



Auto-weighted robust low-rank tensor completion via tensor-train



Chuan Chen ^{a,b}, Zhe-Bin Wu ^{a,b}, Zi-Tai Chen ^{a,b}, Zi-Bin Zheng ^{a,b,*}, Xiong-Jun Zhang ^{c,*}

^a The School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China

^b National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China

^c The School of Mathematics and Statistics and Hubei Key Laboratory of Mathematical Sciences, Central China Normal University, Wuhan 430079, China

ARTICLE INFO

Article history:

Received 23 January 2020

Received in revised form 12 January 2021

Accepted 12 March 2021

Available online 18 March 2021

Keywords:

Robust tensor completion

Tensor-train rank

Ket augmentation

Auto-weighted

ADMM

ABSTRACT

Nowadays, multi-dimensional data (tensor data) have shown their capability of preserving multilinear structures. Due to the measuring error or other non-human factors, these data often suffer signal corruptions or missing values, or even both. To address these issues simultaneously, this paper studies the Robust Tensor Completion (RTC) problem, a mixed problem of the known Low-Rank Tensor Completion (LRTC) and Robust Principal Component Analysis (RPCA). Based on Tensor-Train rank (TT rank), the proposed model is able to capture the latent structure information of tensor data by recovering the low-rank component and separating the sparse component from the partial observations. To make TT rank more effective, an auto-weighted mechanism is utilized to balance the importance of different matricizations from the same tensor. We also propose a more flexible tensor augmentation approach called Tree Ket Augmentation (Tree-KA) to obtain a higher-order tensor from a lower one with a new general explanation. Alternating direction method of multipliers (ADMM) is employed to solve the resulting model and extensive numerical experiments have verified the effectiveness of the proposed model compared with other state-of-the-art methods.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

Tensors, also known as multi-arrays, have received much attention in a great number of fields in a recent decade, such as computer vision [1–3], machine learning [4,5], data mining [6], and hyperspectral image processing [7,8]. The tensor datas are prone to missing values and sparse noise when acquiring and processing them. While the low-rank property of tensor makes it possible to reconstruct the low-rank component from the damaged data. Two major previous works, Robust Principal Component Analysis (RPCA) [9] and Low-Rank Tensor Completion (LRTC) [2], can only address the outlier problem and the missing value problem alone, respectively. Thus, in this paper, we mainly focus on the Robust Tensor Completion (RTC) problem [10,11], which aims to decompose a given tensor with partial observations into a low rank tensor and a sparse tensor [12].

* Corresponding authors at: The School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China (Zi-Bin Zheng); The School of Mathematics and Statistics and Hubei Key Laboratory of Mathematical Sciences, Central China Normal University, Wuhan 430079, China (Xiong-Jun Zhang).

E-mail addresses: chenchuan@mail.sysu.edu.cn (C. Chen), wuzhb6@mail2.sysu.edu.cn (Z.-B. Wu), chenzt25@mail2.sysu.edu.cn (Z.-T. Chen), zhzbin@mail.sysu.edu.cn (Z.-B. Zheng), xjzhang@mail.ccnu.edu.cn (X.-J. Zhang).

As it's well known that multi-channel data prevails in the real world and the two-dimensional array is difficult to meet the structural requirements of storing higher-dimensional data. For example, the color image not only has the height and width of the plane but also has three channels of Red, Green, and Blue (RGB) in depth, while hyperspectral images own more channels. Apparently, if a one-dimensional or two-dimensional array is used for data storage at this time, the structured information of the original data may be lost. Besides, the common tensors are nevertheless of low-rank, and tensor decomposition is usually the core of high-dimensional data processing. In this way, they can be effectively transformed into much smaller subspaces through powerful decompositions such as the CANDECOMP/PARAFAC (CP) [13], Tucker [14] and Tensor-Train (TT) [15]. And these decompositions are usually used to analyze multi-linear datas to obtain a higher-precision result.

Among the images or video data which lies in high dimensional spaces, the characteristics of low-rank structure play an essential role. The low-rank component shows a high correlation in various parts of a picture or frames of a video, which offers potential inherent information of structure. The principal component analysis (PCA) for dimensionality reduction is the most popular data analysis tool to perform low-rank approximation on the input matrix or tensor. Because its implementation can be achieved by eigenvalue decomposition, it is especially sensitive to outliers.

To overcome this drawback, the matrix-based RPCA [9] is introduced to eliminate sparse noise in the input matrix in order to obtain a low-rank approximation. The known observations are not accurate and randomly contain variance in different strengths. Hence, to recover potential low-rank and sparse noise components from the partially corrupted entities is knotty. Suppose we have an observation matrix $Y = X + N$, where $X \in \mathbb{R}^{n_1 \times n_2}$ represents the low-rank component and $N \in \mathbb{R}^{n_1 \times n_2}$ represents the sparse noise component. Then the RPCA problem can be written as follow [16]:

$$\min_{X, N} \text{rank}(X) + \lambda \|N\|_1 \quad \text{s.t.} \quad Y = X + N, \quad (1)$$

where $\|N\|_1$ denotes the ℓ_1 -norm (sum of the absolute values of all the entries in N). λ is the regularization coefficient. Moreover, the larger λ is, the greater the sparsity of N is (there are more zero elements). It's a convex optimization problem with equality constraints inherently. In this way, RPCA is able to address the sparse noise problem with the observations, whose structure is the sum of low-rank and sparse components. In tensor case, the given observed tensor $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ can be also decomposed as $\mathcal{Y} = \mathcal{X} + \mathcal{N}$, and the RPCA model becomes tensor RPCA (TRPCA):

$$\min_{\mathcal{X}, \mathcal{N}} \text{rank}(\mathcal{X}) + \lambda \|\mathcal{N}\|_1 \quad \text{s.t.} \quad \mathcal{Y} = \mathcal{X} + \mathcal{N} \quad (2)$$

To overcome the missing value estimation problem, i.e., obtaining a completed object from its partial components, a lot of attempts have been made [17–19]. The core of these problems is to find the relationship between missing entries and retained entries, deriving many low-rank matrix completion (LRMC)[20] methods. They aim to recover missing values from partially observed entries. Given an observation subset Ω , then recovering the missing entries from the known is a well-known low-rank matrix optimization problem:

$$\min_X \text{rank}(X) \quad \text{s.t.} \quad \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(T) \quad (3)$$

where T is the original data and $\mathcal{P}_\Omega(\cdot)$ is the projection operator (see definition in Section 3). Similarly, the tensor-based completion problem, i.e. LRTC, to recover a multi-dimensional data $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is:

$$\min_{\mathcal{X}} \text{rank}(\mathcal{X}) \quad \text{s.t.} \quad \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(T) \quad (4)$$

In fact, the situation may go more complicated, i.e., the given entries are not only partial but also inaccurate. Neither of the models above can solve this alone. For TRPCA methods, once they cannot get completed objects, the missing entries may be regarded as outliers. As a result, the missing entries of the low-rank component can be recovered by mean or other values. For LRTC methods, once their partially known entries are not all accurate, the noisy ones can never be modified, which causes the reduction of recovering performance. Therefore, it's worthwhile to solve these two problem at the same time to construct a RTC model, i.e.,

$$\begin{aligned} & \min_{\mathcal{X}, \mathcal{N}} \text{rank}(\mathcal{X}) + \lambda \|\mathcal{N}\|_1 \\ & \text{s.t.} \quad \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{P}_\Omega(\mathcal{X} + \mathcal{N}). \end{aligned} \quad (5)$$

In this way, the mixture model can address the observations with partial noise problem based on TT rank.

The paper's specific contributions are as follows:

1. We proposed a new mixture Robust Tensor Completion model based on TT rank to recover missing entries from partially known corrupted entries. We used Block Coordinate Descent (BCD) [21] and the Alternating Direction Method of Multipliers (ADMM) [22] to optimize the proposed model.
2. The TT rank of different modes varies from each other and the global information contained is various accordingly. Thus, we adopted an auto-weighted mechanism to measure the importance of TT ranks.
3. We also proposed a more flexible Ket Augmentation (KA) [23] named Tree-KA by using the tree structure, which can better fit various sizes of tensors. As a result, the Tree-KA scheme can also represent a low-order tensor by different higher-order tensors without changing the total number of the entries.

The rest of this paper is organized as follows. In Section 2, we illustrate some relevant works. In Section 3, we define the notations about tensor and some definitions in detail. In Section 4, we propose our model RTC-TT along with its optimization process. In Section 5, experimental results are reported and analyzed. The last Section 6 concludes and plan for future work.

2. Related work

Since the minimization of the rank of the tensor in Eq. (2) is NP-hard [24], the proved best convex approximation, the tensor nuclear norm, could be a surrogate for the function $\text{rank}(\cdot)$ [25], which leads to the following optimization problem [16]:

$$\min_{\mathcal{X}, \mathcal{N}} \|\mathcal{X}\|_* + \lambda \|\mathcal{N}\|_1 \quad \text{s.t.} \quad \mathcal{Y} = \mathcal{X} + \mathcal{N} \quad (6)$$

where $\|\mathcal{X}\|_*$ denotes the tensor nuclear norm (see definition in Section 3). This is the well-known robust principal component analysis problem, which was originally proposed in [9] based on matrix. While Lu et al. [16] leveraged the emerging numerical algebra of tensors, e.g. t-product [26], to extend it to the tensor case as in Eq. (6). And actually t-product is a convolution-like operation, which can be implemented by using Discrete Fourier Transform (DFT). Thus, he further proposed a more general TRPCA model in [27]. The best transform is different for different tasks, which leads to different definitions of tensor rank and tensor nuclear norm according to the invertible linear transforms used.

Low-rank tensor completion based on Sum of Nuclear Norm (SNN) was first proposed in [2], while it is mainly based on Tucker rank [14]. Goldfarb et al. [28] proposed to use the SNN for robust tensor completion, which is given as follows:

$$\begin{aligned} \min_{\mathcal{X}, \mathcal{N}} \quad & \sum_{i=1}^N \|\mathcal{X}_{(i)}\|_* + \lambda \|\mathcal{N}\|_1 \\ \text{s.t.} \quad & \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{P}_\Omega(\mathcal{X} + \mathcal{N}). \end{aligned} \quad (7)$$

However, the SNN is just suboptimal to approximate the Tucker rank minimization [19]. Moreover, the Tucker rank was proved not so good as TT rank on capturing global correlations in [17]. TT rank originally came from Tensor-Train decomposition, which was proposed by Oseledets [15] in 2011. Based on tensor tubal rank, Lu proposed to use the tensor nuclear norm in [29] for third-order tensors. Nevertheless, the tensor nuclear norm method is just suitable for third-order tensors, while many real-world datasets are higher-order tensors, such as color video. On the other hand, Xu et al. [30] proposed an unfolding method along each mode based on matrix factorization for noisy low-rank tensor completion, while it also owns the shortcomings of the SNN method since it utilizes the Tucker rank minimization of a tensor.

3. Notations and preliminaries

In this section, the representations notations and preliminaries are described in detail.

3.1. Notations

Throughout this paper, we denote an n -mode tensor by calligraphic letters, e.g., $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$, where $I_k, k = 1, 2, \dots, n$ is the dimension of mode k . And matrices are denoted by the upper case letters, e.g., X . Especially, the element (i_1, i_2, \dots, i_n) of tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$ is denoted by $x_{i_1 i_2 \dots i_n}$. Vectors are denoted by boldface lowercase letters, e.g. \mathbf{x} and scalars by lowercase letters, e.g. x .

Some mathematical operations of matrix and tensor are used. The inner product of $X \in \mathbb{R}^{n \times m}$ and $Y \in \mathbb{R}^{n \times m}$ is defined as $\langle X, Y \rangle = \text{tr}(X^T Y)$, where X^T is the transpose of matrix X and $\text{tr}()$ is matrix trace. The ℓ_1 -norm is defined as $\|\mathcal{X}\|_1 = \sum_{i_1 i_2 \dots i_n} |x_{i_1 i_2 \dots i_n}|$, the Frobenius norm as $\|\mathcal{X}\|_F = \sqrt{\sum_{i_1 i_2 \dots i_n} x_{i_1 i_2 \dots i_n}^2}$ and the nuclear norm of matrix as $\|X\|_* = \sum_i \sigma_i(X)$, where $\sigma_i(X)$ is the i -th singular value of matrix X . $\text{fold}(\cdot)$ is the operation to convert a lot of matrices into a tensor and it is opposite to $\text{unfold}(\cdot)$. For a vector \mathbf{x} , the ℓ_2 -norm is $\|\mathbf{x}\|_2 = \sqrt{\sum_i x_i^2}$.

3.2. Preliminaries

Definition 1 (Projection Operator $\mathcal{P}_\Omega(\cdot)$). [31] Suppose given an index set Ω , the projection operator $\mathcal{P}_\Omega(\mathcal{T})$ is defined as the tensor that exacts the entries from $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$ in set $\Omega, n = 1, 2, 3, \dots$. And the other were set to zeros, i.e.,

$$\mathcal{P}_\Omega(\mathcal{T})_{i_1 i_2 \dots i_N} = \begin{cases} \mathcal{T}_{i_1 i_2 \dots i_N}, & i_1 i_2 \dots i_N \in \Omega, \\ 0, & i_1 i_2 \dots i_N \notin \Omega. \end{cases}$$

Definition 2 (Tensor-Train Rank). [15] A tensor-train rank is generated by tensor-train decomposition of tensor $\mathcal{X} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_N}$, which owns a succinct format:

$$\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N),$$

where $r_k = \text{rank}(X_{[k]})$ and $X_{[k]} \in \mathbb{R}^{m \times n}$ ($m = \prod_{i=1}^k d_i, n = \prod_{s=k+1}^N d_s$), $k = 1, 2, \dots, N$.

Definition 3 (Tensor Nuclear Norm). [32] The tensor nuclear norm of $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\|\mathcal{X}\|_*$, is defined as the sum of the singular values of all the frontal slices of $\hat{\mathcal{X}}$, where $\hat{\mathcal{X}}$ is obtained by performing the fast Fourier transformation (FFT) along the tube fibers of \mathcal{X} . Indeed,

$$\|\mathcal{X}\|_* = \sum_{i=1}^{n_3} \|\hat{X}^{(i)}\|_*,$$

where $\hat{X}^{(i)}$ denotes the i -th frontal slice of $\hat{\mathcal{X}}$.

Definition 4 (Tensor-Train (TT) Nuclear Norm). [17] The tensor trace norm of tensor $\mathcal{X} \in \mathbb{R}^{l_1 \times l_2 \times \dots \times l_n}$, denoted by $\|\mathcal{X}\|_*$, is defined as a convex combination of the tensor-train nuclear norms of different matrices flattened along each mode, that is, $\|\mathcal{X}\|_* = \sum_{k=1}^n \alpha_k \|X_{[k]}\|_*$, where α_k is the weight of the k -th norm and $\sum_{k=1}^n \alpha_k = 1, \alpha_k > 0$.

4. The proposed algorithm

4.1. Robust tensor completion based on tensor-train rank (RTC-TT)

The main problem of tensor model is the definition of tensor rank due to the exist of a common dilemma. Unlike the several “good” properties of matrix rank, the properties of tensor rank are difficultly satisfied. Moreover, the definition of nuclear norm and the convex envelope of the chosen rank are also ambiguous. The several popular definitions of tensor rank are CP rank, Tucker rank, tubal rank [33] and tensor-train rank (TT rank). Compared with the others, the TT-rank can capture the global correlation of a tensor as it provides a correlation between a few modes and the rest [17]. For example, suppose there is a tensor with all the same dimension ($d_1 = d_2 = d_3 = \dots = d_n = d$), then matrix $X_{[k]}$ has a dimension of $\prod_{i=1}^k d_i \times \prod_{s=k+1}^n d_s$. It's a relatively balanced metricizing scheme against Tucker rank with a dimension of $d \times d^{n-1}$.

Thus, we aim to study the Tensor-Train rank optimization problem in this paper, and Eq. (6) turns to the TT Nuclear Norm [17] form as:

$$\min_{X_{[k]}} \sum_{k=1}^{N-1} \alpha_k \|X_{[k]}\|_* + \lambda \|\mathcal{N}\|_1 \quad \text{s.t.} \quad \mathcal{Y} = \mathcal{X} + \mathcal{N} \quad (8)$$

where α_k denotes the weight of $X_{[k]}$, matirx flattened along the k -th mode. Similarly, the low-rank tensor completion (LRTC) problem (4) solved by TT rank optimization becomes:

$$\min_{X_{[k]}} \sum_{k=1}^{N-1} \alpha_k \|X_{[k]}\|_* \quad \text{s.t.} \quad \mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{T}) \quad (9)$$

As mentioned in Section 1, neither RPCA nor LRTC can solve the situation with both missing entries and sparse noise, i.e., in addition to being grossly corrupted, the observed entities are only partial. Thus the tensor case of the exact recovery of low-rank component is considered as a combination of Eq. (8) and (9), whose model (RTC-TT) can be obtained as follow:

$$\begin{aligned} \min_{X_{[k]}, \mathcal{N}} \quad & \sum_{k=1}^{N-1} \alpha_k \|X_{[k]}\|_* + \lambda \|\mathcal{N}\|_1 \\ \text{s.t.} \quad & \mathcal{P}_{\Omega}(\mathcal{Y}) = \mathcal{P}_{\Omega}(\mathcal{Z}), \quad \mathcal{Z} = \mathcal{X} + \mathcal{N}, \quad \boldsymbol{\alpha}^T \mathbf{1} = 1, \quad \boldsymbol{\alpha} \geq 0, \end{aligned} \quad (10)$$

where Ω is the set of partial observations with unknown noise, \mathcal{Y} is the corrupted object without missing entries and \mathcal{Z} is the recovering object. The ℓ_1 -norm is utilized to separate the sparse component from observations, as suggested in some former works [9,16].

4.2. Auto-weighted mechanism

The importance of TT rank of different modes varies and the nuclear norm can be a surrogate for the rank approximately, which inspires us to use different weights to measure their importance. However, how to select α_k remains pendent.

In this paper, we introduce an auto-weighted mechanism [34] to adapt the TT ranks of different modes. Due to the particularity of matrix rank, the larger rank contains more information, which should deserve larger weight. We construct a maximizing objective function to generate adaptive weights with the variation of matrix rank based on Eq. (10) ℓ_2 -norm term is added:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{k=1}^{N-1} \alpha_k \|X_{[k]}\|_* - \gamma \|\boldsymbol{\alpha}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{\alpha}^T \mathbf{1} = 1, \quad \boldsymbol{\alpha} \geq 0, \end{aligned} \quad (11)$$

where $\gamma, \lambda > 0$. The first term of Eq. (11) comes from Eq. (10) by concluding the information relevant to weights α . The larger the nuclear norm of flattened matrix $X_{[k]}$ is, the larger weight is forced to maintain more information of underlying tensor data. And the new term in Eq. (11) is a penalty term as well as used to smoothen the weight distribution. In other word, this term can avoid the occurrence of the situation like this, the most important mode will get a weight of 1 while the others of 0s when $\gamma \rightarrow 0$. On the contrary, when $\gamma \rightarrow \infty$, each weight will be equal. Without loss of generality, a minus sign is exploited here to simplify the solving process in the meanwhile to avoid trivial solutions.

4.3. Optimization

Given that proper correlations γ and λ , a set of partially observed corrupted entries Ω , and in which there exists a structure like $\mathcal{Y} = \mathcal{X} + \mathcal{N}$, where \mathcal{X} is low-rank and \mathcal{N} is sparse, we can optimize the problem (11) by two steps. The outer framework of optimization is BCD and the inner framework is ADMM.

4.3.1. Outer optimization framework (BCD)

We divide the updating parameters into two blocks. The first block is $\boldsymbol{\alpha}$ and the second block is the others (\mathcal{X} and \mathcal{N}). In the first block, we solve the objective function (11) by rewriting it as the minimization form initially:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & -\boldsymbol{\mu}^T \boldsymbol{\alpha} + \gamma \|\boldsymbol{\alpha}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{\alpha}^T \mathbf{1} = 1, \boldsymbol{\alpha} \geq 0, \end{aligned} \quad (12)$$

where vector $\boldsymbol{\mu} = (\|X_{[1]}\|_*, \|X_{[2]}\|_*, \dots, \|X_{[N]}\|_*)$. Obviously, it's a convex Quadratic Programming (QP) with equality and nonequality constraints, and it can be solved by any QP solvers. The Lagrangian function of Eq. (12) is

$$L = -\boldsymbol{\mu}^T \boldsymbol{\alpha} + \gamma \|\boldsymbol{\alpha}\|_2^2 + \eta(\boldsymbol{\alpha}^T \mathbf{1} - 1) - \delta^T \boldsymbol{\alpha}, \quad (13)$$

where the Lagrangian multipliers are $\eta \geq 0$ and $\boldsymbol{\sigma} \geq 0$. By taking the derivatives of Eq. (13) to $\boldsymbol{\alpha}$ and set it as 0, we have

$$\alpha_k = \frac{\delta_k - \eta + \mu_k}{2\gamma}, \quad \text{for } k = 1, \dots, N. \quad (14)$$

At the optimal solution, the KKT complimentary conditions should be satisfied because the QP here owns a globally optimal solution. First we consider the case of $\mu_k - \eta > 0$. We can know that $\alpha_k > 0$ because of $\sigma_k \geq 0$. According to the KKT complimentary conditions, $\sigma_k \alpha_k = 0$, so we have $\sigma_k = 0$, that is, $\boldsymbol{\sigma} = 0$. Then if $\mu_k - \eta < 0$, $\alpha_k > 0$ satisfies the nonnegative constraint of α_k . Again according to the KKT complimentary conditions, $\alpha_k = 0$. Similarly, if $\mu_k - \eta = 0$, we have $\alpha_k = 0$. As a result,

$$\alpha_k = \begin{cases} \frac{\mu_k - \eta}{2\gamma}, & \mu_k - \eta > 0, \\ 0, & \mu_k - \eta \leq 0, \end{cases} \quad (15)$$

where

$$\eta = \frac{\sum_{k=1}^N \mu_k - 2\gamma}{n}, \quad (16)$$

because of the constraint condition $\boldsymbol{\alpha}^T \mathbf{1} = 1$, and n denotes the number of nonvanishing elements in vector $\boldsymbol{\alpha}$.

Due to the connection between $\{X_{[k]}\}_{k=1}^{N-1}$, a split variable is needed extra to solve the problem, leading to exploit ADMM to optimize our inner framework.

4.3.2. Inner optimization framework (ADMM)

In this part, we introduce matrices M_k to disentangle the relationship between $\{X_{[k]}\}_{k=1}^{N-1}$, thus the objective function with a new constrained condition $M_k = X_{[k]}$ is rewritten as follow:

$$\begin{aligned} \min_{X_{[k]}, \mathcal{N}} \quad & \sum_{k=1}^{N-1} \alpha_k \|M_k\|_* + \lambda \|\mathcal{N}\|_1 \\ \text{s.t.} \quad \mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{P}_\Omega(\mathcal{Z}), \mathcal{Z} = \mathcal{X} + \mathcal{N}, M_k = X_{[k]}, \end{aligned} \quad (17)$$

whose terms are only relevant to \mathcal{X} and \mathcal{N} . Then we can obtain the unconstrained augmented Lagrange function of Eq. (17):

$$\begin{aligned} \mathcal{L}(\mathcal{X}, \mathcal{N}, \mathcal{Z}, M_k, \Delta, Q_k) = & \sum_{k=1}^{N-1} \alpha_k \|M_k\|_* + \lambda \|\mathcal{N}\|_1 + \frac{\rho}{2} \|\mathcal{Z} - \mathcal{X} - \mathcal{N}\|_F^2 + \langle \Delta, (\mathcal{Z} - \mathcal{X} - \mathcal{N}) \rangle + \sum_{k=1}^{N-1} \frac{\beta_k}{2} \|X_{[k]} - M_k\|_F^2 \\ & + \text{tr}(Q_k^T (X_{[k]} - M_k)), \end{aligned} \quad (18)$$

where ρ, β_k are the penalty factors and Δ, Q_k are the dual multipliers, and the updates of $\mathcal{X}, \mathcal{N}, \mathcal{Z}, M_k, \Delta$ and Q_k are given respectively.

Hence, we can update \mathcal{X} with fixed M_k, \mathcal{N} and \mathcal{Z} by taking the derivative of \mathcal{X} in Eq. (18):

$$\mathcal{X}^{t+1} = \frac{\rho}{\rho + \bar{\beta}} \left(\mathcal{Z}^t - \mathcal{N}^t + \frac{\Delta^t}{\rho} \right) + \frac{\bar{\beta}}{\rho + \bar{\beta}} \frac{\sum_{k=1}^{N-1} \beta_k \text{fold} \left(M_k^t - \frac{Q_k^t}{\beta_k} \right)}{\sum_{k=1}^{N-1} \beta_k}, \quad (19)$$

where $\bar{\beta}$ denotes the average value of $\{\beta_k\}_{k=1}^{N-1}$.

Similarly, updating \mathcal{N} with fixed \mathcal{X}, \mathcal{Z} is as follow:

$$\mathcal{N}^{t+1} = \arg \min \lambda \|\mathcal{N}\|_1 + \frac{\rho}{2} \|\mathcal{Z}^t - \mathcal{X}^{t+1} - \mathcal{N}\|_F^2 + \langle \Delta^t, (\mathcal{Z}^t - \mathcal{X}^{t+1} - \mathcal{N}) \rangle = \text{softthresholding} \left(\mathcal{Z}^t - \mathcal{X}^{t+1} + \frac{\Delta^t}{\rho}, \frac{\lambda}{\rho} \right), \quad (20)$$

where the *softthresholding* algorithm can be found in [35].

Then, the update of M_k with fixed \mathcal{X} is

$$M_k^{t+1} = \arg \min \alpha_k \|M_k\|_* + \frac{\beta_k}{2} \|X_{[k]}^{t+1} - M_k\|_F^2 + \text{tr} \left((Q_k^t)^T (X_{[k]}^{t+1} - M_k) \right). \quad (21)$$

And its closed-form solution is:

$$M_k^{t+1} = \mathbf{D}_{\tau_k} \left(X_{[k]}^{t+1} + \frac{Q_k^t}{\beta_k} \right) = U \text{diag}(\max(\lambda_l - \tau_k, 0)) V^T \quad (22)$$

where $\tau_k = \frac{\alpha_k}{\beta_k}$, $\mathbf{D}_{\tau_k}(X_{[k]})$ denotes the thresholding singular value decomposition (SVD) of $X_{[k]}$ [20] and $\text{diag}(\cdot)$ denotes a diagonal matrix. Specially, if the SVD of $X_{[k]} = U \Lambda V^T$, then its thresholding SVD is defined as:

$$D_{\tau_k}(X_{[k]}) = U \Lambda_{\tau_k} V^T,$$

where $\Lambda_{\tau_k} = \text{diag}(\max(\Lambda_{l-\tau_k}, 0))$.

Moreover, \mathcal{Z} should satisfy the constraint conditions $\mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{P}_\Omega(\mathcal{Z})$ and $\mathcal{Z} = \mathcal{X} + \mathcal{N}$, so it can be updated by:

$$\mathcal{Z}_{i_1 i_2 \cdots i_N}^{t+1} = \begin{cases} \mathcal{X}_{i_1 i_2 \cdots i_N}^{t+1} + \mathcal{N}_{i_1 i_2 \cdots i_N}^{t+1}, & i_1 i_2 \cdots i_N \notin \Omega, \\ \mathcal{Y}_{i_1 i_2 \cdots i_N}, & i_1 i_2 \cdots i_N \in \Omega. \end{cases} \quad (23)$$

Lastly, we update dual multipliers Δ and Q_k by gradient descent:

$$\Delta^{t+1} = \Delta^t + \rho * (\mathcal{Z}^{t+1} - \mathcal{X}^{t+1} - \mathcal{N}^{t+1}), \quad (24)$$

$$Q_k^{t+1} = Q_k^t + \beta_k * (X_{[k]}^{t+1} - M_k^{t+1}). \quad (25)$$

The pseudocode of RTC-TT algorithm is given in Algorithm 1.

Algorithm 1. Robust Tensor Completion based on Tensor-Train (RTC-TT)

Input: The observed data $\mathcal{Y} \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_n}$, index set Ω , original data \mathcal{O}

Parameters: $\lambda, \rho, \gamma, \alpha_k, \beta_k$ where $k = 1, 2, \dots, N - 1$

Initialization: $\mathcal{X}^0 = \mathcal{Z}^0$, with $\mathcal{Z}_\Omega^0 = \mathcal{Y}_\Omega$

whilenot converged

 Update α_k by Eq. (15);

 for $i = 1$ to MaxIterations

 Update \mathcal{X} by Eq. (19);

 Update \mathcal{N} by Eq. (20);

 Update M_k by Eq. (22);

 Update \mathcal{Z} by Eq. (23);

 Update Δ by Eq. (24);

 Update Q_k by Eq. (25);

 if $\frac{\|\mathcal{X} - \mathcal{O}\|_F^2}{\|\mathcal{O}\|_F^2} < \epsilon$

 break;

 end if

 if $\rho < 1 \times 10^{10}$

$\rho = 1.1 \times \rho$;

 end if

 end for

 end while

Output: The recovered tensor \mathcal{X} and \mathcal{N}

5. Experiment

In this section, we use the popular color image datasets known as *Lena*, *Peppers* and *Starfish* which can be represented by a third-order tensor ($256 \times 256 \times 3$), to verify the availability of our RTC-TT algorithm. Besides, one of the multispectral unmixing datasets, MRI image¹ ($144 \times 192 \times 144$), and video datasets, *Suzie*² ($176 \times 144 \times 150$) and *News1* ($144 \times 144 \times 144$) are also used. Meanwhile, several state-of-the-art recovery LRTC methods and RPCA algorithms are compared under different missing percentages and salt-and-pepper noise [36,37] ratios:

- LRTC: SiLRTC [2] unfolded tensor as matrices along each mode and used the truncated SVD algorithm to complete the estimating task of missing values. Further in this work, SiLRTC-TT [17] first used tensor-train rank to take place of Tucker rank based on the SiLRTC. Similarly, TMac-TT [17] was also improved from TMac [30] by using TT rank.

- RPCA: TRPCA-SNN [28] focused on the sum of the nuclear norm to tackle RPCA problems in the unfolding matrices perspective. TRPCA-TNN [16] deduced the Tensor Nuclear Norm (TNN) from t-product and then proposed tensor singular value thresholding (t-SVT), a generalization of thresholding SVD of matrices, to compute TNN approximatively. Considering the comparability between RTC and TRPCA, we modified the constrained conditions of the latter one as $\mathcal{P}_\Omega(\mathcal{Y}) = \mathcal{P}_\Omega(\mathcal{X} + \mathcal{N})$ to obtain two new models, TRPCA-SNN2 and TRPCA-TNN2, respectively. Moreover, TRPCA-TNN2 was recently proposed in [38], which is based on tubal rank.

5.1. Data processing

The salt-and-pepper noise of a certain ratio is used to produce the noisy image, i.e., a specific percentage of the image entries will be randomly picked so as to set to be 0 or 1. And based on the dirty image, we execute the missing process to obtain the observed image under a specific percentage. As for the value of missing entries, the mean of the values of known entries is adopted in initializations. We set the value of missing entries to mean value instead of zeros to improve the RPCA competitors meanwhile accelerate the recovery process.

All the experiments are simulated with respect to different missing ratios mr of various datasets, here mr is defined as

$$mr = \frac{p}{\prod_{k=1}^N d_k},$$

where p is the total number of missing entries and d_k denotes the dimension of k -th order of a tensor.

¹ https://brainweb.bic.mni.mcgill.ca/brainweb/selection_normal.html

² <http://trace.eas.asu.edu/yuv/>.

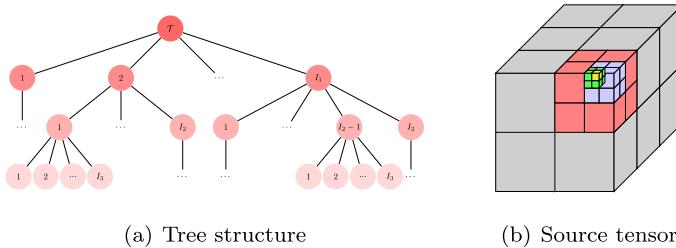


Fig. 1. Explanation of tree-KA.

5.2. Tensor augmentation

It's proved that TT rank can capture better global information on higher-order tensors. And it's a special case of Tucker rank at the level of third-order tensors. However, if we use TT rank of a third-order tensor, the local structure of its original tensor is hardly exploited under TT decomposition. On the contrary, a low-rank TT can represent the augmented tensor once the original tensor is slightly correlated [39]. Thus, higher-order tensors are needed to show the superiority of TT rank. As a result, we use Tree-KA to extend the original data magnitude and introduce a more general tree structure explanation for KA. Given a tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, what Tree-KA do is to cast it into another size of $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n}$, where $\prod_{j=1}^3 n_j = \prod_{i=1}^n I_i$. Fig. 1 gives our explanation of Tree-KA. Corresponding to Fig. 1(a) and (b) shows the different divided granularities of source tensor $\mathcal{T} \in \mathbb{R}^{16 \times 16 \times 24}$. The twelve gray parts make up the first generations of the left tree, i.e., $I_1 = 12$. Samely, the eight red parts from the gray part 2 correspond to the second generations, i.e., $I_2 = 8$. And the rest can be done for the same reason. Eventually, the source tensor is transformed to a new tensor $\mathcal{T}' \in \mathbb{R}^{12 \times 8 \times 8 \times 8}$, for instance, the yellow unit can be searched as (2, 2, 1, 2).

5.3. Color image experiment

5.3.1. Random missing

We apply RTC-TT for image recovery in the numeric experiments, in which we set different missing ratios and different degrees of noise to show the performances of various algorithms and adapt the $\lambda = 1/\sqrt{\max(n_1, n_2)n_3}$ suggested in [16]. First, we use Tree-KA to extend the image of $\mathcal{T} \in \mathbb{R}^{256 \times 256 \times 3}$ to $\hat{\mathcal{T}} \in \mathbb{R}^{4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 3}$. According to the theory in [17], the Tucker-based SiLRTC performs considerably worse when using Tree-KA because of their property of unbalance. So our model RTC-TT, RTC-TT without Tree-KA, SiLRTC-TT, TMac-TT use the TT rank in experiments, while Tucker rank for the others, because they are mainly based on third-order tensors. The TRPCA-TNN and TRPCA-SNN can hardly recover the low-rank components when missing ratio $mr \geq 0.6$, they treat the missing entries as sparse component. As shown in Fig. 2, the results of them are not pleasant when $mr = 0.6$ and noise ratio $nr = 0.1$. So we perform the simulation for the *Lena*, *Peppers* and *Starfish* images where missing entries of the images are chosen according to a random distribution, the missing ratio mr varies from 0.1 to 0.6 and nr from 0.05 to 0.3. For every experiment, we set the TRPCA-TNN with the best $\lambda = 1/\sqrt{\max(n_1, n_2)n_3}$. And we apply $\lambda_1 = \sqrt{\max(n_1, n_2)n_3}/3$, $\lambda_2 = \sqrt{\max(n_2, n_1)n_3}/3$ and $\lambda_3 = \sqrt{\max(n_3, n_1n_2)/3}$ which make TRPCA-SNN perform well in most cases. For the LRTC algorithms, we use the same index set Ω of observations.

In Fig. 2, performance of the algorithms on completing the *Lena*, *Peppers*, *Starfish* images is shown. Both low-rank component \mathcal{X} and sparse component \mathcal{N} are well separated under the condition of the $mr = 0.4$ and $nr = 0.1$. However, the LRTC algorithms can not wipe out the noise permanently once there exists any noise in Ω . That is because the elements in set Ω will never change. Besides, the two modified model of TRPCA perform better than their raw ones, which also verify the effectiveness of our improvement. Even though the SSIM values of RTC-TT and TRPCA-TNN2 are close, the details recovered by the latter are inferior to RTC-TT, which can be obviously distinguished in the labeled boxes. What's more, the separated sparse components in the lower row can also reflect the denoising ability under this certain circumstance. The ones separated by RTC-TT method hardly contain the content of the original image, on the contrary, the skeleton of the image is captured by other methods, which is not a clean job exactly.

For color images, we adopt Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) as the evaluation metric. Fig. 3 shows the variation trends of PSNR values of each algorithm. As we can see that in Fig. 3(a), RTC-TT method holds superior PSNR values of different missing ratios. TRPCA-TNN2 and TRPCA-SNN2 are second and third to RTC-TT. At a low missing level, TRPCA-TNN performs well. Because at that point, the missing values are regarded as noise by TRPCA-TNN, and the low missing ratio corresponds to the sparsity, which satisfied the assumptions of RPCA problem properly. However, once the missing ratio rises, the assumption of sparsity is broken. In the meantime, a conflict between high missing ratio and the functionality of ℓ_0 -norm occurs, causing an unpleasant recovery result. LRTC methods invariably stay at a low PSNR level due to their insolvability to outliers. TRPCA-TNN2 performs better than RTC-TT when $nr > 0.15$ as well as $mr = 0.4$ in Fig. 3(b), which shows their extraordinary solving ability to noisy images. The reason why TMac-TT does not

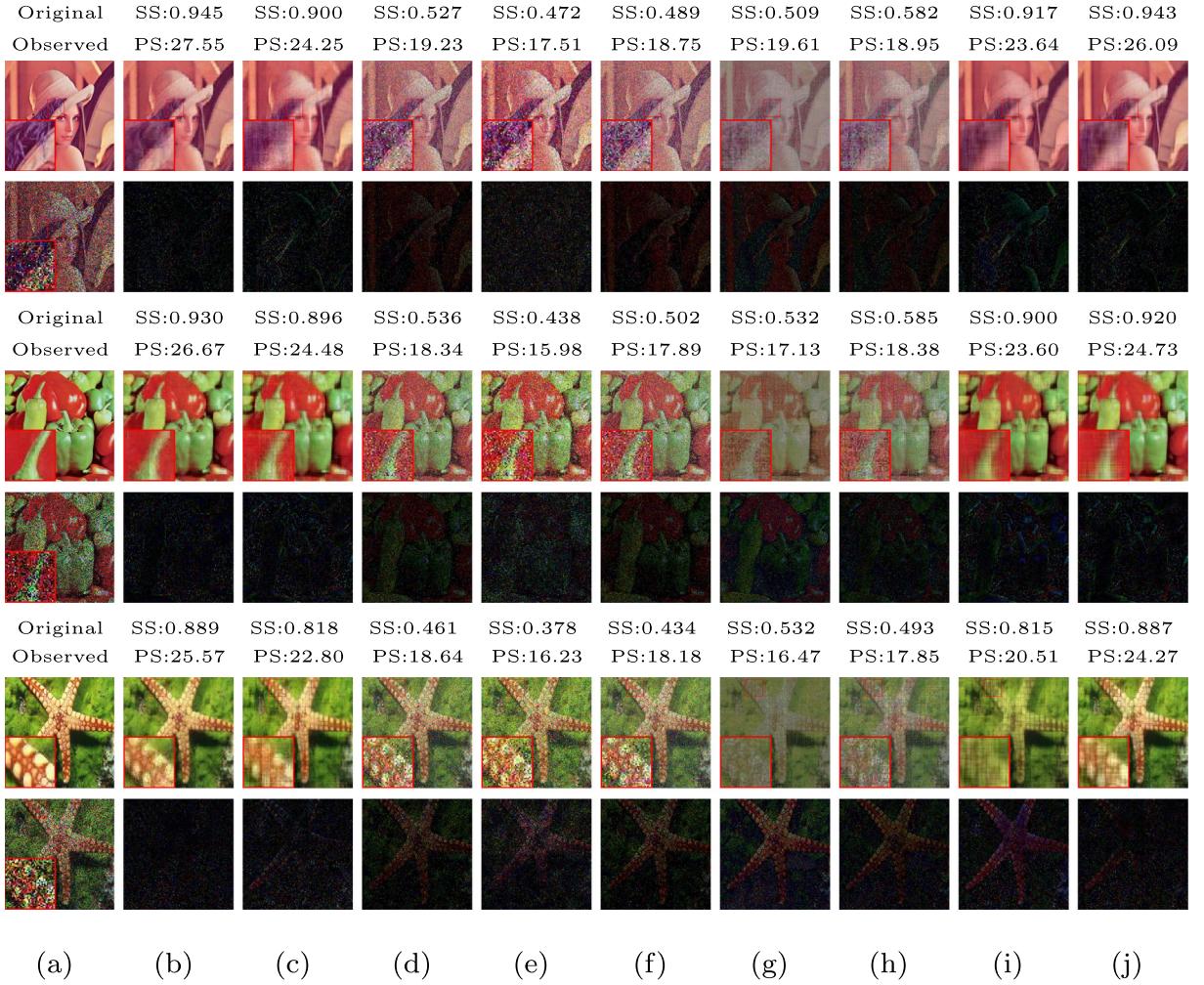


Fig. 2. Recover the Lena, Peppers, Starfish images with 40% of the missing entries and 10% of the corrupted entries using different algorithms. SS denotes SSIM and PS denotes PSNR. (a) From top to bottom: Original image and Observed image; (b)–(j) From left to right: recovered images by RTC-TT, RTC-TT without Tree-KA, SILRTC-TT, TMac-TT, SILRTC, TRPCA-SNN, TRPCA-TNN, TRPCA-SNN2 and TRPCA-TNN2. The low-rank component \mathcal{X} 's are shown in upper row by different algorithms. And the second-row images are the sparse component \mathcal{N} 's.

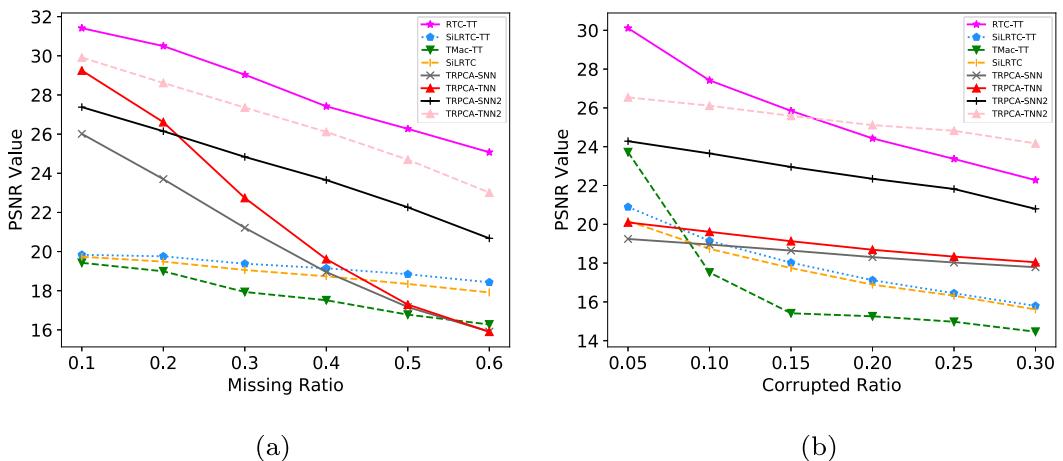


Fig. 3. The PSNR values of all algorithms on Lena image. (a) Under the condition of $nr = 0.1$, the PSNR values vary by the growth of missing ratio; (b) Under the condition of $mr = 0.4$, the PSNR values vary by the growth of noise ratio.

get a pleasant result is that its core is matrix factorization which is also extremely sensitive to outliers. We also note that when $mr < 0.3$ and $nr = 0.1$, the TRPCA-TNN and TRPCA-SNN hold a better performance in most cases. At this time, the missing entries are regarded as sparse noise. So we choose the missing percentage of 40% to compare the performance of all algorithms under different noisy ratios in Fig. 3(b).

The effects of Tree-KA are shown in Fig. 2 compared with RTC-TT without Tree-KA. It can be seen easily that the performance of the method without Tree-KA is inferior to the one using Tree-KA. That is because Tree-KA is able to capture the latent structure information of TT rank in low-order tensors. According to the format of the TT rank, the size of the matrix flattened along the third mode is (196608×1) , which hardly owns structure. And what Tree-KA does is to break

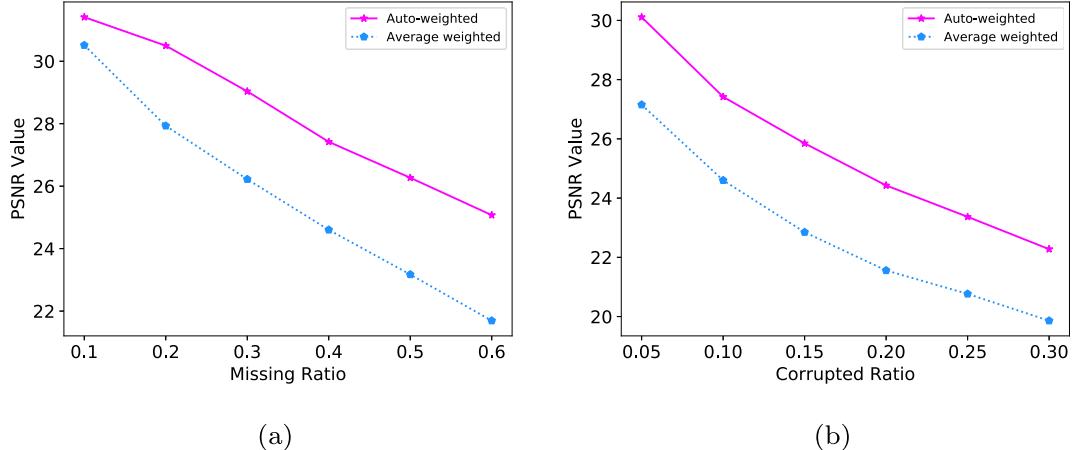


Fig. 4. The average PSNR values of two setting approaches of weight on the three color images. (a) The PSNR values vary by the growth of missing ratio at $mr = 0.1$; (b) The PSNR values vary by the growth of noise ratio at $nr = 0.1$.

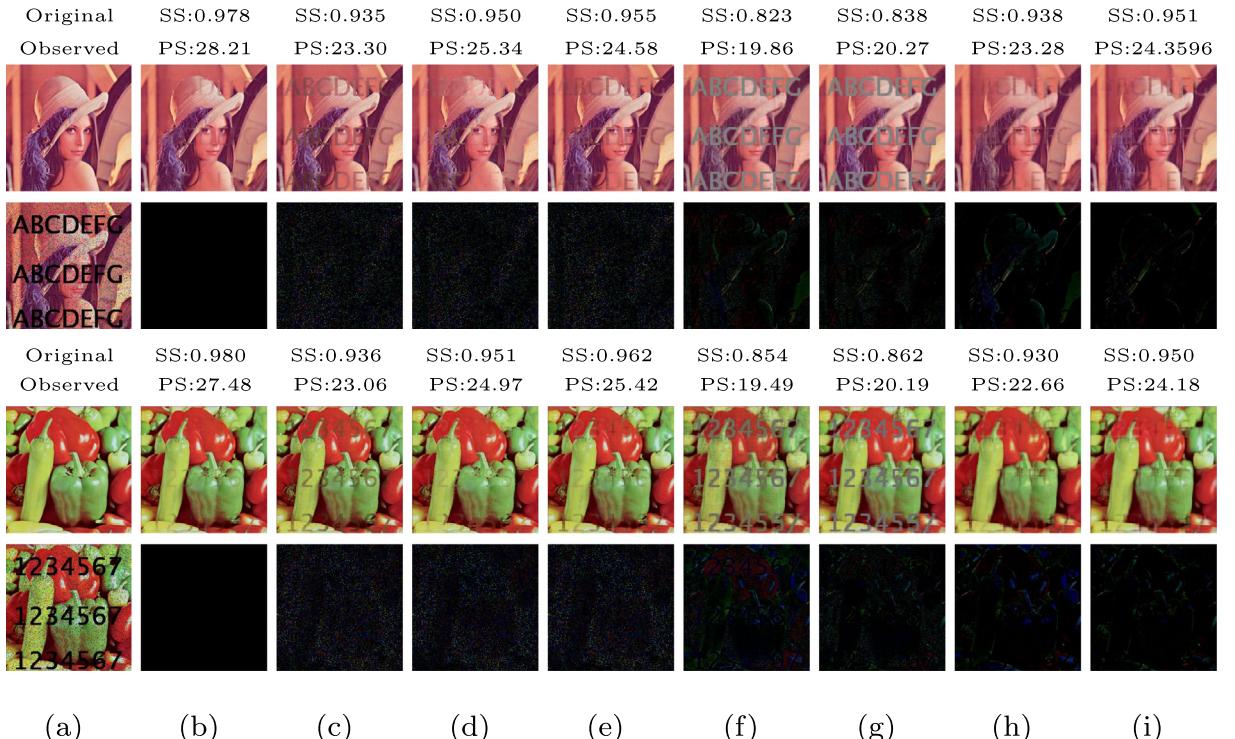


Fig. 5. Recover the *Lena*, *Peppers* images with structural missing entries and 10% of the noisy entries using different algorithms. SS denotes SSIM and PSNR denotes PSNR. (a) From top to bottom: Original image and Observed image; (b)–(i) From left to right: recovered images by RTC-TT, SiRTC-TT, TMac-TT, SiLRTC, TRPCA-SNN, TRPCA-TNN, TRPCA-SNN2 and TRPCA-TNN2. The low-rank component \mathcal{X} 's are shown in upper row by different algorithms. And the second-row images are the sparse component \mathcal{N} 's.

the dimensional limitation of TT rank. Additionally, we compared our auto-weighted method against the average weight based on the three color images in various circumstances. Both Fig. 4(a) and (b) show the superiority of the auto-weighted method, which verifies the effectiveness of the auto-weighted mechanism.

5.3.2. Structural missing

Fig. 5 shows the recovered performance on color images for the structural missing entries. In this case, the proposed method produces a much better visual effect against the others. No matter the global noise or local features, RTC-TT performs the most similar result to the original image. By taking the mean value as the missing entries in RPCA methods for a relatively fair comparison, both TRPCA-SNN and TRPCA-TNN can hardly erase the text from observations. However, their modified versions, i.e., TRPCA-SNN2 and TRPCA-TNN2, are robust to RTC problems and obtain better results.

5.4. MRI image experiment

In multispectral image recovery, we also benchmark the algorithms introduced above on MRI image. We do the same initializations on the original image as color images do, and the same setting approaches of λ and other parameters of each algorithms. The only difference is that the parameters in TRPCA-SNN are set to be $[\lambda_1 \ \lambda_2 \ \lambda_3] = [30 \ 35 \ 30]$, which can hold a better performance.

Using Tree-KA to make best use of the potential of TT rank, we extend the low-order multispectral image of $P \in \mathbb{R}^{144 \times 192 \times 144}$ to a higher-order one of $\hat{P} \in \mathbb{R}^{4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 3 \times 3 \times 3 \times 3}$, despite of low-diamention structure. And this high-order tensor is directly used for the robust tensor completion experiments. PSNR is again used to evaluate the quality of recovery.

In Fig. 6, the recovered results of MRI image are given. The proposed method holds a relatively close PSNR value with TRPCA-TNN2 under $mr = 0.5$ and $nr = 0.1$. What's more, it's difficult to recognize the superior one by ocular estimate, too. However, the results of each band of MRI image are shown by PSNR values in Fig. 7(a). It can be seen that the proposed

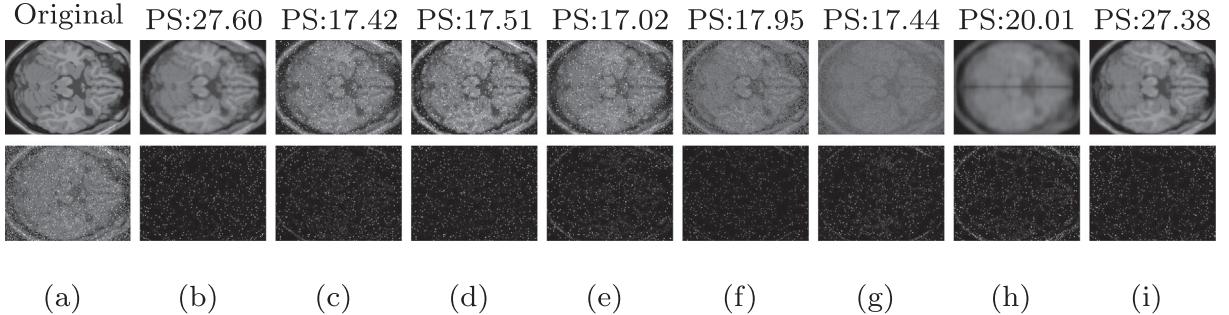


Fig. 6. Recovered results of MRI image with 50% of the missing entries and 10% of the corrupted entries using different algorithms. PS denotes PSNR. (a) From top to bottom: Original image and Observed image; (b)–(i) From left to right: recovered images by RTC-TT, SiLRTC-TT, TMac-TT, SiLRTC, TRPCA-SNN, TRPCA-TNN, TRPCA-SNN2 and TRPCA-TNN2. The low-rank component \mathcal{X} 's are shown in upper row by different algorithms. And the second-row images are the sparse component \mathcal{N} 's.

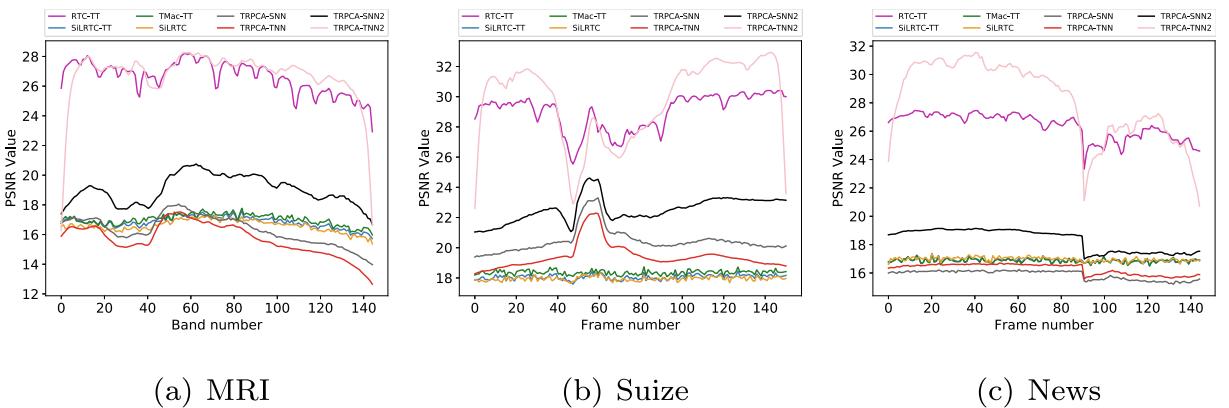


Fig. 7. Under the condition of $nr = 0.1$ and $mr = 0.5$, the PSNR values of all algorithms.

method performs better than TRPCA-TNN2 at both ends where the neighbor bands are lacking. It implies that TRPCA-TNN2 needs more reference information than RTC-TT.

5.5. Video experiment

We also apply our RTC-TT algorithms to video datasets. Similarly, using Tree-KA to reshape the original video *Suzie* of $\mathcal{T} \in \mathbb{R}^{176 \times 144 \times 150}$ to a higher-order one of $\hat{\mathcal{T}} \in \mathbb{R}^{4 \times 4 \times 5 \times 4 \times 4 \times 5 \times 11 \times 9 \times 6}$ and *News* of $\mathcal{Y} \in \mathbb{R}^{144 \times 144 \times 144}$ to $\hat{\mathcal{Y}} \in \mathbb{R}^{4 \times 4 \times 4 \times 4 \times 4 \times 3 \times 3 \times 3 \times 3 \times 3}$. Rather than performing the algorithms on each frame, we perform our benchmarks on the entire videos.

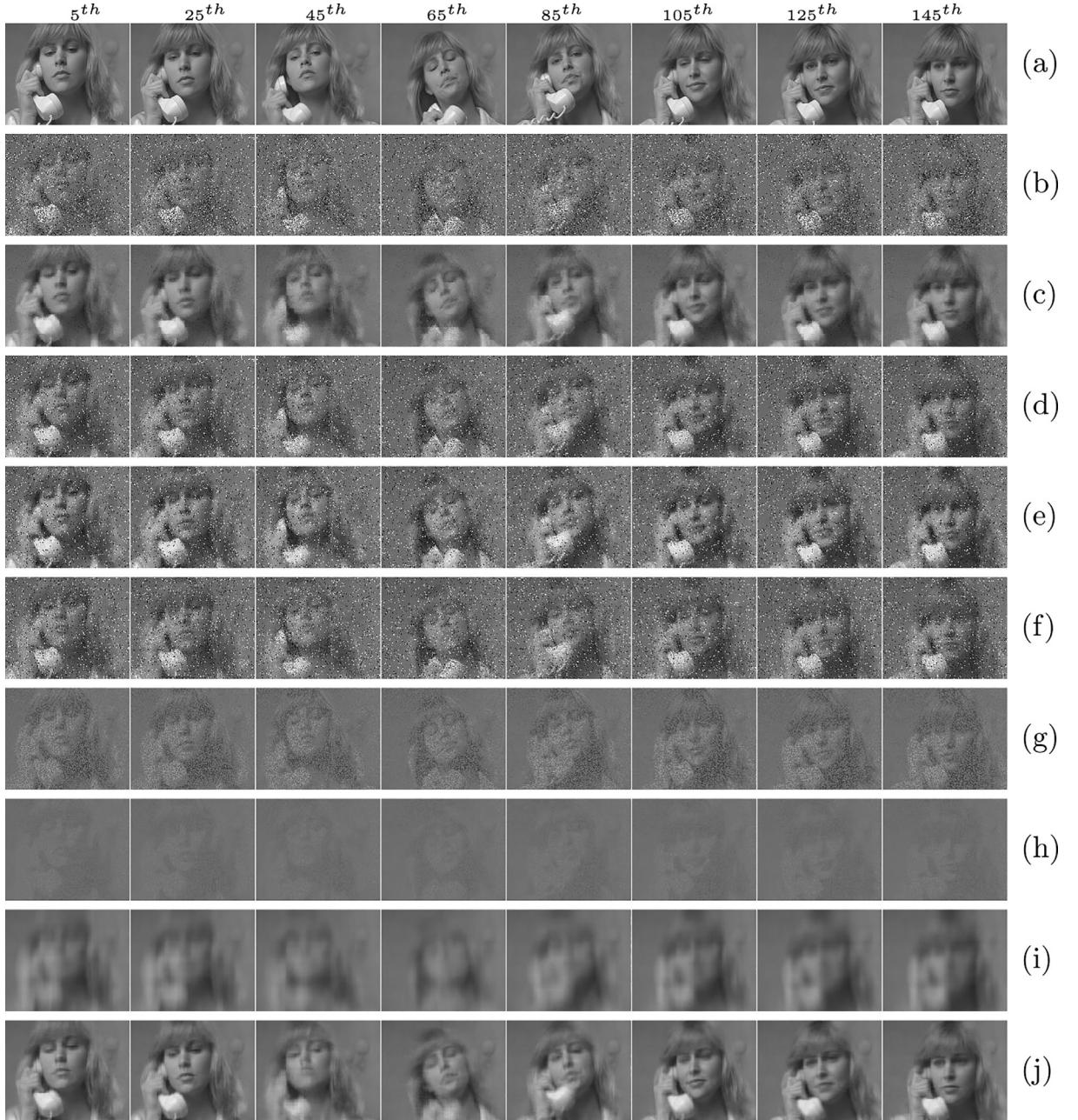


Fig. 8. The 5th, 25th, 45th, 65th, 85th, 105th, 125th and 145th frames in *Suzie* video recovery of all algorithms with $mr = 0.5$ and $nr = 0.1$. (a) Original video; (b) Observed video; (c) RTC-TT; (d) SiLRTC-TT; (e) TMac-TT; (f) SiLRTC; (g) TRPCA-SNN; (h) TRPCA-TNN; (i) TRPCA-SNN2; (j) TRPCA-TNN2.

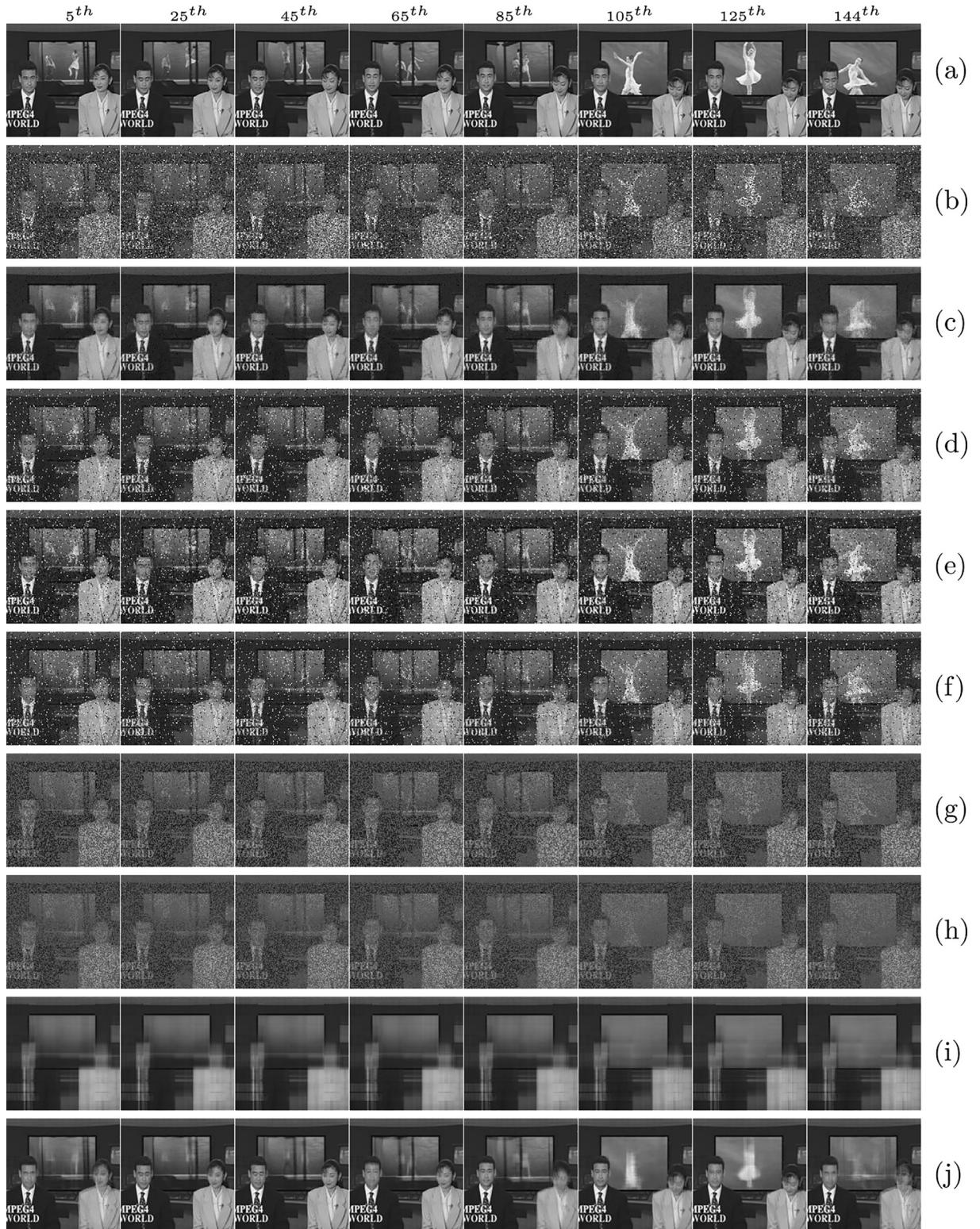


Fig. 9. The 5th, 25th, 45th, 65th, 85th, 105th, 125th and 144th frames in *News* video recovery of all algorithms with $mr = 0.5$ and $nr = 0.1$. (a) Original video; (b) Observed video; (c) RTC-TT; (d) SiLRTC-TT; (e) TMac-TT; (f) SiLRTC; (g) TRPCA-SNN; (h) TRPCA-TNN; (i) TRPCA-SNN2; (j) TRPCA-TNN2.

Figs. 8 and 9 show several frames uniformly selected from recovery videos. For TRPCA-SNN, $[\lambda_1 \ \lambda_2 \ \lambda_3] = [30 \ 30 \ 30]$ is empirically set to improve its performance. For the others, we continue using the same setting approaches of parameters. It can be seen that basic TRPCA algorithms are both almost incomprehensible and LRTC algorithms can not denoise the video effectively under the case of $mr = 0.5$, $nr = 0.1$. While our RTC-TT method and TRPCA-TNN2 outperform all algorithms for a clean and elegant job. Actually as shown in Fig. 7(b) and (c), each of the two have its own advantages. RTC-TT can hold a better performance in the place of great fluctuation or both ends of the videos. By way of contrast, TRPCA-TNN2 is more suitable for the mild changing datasets. Recently, [38] claimed that compared with the traditional multilinear algebraic setup, the t-SVD algebraic framework is more suitable for natural image processing, which is corresponding to our results. Even though, our model works better than all the rest, except TRPCA-TNN2. Between frame 40th and 60th in video *Suzie*, the proposed method shows its superiority to TRPCA-TNN2 when the video varies tempestuously. That is the part that *Suzie* flipped her hair when calling, in contrast to the other frames that she nearly stand still all the time. In a way, our method does not rely as much on the information of the front or back frames as the other one, which implies that RTC-TT will perform well in a low-frame video. And as shown in color image experiments, the three-channel images explained what the low-frame video is exactly like. What happened in video *News* similarly is that the part of the 91st frame the girl in the background is amplified suddenly, which also implies RTC-TT's relative independence to neighbors.

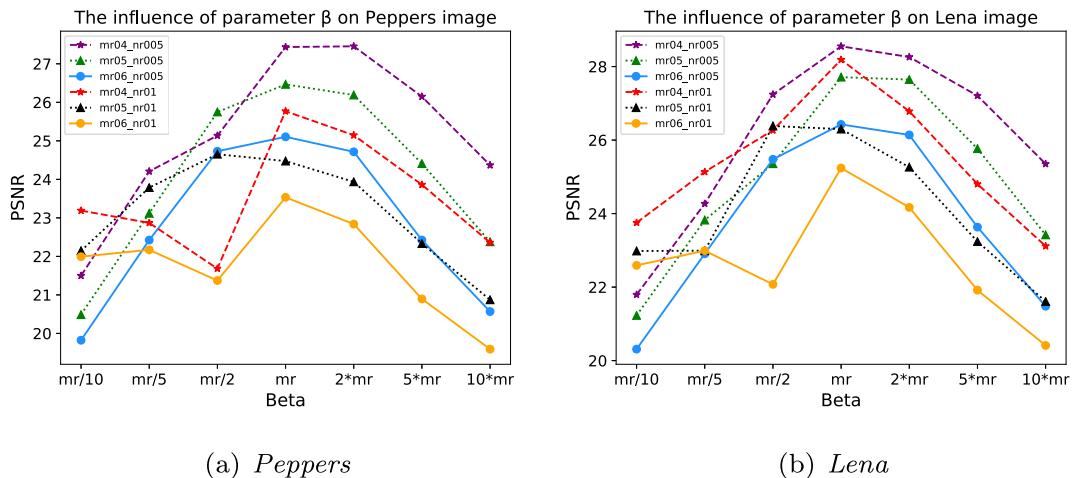


Fig. 10. The PSNR values of our methods with different β settings on color image *Lena* and *Peppers*. The legend 'mr04_nr005' means the result is obtained at $mr = 0.4$, $nr = 0.05$, and the same to the rest. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

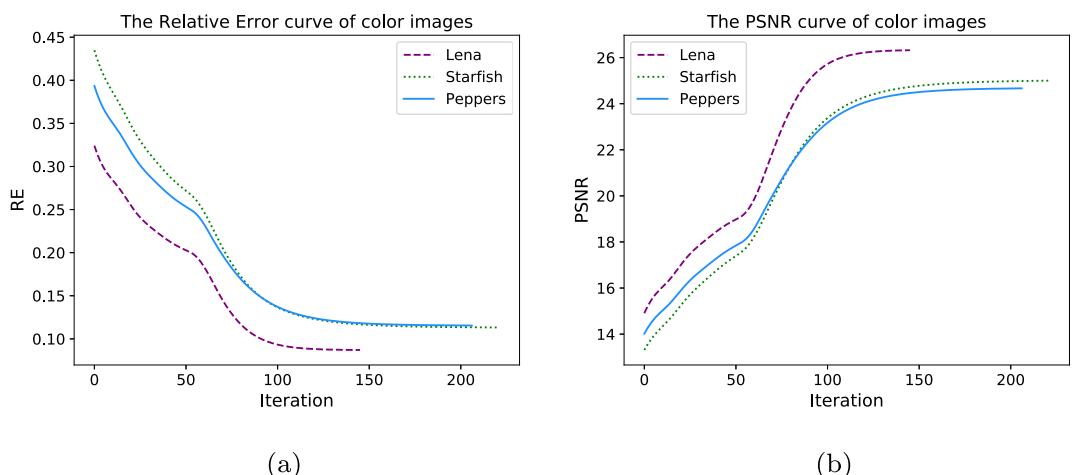


Fig. 11. The convergence behavior of the proposed algorithm on RE and PSNR values at $mr = 0.5$, $nr = 0.05$.

5.6. Parameters analysis

In the experiments, all the elements of vector β share the same value, i.e., $\beta_1 = \beta_2 = \dots = \beta_{N-1}$. The tradeoff between parameter $\beta_k (k = 1, 2, \dots)$ and ρ in Eq. (19) determine the final recovering result of the proposed model. To gain better performance, β_k and ρ are expected to match the missing ratio and noise ratio in different scenarios, respectively. In other words, β_k deserves a larger value at a high missing ratio circumstance. Thus, we consider that the value of β_k should be the multiple of missing ratio. Based on that, we construct a small parameter experiment on two color images *Lena* and *Peppers* to select the most appropriate β_k . Fig. 10 shows that the proposed model can perform well in most situations when $\beta_k = mr \times 10^{-3}$. Besides, the value of ρ is assigned as $\rho = \beta_k \times \frac{nr}{mr}$.

5.7. Convergency analysis

The iterations of our algorithm will be stopped when the relative change of the variables, i.e., $\frac{\|\mathcal{X}^k - \mathcal{X}^{k-1}\|_F}{\|\mathcal{T}\|_F}$, is smaller than 10^{-5} . Fig. 11(a) and (b) show the variation tendency of Relative Error (RE) values, i.e., $RE = \frac{\|\mathcal{X} - \mathcal{T}\|_F}{\|\mathcal{T}\|_F}$, and PSNR values on three color images, *Lena*, *Peppers*, and *Starfish*, respectively. The overall downward trend of all the curves in Fig. 11 illustrates that the proposed algorithm converges to an optimal point.

6. Conclusion and future works

In this paper, to decompose a given tensor with partial observations into a low-rank component and a sparse component, we have proposed a mixture model for LRTC and RPCA. Due to the unbalanced property of Tucker rank, we use TT rank to capture more global information. Moreover, an auto-weighted mechanism is appended into the composite model to maximize the structure potential of TT rank instead of average weight. The superiority of the auto-weighted mechanism is verified in numerical experiments. Compared with TRPCA-TNN2, the proposed method performs better under the low noise level circumstance. Based on several datasets, experimental results also show the effectiveness of our algorithm.

In future work, it would be of great interest to leverage the prior information of the image data to remove the block-artifacts shown in recovered color images.

CRediT authorship contribution statement

Chuan Chen: Conceptualization, Methodology, Writing - original draft, Writing - review & editing. **Zhe-Bin Wu:** Methodology, Software, Writing - original draft, Writing - review & editing. **Zi-Tai Chen:** Methodology, Writing - review & editing, Formal analysis. **Zi-Bin Zheng:** Supervision, Funding acquisition. **Xiong-Jun Zhang:** Methodology, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The work described in this paper was supported by the Key-Area Research and Development Program of Guangdong Province (2018B010109001), the National Natural Science Foundation of China (11801595), the Guangdong Basic and Applied Basic Research Foundation (2019A1515011043), the Natural Science Foundation of Guangdong (2018A030310076) and the Tencent Wechat Rhino-bird project No. 2021321.

References

- [1] X. Zhang, A nonconvex relaxation approach to low-rank tensor completion, *IEEE Trans. Neural Netw. Learn. Syst.* 30 (6) (2019) 1659–1671.
- [2] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, in: 2009 IEEE 12th International Conference on Computer Vision, 2009, pp. 2114–2121, <https://doi.org/10.1109/ICCV.2009.5459463>.
- [3] W.K. Wong, Z. Lai, Y. Xu, J. Wen, C.P. Ho, Joint tensor feature analysis for visual object recognition, *IEEE Trans. Cybern.* 45 (11) (2014) 2425–2436.
- [4] M. Signoretto, Q.T. Dinh, L.D. Lathauwer, J.A.K. Suykens, Learning with tensors: a framework based on convex optimization and spectral regularization, *Mach. Learn.* 94 (3) (2014) 303–351.
- [5] Y. Li, C. Chen, F. Ye, Z. Zheng, G. Ling, A semi-supervised classification approach for multiple time-varying networks with total variation, in: International Conference on Database Systems for Advanced Applications, Springer, 2019, pp. 334–337.
- [6] P. Symeonidis, ClustHOSVD: item recommendation by combining semantically enhanced tag clustering with tensor HOSVD, *IEEE Trans. Man Cybern. Syst.* 46 (9) (2016) 1240–1251.
- [7] M.K. Ng, Q. Yuan, Y. Li, S. Jing, 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.

- [8] 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.
- [9] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis?, *J ACM* 58 (3) (2011) 11.
- [10] B. Huang, C. Mu, D. Goldfarb, J. Wright, Provable models for robust low-rank tensor completion, *Pac. J. Optim.* 11 (2015) 339–364.
- [11] A. Wang, Z. Lai, Z. Jin, Noisy low-tubal-rank tensor completion, *Neurocomputing* 330 (2019) 267–279.
- [12] Z. Lai, Y. Xu, J. Yang, J. Tang, D. Zhang, Sparse tensor discriminant analysis, *IEEE Trans. Image Process.* 22 (10) (2013) 3904–3915.
- [13] J.D. Carroll, Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition, *Psychometrika* 35 (3) (1970) 283–319.
- [14] L. Tucker, Some mathematical notes on three-mode factor analysis, *Psychometrika* 31 (3) (1966) 279–311.
- [15] I. Oseledets, Tensor-train decomposition, *SIAM J. Sci. Comput.* 33 (5) (2011) 2295–2317, <https://doi.org/10.1137/090752286>.
- [16] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, S. Yan, Tensor robust principal component analysis: exact recovery of corrupted low-rank tensors via convex optimization, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2016 (2016) 5249–5257, <https://doi.org/10.1109/CVPR.2016.567>.
- [17] J.A. Bengua, H.N. Phien, H.D. Tuan, M.N. Do, Efficient tensor completion for color image and video recovery: llow-rank tensor train, *IEEE Trans. Image Process.* 26 (5) (2017) 2466–2479, <https://doi.org/10.1109/TIP.2017.2672439>.
- [18] C. Chen, M.K. Ng, X.-L. Zhao, Alternating direction method of multipliers for nonlinear image restoration problems, *IEEE Trans. Image Process.* 24 (1) (2014) 33–43.
- [19] B. Romera-Paredes, M. Pontil, A new convex relaxation for tensor completion, *Proc. Adv. Neural Inf. Process. Syst.* (2013) 2967–2975.
- [20] J.F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, *Siam J. Optim.* 20 (4) (2008) 1956–1982.
- [21] P. Tseng, Convergence of a block coordinate descent method for nondifferentiable minimization, *J. Optim. Theory Appl.* 109 (3) (2001) 475–494.
- [22] P. Sutov, Jr., T. Goldstein, The alternating direction method of multipliers (2015)..
- [23] J.I. Latorre, Image compression and entanglement, *arXiv preprint quant-ph/0510031* (2005).
- [24] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, *SIAM Rev.* 51 (3) (2009) 455–500.
- [25] C. Lu, J. Feng, W. Liu, Z. Lin, S. Yan, et al, Tensor robust principal component analysis with a new tensor nuclear norm, *IEEE Trans. Pattern Anal. Mach. Intell.* (2019).
- [26] M.E. Kilmer, C.D. Martin, Factorization strategies for third-order tensors, *Linear Algebra Appl.* 435 (3) (2011) 641–658.
- [27] C. Lu, P. Zhou, Exact recovery of tensor robust principal component analysis under linear transforms, *arXiv preprint arXiv:1907.08288* (2019)..
- [28] D. Goldfarb, Z. Qin, Robust low-rank tensor recovery: models and algorithms, *SIAM J. Matrix Anal. Appl.* 35 (1) (2014) 225–253.
- [29] M.E. Kilmer, C.D. Martin, Factorization strategies for third-order tensors, *Linear Algebra Appl.* 435 (3) (2011) 641–658, Special Issue: Dedication to Pete Stewart on the occasion of his 70th birthday. doi: 10.1016/j.laa.2010.09.020. <http://www.sciencedirect.com/science/article/pii/S0024379510004830..>
- [30] Y. Xu, R. Hao, W. Yin, Z. Su, Parallel matrix factorization for low-rank tensor completion, *Inverse Probl. Imag.* 9 (2) (2015) 601–624.
- [31] 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.* S0020025516318734.
- [32] Z. Zhang, G. Ely, S. Aeron, N. Hao, M. Kilmer, Novel methods for multilinear data completion and de-noising based on tensor-svd, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3842–3849.
- [33] M.E. Kilmer, K. Braman, N. Hao, R.C. Hoover, Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging, *SIAM J. Matrix Anal. Appl.* 34 (1) (2013) 148–172.
- [34] C. Chen, H. Qian, W. Chen, Z. Zheng, H. Zhu, Auto-weighted multi-view constrained spectral clustering, *Neurocomputing* 366 (2019) 1–11.
- [35] D.L. Donoho, De-noising by soft-thresholding, *IEEE Trans. Inf. Theory* 41 (3) (1995) 613–627.
- [36] R.H. Chan, C.-W. Ho, M. Nikolova, Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization, *IEEE Trans. Image Process.* 14 (10) (2005) 1479–1485.
- [37] I. Djurović, Bm3d filter in salt-and-pepper noise removal, *EURASIP J. Image Video Process.* 2016 (1) (2016) 13.
- [38] Q. Jiang, M. Ng, Robust low-tubal-rank tensor completion via convex optimization, in: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization*, 2019, pp. 2649–2655. doi:10.24963/ijcai.2019/368. <https://doi.org/10.24963/ijcai.2019/368>.
- [39] J.A. Bengua, H.D. Tuan, H.N. Phien, M.N. Do, Concatenated image completion via tensor augmentation and completion, in: *2016 10th International Conference on Signal Processing and Communication Systems (ICSPCS)*, 2016, pp. 1–7, doi:10.1109/ICSPCS.2016.7843326.