) (2016) 442–446.
- [44] Y.-B. Zheng, T.-Z. Huang, T.-Y. Ji, X.-L. Zhao, T.-X. Jiang, T.-H. Ma, Low-rank tensor completion via smooth matrix factorization, *Appl. Math. Model.* 70 (2019) 677–695.
- [45] Z. Zhang, S. Aeron, Exact tensor completion using t-SVD, *IEEE Trans. Signal Process.* 65 (6) (2017) 1511–1526.
- [46] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering, *IEEE Trans. Image Process.* 16 (8) (2007) 2080–2095.
- [47] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space, in: *Proceedings of the IEEE International Conference on Image Processing*, 2007, pp. 313–316.
- [48] A. Buades, B. Coll, J. Morel, A non-local algorithm for image denoising, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 60–65.
- [49] I. Daubechies, M. Debrise, C. De Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, *Commun. Pure Appl. Math.* 57 (11) (2004) 1413–1457.
- [50] 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.
- [51] Y.-F. Xiang, Y.-F. Jing, T.-Z. Huang, A new projected variant of the deflated block conjugate gradient method, *J. Sci. Comput.* (2019), doi:10.1007/s10915-019-00969-4.
- [52] L.-B. Cui, M.-H. Li, Y. Song, Preconditioned tensor splitting iterations method for solving multi-linear systems, *Appl. Math. Lett.* 96 (2019) 89–94.
- [53] S.V. Venkatakrishnan, C.A. Bouman, B. Wohlberg, Plug-and-play priors for model based reconstruction, in: *Proceedings of the IEEE Global Conference on Signal and Information Processing*, 2013, pp. 945–948.
- [54] S.H. Chan, X. Wang, O.A. Elgindy, Plug-and-play ADMM for image restoration: fixed-point convergence and applications, *IEEE Trans. Comput. Imaging* 3 (1) (2017) 84–98.
- [55] A.M. Teodoro, J.M. Bioucas-Dias, M.A.T. Figueiredo, A convergent image fusion algorithm using scene-adapted gaussian-mixture-based denoising, *IEEE Trans. Image Process.* 28 (1) (2019) 451–463.
- [56] Z. Kai, W. Zuo, S. Gu, Z. Lei, Learning deep CNN denoiser prior for image restoration, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2808–2817.
- [57] X. Wang, S.H. Chan, Parameter-free plug-and-play ADMM for image restoration, in: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 1323–1327.
- [58] Y. Sun, B. Wohlberg, U.S. Kamilov, An online plug-and-play algorithm for regularized image reconstruction, *IEEE Trans. Comput. Imaging* (2019), doi:10.1109/TCI.2019.2893568.
- [59] M. Maggioni, G. Boracchi, A. Foi, K. Egiazarian, Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms, *IEEE Trans. Image Process.* 21 (9) (2012) 3952–3966.
- [60] Y. Chen, T. Pock, Trainable nonlinear reaction diffusion: a flexible framework for fast and effective image restoration, *IEEE Trans. Pattern Anal. Mach. Intell.* 39 (6) (2017) 1256–1272.
- [61] M. Razaviyayn, M. Hong, Z.Q. Luo, A unified convergence analysis of block successive minimization methods for nonsmooth optimization, *SIAM J. Optimiz.* 23 (2) (2013) 1126–1153.
- [62] 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.
- [63] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (4) (2004) 600–612.
- [64] K. Dabov, A. Foi, K. Egiazarian, Video denoising by sparse 3D transform-domain collaborative filtering, in: *Proceedings of the 15th European Signal Processing Conference*, 2007, pp. 145–149.
- [65] A. Teodoro, J. M. Bioucas-Dias, M. Figueiredo, Scene-adapted plug-and-play algorithm with convergence guarantees, in: *Proceedings of the IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2017, pp. 1–6.
- [66] J.J. Moreau, Proximité et dualité dans un espace hilbertien, in: *Proceedings of the Bull Soc Math France*, 2018, pp. 273–299.



Xiao-Tong Li is currently a undergraduate in information and computing science from the University of Electronic Science and Technology of China (UESTC), Chengdu, China. His research interests include models and algorithms of high dimensional image processing problems.



Xi-Le Zhao received the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009 and 2012. He is currently a Professor with the School of Mathematical Sciences, UESTC. His research interests include the sparse and low-rank modeling for image processing problems.



Tai-Xiang Jiang received the B.S. degrees in mathematics and applied mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2013. He is currently pursuing the Ph.D. degree with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China. From 2017 to 2018, supported by the China Scholarship Council, he was a co-training Ph.D. student with the Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal. In 2019, he has been a research assistant in the department of mathematic, Hong Kong Baptist University for three months. His research interests include sparse and low-rank modeling, tensor decomposition and

deep learning. <https://sites.google.com/view/taixiangjiang/>.



Teng-Yu Ji received the B.S. and Ph.D. degrees from the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China, in 2012 and 2018, respectively. He is currently an Assistant Professor with the Department of Applied Mathematics, Northwestern Polytechnical University, Xi'an, China. His research interests include tensor decomposition and applications, including tensor completion and remotely sensed image reconstruction.



Yu-Bang Zheng received the B.S. degree in information and computing science from Anhui University of Finance and Economics, Bengbu, China, in 2017. He is currently working toward the Ph.D. degree in School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China. His research interests include tensor analysis and high dimensional image processing.



Ting-Zhu Huang received the B.S., M.S., and Ph.D. degrees in computational mathematics from the Department of Mathematics, Xian Jiaotong University, Xian, China. He is currently a Professor with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China. His research interests include scientific computation and applications, numerical algorithms for image processing, numerical linear algebra, preconditioning technologies, and matrix analysis with applications. Dr. Huang is an Editor for the Scientific World Journal, Advances in Numerical Analysis, the Journal of Applied Mathematics, the Journal of Pure and Applied Mathematics: Advances in Applied Mathematics, and the Journal of Electronic Science and Technology, China.

and the Journal of Electronic Science and Technology, China.