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

Xiao-Tong Li, Xi-Le Zhao, Tai-Xiang Jiang, Yu-Bang Zheng, Teng-Yu Ji, Ting-Zhu Huang

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.

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

\[
\min_{\mathcal{Y}} \quad \text{rank} (\mathcal{Y}) \\
\text{s.t.} \quad \mathcal{P}_{\Omega} (\mathcal{Y}) = \mathcal{F},
\]

where \( \mathcal{Y} \) is the underlying tensor, \( \mathcal{F} \) is the observed data, \( \Omega \) 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 \( \Omega \) 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 \( \mathcal{Y} \in \mathbb{R}^{d_1 \times \cdots \times d_k} \), 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 \( (\text{rank}(Y_{(1)}), \text{rank}(Y_{(2)}), \ldots, \text{rank}(Y_{(N)})) \), 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: xzhao122003@163.com (X.-L. Zhao).

https://doi.org/10.1016/j.neucom.2019.07.092
0925-2312/© 2019 Elsevier B.V. All rights reserved.
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:

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

(1.2)

where $\alpha_n \geq 0$ ($n = 1, 2, \ldots, N$), $\sum_{n=1}^{N} \alpha_n = 1$, 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.,

$$\min_{\mathcal{Y}, \mathcal{A}} \sum_{n=1}^{N} \alpha_n \| Y_{(n)} - A_nX_n \|_F^2 \quad \text{s.t.} \quad \mathcal{P}_T(Y) = \mathcal{F},$$

(1.3)

where $A = (A_1, A_2, \ldots, A_N)$ and $X = (X_1, X_2, \ldots, X_N)$ 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:

$$\min_{\mathcal{Y}, \mathcal{X}} \sum_{n=1}^{N} \frac{\alpha_n}{2} \| Y_{(n)} - A_nX_n \|_F^2 + \lambda \text{TV}(X_3) \quad \text{s.t.} \quad \mathcal{P}_T(Y) = \mathcal{F},$$

(1.4)

where TV$(X_3)$ is the TV regularizer of factor matrix $X_3$ using piece-wise smooth prior and $\lambda$ 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:

$$\min_{\mathcal{Y}, \mathcal{G}(U^{(n)})} \sum_{n=1}^{N} \frac{\lambda_n}{2} \| F_nY_{(n)} \|_1 + \frac{\mu}{2} \sum_{n=1}^{N} \| U^{(n)} \|_* + \lambda_2 \| \mathcal{G} \|_F^2 \quad \text{s.t.} \quad \mathcal{P}_T(Y) = \mathcal{F}, \quad Y = \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \cdots \times_N U^{(N)},$$

(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 $F_n(i, i) = F_n(i, i + 1) = -1$, 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:

$$\min_{\mathcal{Y}, \mathcal{A}} \sum_{n=1}^{3} \frac{\alpha_n}{2} \| Y_{(n)} - A_nX_n \|_F^2 + \lambda \Phi(Y) \quad \text{s.t.} \quad \mathcal{P}_T(Y) = \mathcal{F},$$

(1.6)

where $\sum_{n=1}^{3} \frac{\alpha_n}{2} \| Y_{(n)} - A_nX_n \|_F^2$ is to guarantee the global low-dimensionality along each mode and $\lambda$ is a regularization parameter. Instead of investing efforts in tailoring non-local regularizers, we develop an implicit regularizer $\Phi(Y)$ using Plag and Play framework (see details in Section 2.3). The implicit regularizer $\Phi(Y)$ 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 Plag 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 lower-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 $\mathcal{X} \in \mathbb{R}^{d_1 \times \cdots \times d_N}$, whose $(i_1, i_2, \ldots, i_N)$th component is denoted as $x_{i_1, i_2, \ldots, i_N}$. A fiber of a tensor is defined by fixing every index but one. The mode-$n$ fibers are vectors $\mathcal{X}(i_1, \ldots, i_{n-1}, i_n, \ldots, i_N)$ respectively for all $i_1, i_2, \ldots, i_N$.

The mode-$n$ unfolding of a tensor $\mathcal{X}$ is denoted as $X_{(n)} \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$, which is a matrix with columns being the mode-$n$ fibers of $\mathcal{X}$. The element $(i_1, i_2, \ldots, i_N)$th of the tensor $\mathcal{X}$ maps to the $(i_n, j)$th element of the matrix $X_{(n)}$ in the lexicographical order, where

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

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.

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

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

$$\langle \mathcal{X}, \mathcal{Y} \rangle := \sum_{i_1, i_2, \ldots, i_N} x_{i_1, i_2, \ldots, i_N} y_{i_1, i_2, \ldots, i_N}.$$
Table 1: An introduction of the related LRTC methods with their properties.

<table>
<thead>
<tr>
<th>Method</th>
<th>Low-rankness or minimizing the rank</th>
<th>Spatial smoothness Constraints on factors/tensors</th>
<th>The solving algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>HALRTC [33]</td>
<td>The Tucker rank</td>
<td>-</td>
<td>An alternating direction method of multipliers based algorithm</td>
</tr>
<tr>
<td>LRTC-TV-I [42]</td>
<td>The Tucker rank</td>
<td>Isotropic TV for underlying tensors</td>
<td>An alternating direction method of multipliers based algorithm</td>
</tr>
<tr>
<td>LRTC-TV-II [42]</td>
<td>The Tucker decomposition</td>
<td>Isotropic TV for underlying tensors</td>
<td>An alternating direction method of multipliers based algorithm</td>
</tr>
<tr>
<td>TMac [35]</td>
<td>Low-rank matrix factorization</td>
<td>-</td>
<td>An alternating direction method of multipliers based algorithm</td>
</tr>
<tr>
<td>MF-TV [37]</td>
<td>Low-rank matrix factorization</td>
<td>Isotropic TV for factors</td>
<td>A block successive upper-bound minimization based algorithm</td>
</tr>
<tr>
<td>MF-Framelet [4]</td>
<td>Low-rank matrix factorization</td>
<td>Framelet for factors</td>
<td>A block successive upper-bound minimization based algorithm</td>
</tr>
<tr>
<td>SMF-LRTC [44]</td>
<td>Low-rank matrix factorization</td>
<td>Unidirectional TV for underlying tensors</td>
<td>A block successive upper-bound minimization based algorithm</td>
</tr>
<tr>
<td>SPC-TV/VO [36]</td>
<td>The PARAFAC/polyadic decomposition rank</td>
<td>Unidirectional TV/VO for underlying tensors</td>
<td>A hierarchical alternating least squares based algorithm</td>
</tr>
<tr>
<td>TNN [45]</td>
<td>The tubal rank</td>
<td>-</td>
<td>An alternating direction method of multipliers based algorithm</td>
</tr>
</tbody>
</table>

The Frobenius norm is defined as:
\[ \|X\|_F := \sqrt{\langle X, X \rangle}. \] (2.4)

The tensor inner product follows the exchange law and combination law, which can be proven as follows.
\[ \langle X, Y \rangle = \sum_{i_1, i_2, \ldots, i_n} x_{i_1, i_2, \ldots, i_n} y_{i_1, i_2, \ldots, i_n} = \sum_{i_1, i_2, \ldots, i_n} y_{i_1, i_2, \ldots, i_n} x_{i_1, i_2, \ldots, i_n} \]
\[ = \langle Y, X \rangle. \] (2.5)

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

2.2. Projected and proximal operators

The projected operator \( P_\Omega (y) \) is a function that keeps the entries of \( y \) in \( \Omega \) while making others be zeros, i.e.,
\[ (P_\Omega (y))_{i_1, i_2, \ldots, i_n} := \begin{cases} y_{i_1, i_2, \ldots, i_n}, & (i_1, i_2, \ldots, i_n) \in \Omega, \\ 0, & \text{otherwise}. \end{cases} \] (2.7)

The proximal operator of a convex function \( f(x) \) is defined as:
\[ \text{prox}_f (y) = \text{arg min}_x \left\{ f(x) + \frac{\rho}{2} \| x - y \|^2 \right\}. \] (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:
\[ x^{(k+1)} = \arg \min_x \frac{\rho}{2} \| x - z \|^2_2 + \lambda \Phi(x). \] (2.9)

where \( \Phi(x) \) is the regularizer and \( \lambda \) is a regularization parameter that trades off the fidelity term and the regularizer. If we define \( \sigma = \sqrt{\frac{\rho}{\lambda}} \), the problem can be rewritten as:
\[ x^{(k+1)} = \arg \min_x \frac{1}{2\sigma^2} \| x - z \|^2_2 + \Phi(x). \] (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:
\[ x^{(k+1)} = D(z, \sigma). \] (2.11)

where \( D \) is the denoiser engine and \( \sigma \) 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 \( Y \in \mathbb{R}^{d_1 \times d_2 \times d_3} \), the objective function of the proposed model (1.6) is:
\[ f(X, A, Y) = \sum_{n=1}^{3} \alpha_n \left\| Y_{(n)} - A_n X_n \right\|^2_2 + \lambda \Phi(Y) + \iota(Y), \] (3.1)

where \( \alpha_n \) 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 $Y$, $\lambda$ is a regularization parameter, $\Phi(Y)$ is a non-local denoiser regularizer, and $\iota(Y)$ is the indicator function, i.e.,
\[
\iota(Y) := \begin{cases} 0, & \text{if } P_{\Omega}(Y) = F, \\ \infty, & \text{otherwise.} \end{cases} \tag{3.2}
\]

The proposed model has two parts, one is the low-rank regularizer term $\sum_{n=1}^{N} \frac{\alpha_n}{2} \left\| Y_n - A_nX_n \right\|^2_F$. We assume the Tucker rank of tensor $Y \in \mathbb{R}^{d_1 \times d_2 \times d_3}$ is $(r_1, r_2, r_3)$, which is given as a low-rank prior. $A_n \in \mathbb{R}^{d_n \times r_n}$, $X_n \in \mathbb{R}^{r_n \times d_n}$ 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 $Y$.

Another part is the regularizer $\Phi(Y)$, which is used to promote the self-similar property. Instead of tailoring non-local regularizers, the implicit regularizer $\Phi(Y)$ 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 $Z = (X, A, Y)$, $Z^k = (X^k, A^k, Y^k)$, and $h(Z, Z^k) = f(Z) + \frac{\rho}{2} \left\| Z - Z^k \right\|^2_F$. By introducing the proximal operator, we can solve the optimization problem (3.1) as follows:
\[
z^{k+1} = \arg \min_{Z} h(Z, Z^k) = \arg \min_{Z} f(Z) + \frac{\rho}{2} \left\| Z - Z^k \right\|^2_F, \tag{3.3}
\]
where $Z = (X, A, Y)$, $Z^k = (X^k, A^k, Y^k)$, and $\rho$ is the proximal parameter. With utilization of the BSUM algorithm [61], the optimization variables of the objective function can be decomposed into multiblocks. Then Eq. (3.3) can be iteratively solved by:
\[
\begin{align*}
X^{k+1} &= \arg \min_{X} \left\{ h_1(X, Z^k) = f(X, A^k, Y^k) + \frac{\rho}{2} \left\| X - X^k \right\|^2_F \right\} \\
A^{k+1} &= \arg \min_{A} \left\{ h_2(A, Z^k) = f(X^{k+1}, A, Y^k) + \frac{\rho}{2} \left\| A - A^k \right\|^2_F \right\} \\
Y^{k+1} &= \arg \min_{Y} \left\{ h_3(Y, Z^k) = f(X^{k+1}, A^{k+1}, Y) + \frac{\rho}{2} \left\| Y - Y^k \right\|^2_F \right\}.
\end{align*}
\]
\[
\tag{3.4}
\]

The $X_n$-subproblem and $A_n$-subproblem can be written as:
\[
\begin{align*}
X_n^{k+1} &= \arg \min_{X_n} \frac{1}{2} \left\| Y_n - A_nX_n^k \right\|^2_F + \frac{\rho}{2} \left\| X_n - X_n^k \right\|^2_F, \\
A_n^{k+1} &= \arg \min_{A_n} \frac{1}{2} \left\| Y_n - A_nX_n^{k+1} \right\|^2_F + \frac{\rho}{2} \left\| A_n - A_n^k \right\|^2_F.
\end{align*}
\]
\[
\tag{3.5}
\]

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 $Y$-subproblem have closed-formed solutions, the core problem turns out to be how to solve the $Y$-subproblem, which can be written as:
\[
Y^{k+1} = \arg \min_{Y} \sum_{n=1}^{N} \frac{\alpha_n}{2} \left\| Y_n - A_nX_n^k \right\|^2_F + \frac{\rho}{2} \left\| Y - Y^k \right\|^2_F + \lambda \Phi(Y) + \iota(Y).
\]
\[
\tag{3.6}
\]

The matrix Frobenius norm $\left\| Y_n - A_nX_n^k \right\|^2_F$ is to calculate the square root of the sum of the squares of all elements. If we define $M_n^k$ as the fold-$n$ tensor of $A_nX_n^k$, it’s not hard to find that the matrix Frobenius norm $\left\| Y_n - A_nX_n^k \right\|^2_F$ is equal to tensor Frobenius norm $\left\| Y - M_n^k \right\|^2_F$, which is also to calculate the square root of the sum of the squares of all elements. Thus we can obtain the $Y$-subproblem as follows:
\[
Y^{k+1} = \arg \min_{Y} \sum_{n=1}^{N} \frac{\alpha_n}{2} \left\| Y_n - A_nX_n^k \right\|^2_F + \frac{\rho}{2} \left\| Y - Y^k \right\|^2_F + \lambda \Phi(Y) + \iota(Y).
\]
\[
\tag{3.7}
\]

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:
\[
Y^{k+1} = \arg \min_{Y} \sum_{n=1}^{N} \frac{\alpha_n}{2} \left\| Y_n - A_nX_n^k \right\|^2_F + \frac{\rho}{2} \left\| Y - Y^k \right\|^2_F + \lambda \Phi(Y) + \iota(Y).
\]
\[
\tag{3.8}
\]

where $M_n^k = fold_n(A_n^kX_n^k)$ is denoted as the fold-$n$ tensor of $A_n^kX_n^k$. The exchange law and combination law of tensor inner product used in the derivation are proven previously in Section 2.1. Let $Y(k) = \frac{1}{\sqrt{\alpha}} \sum_{n=1}^{N} \alpha_nM_n^k + \rho Y^k$, and $\sigma = \sqrt{\frac{\lambda}{1 + \rho}}$, the $Y$-subproblem will be rewritten as:
\[
\min_{Y} \frac{1}{2\sigma^2} \left\| Y - Y(k) \right\|^2_F + \Phi(Y)
\]
\[
\tag{3.9}
\]

s.t. $P_{\Omega}(Y) = F$.

Then the question is formalized as a standard constrained optimization. Factly, $\Phi(Y)$ is the denoiser regularizer, which can be solved in Plug and Play method. By using the project operator to meet the constrain, the $Y$-subquestion can be solved as:
\[
Y^{k+1} = P_{\Omega}(D(Y(k), \sigma) + F),
\]
\[
\tag{3.10}
\]

where $P_{\Omega}$ is the project function to keep constraint condition, $D$ is the denoiser engine, and $\sigma$ is denoted as the denoiser parameter to control the strength of denoising. Note that the denoiser parameter $\sigma$ is linked to the noise level in i.i.d. Gaussian denoising, but in our model the $\sigma$ is linked to the general system error between $Y(k)$ and the ground truth. Thus, in our model $\sigma$ 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 $r_0 = (r_0^1, r_0^2, r_0^3)$, when the relative error becomes less than the tolerance we set, i.e.,
\[
\left| 1 - \frac{P_{\Omega}(A_n^{k+1}X_n^{k+1})}{P_{\Omega}(A_n^kX_n^k)} \right|_F < \gamma, \quad n = 1, 2, 3
\]
\[
\tag{3.11}
\]
then increase the corresponding rank $r_n$ to $\min(r_n + \Delta r_n, r_n^{\max})$ at iteration $k + 1$, where $\Delta r_n$ is a positive integer and $r_n^{\max}$ is the
maximum Tucker rank. More specifically, when the $r_n$ increased at iteration $k+1$, the $A_n^{k+1}$ will be updated to $A_n^{k} \ast \text{rand}(d_n, \Delta r_n)$ and $X_n^{k+1}$ will be updated to $X_n^{k} \ast \text{rand}(\Delta r_n, s_n)$ with $s_n = \prod_{m \neq n} d_m$. i.e., adding randomly generated columns $\Delta r_n$ to $A_n^{k}$ and randomly generated rows $\Delta r_n$ 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 $\gamma = 10^{-4}$ 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., RelCha $= \frac{\|X_n^{k+1} - X_n^{k}\|_F}{\|X_n^{k}\|_F} < \varepsilon$, where $\varepsilon$ 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 $\varepsilon = 3 \times 10^{-4}$, $\Delta r_n = (5, 5, 5)$, and the denoiser parameter $\sigma$ is selected from the set $[5, 10, 15, 20]$. By adopting the Plug and Play framework, we can flexibly choose the denoiser $D$ with the corresponding data. In all experiments, TMac [35] and MF-TV [37] are implemented using the default setting in [37]. TNN [45],
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 incompletely generated tensors are randomly sampled elements and the sampling rates (SRs) are set to be 5%, 10%, and 20%, respectively. The initial Tucker rank $\rho = (10, 10, 3)$ and the maximum Tucker rank $\rho_{max} = (125, 125, 3)$. For color image data, we adopt an off-the-shelf 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.
Meanwhile, to shown the results obtained 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.

<table>
<thead>
<tr>
<th>Type</th>
<th>Text masked image</th>
<th>Gridlines damaged image</th>
<th>Scratched image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>PSNR SSIM Time</td>
<td>PSNR SSIM Time</td>
<td>PSNR SSIM Time</td>
</tr>
<tr>
<td>TMac</td>
<td>13.042 0.522 27.313</td>
<td>13.725 0.611 25.462</td>
<td>11.884 0.538 30.598</td>
</tr>
<tr>
<td>MF-TV</td>
<td>20.359 0.717 660.750</td>
<td>15.947 0.660 604.815</td>
<td>12.597 0.575 630.061</td>
</tr>
<tr>
<td>TNN</td>
<td>25.753 0.820 4.420</td>
<td>26.259 0.831 4.389</td>
<td>23.727 0.777 4.602</td>
</tr>
<tr>
<td>LRTC-TV-II</td>
<td>28.933 0.918 40.315</td>
<td>29.453 0.925 40.643</td>
<td>26.512 0.849 40.791</td>
</tr>
<tr>
<td>SPC-QV</td>
<td>28.095 0.889 18.721</td>
<td>27.208 0.866 14.629</td>
<td>25.228 0.818 25.287</td>
</tr>
<tr>
<td>NLS-LR</td>
<td>32.022 0.956 46.533</td>
<td>31.167 0.946 68.442</td>
<td>27.447 0.877 132.737</td>
</tr>
</tbody>
</table>

The reconstructed results of three kinds of incomplete color image data are displayed in Fig. 3. Table 3 summaries 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 $r^{\text{max}} = (105, 115, 75)$. 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 database\(^1\) 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 $r^{\text{max}} = (185, 185, 5)$. The SRs are set to be 5%, 10%, and 20%, respectively. For MSI data, which is consists 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 summaries 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, $\rho$ represents the

\(^1\) http://www.cs.columbia.edu/CUBE/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.

<table>
<thead>
<tr>
<th>Video</th>
<th>SR</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PSNR</td>
<td>SSIM</td>
<td>PSNR</td>
<td>SSIM</td>
</tr>
<tr>
<td></td>
<td>Method</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hall</td>
<td>TMac</td>
<td>14.681</td>
<td>0.480</td>
<td>24.916</td>
<td>0.832</td>
</tr>
<tr>
<td></td>
<td>MF-TV</td>
<td>16.706</td>
<td>0.523</td>
<td>28.905</td>
<td>0.880</td>
</tr>
<tr>
<td></td>
<td>TNN</td>
<td>29.720</td>
<td>0.517</td>
<td>32.434</td>
<td>0.946</td>
</tr>
<tr>
<td></td>
<td>LRTC-TV-II</td>
<td>20.288</td>
<td>0.603</td>
<td>21.931</td>
<td>0.705</td>
</tr>
<tr>
<td></td>
<td>SPC-QV</td>
<td>27.492</td>
<td>0.877</td>
<td>29.024</td>
<td>0.905</td>
</tr>
<tr>
<td></td>
<td>NLS-LR</td>
<td>31.219</td>
<td>0.937</td>
<td>34.817</td>
<td>0.965</td>
</tr>
<tr>
<td>Coastguard</td>
<td>TMac</td>
<td>7.331</td>
<td>0.018</td>
<td>8.366</td>
<td>0.031</td>
</tr>
<tr>
<td></td>
<td>MF-TV</td>
<td>7.962</td>
<td>0.035</td>
<td>9.338</td>
<td>0.060</td>
</tr>
<tr>
<td></td>
<td>TNN</td>
<td>23.541</td>
<td>0.551</td>
<td>25.180</td>
<td>0.648</td>
</tr>
<tr>
<td></td>
<td>LRTC-TV-II</td>
<td>20.743</td>
<td>0.466</td>
<td>21.935</td>
<td>0.549</td>
</tr>
<tr>
<td></td>
<td>SPC-QV</td>
<td>23.855</td>
<td>0.586</td>
<td>25.193</td>
<td>0.666</td>
</tr>
<tr>
<td></td>
<td>NLS-LR</td>
<td>25.775</td>
<td>0.675</td>
<td>28.230</td>
<td>0.769</td>
</tr>
<tr>
<td>News</td>
<td>TMac</td>
<td>10.323</td>
<td>0.088</td>
<td>12.447</td>
<td>0.225</td>
</tr>
<tr>
<td></td>
<td>MF-TV</td>
<td>11.228</td>
<td>0.117</td>
<td>14.336</td>
<td>0.263</td>
</tr>
<tr>
<td></td>
<td>TNN</td>
<td>27.394</td>
<td>0.816</td>
<td>29.610</td>
<td>0.873</td>
</tr>
<tr>
<td></td>
<td>LRTC-TV-II</td>
<td>20.878</td>
<td>0.688</td>
<td>22.392</td>
<td>0.764</td>
</tr>
<tr>
<td></td>
<td>SPC-QV</td>
<td>27.243</td>
<td>0.845</td>
<td>29.215</td>
<td>0.890</td>
</tr>
<tr>
<td></td>
<td>NLS-LR</td>
<td>29.978</td>
<td>0.904</td>
<td>33.347</td>
<td>0.946</td>
</tr>
<tr>
<td>Foreman</td>
<td>TMac</td>
<td>7.586</td>
<td>0.018</td>
<td>13.634</td>
<td>0.270</td>
</tr>
<tr>
<td></td>
<td>MF-TV</td>
<td>8.560</td>
<td>0.030</td>
<td>17.228</td>
<td>0.400</td>
</tr>
<tr>
<td></td>
<td>TNN</td>
<td>22.738</td>
<td>0.515</td>
<td>25.419</td>
<td>0.650</td>
</tr>
<tr>
<td></td>
<td>LRTC-TV-II</td>
<td>20.473</td>
<td>0.631</td>
<td>22.052</td>
<td>0.740</td>
</tr>
<tr>
<td></td>
<td>SPC-QV</td>
<td>26.097</td>
<td>0.755</td>
<td>27.702</td>
<td>0.808</td>
</tr>
<tr>
<td></td>
<td>NLS-LR</td>
<td>27.734</td>
<td>0.828</td>
<td>30.873</td>
<td>0.895</td>
</tr>
<tr>
<td>Suzie</td>
<td>TMac</td>
<td>11.663</td>
<td>0.047</td>
<td>17.706</td>
<td>0.488</td>
</tr>
<tr>
<td></td>
<td>MF-TV</td>
<td>13.712</td>
<td>0.092</td>
<td>22.308</td>
<td>0.609</td>
</tr>
<tr>
<td></td>
<td>TNN</td>
<td>25.801</td>
<td>0.666</td>
<td>28.116</td>
<td>0.748</td>
</tr>
<tr>
<td></td>
<td>LRTC-TV-II</td>
<td>24.900</td>
<td>0.725</td>
<td>28.253</td>
<td>0.814</td>
</tr>
<tr>
<td></td>
<td>SPC-QV</td>
<td>29.127</td>
<td>0.812</td>
<td>30.786</td>
<td>0.851</td>
</tr>
<tr>
<td></td>
<td>NLS-LR</td>
<td>30.030</td>
<td>0.840</td>
<td>32.689</td>
<td>0.886</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>MSI</th>
<th>SR</th>
<th>5% PSNR</th>
<th>5% SSIM</th>
<th>10% PSNR</th>
<th>10% SSIM</th>
<th>20% PSNR</th>
<th>20% SSIM</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloth</td>
<td>TMac</td>
<td>11.164</td>
<td>0.109</td>
<td>12.196</td>
<td>0.237</td>
<td>29.346</td>
<td>0.829</td>
<td>1.566</td>
</tr>
<tr>
<td></td>
<td>MF-TV</td>
<td>14.493</td>
<td>0.209</td>
<td>19.102</td>
<td>0.445</td>
<td>29.346</td>
<td>0.829</td>
<td>107.226</td>
</tr>
<tr>
<td></td>
<td>TNN</td>
<td>21.972</td>
<td>0.532</td>
<td>25.266</td>
<td>0.736</td>
<td>29.835</td>
<td>0.882</td>
<td>0.979</td>
</tr>
<tr>
<td></td>
<td>LRTC-TV-II</td>
<td>19.605</td>
<td>0.350</td>
<td>21.402</td>
<td>0.502</td>
<td>24.081</td>
<td>0.692</td>
<td>9.975</td>
</tr>
<tr>
<td></td>
<td>SPC-QV</td>
<td>23.639</td>
<td>0.627</td>
<td>25.431</td>
<td>0.748</td>
<td>27.901</td>
<td>0.853</td>
<td>21.297</td>
</tr>
<tr>
<td></td>
<td>NLS-LR</td>
<td>25.082</td>
<td>0.709</td>
<td>28.223</td>
<td>0.843</td>
<td>32.640</td>
<td>0.935</td>
<td>15.802</td>
</tr>
<tr>
<td>Beads</td>
<td>TMac</td>
<td>13.942</td>
<td>0.143</td>
<td>14.516</td>
<td>0.195</td>
<td>16.207</td>
<td>0.383</td>
<td>1.175</td>
</tr>
<tr>
<td></td>
<td>MF-TV</td>
<td>16.345</td>
<td>0.234</td>
<td>18.490</td>
<td>0.363</td>
<td>22.679</td>
<td>0.587</td>
<td>94.557</td>
</tr>
<tr>
<td></td>
<td>TNN</td>
<td>20.202</td>
<td>0.449</td>
<td>23.516</td>
<td>0.660</td>
<td>28.154</td>
<td>0.842</td>
<td>0.690</td>
</tr>
<tr>
<td></td>
<td>LRTC-TV-II</td>
<td>17.897</td>
<td>0.387</td>
<td>20.271</td>
<td>0.588</td>
<td>24.373</td>
<td>0.792</td>
<td>4.958</td>
</tr>
<tr>
<td></td>
<td>SPC-QV</td>
<td>24.668</td>
<td>0.745</td>
<td>27.174</td>
<td>0.840</td>
<td>29.970</td>
<td>0.906</td>
<td>12.469</td>
</tr>
<tr>
<td></td>
<td>NLS-LR</td>
<td>25.859</td>
<td>0.801</td>
<td>28.915</td>
<td>0.894</td>
<td>32.780</td>
<td>0.951</td>
<td>37.881</td>
</tr>
<tr>
<td>Toy</td>
<td>TMac</td>
<td>12.130</td>
<td>0.461</td>
<td>14.745</td>
<td>0.660</td>
<td>24.682</td>
<td>0.899</td>
<td>2.641</td>
</tr>
<tr>
<td></td>
<td>MF-TV</td>
<td>13.838</td>
<td>0.519</td>
<td>18.929</td>
<td>0.742</td>
<td>38.159</td>
<td>0.971</td>
<td>157.783</td>
</tr>
<tr>
<td></td>
<td>TNN</td>
<td>28.883</td>
<td>0.843</td>
<td>32.599</td>
<td>0.917</td>
<td>37.552</td>
<td>0.967</td>
<td>0.726</td>
</tr>
<tr>
<td></td>
<td>LRTC-TV-II</td>
<td>23.433</td>
<td>0.774</td>
<td>26.983</td>
<td>0.893</td>
<td>30.489</td>
<td>0.935</td>
<td>4.925</td>
</tr>
<tr>
<td></td>
<td>SPC-QV</td>
<td>29.821</td>
<td>0.895</td>
<td>32.875</td>
<td>0.940</td>
<td>35.156</td>
<td>0.960</td>
<td>8.707</td>
</tr>
<tr>
<td></td>
<td>NLS-LR</td>
<td>31.957</td>
<td>0.936</td>
<td>35.439</td>
<td>0.965</td>
<td>39.533</td>
<td>0.984</td>
<td>11.774</td>
</tr>
</tbody>
</table>

Fig. 5. The pseudo-color images (R:10G-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 σ = \sqrt{\frac{\lambda}{\sigma^2}}. The value of λ can be calculated by λ = σ^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 \( \lambda \)-subproblem and \( \Lambda \)-subproblem at each iteration is \( O\left( \sum_{n=1}^{3} (d_n r_1 s_n + r_2^2 s_n + d_n r_2^2 + r_3^2) \right) \), where \( s_n = \sum_{m=1, m \neq n} d_m \) and \( (r_1, r_2, r_3) \) is the Tucker rank of \( \mathcal{Y} \). For the \( \psi \)-subproblem, the complexity at each iteration is \( O\left( \sum_{n=1}^{3} (d_n r_1 s_n + r_2^2 s_n + d_n r_2^2 + r_3^2) \right) \), 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 \( O\left( \sum_{n=1}^{3} (d_n r_1 s_n + r_2^2 s_n + d_n r_2^2 + r_3^2 + C d_1 d_2 d_3) \right) \), 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 RELCh 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 $\rho$ and denoiser parameter $\sigma$.

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 constrains 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-IV-II [42], and TNN [45] algorithm. This research is supported by NSFC (61872603, 61772003), the Fundamental Research Funds for the Central Universities (31020180QD126), and Science Strength Promotion Programme of UESTC.

References


M. Che, A. Cichocki, Y. Wei, Neural networks for computing best rank-one approximations of tensors and its applications, Neurocomputing 267 (2017) 114–133.


S. Gandy, B. Recht, I. Yamada, Tensor completion and low-rank tensor recovery via convex optimization, Inverse Probl. 27 (2) (2011) 025010.


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 mathematics, 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/.

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.

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.

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.