Patched-tube unitary transform for robust tensor completion

Michael K. Ng [a,1], Xiongjun Zhang [b,2,*], Xi-Le Zhao [c,3]

[a] *Department of Mathematics, The University of Hong Kong, Hong Kong*
[b] *School of Mathematics and Statistics and Hubei Key Laboratory of Mathematical Sciences, Central China Normal University, Wuhan 430079, China*
[c] *School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*

ARTICLE INFO

ABSTRACT

The aim of the robust tensor completion problem for third-order tensors is to recover a low-rank tensor from incomplete and/or corrupted observations. In this paper, we develop a patched-tubes unitary transform method for robust tensor completion. The proposed method is to extract similar patched-tubes to form a third-order sub-tensor, and then a transformed tensor singular value decomposition is employed to recover such low-rank incomplete and/or corrupted sub-tensor. Here the unitary transform matrix for transformed tensor singular value decomposition is constructed by using left singular vectors of the unfolding matrix arising from such sub-tensor. Moreover, we establish the perturbation results of the transformed tensor singular value decomposition for patched-tubes tensor completion. Extensive numerical experiments on hyperspectral, video and face data sets are presented to demonstrate the superior performance of the proposed patched-tubes unitary transform method over testing state-of-the-art robust tensor completion methods.

1. Introduction

Robust tensor completion has become a rapidly growing field recently in signal and image processing [1,2], remote sensing [3,4], computer vision [5], and machine learning [6], etc. Since high-dimensional data is always embedded in a low dimensional subspace, low-rank tensors have been used to exploit the low-dimensional structure information of high dimensional data successfully. In this paper, we mainly focus on the robust tensor completion problem for third-order tensors, which is used to recover a tensor from partial observations with/without sparse noise.

Tensors are multidimensional arrays, which are high-order generalizations of vectors and matrices and have been applied in many fields such as hyperspectral image processing [7] and feature representation [8]. The tensors of order three or higher are called higher-order tensors [9]. In particular, when the order of the tensor is two, robust tensor completion reduces to robust matrix completion, which has received a considerable amount of attention in both theoretical and applied areas, see [10,11] and references therein. Recent effort has been made to extend robust matrix completion to robust tensor completion, where the main issue is the rank of a tensor. However, there is a great difference between the ranks of matrices and tensors. Several definitions of tensor ranks have been proposed, such as CANDECOMP/PARAFAC (CP) rank [12], Tucker rank [13], tensor train (TT) rank [14], and tubal rank [15]. The CP rank is defined as the smallest number of rank one tensor decomposition, where the computation is generally NP-hard [16] and its convex relaxation is intractable. For the Tucker rank minimization, Liu et al. [17] first proposed the sum of nuclear norms of unfolding matrices of a tensor (SNN) to approximate the sum of Tucker rank of a tensor for tensor completion. However, Romera-Paredes et al. [18] showed that the SNN is not the convex envelope of the sum of ranks of unfolding matrices of a tensor and the SNN method is substantially suboptimal. Mu et al. [19] proposed a square deal method for fourth or higher-order tensors, while the square deal method may not get a high accuracy solution and has the similar limitations of SNN since it just utilizes the nuclear norm of one unfolding matrix of a tensor for third-order tensors. Afterwards, based on TT rank, Bengua et al. [20] proposed a TT method for color images and video recovery without noise. However, the TT method is just suitable to the tensor with the third-dimension being three and also needs to reshape the original

tensor into a higher-order tensor, which may destroy the structure of the original tensor.

Recently, Kilmer et al. [15] proposed the tubal rank of a third-order tensor, which is based on tensor-tensor product (t-product) and its algebraic framework [21], where the t-product allows tensor factorizations. Moreover, the t-product and its algebra operations by using Fourier transform for third-order tensors are studied and analyzed, which lead to other matrix factorizations to be extended to tensors [22]. With this algebra framework, the tensor singular value decomposition (t-SVD) is proposed for third-order tensors [15] and higher-order tensors [22], which is analogue of the SVD of a matrix. By using t-SVD, Kilmer et al. [15] defined the tubal rank and multi-rank of a tensor. Moreover, the t-product and t-SVD are generalized to any inverse transform in [23]. By making use of t-product and t-SVD, low-rank tensor recovery and related problems are studied in image processing, see [24] and references therein.

Based on t-product in the Fourier domain, robust tensor completion [25] and tensor robust principal component analysis [26] have been proposed and studied for third-order tensors. For example, by using t-SVD, Zhang et al. [27] proposed a tensor nuclear norm based on t-product for tensor completion and showed that one can guarantee exact recovery with high probability by minimizing a convex surrogate to the tubal rank, and Lu et al. [26] proposed a tensor nuclear norm based on t-SVD by using Fourier transform for tensor robust principal component analysis and provided the theoretical guarantee for exact recovery with low tubal rank and sparse components. But using Fourier transform along each tube may not get a low tubal rank tensor due to the limitations of Fourier transform. Recently, Song et al. [28] studied the unitary transformed t-SVD for robust tensor completion, where they used any unitary matrix instead of using Fourier transform for t-product and t-SVD. Moreover, they showed that one can recover a low transformed tubal rank tensor exactly with overwhelming probability provided that its transformed tubal rank is sufficiently small and its corrupted entries are reasonably sparse. Numerical examples have demonstrated that the use of the other unitary transform matrices for robust tensor completion can be better than that of the Fourier transform matrix. However, the robust tensor completion method by using t-SVD with any unitary transform just uses the global low rank property of a tensor and is not sufficient enough to exploit the structure of a tensor.

In recent years, the nonlocal self-similarity, which exploits the spatial redundancy of nonlocal similar patches, has emerged as one of the most successful regularization priors for natural images [29]. It has been applied in image processing and sparse representation [7,30] successfully, see also [31,32] and references therein. In this paper, by jointly exploring global low-rankness and nonlocal similarity of a tensor, we propose a patched-tubes unitary transform method for robust tensor completion, where the patched-tubes mean that we search similar cubes along the third-dimension. By extracting similar cubes of the tensor and grouping them into a sub-tensor, the singular values generated by the patched-tubes unitary transform may be much smaller than those generated by the unitary transformed t-SVD method. Then a transformed t-SVD is employed to recover such incomplete and/or corrupted sub-tensor. Here the unitary transform matrix for transformed t-SVD is constructed by using left singular vectors of the unfolding matrix arising from such sub-tensor. Moreover, we show that the resulting sub-tensor can be regarded as a low tubal rank tensor with a small perturbation. Numerical experiments will conform the advantage of the patched-tubes unitary transform method by exploiting global low-rankness and nonlocal self-similarity of a tensor in robust tensor completion.

The contributions of this paper are summarized as follows: (i) We develop a patched-tubes unitary transform method for robust tensor completion. The proposed method exploits the global low-rankness and non-local self-similarity of a tensor based on the transformed t-SVD. (ii) We establish the perturbation results of the transformed t-SVD in each sub-tensor for patched-tubes tensor completion. (iii) Extensive numerical experiments on hyperspectral, video and face data sets are presented to demonstrate the superior performance of the proposed patched-tubes unitary transform method over several state-of-the-art robust tensor completion methods.

The remainder of this paper is organized as follows. In Section 2, some preliminaries of transformed t-SVD are reviewed. In Section 3, we propose the patched-tubes unitary transform method for robust tensor completion, where the perturbation results of the transformed t-SVD for patched-tubes tensor completion are established. Moreover, a symmetric Gauss-Seidel based multi-block alternating direction method of multipliers (sGS-ADMM) is developed to solve our proposed model. In Section 4, some numerical experiments are presented to demonstrate the effectiveness of the patched-tubes unitary transform method. Finally, we conclude the paper in Section 5.

2. Tensor singular value decomposition

2.1. Preliminaries

Let us start with notations and definitions of tensors used throughout this paper. The lowercase and boldface lowercase letters represent scalars and vectors, respectively, e.g., a and $\mathbf{a}$. Matrices are denoted by boldface capital letters, e.g., $\mathbf{A}$. Tensors are denoted by calligraphy letters, e.g., $\mathcal{A}$. The spaces with real and complex entries are denoted by $\mathbb{R}$ and $\mathbb{C}$, respectively. A tensor is a multidimensional array and its order is the number of dimensions of this array, also known as ways or modes. For any third-order tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, its (i, j, k)th component is denoted by a_{ijk}. Slices of a tensor are defined by fixing all but two indices, which are two-dimensional arrays of the tensor. For simplicity, we use the Matlab indexing notation $\mathcal{A}(:, :, i), \mathcal{A}(:, i, :), \mathcal{A}(i, :, :)$ to denote the ith frontal, lateral, and horizontal slices of $\mathcal{A}$. For the sake of convenience, the ith frontal slice $\mathcal{A}(:, :, i)$ is also denoted by $\mathbf{A}^{(i)}$. A tube of a third-order tensor is a vector defined by fixing the first two indices and varying the third one. For instance, using Matlab indexing notation, the (i, j)th tube of $\mathcal{A}$ is denoted by $\mathcal{A}(i, j, :)$.

For any $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, their inner product is defined by $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} a_{ijk}^* b_{ijk}$, where $\cdot^*$ denotes its conjugate entry. The nuclear norm of a matrix $\mathbf{A}$, denoted by $\|\mathbf{A}\|_*$, is the sum of all singular values of $\mathbf{A}$. The spectral norm of a matrix $\mathbf{A}$, denoted by $\|\mathbf{A}\|$, is the largest singular value of $\mathbf{A}$. The ℓ_1 norm of a tensor $\mathcal{A}$ is denoted by $\|\mathcal{A}\|_1 = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} |a_{ijk}|$. The Frobenius norm of a tensor $\mathcal{A}$ is denoted by $\|\mathcal{A}\|_F := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$.

A convex function f is said to be proper if $f(\mathcal{X}) < +\infty$ for at least one $\mathcal{X}$ and $f(\mathcal{X}) > -\infty$ for every $\mathcal{X}$. Let $\mathbb{X}$ be a finite-dimensional Euclidean space. For any given closed proper convex function $f : \mathbb{X} \to \mathbb{R} \cup \{+\infty\}$, the proximal mapping $\text{Prox}_f(\cdot)$ associated with f is defined by

$$\text{Prox}_f(\mathcal{Y}) := \arg\min_{\mathcal{X} \in \mathbb{X}} \left\{ f(\mathcal{X}) + \frac{1}{2} \|\mathcal{X} - \mathcal{Y}\|_F^2 \right\}.$$

2.2. Review of transformed t-SVD

The t-product and t-SVD have received a considerable amount of attention in robust tensor completion. The t-product was first proposed via using Fourier transform in [15]. Furthermore, Kernfeld et al. [23] extended the Fourier transform to any invertible transform for t-product and t-SVD. Very recently, Song et al. [28] studied t-product and t-SVD via unitary transform for the robust ten-

sor completion problem. We are now ready to review the t-product and t-SVD with any unitary transform briefly.

For any $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ and given unitary matrix $\mathbf{U} \in \mathbb{C}^{n_3 \times n_3}$, $\widehat{\mathcal{A}}$ denotes a third-order tensor via multiplying by $\mathbf{U}$ on all tubes of $\mathcal{A}$, i.e.,

$$\widehat{\mathcal{A}}(i, j, :) = \mathbf{U}(\mathcal{A}(i, j, :)), \ i = 1, \ldots, n_1, \ j = 1, \ldots, n_2.$$

For the sake of simplicity, we denote $\widehat{\mathcal{A}} = \mathbf{U}[\mathcal{A}]$. Conversely, one can get $\mathcal{A}$ from $\widehat{\mathcal{A}}$ via multiplying by $\mathbf{U}^H$ on all tubes of $\widehat{\mathcal{A}}$, i.e., $\mathcal{A} = \mathbf{U}^H[\widehat{\mathcal{A}}]$, where $\cdot^H$ denotes the conjugate transpose.

Now we are ready to give the definition of t-product between two tensors of appropriate dimension by using any unitary transform matrix.

Definition 2.1 [28, Definition 1]**.** Let $\mathbf{U}$ be an $n_3 \times n_3$ unitary matrix, $\mathcal{A}$ be an $n_1 \times n_2 \times n_3$ tensor, and $\mathcal{B}$ be an $n_2 \times n_4 \times n_3$ tensor. Then the t-product of $\mathcal{A}$ and $\mathcal{B}$ via unitary transform, denoted by $\mathcal{A} \diamond_{\mathbf{U}} \mathcal{B}$, is an $n_1 \times n_4 \times n_3$ tensor, defined by

$$\mathcal{A} \diamond_{\mathbf{U}} \mathcal{B} = \mathbf{U}^H\big[\text{foldblock}\big(\text{blockdiag}(\widehat{\mathcal{A}}) \cdot \text{blockdiag}(\widehat{\mathcal{B}})\big)\big], \tag{1}$$

where

$$\text{blockdiag}(\widehat{\mathcal{A}}) = \begin{pmatrix} \widehat{\mathbf{A}}^{(1)} & & & \\ & \widehat{\mathbf{A}}^{(2)} & & \\ & & \ddots & \\ & & & \widehat{\mathbf{A}}^{(n_3)} \end{pmatrix}$$

and $\text{foldblock}(\text{blockdiag}(\widehat{\mathcal{A}})) = \widehat{\mathcal{A}}$.

Let $\overline{\mathcal{A}} = \text{blockdiag}(\widehat{\mathcal{A}})$. We obtain that $\mathcal{Z} = \mathcal{A} \diamond_{\mathbf{U}} \mathcal{B}$ is equivalent to $\overline{\mathcal{Z}} = \overline{\mathcal{A}} \ \overline{\mathcal{B}}$. Before giving the transformed t-SVD, we first review relevant mathematical concepts including the conjugate transpose of a tensor, identity tensor, and unitary tensor.

Definition 2.2 [28, Definition 2]**.** For any tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, its conjugate transpose in the sense of the unitary matrix $\mathbf{U}$, denoted by $\mathcal{A}^H \in \mathbb{C}^{n_2 \times n_1 \times n_3}$, is obtained by

$$\mathcal{A}^H = \mathbf{U}^H\big[\text{foldblock}\big(\text{blockdiag}(\widehat{\mathcal{A}})^H\big)\big].$$

We now introduce the identity tensor with respect to any unitary matrix, which is given in [23,28]. Suppose that each frontal slice $\mathbf{I}^{(i)}$ of the tensor $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$, $i = 1, \ldots, n_3$, is the $n \times n$ identity matrix. The identity tensor with respect to $\mathbf{U}$, denoted by $\mathcal{I}_{\mathbf{U}} \in \mathbb{C}^{n \times n \times n_3}$, is defined to be a tensor such that $\mathcal{I}_{\mathbf{U}} = \mathbf{U}^H[\mathcal{I}]$ [23, Proposition 4.1]. We know that all frontal slices of $\mathcal{I}_{\mathbf{U}}$ under unitary transform are identity matrices. In particular, the identity tensor by using Fourier transform [15, Definition 3.4] is a special case of this definition. Based on the identity tensor, the unitary tensor with respect to any unitary matrix can be defined as follows:

Definition 2.3 [23, Definition 5.1]**.** A tensor $\mathcal{U} \in \mathbb{C}^{n \times n \times n_3}$ is unitary with respect to $\mathbf{U}$ if it satisfies

$$\mathcal{U}^H \diamond_{\mathbf{U}} \mathcal{U} = \mathcal{U} \diamond_{\mathbf{U}} \mathcal{U}^H = \mathcal{I}_{\mathbf{U}}.$$

In order to give the transformed t-SVD, we need to define the diagonal tensor. A tensor is called diagonal if each frontal slice is a diagonal matrix [15]. Now based on the above definitions, we are ready to state the transformed t-SVD of a tensor with any unitary transform.

Theorem 2.1. *[23, Theorem 5.1] For any $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, the transformed t-SVD is given by*

$$\mathcal{A} = \mathcal{U} \diamond_{\mathbf{U}} \mathcal{S} \diamond_{\mathbf{U}} \mathcal{V}^H, \tag{2}$$

where $\mathcal{U} \in \mathbb{C}^{n_1 \times n_1 \times n_3}, \mathcal{V} \in \mathbb{C}^{n_2 \times n_2 \times n_3}$ are unitary tensors, and $\mathcal{S} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ is a diagonal tensor.

One can obtain the transformed t-SVD by computing the SVDs of matrices efficiently in the transformed domain. More details can

be referred to [28] and references therein. The transformed t-SVD is useful to compute the proximal mapping of the transformed tensor nuclear norm, which will be defined later. Now we define the tubal rank and multi-rank of a tensor.

Definition 2.4 [28, Definition 6]**.** The transformed multi-rank of a tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ is a vector $\mathbf{r} \in \mathbb{R}^{n_3}$ with its ith entry being the rank of the ith frontal slice $\widehat{\mathbf{A}}^{(i)}$ of $\widehat{\mathcal{A}}$, i.e., $r_i = rank(\widehat{\mathbf{A}}^{(i)})$. The transformed tubal rank, denoted by $rank_{tt}(\mathcal{A})$, is defined as the number of nonzero singular tubes of $\mathcal{S}$, where $\mathcal{S}$ is obtained from the transformed t-SVD of $\mathcal{A} = \mathcal{U} \diamond_{\mathbf{U}} \mathcal{S} \diamond_{\mathbf{U}} \mathcal{V}^H$, i.e.,

$$rank_{tt}(\mathcal{A}) = \text{card}\{i : \mathcal{S}(i, i, :) \neq \mathbf{0}\} = \max_i\{r_i\}, \tag{3}$$

where $\text{card}\{\cdot\}$ denotes the cardinality of a set.

Definition 2.5 [28, Definition 7]**.** The transformed tensor nuclear norm of a tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, denoted by $\|\mathcal{A}\|_{\text{TTNN}}$, is defined as the sum of the nuclear norms of all frontal slices $\widehat{\mathbf{A}}^{(i)}$ of $\widehat{\mathcal{A}}$, i.e.,

$$\|\mathcal{A}\|_{\text{TTNN}} = \sum_{i=1}^{n_3} \|\widehat{\mathbf{A}}^{(i)}\|_*.$$

The tensor spectral norm of a tensor $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, denoted by $\|\mathcal{A}\|$, is defined as $\|\mathcal{A}\| = \|\overline{\mathcal{A}}\|$, which is the spectral norm of the block diagonal matrix $\overline{\mathcal{A}}$. Note that the transformed tensor nuclear norm of a tensor is the convex envelope of the sum of ranks of each frontal slices $\widehat{\mathbf{A}}^{(i)}$ of $\widehat{\mathcal{A}}$ on the set $\{\mathcal{A}|\|\mathcal{A}\| \leq 1\}$ [28, Lemma 1].

3. Patched-tubes tensor completion

In this section, we propose a patched-tubes unitary transform method for robust tensor completion by jointly exploring global low-rankness and nonlocal similarity of a tensor. We remark that the nonlocal strategy for two-dimensional image denoising and deblurring was already discussed in [29]. The patched-tubes unitary transform method is used to exploit the low-rankness of similar patches of tubes in each group. Compared with the low-rankness of the original tensor, the resulting sub-tensor achieved by grouping similar patches together can obtain lower tubal rank. The flowchart of the patched-tubes unitary transform method is shown in Fig. 1. In Algorithm 1, the first step is to generate an initial estimator $\widetilde{\mathcal{X}}$ for the missing data by the transformed t-SVD in [28]. Then we search all patches of tubes of $\widetilde{\mathcal{X}}$ and group the similar patches of tubes together. In this case, the tubal rank of each patched sub-tensor can be lower than the original tensor since each patched sub-tensor is constructed by similar patched-tubes, see the results in the next section.

Algorithm 1 A Patched-Tubes Unitary Transform Method via t-SVD for Robust Tensor Completion.

Step 0. Input index set Ω and $\mathcal{P}_{\Omega}(\mathcal{A})$, patch size s, overlap p, number of similar patches l in each group.
Step 1. Estimate an initial estimator $\widetilde{\mathcal{X}}$ by the transformed t-SVD.
Step 2. Find similar patches of $\widetilde{\mathcal{X}}$ and get the unitary matrices $\mathbf{U}_i$ in each sub-tensor, where the index set Ω_i of each sub-tensor can be attained according to the same coordinates of patches in $\widetilde{\mathcal{X}}$.
Step 3. Solve the robust tensor completion problem (3.4) for each sub-tensor.
Step 4. Reconstructed the final estimate by aggregating all patches in $\mathcal{X}_i$.

Mathematically, given an initial estimator $\widetilde{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we first partition it into many overlapped groups of tubes, where the size of each group of tubes is $s \times s \times n_3$. The groups of tubes are denoted by $\{\mathcal{C}_{mn}\} \in \mathbb{R}^{s \times s \times n_3}$, where $m = \lceil (n_1 - s)/(s - p) \rceil$ and $n = \lceil (n_2 - s)/(s - p) \rceil$, and p is the size of overlaps. Here $\lceil \cdot \rceil$ denotes the rounding up number. Now we need to search key groups
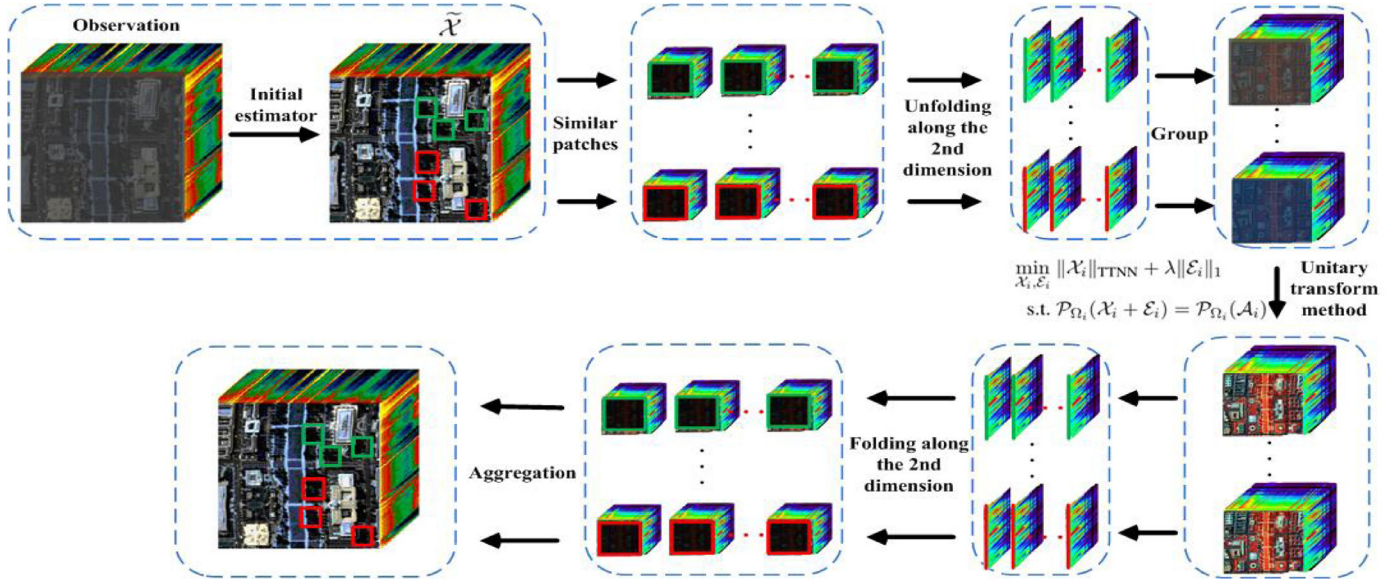
Fig. 1. Flowchart of the patched-tubes unitary transform method for robust tensor completion.

of tubes and patch several groups of tubes together. More precisely, we determine l similar groups and patch them together to form patched-tubes. By reshaping such patched-tubes, we formulate it a third-order sub-tensor $\widetilde{\mathcal{X}}_i$ from the initial estimator. Then the size of the resulting tensor is $s^2 \times l \times n_3$, where l is the number of patches involved. Now we unfold the tensor $\widetilde{\mathcal{X}}_i$ along the third-dimension to get the matrix $\widetilde{\mathbf{X}}_i$. Let $\widetilde{\mathbf{X}}_i = \mathbf{U}_i \Sigma_i \mathbf{V}^H$. Then the unitary matrix of the transformed t-SVD for the ith sub-tensor is $\mathbf{U}_i^H$, which is the transformation to obtain a low rank matrix of $\widetilde{\mathbf{X}}_i$ [28].

For the observed tensor $\mathcal{P}_\Omega(\mathcal{A})$ and index set Ω, the same coordinates of patches $\widetilde{\mathcal{X}}_i$ are used to get the index set Ω_i and $\mathcal{P}_{\Omega_i}(\mathcal{A}_i)$ in each sub-tensor. The underlying sub-tensor $\mathcal{A}_i$ can be decomposed into a low transformed multi-rank tensor $\mathcal{X}_i$ and a sparse tensor $\mathcal{E}_i$, i.e., $\mathcal{A}_i = \mathcal{X}_i + \mathcal{E}_i$. Note that the underlying sub-tensor $\mathcal{A}_i$ can only be available in the index set Ω_i. Then we get $\mathcal{P}_{\Omega_i}(\mathcal{X}_i + \mathcal{E}_i) = \mathcal{P}_{\Omega_i}(\mathcal{A}_i)$ in Fig. 1. Therefore, the robust tensor completion problem in each sub-tensor can be written as follows:

$$
\begin{aligned}
\min_{\mathcal{X}_i, \mathcal{E}_i} \quad & \|\mathcal{X}_i\|_{\mathrm{TTNN}} + \lambda \|\mathcal{E}_i\|_1 \\
\text{s.t.} \quad & \mathcal{P}_{\Omega_i}(\mathcal{X}_i + \mathcal{E}_i) = \mathcal{P}_{\Omega_i}(\mathcal{A}_i),
\end{aligned} \tag{4}
$$

where i denotes the ith sub-tensor.

After recovering all sub-tensors $\mathcal{X}_i$, we need to collect $\mathcal{X}_i$ to obtain a final estimate of the true tensor. We first fold all lateral slices of each sub-tensor into a third-order tensor along the second dimension, where the sizes of the resulting tensors are $s \times s \times n_3$. Then we reconstruct the final tensor based on these recovering patches. Since the patches are overlapping at some positions, we reconstruct the finial estimate of the true tensor by aggregating all of the obtained local estimated patches using a weighting average at those positions where there are overlapping cube estimates. Based on the previous discussions, the patched-tubes unitary transform method via t-SVD for robust tensor completion can be stated in Algorithm 1.

Remark 3.1. The motivation of Algorithm 1 is to extract similar patches and group them together. Then the resulting sub-tensor would be lower tubal rank in each group than the original tensor. However, if there is little similar patches in the original tensor, the sub-tensor is not low rank and the proposed method is not more efficient than the transformed t-SVD in [28]. In the extreme case, if the sizes of patches are the same as that of the original

tensor, then the patched-tubes unitary transform method reduces to the transformed t-SVD method. In this case, the performance of Algorithm 1 is the same as that of the transformed t-SVD. We find that the performance of patched-based method is better than that of the transformed t-SVD (the extreme case) for real data sets, see Tables 1, 2 and 3 in Section 4.

3.1. The perturbation analysis

Since the similar patches are established based on the minimum distances from the key patch in each sub-tensor, the distances between one patch and the other patches are increasing. Let $\mathcal{X} \in \mathbb{R}^{s^2 \times l \times n_3}$ be a sub-tensor. Now we can establish an upper bound between $\mathcal{X}$ and a tubal rank-1 tensor.

Theorem 3.1. Let $\mathbf{Z}$ be the average lateral slice of $\mathcal{X}$, i.e., $\mathbf{Z} = (\sum_{j=1}^l \mathcal{X}(:, j, :))/l$, and $\mathcal{Y}(:, j, :) = \mathbf{Z}$, $j = 1, \dots, l$. Suppose that
$$\max_j \{\|\mathcal{X}(:, j, :) - \mathbf{Z}\|_F\} \leq \varepsilon.$$
Let $\mathcal{X} = \mathcal{Y} + \mathcal{E}$. Then $\|\mathcal{E}\|_F \leq \sqrt{l}\varepsilon$.

Proof. Note that
$$\|\mathcal{E}\|_F^2 = \|\mathcal{X} - \mathcal{Y}\|_F^2 = \sum_{j=1}^l \|\mathcal{X}(:, j, :) - \mathbf{Z}\|_F^2 \leq l\varepsilon^2.$$
Therefore, $\|\mathcal{E}\|_F \leq \sqrt{l}\varepsilon$. This completes the proof. □

Remark 3.2. From Theorem 3.1, we obtain that $\mathcal{X}$ can be approximated by a tubal rank-1 tensor. In practice, ε is very small, the generated sub-tensor is a tensor with approximatively tubal rank-1. Moreover, when all patches are the same, then the generating sub-tensor is a tubal rank-1 tensor.

Theorem 3.1 shows an upper bound between $\mathcal{X}$ and a tubal rank-1 tensor. Now we consider the tubal rank-k approximation of $\mathcal{X}$. Let $\mathfrak{D} := \{1, \dots, k\}$.

Theorem 3.2. Let $\mathcal{X} = \widetilde{\widetilde{\mathcal{Y}}} + \widetilde{\widetilde{\mathcal{E}}}$, where $\widetilde{\widetilde{\mathcal{Y}}}(:, i, :) = \widehat{\mathcal{X}}(:, i, :)$, $i = 1, \dots, k$, $\widetilde{\widetilde{\mathcal{Y}}}(:, j, :) = \widehat{\mathcal{X}}(:, 1, :)$, $j = k+1, \dots, l$, i.e., $\widetilde{\widetilde{\mathcal{Y}}} = [\widehat{\mathcal{X}}(:, 1, :), \widehat{\mathcal{X}}(:, 2, :), \dots, \widehat{\mathcal{X}}(:, k, :), \widehat{\mathcal{X}}(:, 1, :), \dots, \widehat{\mathcal{X}}(:, 1, :)] \in \mathbb{R}^{s^2 \times l \times n_3}$. Suppose that $\|\mathcal{X}(:, q, :) - \mathcal{X}(:, j, :)\|_F \leq \widetilde{\varepsilon}$ for $q \in \mathfrak{D}$ and $j = 1, \dots, l$. Then
$$\|\widetilde{\widetilde{\mathcal{E}}}\|_F \leq \sqrt{(l-k)}\widetilde{\varepsilon}.$$

Table 1
The PSNR and SSIM values of different methods for the hyperspectral data in robust tensor completion.

	SR	γ	PSNR					SSIM				
			SNN	t-SVD (FFT)	t-SVD (Wavelet)	t-SVD (Data)	Patched-t-SVD	SNN	t-SVD (FFT)	t-SVD (Wavelet)	t-SVD (Data)	Patched-t-SVD
Japser Ridge	0.6	0.1	30.13	39.22	40.57	44.97	**47.62**	0.882	0.977	0.982	0.992	**0.995**
		0.2	27.92	36.39	37.25	41.15	**43.91**	0.793	0.966	0.966	0.987	**0.991**
		0.3	26.00	33.57	33.57	37.33	**40.92**	0.697	0.956	0.943	0.977	**0.988**
	0.8	0.1	31.98	40.82	42.62	46.63	**49.35**	0.934	0.985	0.986	0.994	**0.996**
		0.2	29.61	37.89	38.63	43.06	**46.41**	0.867	0.978	0.968	0.990	**0.994**
		0.3	27.49	35.13	35.75	39.20	**43.10**	0.775	0.964	0.959	0.983	**0.991**
Urban	0.6	0.1	27.88	38.78	39.10	45.80	**49.03**	0.848	0.986	0.986	0.997	**0.999**
		0.2	26.13	36.08	35.70	41.87	**44.62**	0.786	0.977	0.975	0.995	**0.997**
		0.3	24.69	33.39	32.16	37.22	**42.50**	0.718	0.967	0.948	0.983	**0.995**
	0.8	0.1	30.31	40.76	42.94	46.06	**50.12**	0.916	0.992	0.994	0.997	**0.999**
		0.2	27.89	37.77	38.44	43.27	**47.20**	0.857	0.986	0.986	0.996	**0.998**
		0.3	26.06	34.98	33.37	39.68	**44.43**	0.790	0.975	0.954	0.991	**0.997**
Scene 5	0.6	0.1	32.81	36.12	34.77	43.72	**48.82**	0.913	0.953	0.936	0.993	**0.998**
		0.2	31.18	34.13	32.71	40.48	**45.37**	0.873	0.934	0.918	0.985	**0.995**
		0.3	29.83	32.15	31.02	36.58	**42.29**	0.840	0.928	0.876	0.961	**0.992**
	0.8	0.1	34.27	38.00	37.50	46.58	**53.58**	0.932	0.977	0.975	0.995	**0.999**
		0.2	32.33	35.70	34.60	42.55	**49.49**	0.912	0.961	0.944	0.989	**0.998**
		0.3	30.95	33.83	32.31	37.81	**47.06**	0.871	0.937	0.901	0.959	**0.997**
Scene 7	0.6	0.1	31.41	38.94	34.94	44.15	**48.18**	0.919	0.982	0.944	0.994	**0.998**
		0.2	29.79	36.69	33.78	40.13	**43.81**	0.870	0.970	0.954	0.987	**0.994**
		0.3	28.43	34.56	30.53	36.62	**41.11**	0.842	0.961	0.883	0.970	**0.991**
	0.8	0.1	33.90	40.90	38.70	46.71	**52.37**	0.935	0.991	0.979	0.996	**0.999**
		0.2	31.32	38.52	35.09	42.38	**47.75**	0.914	0.983	0.953	0.991	**0.997**
		0.3	29.81	36.17	32.22	38.77	**43.42**	0.874	0.970	0.911	0.983	**0.994**

Table 2
The PSNR and SSIM values of video data for robust tensor completion.

	SR	γ	PSNR					SSIM				
			SNN	t-SVD (FFT)	t-SVD (Wavelet)	t-SVD (Data)	Patched-t-SVD	SNN	t-SVD (FFT)	t-SVD (Wavelet)	t-SVD (Data)	Patched-t-SVD
Carphone	0.6	0.1	26.80	30.72	31.24	32.38	**34.83**	0.833	0.884	0.934	0.926	**0.970**
		0.2	24.88	28.83	28.56	30.13	**32.24**	0.766	0.857	0.913	0.915	**0.950**
		0.3	23.39	27.18	27.29	28.02	**31.25**	0.691	0.880	0.845	0.839	**0.939**
	0.8	0.1	28.74	32.55	32.70	34.03	**36.62**	0.876	0.931	0.932	0.945	**0.980**
		0.2	26.35	30.16	30.26	31.24	**34.08**	0.826	0.904	0.931	0.910	**0.965**
		0.3	24.68	28.14	28.03	29.10	**32.40**	0.745	0.869	0.904	0.880	**0.951**
Announcer	0.6	0.1	29.58	37.96	38.31	39.46	**41.49**	0.901	0.977	0.982	0.985	**0.991**
		0.2	27.55	35.41	34.98	36.34	**38.54**	0.845	0.968	0.962	0.979	**0.984**
		0.3	26.36	32.72	31.94	33.95	**37.55**	0.828	0.960	0.951	0.964	**0.981**
	0.8	0.1	31.57	39.65	40.41	41.27	**43.40**	0.906	0.987	0.985	0.990	**0.994**
		0.2	28.67	36.68	35.84	38.09	**40.58**	0.891	0.979	0.948	0.983	**0.990**
		0.3	27.50	34.17	33.58	35.19	**38.76**	0.801	0.967	0.955	0.973	**0.985**
Grandma	0.6	0.1	29.42	37.69	38.48	39.07	**41.67**	0.894	0.962	0.975	0.972	**0.988**
		0.2	27.24	35.58	35.36	37.10	**39.06**	0.838	0.949	0.962	0.967	**0.979**
		0.3	25.35	33.91	33.27	34.88	**37.55**	0.801	0.952	0.942	0.956	**0.972**
	0.8	0.1	31.99	39.94	40.50	41.28	**42.99**	0.891	0.978	0.981	0.984	**0.991**
		0.2	28.96	37.22	37.53	38.50	**39.85**	0.880	0.966	0.973	0.978	**0.982**
		0.3	27.07	34.77	34.78	35.99	**38.51**	0.733	0.950	0.945	0.967	**0.977**

Table 3
The PSNR and SSIM values obtained by different methods for the extended Yale face database B in robust tensor completion.

SR	γ	PSNR					SSIM				
		SNN	t-SVD (FFT)	t-SVD (Wavelet)	t-SVD (Data)	Patched-t-SVD	SNN	t-SVD (FFT)	t-SVD (Wavelet)	t-SVD (Data)	Patched-t-SVD
0.6	0.1	24.57	25.92	26.92	28.41	**31.16**	0.771	0.774	0.841	0.861	**0.954**
	0.2	22.89	24.12	24.77	26.09	**28.88**	0.646	0.723	0.764	0.808	**0.931**
	0.3	21.76	22.23	23.01	23.68	**26.83**	0.611	0.708	0.710	0.744	**0.897**
0.8	0.1	26.59	28.06	28.98	30.29	**34.28**	0.835	0.868	0.877	0.888	**0.980**
	0.2	24.53	25.53	26.05	27.33	**31.28**	0.737	0.805	0.781	0.820	**0.963**
	0.3	23.06	23.57	24.09	25.05	**29.23**	0.648	0.736	0.718	0.754	**0.940**

Proof. It follows from the definition of $\widetilde{\mathcal{E}}$ that

$$\|\widetilde{\mathcal{E}}\|_F^2 = \|\mathcal{X} - \widetilde{\mathcal{Y}}\|_F^2 = \|\widehat{\mathcal{X}} - \widehat{\widetilde{\mathcal{Y}}}\|_F^2 \leq \sum_{j=k+1}^{l} \|\widehat{\mathcal{X}}(:, j, :) - \widehat{\mathcal{X}}(:, 1, :)\|_F^2$$

$$= (l - k)\widetilde{\varepsilon}^2.$$

where the last equality holds by the unitary invariance of Frobenius. Hence $\|\widetilde{\mathcal{E}}\|_F \leq \sqrt{l - k}\widetilde{\varepsilon}$. $\square$

Remark 3.3. Note that $\widetilde{\mathcal{Y}}$ is a tubal rank-k tensor. We can see that $\mathcal{X}$ is an approximation of $\widetilde{\mathcal{Y}}$ with an error. Since $\widetilde{\varepsilon} < \varepsilon$ in general, the error bound of Theorem 3.2 is lower than that of Theorem 3.1.

Let $\mathcal{X} := \mathcal{Y} + \mathcal{E} = \mathcal{U} \diamond_{\mathbf{U}} \mathcal{S} \diamond_{\mathbf{U}} \mathcal{V}^H$, $\mathcal{Y} = \mathcal{U}_1 \diamond_{\mathbf{U}} \mathcal{S}_1 \diamond_{\mathbf{U}} \mathcal{V}_1^H$, and $\mathcal{E} = \mathcal{U}_2 \diamond_{\mathbf{U}} \mathcal{S}_2 \diamond_{\mathbf{U}} \mathcal{V}_2^H$. We establish the perturbation results for the transformed t-SVD.

Theorem 3.3. *If $\mathcal{Y} + \mathcal{E} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ with $n_1 \geq n_2$. Then*

$$\|\mathcal{S} - \mathcal{S}_1\|_F \leq \|\mathcal{E}\|_F.$$

Proof. Denote

$$\widetilde{\mathcal{Y}} := \begin{pmatrix} 0 & \mathcal{Y}^H \\ \mathcal{Y} & 0 \end{pmatrix} \in \mathbb{R}^{(n_1+n_2) \times (n_1+n_2) \times n_3} \quad \text{and}$$

$$\widetilde{\mathcal{Y}} + \widetilde{\mathcal{E}} := \begin{pmatrix} 0 & (\mathcal{Y}+\mathcal{E})^H \\ \mathcal{Y}+\mathcal{E} & 0 \end{pmatrix} \in \mathbb{R}^{(n_1+n_2) \times (n_1+n_2) \times n_3}.$$

Let $\mathcal{U}_1 = (\mathcal{U}_{1n_2}, \mathcal{U}_{1\widetilde{n}_2})$, where $\mathcal{U}_{1n_2} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. Denote

$$\mathcal{Q}_1 : \quad = \frac{1}{\sqrt{2}} \begin{pmatrix} \mathcal{V}_1 & \mathcal{V}_1 & 0 \\ \mathcal{U}_{1n_2} & -\mathcal{U}_{1n_2} & \sqrt{2}\mathcal{U}_{1\widetilde{n}_2} \end{pmatrix} \in \mathbb{R}^{(n_1+n_2) \times (n_1+n_2) \times n_3}.$$

Since $\mathcal{U}_1^H \diamond_{\mathbf{U}} \mathcal{Y} \diamond_{\mathbf{U}} \mathcal{V}_1 = \mathcal{S}_1$, we obtain that

$$\begin{pmatrix} \mathcal{U}_{1n_2}^H \diamond_{\mathbf{U}} \mathcal{Y} \diamond_{\mathbf{U}} \mathcal{V}_1 \\ \mathcal{U}_{1\widetilde{n}_2}^H \diamond_{\mathbf{U}} \mathcal{Y} \diamond_{\mathbf{U}} \mathcal{V}_1 \end{pmatrix} = \begin{pmatrix} \widetilde{\mathcal{S}}_1 \\ 0 \end{pmatrix}$$

where $\widetilde{\mathcal{S}}_1 \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ is the first n_2 horizontal slices of $\mathcal{S}_1$, i.e., $\widetilde{\mathcal{S}}_1 = \mathcal{S}_1(1:n_2,:,:)$. Note that $\widetilde{\mathcal{S}}_1^H = \widetilde{\mathcal{S}}_1$. Therefore,

$$\mathcal{U}_{1n_2}^H \diamond_{\mathbf{U}} \mathcal{Y} \diamond_{\mathbf{U}} \mathcal{V}_1 = \mathcal{V}_1^H \diamond_{\mathbf{U}} \mathcal{Y}^H \diamond_{\mathbf{U}} \mathcal{U}_{1n_2}.$$

Then we obtain that

$$\mathcal{Q}_1^H \diamond_{\mathbf{U}} \widetilde{\mathcal{Y}} \diamond_{\mathbf{U}} \mathcal{Q}_1 = \begin{pmatrix} \widetilde{\mathcal{S}}_1 & 0 & 0 \\ 0 & -\widetilde{\mathcal{S}}_1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Let $\mathcal{U} = (\mathcal{U}_{n_2}, \mathcal{U}_{\widetilde{n}_2})$, where $\mathcal{U}_{n_2} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, and

$$\mathcal{Q} := \frac{1}{\sqrt{2}} \begin{pmatrix} \mathcal{V} & \mathcal{V} & 0 \\ \mathcal{U}_{n_2} & -\mathcal{U}_{n_2} & \sqrt{2}\mathcal{U}_{\widetilde{n}_2} \end{pmatrix} \in \mathbb{R}^{(n_1+n_2) \times (n_1+n_2) \times n_3}.$$

In the same argument, we get that

$$\mathcal{Q}^H \diamond_{\mathbf{U}} (\widetilde{\mathcal{Y}} + \widetilde{\mathcal{E}}) \diamond_{\mathbf{U}} \mathcal{Q} = \begin{pmatrix} \widetilde{\mathcal{S}} & 0 & 0 \\ 0 & -\widetilde{\mathcal{S}} & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

where $\widetilde{\mathcal{S}}$ is the first n_2 horizontal slices of $\mathcal{S}$, i.e., $\widetilde{\mathcal{S}} = \mathcal{S}(1:n_2,:,:)$. By [33, Theorem 8.1.4], we obtain that for $k = 1, \ldots, n_2$,

$$\sum_{k=1}^{n_2} \left(\widehat{\widetilde{\mathcal{S}}}(k,k,j) - \widehat{\widetilde{\mathcal{S}}}_1(k,k,j) \right)^2 \leq \|\widehat{\mathcal{E}}(:,:,j)\|_F^2, \quad j = 1, \ldots, n_3,$$

which yields that

$$\|\mathcal{S} - \mathcal{S}_1\|_F \leq \|\mathcal{E}\|_F.$$

We complete the proof. □

Remark 3.4. For the similar patches in each sub-tensor, the distances between the first and other lateral slices are increasing. We can regard the generated sub-tensor as a low rank tensor plus a perturbation tensor. In this case, Theorem 3.3 gives an upper bound of the singular values of the perturbation tensor. When $\mathcal{E}$ is small and $\mathcal{Y}$ is a low-rank tensor, the generated sub-tensor approximates a low-rank tensor. In particular, for $\mathcal{Y}$ in Theorems 3.1 and 3.2, it is just the perturbation bounds of the tubal rank-1 and tubal rank-k tensor approximations, respectively.

3.2. Optimization algorithm

In this subsection, an sGS-ADMM [34,35] is developed to solve problem (4). The sGS-ADMM is very efficient to solve multi-block objective function involving equality constraint with convergence guarantee and has been applied in various fields, e.g., semidefinite programming [34,35] and tensor completion [36,37]. By introducing an auxiliary variable $\mathcal{Y}_i = \mathcal{X}_i + \mathcal{E}_i$, solving problem (4) is equivalent to solving

$$\begin{aligned} \min_{\mathcal{X}_i, \mathcal{E}_i, \mathcal{Y}_i} \quad & \|\mathcal{X}_i\|_{\text{TTNN}} + \lambda\|\mathcal{E}_i\|_1 \\ \text{s.t.} \quad & \mathcal{X}_i + \mathcal{E}_i = \mathcal{Y}_i, \quad \mathcal{P}_{\Omega_i}(\mathcal{Y}_i) = \mathcal{P}_{\Omega_i}(\mathcal{A}_i). \end{aligned} \tag{5}$$

The augmented Lagrangian function associated with (5) is defined as follows:

$$\begin{aligned} L(\mathcal{X}_i, \mathcal{E}_i, \mathcal{Y}_i, \mathcal{W}_i) := \quad & \|\mathcal{X}_i\|_{\text{TTNN}} + \lambda\|\mathcal{E}_i\|_1 - \langle \mathcal{W}_i, \mathcal{X}_i + \mathcal{E}_i - \mathcal{Y}_i \rangle \\ & + \frac{\beta}{2} \|\mathcal{X}_i + \mathcal{E}_i - \mathcal{Y}_i\|_F^2, \end{aligned} \tag{6}$$

where $\beta > 0$ is a given penalty parameter and $\mathcal{W}_i$ is the Lagrangian multiplier.

The iteration template of the sGS-ADMM at the $(k+1)$th iteration is described as follows:

$$\mathcal{Y}_i^{k+\frac{1}{2}} = \arg\min_{\mathcal{Y}_i} \left\{ L(\mathcal{X}_i^k, \mathcal{E}_i^k, \mathcal{Y}_i, \mathcal{W}_i^k), \mathcal{P}_{\Omega_i}(\mathcal{Y}_i) = \mathcal{P}_{\Omega_i}(\mathcal{A}_i) \right\}, \tag{7}$$

$$\mathcal{X}_i^{k+1} = \arg\min_{\mathcal{X}_i} \left\{ L(\mathcal{X}_i, \mathcal{E}_i^k, \mathcal{Y}_i^{k+\frac{1}{2}}, \mathcal{W}_i^k) \right\}, \tag{8}$$

$$\mathcal{Y}_i^{k+1} = \arg\min_{\mathcal{Y}_i} \left\{ L(\mathcal{X}_i^{k+1}, \mathcal{E}_i^k, \mathcal{Y}_i, \mathcal{W}_i^k), \mathcal{P}_{\Omega_i}(\mathcal{Y}_i) = \mathcal{P}_{\Omega_i}(\mathcal{A}_i) \right\}, \tag{9}$$

$$\mathcal{E}_i^{k+1} = \arg\min_{\mathcal{E}_i} \left\{ L(\mathcal{X}_i^{k+1}, \mathcal{E}_i, \mathcal{Y}_i^{k+1}, \mathcal{W}_i^k) \right\}, \tag{10}$$

$$\mathcal{W}_i^{k+1} = \mathcal{W}_i^k - \tau\beta(\mathcal{X}_i^{k+1} + \mathcal{E}_i^{k+1} - \mathcal{Y}_i^{k+1}), \tag{11}$$

where $\tau \in (0, (1+\sqrt{5})/2)$ is the step-length. Now we solve each subproblem explicitly in the sGS-ADMM.

- The subproblem with respect to $\mathcal{Y}_i$. Since the augmented Lagrangian function L about $\mathcal{Y}_i$ is differentiable, its optimal solution can be given explicitly by

$$\mathcal{Y}_i = \mathcal{P}_{\Omega_i}(\mathcal{A}_i) + \mathcal{P}_{\overline{\Omega}_i}\left(\mathcal{X}_i + \mathcal{E}_i - \frac{1}{\beta}\mathcal{W}_i \right),$$

where $\overline{\Omega}_i$ represents the complementary set of Ω_i.

- The subproblem with respect to $\mathcal{X}_i$. The problem (8) can be equivalently reformulated as

$$\min_{\mathcal{X}_i} \|\mathcal{X}_i\|_{\text{TTNN}} + \frac{\beta}{2} \left\| \mathcal{X}_i - \left(\mathcal{Y}_i^{k+\frac{1}{2}} + \frac{1}{\beta}\mathcal{W}_i^k - \mathcal{E}_i^k \right) \right\|_F^2. \tag{12}$$

By [28, Theorem 3], the optimal solution $\mathcal{X}_i^{k+1}$ of problem (12) takes the following form:

$$\mathcal{X}_i^{k+1} \quad = \text{Prox}_{\frac{1}{\beta}\|\cdot\|_{\text{TTNN}}}\left(\mathcal{Y}_i^{k+\frac{1}{2}} + \frac{1}{\beta}\mathcal{W}_i^k - \mathcal{E}_i^k \right) = \mathcal{U} \diamond_{\mathbf{U}_i} \mathcal{S}_\beta \diamond_{\mathbf{U}_i} \mathcal{V}^H, \tag{13}$$

where $\mathcal{Y}_i^{k+\frac{1}{2}} + \frac{1}{\beta}\mathcal{W}_i^k - \mathcal{E}_i^k = \mathcal{U} \diamond_{\mathbf{U}_i} \mathcal{S} \diamond_{\mathbf{U}_i} \mathcal{V}^H$ and $\mathcal{S}_\beta = \mathbf{U}_i^H[\max\{\widehat{\mathcal{S}} - \frac{1}{\beta}, 0\}]$. Here $\mathbf{U}_i$ is the unitary transform matrix in the ith sub-tensor.

- The subproblem with respect to $\mathcal{E}_i$. Let $\mathcal{G}^{k+1} := \mathcal{Y}_i^{k+1} + \frac{1}{\beta}\mathcal{W}_i^k - \mathcal{X}_i$. A brief calculation of problem (10) yields the following form:

$$\mathcal{E}_i^{k+1} = \text{Prox}_{\lambda\|\cdot\|_1}(\mathcal{G}^{k+1}) = \text{sign}(\mathcal{G}^{k+1}) \circ \max\left\{ |\mathcal{G}^{k+1}| - \frac{\lambda}{\beta}, 0 \right\}, \tag{14}$$

where $\circ$ and $|\cdot|$ denote the point-wise product and point-wise absolute value, and $\text{sign}(\cdot)$ denotes the signum function taking the form

$$\text{sign}(y) := \begin{cases} 1, & \text{if } y > 0, \\ 0, & \text{if } y = 0, \\ -1, & \text{if } y < 0. \end{cases}$$

Now the sGS-ADMM framework for solving (5) can be described in Algorithm 2.

Algorithm 2 A Symmetric Gauss-Seidel Based Multi-Block ADMM for Solving (5).

Step 0. Given $\tau \in (0, (1+\sqrt{5})/2), \beta > 0$, $\mathbf{U}_i$, index set Ω_i and $\mathcal{P}_{\Omega_i}(\mathcal{A}_i)$.Choose $\mathcal{X}_i^0, \mathcal{E}_i^0, \mathcal{W}_i^0$.Set $k := 0$.

Step 1. Compute $\mathcal{Y}_i^{k+\frac{1}{2}}$ by

$$\mathcal{Y}_i^{k+\frac{1}{2}} = \mathcal{P}_{\Omega_i}(\mathcal{A}_i) + \mathcal{P}_{\overline{\Omega}_i}\left(\mathcal{X}_i^k + \mathcal{E}_i^k - \frac{1}{\beta}\mathcal{W}_i^k\right).$$

Step 2. Compute $\mathcal{X}_i^{k+1}$ via (3.13).

Step 3. Compute $\mathcal{Y}_i^{k+1}$ by

$$\mathcal{Y}_i^{k+1} = \mathcal{P}_{\Omega_i}(\mathcal{A}_i) + \mathcal{P}_{\overline{\Omega}_i}\left(\mathcal{X}_i^{k+1} + \mathcal{E}_i^k - \frac{1}{\beta}\mathcal{W}_i^k\right).$$

Step 4. Compute $\mathcal{E}_i^{k+1}$ via (3.14).

Step 5. Compute $\mathcal{Z}_i^{k+1}$ by (3.11).

Step 6. If a stopping criterion is not satisfied, set $k := k+1$ and go to step 1.

Since only $\|\cdot\|_{\text{TTNN}}$ and $\|\cdot\|_1$ in the objective function of (5) are nonsmooth, the sequence generated by the sGS-ADMM converges to an optimal solution of (5) [35, Theorem 3]. Based on the first-order optimality condition for problem (5), the stopping criterion of Algorithm 2 for each sub-tensor is given by

$$\eta_{sc} := \max\{\eta_x, \eta_e, \eta_y\} \le \text{tol},$$

where

$$\eta_x = \frac{\|\mathcal{X}_i - \text{Prox}_{\|\cdot\|_{\text{TTNN}}}(\mathcal{W}_i + \mathcal{X}_i)\|_F}{1 + \|\mathcal{W}_i\|_F + \|\mathcal{X}_i\|_F}, \quad \eta_e = \frac{\|\mathcal{E}_i - \text{Prox}_{\lambda\|\cdot\|_1}(\mathcal{W}_i + \mathcal{E}_i)\|_F}{1 + \|\mathcal{W}_i\|_F + \|\mathcal{E}_i\|_F},$$

$$\eta_y = \frac{\|\mathcal{X}_i + \mathcal{E}_i - \mathcal{Y}_i\|_F}{1 + \|\mathcal{X}_i\|_F + \|\mathcal{E}_i\|_F + \|\mathcal{Y}_i\|_F}.$$

4. Numerical results

In this section, some numerical results are presented to demonstrate the effectiveness of the patched-tubes t-SVD with unitary transform (Patched-t-SVD) for robust tensor completion. For tensor completion, we compare the Patched-t-SVD with the high accurate low-rank tensor completion (HaLRTC)[4] [17], low-rank tensor completion by parallel matrix factorization (TMac)[5] [38], trace norm regularized CP decomposition method (TNCP)[6] [39], t-SVD by using FFT (t-SVD (FFT))[7] [27], t-SVD by using wavelet transform (t-SVD (Wavelet)), and t-SVD by using data-dependent unitary transform (t-SVD (Data)) [28]. For tensor robust principal component analysis with partial observations, we compare the Patched-t-SVD with the SNN[8] [40], t-SVD (FFT)[9] [26], t-SVD (Wavalet) and t-SVD (Data) [28]. All experiments are performed in the MATLAB 2017a environment and run on a desktop computer (Intel Core i7-6700, @ 3.40GHz, 20G RAM).

The sampling ratio (SR) of observations is the ratio between the number of observed entries and the true dimensionality of a tensor, i.e.,

$$\text{SR} := \frac{|\Omega|}{n_1 n_2 n_3},$$

---

[4] http://www.cs.rochester.edu/u/jliu/.
[5] https://xu-yangyang.github.io/TMac/.
[6] https://sites.google.com/site/fanhua217/home.
[7] https://github.com/jamiezeminzhang.
[8] https://sites.google.com/site/tonyqin/home.
[9] https://canyilu.github.io/.

where $|\Omega|$ denoted the cardinality of Ω and Ω is an index set generated uniformly at random. To evaluate the recovery performance of different methods, we adopt the metric of the peak signal-to-noise ratio (PSNR) defined by

$$\text{PSNR} := 10 \log_{10} \frac{n_1 n_2 n_3 (\mathcal{X}_{\max} - \mathcal{X}_{\min})^2}{\|\mathcal{X} - \mathcal{X}_0\|_F^2},$$

where $\mathcal{X}, \mathcal{X}_0 \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ are the recovered and ground-truth tensors, respectively, $\mathcal{X}_{\max}$ and $\mathcal{X}_{\min}$ are the maximum and minimum entries of $\mathcal{X}_0$. The structural similarity (SSIM) index [41] is used to measure the quality of the recovered and true images for image quality assessment:

$$\text{SSIM} := \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)},$$

where μ_x, σ_x are the mean intensities and standard deviation of the original images, respectively, μ_y, σ_y denote the mean intensities and standard deviation of the recovered images, respectively, σ_{xy} denotes the covariance of the original and recovered images, and c_1, $c_2 > 0$ are constants. For the testing tensor data, we use SSIM to denote the average values of all images.

The intensity range of the real-world tensor is scaled into [0,1] before generating $\mathcal{P}_{\Omega}(\mathcal{A})$. For the sparse tensor $\mathcal{E}$ in robust tensor completion, a fraction γ of its entries are uniformly corrupted by additive independent and identically distributed noise from a standard Gaussian distribution $N(0, 1)$ at random, which generates the sparse tensor $\mathcal{E}$. Finally, a percentage SR is chosen randomly from the noisy tensor as our observations.

For the unitary transform matrix of the Patched-t-SVD, an initial estimator $\widetilde{\mathcal{X}}$ is generated by t-SVD (Data) first. Then we search similar patches of $\widetilde{\mathcal{X}}$ and formulate sub-tensors $\widetilde{\mathcal{X}}_i$. Afterwards, we unfold $\widetilde{\mathcal{X}}_i$ along the third-dimension to get $\mathbf{X}_i = (\widetilde{\mathcal{X}}_i)_{(3)}$, where $(\widetilde{\mathcal{X}}_i)_{(3)}$ denotes the mode-3 unfolding of $\widetilde{\mathcal{X}}_i$. Denote the SVD of $\mathbf{X}_i = \mathbf{U}_i \Sigma_i \mathbf{V}_i^H$. The unitary matrix $\mathbf{U}_i^H$ is used in the ith sub-tensor. In fact, the resulting matrix $\mathbf{U}_i^H$ is an optimal transformation to obtain a low rank matrix of $\mathbf{X}_i$:

$$\min_{\mathbf{\Phi}, \mathbf{B}} \left\{ \|\mathbf{\Phi}\mathbf{X}_i - \mathbf{B}\|_F^2 : \text{rank}(\mathbf{B}) = k, \text{ uniatry matirx } \mathbf{\Phi} \right\},$$

where $\text{rank}(\mathbf{X}_i) = k$.

4.1. Parameters setting

For the TMac method in tensor completion, we use the rank-increasing strategy and initialize the underestimated rank $r_i = 5$ and set $\Delta r_i = 3$, $r_i^{\max} = 50$, where r_i is the rank of the low-rank factorization matrix in [38], $i = 1, 2, 3$. For t-SVD (Wavelet), we use a Daubechies 4 (db4) discrete wavelet transform in the periodic mode [42]. In each sub-tensor of the Patched-t-SVD method, τ is set to 1.618 for the convergence of sGS-ADMM in all experiments. For robust tensor completion problems, β is set to 0.05 in all experiments since β is not too sensitive to the recovered results. In practical implementation, we let Algorithm 2 stop when $\eta_{sc} \le \text{tol} = 5 \times 10^{-4}$ or when the number of iterations exceeds 500. For t-SVD (FFT), t-SVD (Wavelet), t-SVD (Data), the stopping criterions are the same as that of the Patched-t-SVD method.

For λ of t-SVD (FFT), t-SVD (Wavelet), t-SVD (Data), Patched-t-SVD in robust tensor completion, it is chosen based on $1/\sqrt{n_{(1)}n_3}$ [28], where $n_{(1)} = \max\{n_1, n_2\}$. We choose $\lambda = a/\sqrt{n_{(1)}n_3}$ and adjust a near 1 slightly to obtain the best possible results for the testing data sets.

4.2. Performance analysis of patched-t-SVD

The overlap is set to be $p = 1$ in all experiments for the Patched-t-SVD method. The large size of patches will take more
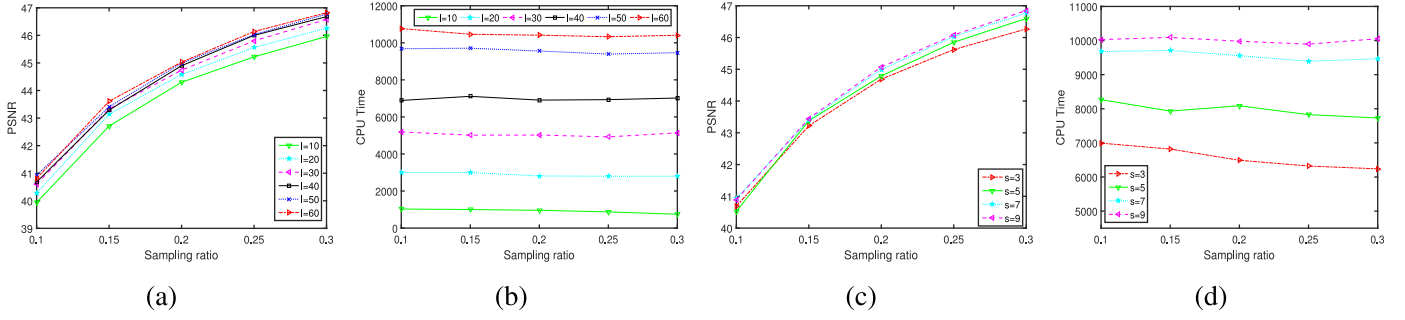
Fig. 2. Performance of different size of patches and number of similar patches for the Japser Ridge data with 5% sampling ratio. (a) PSNR values versus sampling ratios with different size of patches. (b) CPU time(s) versus sampling ratios with different size of patches. (c) PSNR values versus sampling ratios with different number of similar patches. (b) CPU time(s) versus sampling ratios with different number of similar patches.

CPU time (in seconds) while the performance improves only a little. Fig. 2 shows the performance of the Patched-t-SVD with different l and s for the Japser Ridge data, where SR = 5%. From Fig. 2(a) and (b), we can see that large l takes more CPU time while the improvement of PSNR values is very small. Moreover, we test different size of patches with $l = 50$. It can be seen from Fig. 2(c) and (d) that the performance with different size of patches ($s = 3, 5, 7, 9$) is almost the same in terms of PSNR values, while large s takes more CPU time. In order to balance the performance of PSNR values and CPU time, the size of each patch is set to be $s = 7$ in all experiments for the Patched-t-SVD method. The number of similar patches in each sub-tensor is set to 50, i.e., $l = 50$. Therefore, the size of each sub-tensor for all testing data is $49 \times 50 \times n_3$.

4.3. Hyperspectral data

In this subsection, we test the hpyerspectral data sets: including Japser Ridge, Urban[10], Hyperspectral Images of Natural Scenes 2004[11] (Scenes 5 and 7). For the Japser Ridge, a subimage with size 100×100 is recorded. Due to dense water vapor and atmospheric effects, we only remain 198 channels. Therefore, the size of the resulting tensor is $100 \times 100 \times 198$. For the Urban data set, a subimage with 162 spectral bands is chosen, where the size of each band is 105×105. Then the testing tensor of Urban is $105 \times 105 \times 162$. For the hyperspectral images of Natural Scenes 2004, to improve the computational time, we resize the original images of each band to 100×100. Thus, the sizes of the resulting tensors are $100 \times 100 \times 32$ for Scene 5 and $100 \times 100 \times 33$ for Scene 7, respectively.

We give the distributions of all singular value of the hyperspectral data by t-SVD (FFT), t-SVD (Wavelet), t-SVD (Data), and Patched-t-SVD. Let $(\mathcal{X}_0)_i$ be the ith sub-tensor with $(\mathcal{X}_0)_i = \mathcal{U}_i \diamond_{\mathbf{U_i}} \mathcal{S}_i \diamond_{\mathbf{U_i}} \mathcal{V}_i^H$. We show the percentages of all values of $\widehat{\mathcal{S}}_i(j,k)$ in different ranges for $1 \leq j \leq \text{rank}_{tt}((\mathcal{X}_0)_i)$, $1 \leq k \leq n_3$, and all i. These values can be interpreted as the tensor singular values of each sub-tensor $(\mathcal{X}_0)_i$ based on t-SVD in the transformed domain. For t-SVD (FFT), t-SVD (Wavelet), t-SVD (Data), let $\mathcal{X}_0 = \mathcal{U} \diamond \mathcal{S} \diamond \mathcal{V}^H$ be the results by different transforms. In Fig. 3, we show the percentages of all singular values $\widehat{\mathcal{S}}(j,j,k)$ in different ranges for $1 \leq j \leq \text{rank}_{tt}((\mathcal{X}_0))$, $1 \leq k \leq n_3$. It can be seen that the percentages of small singular values in $\widehat{\mathcal{S}}_i(j,j,k)$ are much bigger than those in $\widehat{\mathcal{S}}$. When discarding these small tensor singular values, the resulting patched-tubes tensor may have lower tubal rank since the percentage of large singular value is much smaller. For the robust tensor completion problem, we minimize the tensor nuclear norm

and the recovered tensor can be achieved by truncating small tensor singular values and obtaining a lower tubal rank tensor in each sub-tensor. Therefore, the Patched-t-SVD may be more effective than t-SVD (FFT), t-SVD (Wavelet), and t-SVD (Data).

Fig. 4 shows the recovered images and corresponding zoomed regions of the 20th and 160th bands by different methods with 10% sampling ratio for the Japser Ridge data set. It can be seen that the visual quality of the Patched-t-SVD is better than that of other methods, especially for low frequency bands, e.g., the 20th band. Moreover, the visual quality of t-SVD (Data) is better than that of HaLRTC, TNCP, TMac, t-SVD (FFT), and t-SVD (Wavelet).

In addition, the recovered images and corresponding zoomed regions of the 10th band by different methods are displayed in Fig. 5 for Scene 7 with 10% sampling ratio. The performance of the Patched-t-SVD is slightly better than the other methods in terms of visual quality. Besides, the visual quality of the recovered images obtained by t-SVD (Data) is better than those obtained by HaLRTC, TNCP, TMac, t-SVD (FFT) and t-SVD (Wavelet).

The PSNR values with different bands by SNN, t-SVD (FFT), t-SVD (Wavelet), t-SVD (Data), and Patched-t-SVD are shown in Fig. 6 for the Urban data with 60% sampling ratio and 30% corrupted entries. We can observe that the PSNR values of each band obtained by the Patched-t-SVD are higher than those obtained by SNN, t-SVD (FFT), t-SVD (Wavelet), and t-SVD (Data). The PSNR values of t-SVD (Data) are higher than those of SNN, t-SVD (FFT), and t-SVD (Wavelet) for most bands.

Fig. 7 shows the PSNR values versus sampling ratios of different methods for the hyperspectral data sets. It can be seen that the PSNR values obtained by the Patched-t-SVD are higher than those obtained by other methods for different sampling ratios, especially for Scenes 5 and 7. The performance of t-SVD (FFT) and t-SVD (Wavelet) is almost the same for Japser Ridge and Urban data in terms of PSNR values. Furthermore, the PSNR values obtained by the t-SVD (Data) are higher than those obtained by HaLRTC, TNCP, TMac, t-SVD (FFT), and t-SVD (Wavelet).

The PSNR and SSIM values of different methods are displayed in Table 1 for the hyperspectral data with different sampling ratios and γ in robust tensor completion. It can be seen that the PSNR and SSIM values obtained by the Patched-t-SVD are higher than those obtained by SNN, t-SVD (FFT), t-SVD (Wavelet), and t-SVD (Data). Moreover, the t-SVD (Data) outperforms SNN, t-SVD (FFT), t-SVD (Wavelet) in terms of PSNR and SSIM values for the testing cases. Compared with t-SVD (FFT), the t-SVD (Wavelet) performs better in terms of PSNR and SSIM values for small γ (e.g., 0.1) except Scene 7.

4.4. Video data

In this subsection, we test three video data (length $\times$ width $\times$ frames) to demonstrate the performance

---

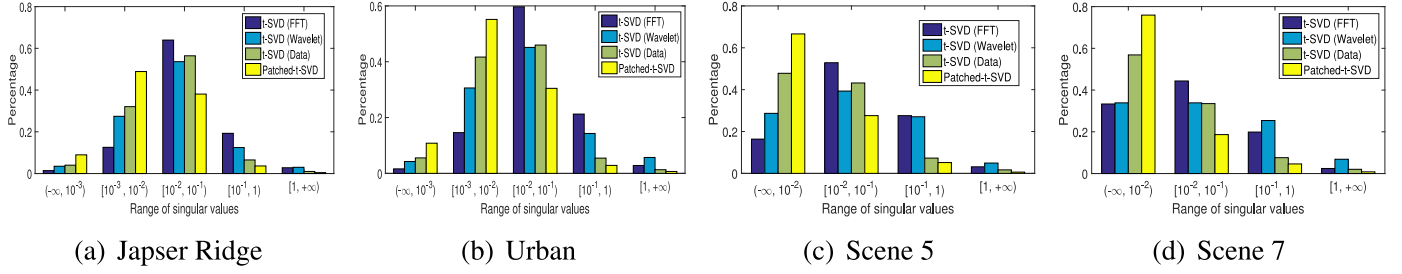(a) Japser Ridge (b) Urban (c) Scene 5 (d) Scene 7

Fig. 3. Distributions of singular values of all frontal slices by different methods for the hyperspectral datasets. (a) Japser Ridge. (b) Urban. (c) Scene 5. (d) Scene 7.
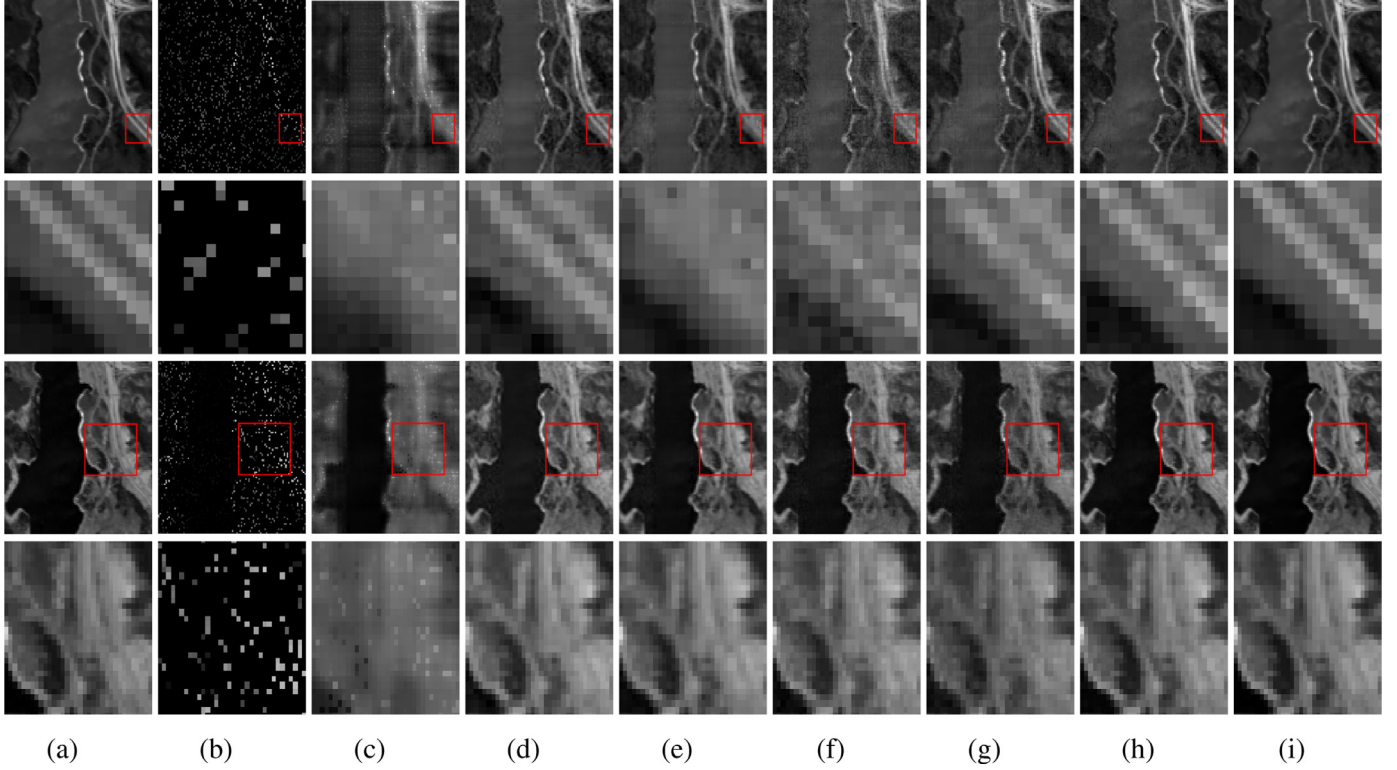

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 4. Recovered images and corresponding zoomed regions with the 20th and 160th bands (from top to bottom) by different methods for the Japser Ridge data set with 10% sampling ratio. (a) Original. (b) Observed. (c) HaLRTC. (d) TMac. (e) TNCP. (f) t-SVD (FFT). (g) t-SVD (Wavelet). (h) t-SVD (Data). (i) Patched-t-SVD.

(a) (b) (c) (d) (e) (f) (g) (h) (i)
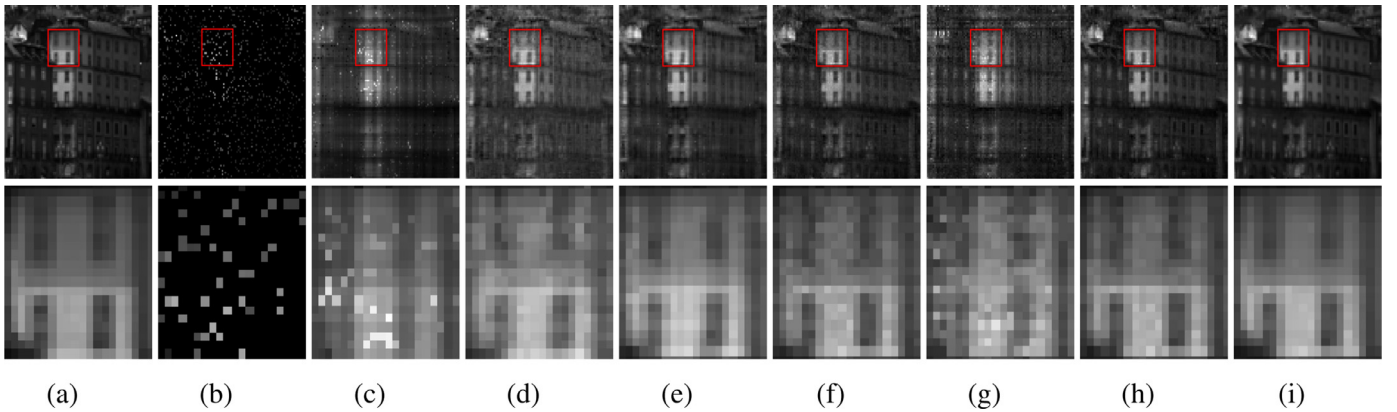
Fig. 5. The 10th band of the recovered images and corresponding zoomed regions by different methods for Scene 7 with 10% sampling ratio. (a) Original. (b) Observed. (c) HaLRTC. (d) TMac. (e) TNCP. (f) t-SVD (FFT). (g) t-SVD (Wavelet). (h) t-SVD (Data). (i) Patched-t-SVD.
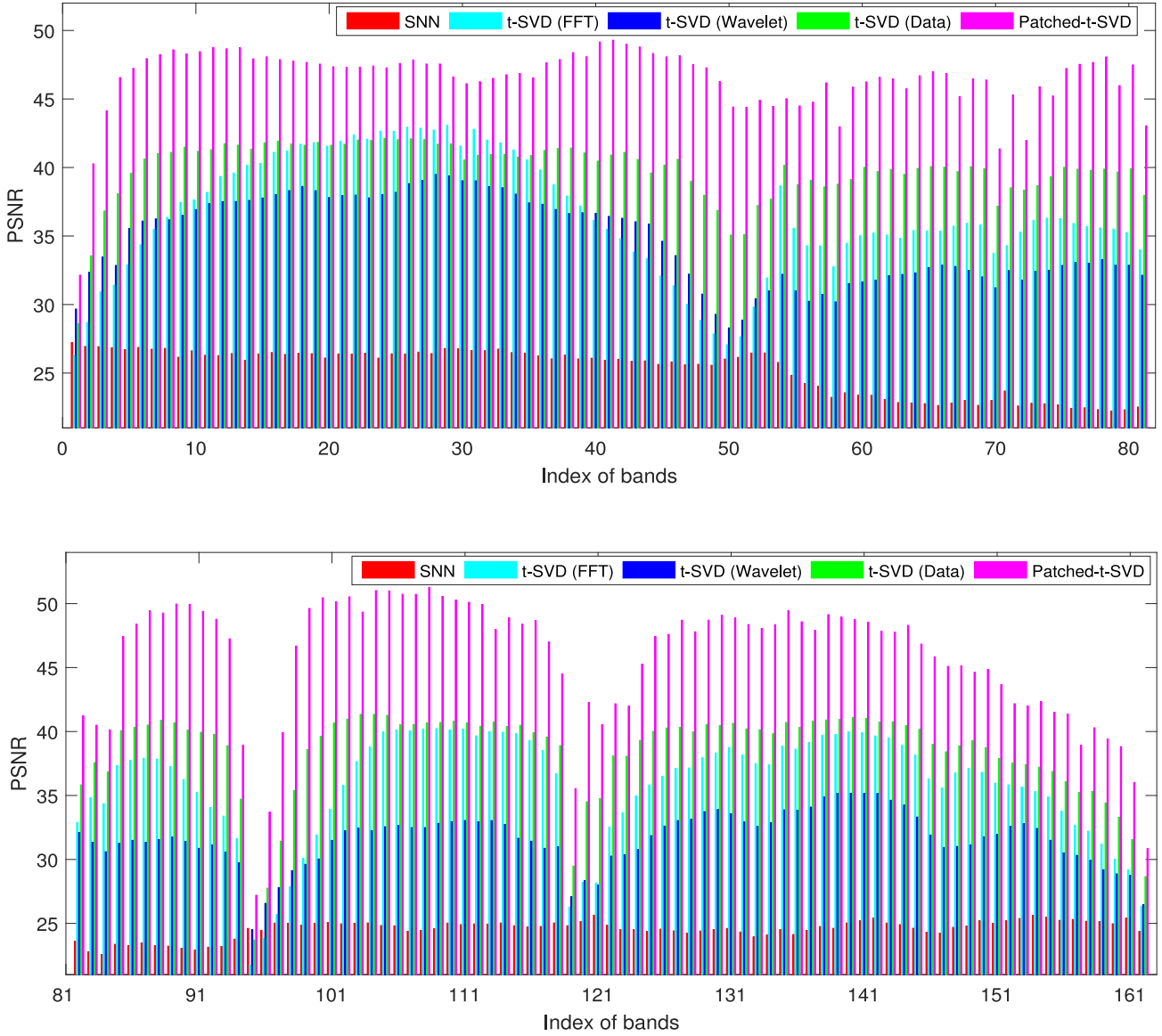
Fig. 6. PSNR values of each band by different methods for the Urban data with 60% sampling ratio and 30% corrupted entries.

(a) Japser Ridge   (b) Urban   (c) Scene 5   (d) Scene 7
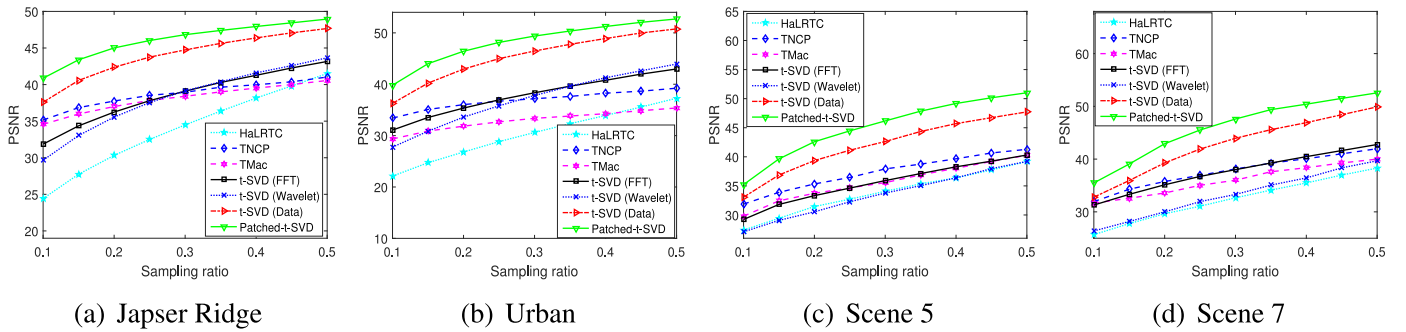
Fig. 7. PSNR values versus sampling ratios by different methods for the hyperspectral data. (a) Japser Ridge. (b) Urban. (c) Scene 5. (d) Scene 7.

of the Patched-t-SVD. In order to improve the computational time, we only use parts of frames of the whole data. The testing videos include Carphone (144 × 176 × 180), Announcer (144 × 176 × 200), and Grandma (144 × 176 × 180)[12].

Similar to the case of hyperspectral data, we generate all singular values of the original tensor in the transformed domain by t-

SVD (FFT), t-SVD (Wavelet), t-SVD (Data), and Patched-t-SVD. Fig. 8 shows the distributions of all singular values of t-SVD (FFT), t-SVD (Wavelet), t-SVD (Data), and Patched-t-SVD in different ranges for the video data. We can see that most of the singular values of Patched-t-SVD lie on the smallest ranges. The percentages of singular values of t-SVD (Data) and Patched-t-SVD in large ranges are smaller than those of t-SVD (FFT) and t-SVD (Wavelet). Therefore, the tubal ranks of the testing video data by Patched-t-SVD

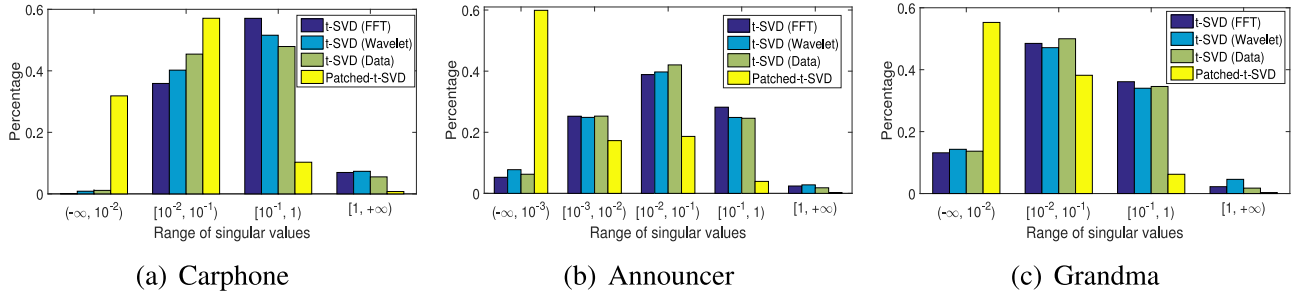[12] https://media.xiph.org/video/derf/.

Fig. 8. Distributions of singular values of all frontal slices by different methods for the video data sets. (a) Carphone. (b) Announcer. (c) Grandma.
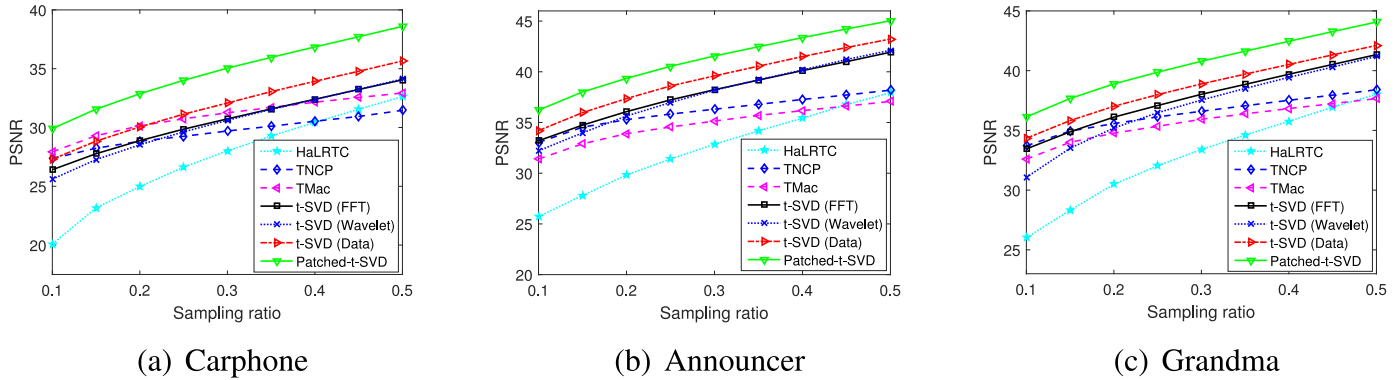

Fig. 9. PSNR values versus sampling ratios by different methods for the video data. (a) Carphone. (b) Announcer. (c) Grandma.
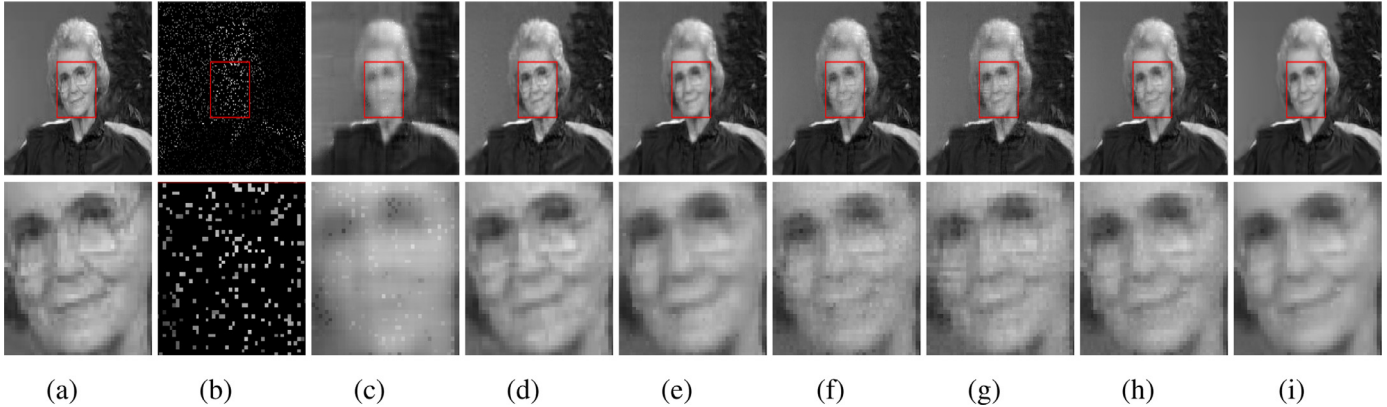

Fig. 10. The 5th frame of recovered images and corresponding zoomed regions by different methods for Grandma data with 10% sampling ratio. (a) Original. (b) Observed. (c) HaLRTC. (d) TMac. (e) TNCP. (f) t-SVD (FFT). (g) t-SVD (Wavelet). (h) t-SVD (Data). (i) Patched-t-SVD.

are much lower than those of other methods. Moreover, the tubal ranks of the three video data by t-SVD (Data) are lower than those of t-SVD (FFT) and t-SVD (Wavelet).

The PSNR values versus sampling ratios of different methods for the testing video data are shown in Fig. 9. It can be seen that the Patched-t-SVD has a great improvement compared with the t-SVD (Data) in terms of PSNR values for different sampling ratios. The PSNR values obtained by the Patched-t-SVD are higher than those obtained by HaLRTC, TNCP, TMac, t-SVD (FFT), t-SVD (Wavelet), and t-SVD (Data). The t-SVD (Data) outperforms HaLRTC, TNCP, TMac, t-SVD (FFT), and t-SVD (Data) in terms of PSNR values for most cases.

Fig. 10 shows the recovered images and corresponding zoomed regions of the 5th frame by different methods for the Grandma data with 10% sampling ratio. It can be observed that the visual quality of the Patched-t-SVD is better than other methods. The Patched-t-SVD method keeps more details of the Grandma data than HaLRTC, TNCP, TMac, t-SVD (FFT), t-SVD (Wavelet), and t-SVD

(Data). In addition, the t-SVD (Data) performs better than HaLRTC, TNCP, TMac, t-SVD (FFT), and t-SVD (Wavelet) in terms of visual quality.

The PSNR and SSIM values of different methods are shown Table 2 for the video data sets in robust tensor completion. It can be seen that the PSNR and SSIM values obtained by the Patched-t-SVD are much higher than those obtained by SNN, t-SVD (FFT), t-SVD (Wavelet), and t-SVD (Data) for different sampling ratios and γ. Moreover, the t-SVD (Data) outperforms SNN, t-SVD (FFT), and t-SVD (Wavelet) in terms of PSNR and SSIM values. The t-SVD (Wavelet) performs better than the t-SVD (FFT) in terms of PSNR and SSIM values for the small γ, e.g., 0.1.

The visual comparisons of the 15th frame by different methods for the video data are shown in Fig. 11, where the sampling ratio of the observed video is 80% and γ is 30%. We can observe that the images obtained by the Patched-t-SVD are more clear and keep more details than those obtained by SNN, t-SVD (FFT), t-SVD (Wavelet), and t-SVD (Data). Furthermore, the t-SVD (Data) out-
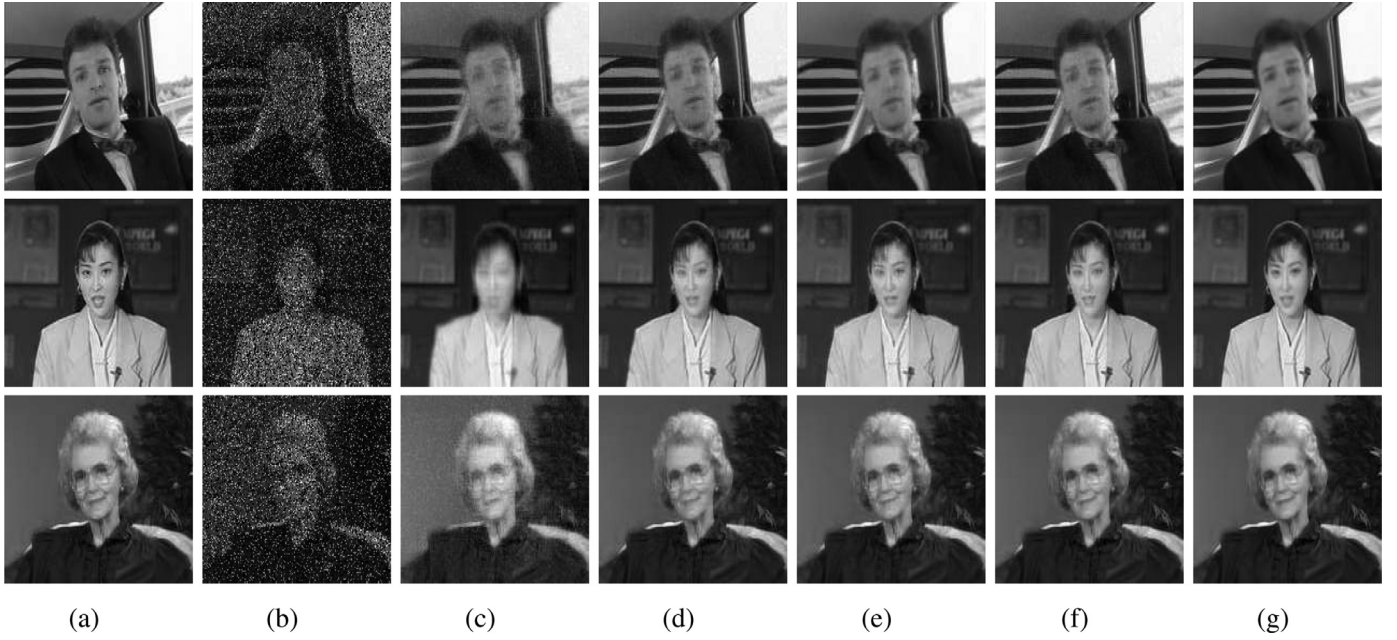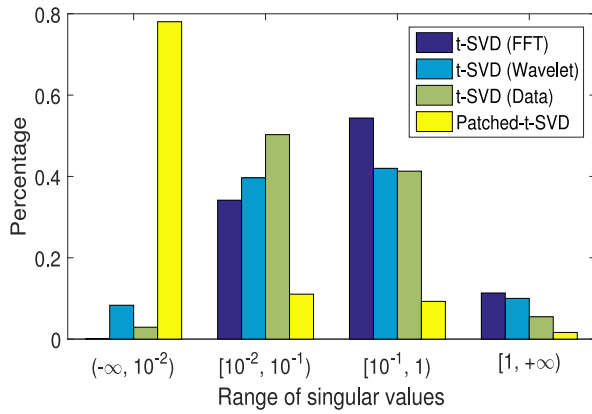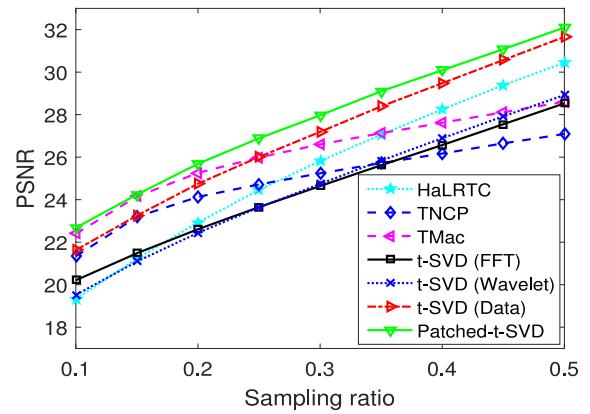
Fig. 11. The 15th frame of the recovered images by different methods for the video data sets with 80% sampling ratio and 30% corrupted entries. From top to bottom: Carphone, Announcer, and Grandma.(a) Original. (b) Observed. (c) SNN. (d) t-SVD (FFT). (e) t-SVD (Wavelet). (f) t-SVD (Data). (g) Patched-t-SVD.

Fig. 12. (a) Distribution of singular values of all frontal slices by using different methods for the extended Yale face database B. (b) PSNR values versus sampling ratios of different methods for the extended Yale face database B.

performs SNN, t-SVD (FFT), and t-SVD (Wavelet) in terms of visual quality for the testing video data.

4.5. Face data

In this subsection, we use the extended Yale face database B[13] to test the robust tensor completion problem, where the data set contains 16,128 images of 28 human subjects under 9 poses (252 subjects) and 64 illumination conditions. To improve the computation time, we crop each image to keep the central area containing the face, and then resize it to 73×55. We only choose the images of the first 30 subjects and 25 illumination conditions in our test for this database. Therefore, the size of the testing tensor is $73 \times 55 \times 750$.

Similar to the case of hyperspectral data, we generate all singular values of the extended Yale face database B in the transformed domain by t-SVD (FFT), t-SVD (Wavelet), t-SVD (Data), and Patched-t-SVD. The distribution of all singular values by different

methods is shown in Fig. 12(a). It can be observed that about 80% singular values of the Patched-t-SVD method are less than 0.01, which is much larger than that of t-SVD (FFT), t-SVD (Wavelet), and t-SVD (Data). This implies that the ranks of frontal slices of the tensor by Patched-t-SVD are lower than those by t-SVD (FFT), t-SVD (Wavelet), and t-SVD (Data) by discarding some small singular values.

Fig. 12(b) shows the PSNR values versus sampling ratios of different methods for tensor completion. It can be seen that the Patched-t-SVD is better than the other methods in terms of PSNR values for different sampling ratios. Moreover, the t-SVD (Data) outperforms HaLRTC, TNCP, t-SVD (FFT), and t-SVD (Wavelet) in terms of PSNR values. The PSNR values obtained by TMac are higher than those obtained by the t-SVD (Data) only when the sampling ratio is low, e.g., 0.1 to 0.25.

In Table 3, we list the PSNR and SSIM values of different methods for the extended Yale face database B in robust tensor completion. We can see that the Patched-t-SVD obtains highest PSNR and SSIM values compared with SNN, t-SVD (FFT), t-SVD (Wavelet), and t-SVD (Data) for different sampling ratios and γ. The PSNR val-

---

[13] http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html.

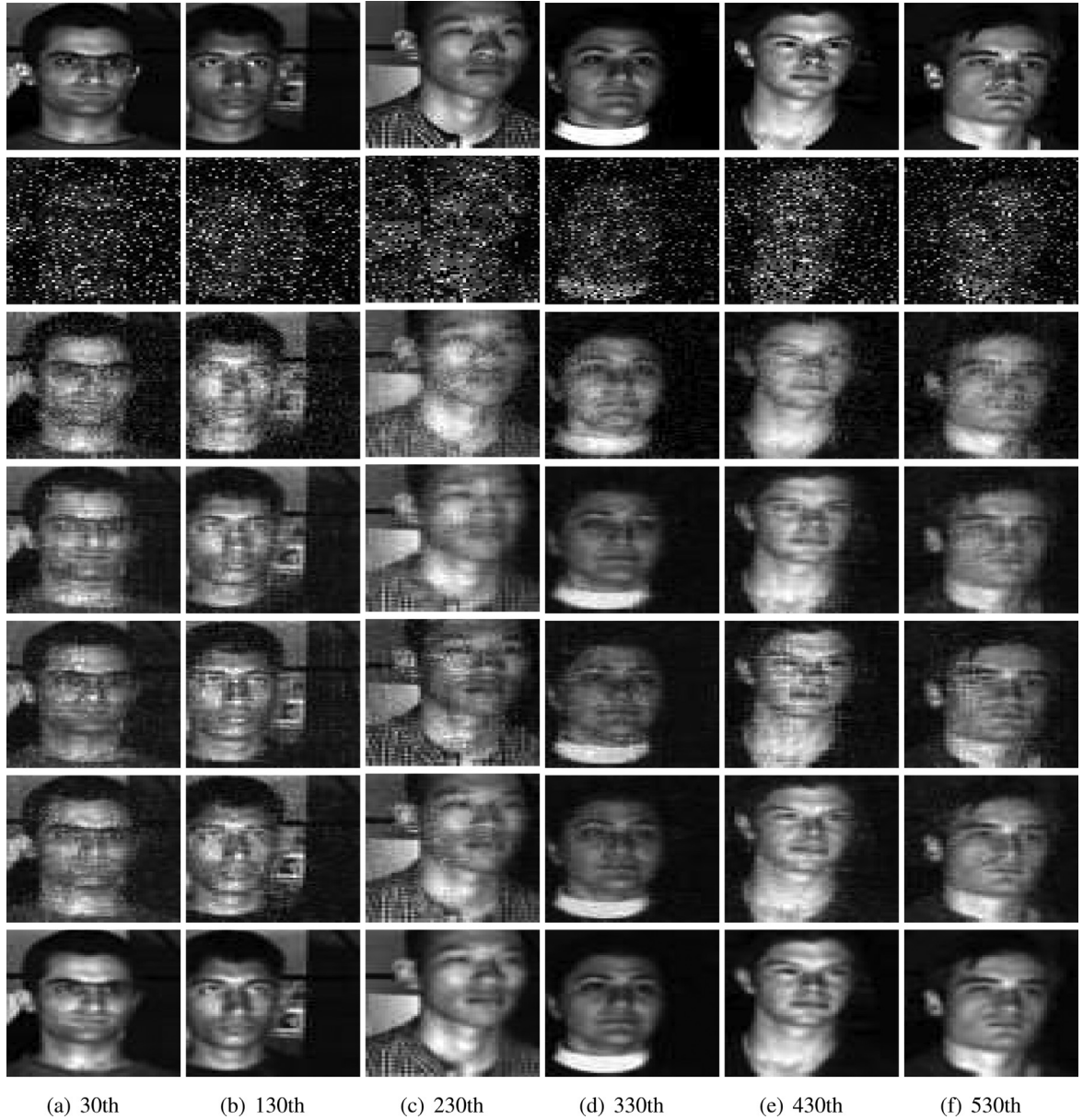| (a) 30th | (b) 130th | (c) 230th | (d) 330th | (e) 430th | (f) 530th |

Fig. 13. Recovered images with different subjects by different methods for the extended Yale face database B with 80% sampling ratio and 30% corrupted entries. From top to bottom: Original images, observed images, recovered images by SNN, recovered images by t-SVD (FFT), recovered images by t-SVD (Wavelet), recovered images by t-SVD (Data), and recovered images by Patched-t-SVD.

ues obtained by the Patched-t-SVD can be improved around 3 to 4dB than those obtained by the t-SVD (Data). Moreover, the t-SVD (Data) outperforms SNN, t-SVD (FFT), and t-SVD (Wavelet) in terms of PSNR and SSIM values.

The visual comparisons by different methods for the extended Yale face database B with the 30th, 130th, 230th, 330th, 430th, 530th images are shown in Fig. 13 in robust tensor completion, where the sampling ratio is 80% and $\gamma = 30\%$. It can be observed that the Patched-t-SVD can get more clear images and keep more details compared with SNN, t-SVD (FFT), t-SVD (Wavelet), and t-SVD (Data). Furthermore, the t-SVD (Data) performs better than SNN, t-SVD (FFT), and t-SVD (Wavelet) in terms of visual quality.

5. Concluding remarks

Inspired from the insights of the unitary transform method for robust tensor completion [28], in this paper, we have proposed a Patched-t-SVD for robust tensor completion. The Patched-t-SVD consists of the following steps: First, we estimate an initial estimator by the t-SVD (Data) for robust tensor completion. Then we search all patched-tubes of the initial estimator, and group the similar patches to form a third-order sub-tensor via unfolding the similar patches into matrices along the second dimension. The unitary matrices for the transformed t-SVD in each sub-tensor can be obtained by the SVDs of the unfolding matrices of the generated sub-tensors along the third-dimension. Then the robust tensor completion problem of each sub-tensor can be formulated. We also establish some perturbation results of transformed t-SVD for the similar patches. An sGS-ADMM is developed to solve the resulting model in each sub-tensor. Our preliminary numerical experiments on many real-world data sets show the proposed Patched-t-SVD outperforms several state-of-the-art approaches.

By leveraging on global low-rankness and nonlocal similarity of a tensor, the proposed Patched-t-SVD could get a lower rank sub-tensor. We have shown that the sub-tensor can be regarded a low tubal rank tensor with a small perturbation. Though numerical

experiments show the effectiveness of Patched-t-SVD, we do not analyze how much it can be improved in theory compared with the transformed t-SVD method [28]. In the future work, it would be of great interest to analyze the improvement of the Patched-t-SVD compared with the t-SVD (Data) (cf. [37]), which would be helpful to understand the performance of Patched-t-SVD. It would be also of great interest to study the exact recovery conditions and sampling size requirement of Pathced-t-SVD for successful recovery (cf. [28]). Due to the limitations of transformed t-SVD, which is just useful for third-order tensors, our Patched-t-SVD method is only effective for third-order tensors. One of the potential future work is to extend the transformed t-SVD from third-order tensors to higher-order tensors. Moreover, since the sGS-ADMM is linearly convergent, the efficiency may be challenged for large-scale data sets. Therefore, future work will be devoted to designing a faster and more efficient algorithm by exploiting the second-order information of the ℓ_1 norm and transformed tensor nuclear norm of a tensor.

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.

Acknowledgments

References

[1] B. Qin, M. Jin, D. Hao, Y. Lv, Q. Liu, Y. Zhu, S. Ding, J. Zhao, B. Fei, Accurate vessel extraction via tensor completion of background layer in X-ray coronary angiograms, Pattern Recognit. 87 (2019) 38–54.

[2] Y. Wang, D. Meng, M. Yuan, Sparse recovery: from vectors to tensors, Natl. Sci. Rev. 5 (5) (2018) 756–767.

[3] M.K. Ng, Q. Yuan, L. Yan, J. Sun, An adaptive weighted tensor completion method for the recovery of remote sensing images with missing data, IEEE Trans. Geosci. Remote Sens. 55 (6) (2017) 3367–3381.

[4] 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.

[5] X. Zhang, A nonconvex relaxation approach to low-rank tensor completion, IEEE Trans. Neural Netw. Learn. Syst. 30 (6) (2019) 1659–1671.

[6] M. Signoretto, Q.T. Dinh, L. De Lathauwer, J.A. Suykens, Learning with tensors: a framework based on convex optimization and spectral regularization, Mach. Learn. 94 (3) (2014) 303–351.

[7] B. Du, M. Zhang, L. Zhang, R. Hu, D. Tao, PLTD: patch-based low-rank tensor decomposition for hyperspectral images, IEEE Trans. Multimed. 19 (1) (2017) 67–79.

[8] L. Zhang, L. Zhang, D. Tao, B. Du, A sparse and discriminative tensor to vector projection for human gait feature representation, Signal Process. 106 (2015) 245–252.

[9] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (3) (2009) 455–500.

[10] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM 58 (3) (2011) 11.

[11] B. Recht, M. Fazel, P.A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, SIAM Rev. 52 (3) (2010) 471–501.

[12] J.D. Carroll, J.-J. Chang, Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition, Psychometrika 35 (3) (1970) 283–319.

[13] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311.

[14] I.V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput. 33 (5) (2011) 2295–2317.

[15] M.E. Kilmer, C.D. Martin, Factorization strategies for third-order tensors, Linear Algebra Appl. 435 (3) (2011) 641–658.

[16] C.J. Hillar, L.-H. Lim, Most tensor problems are NP-hard, J. ACM 60 (6) (2013) 45.

[17] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 208–220.

[18] B. Romera-Paredes, M. Pontil, A new convex relaxation for tensor completion, in: Proceedings Advances Neural Information Processing Systems 26, 2013, pp. 2967–2975.

[19] C. Mu, B. Huang, J. Wright, D. Goldfarb, Square deal: lower bounds and improved relaxations for tensor recovery, in: Proceedings International Conference Machine Learning, 32, 2014, pp. 73–81.

[20] J.A. Bengua, H.N. Phien, H.D. Tuan, M.N. Do, Efficient tensor completion for color image and video recovery: low-rank tensor train, IEEE Trans. Image Process. 26 (5) (2017) 2466–2479.

[21] K. Braman, Third-order tensors as linear operators on a space of matrices, Linear Algebra Appl. 433 (7) (2010) 1241–1253.

[22] C.D. Martin, R. Shafer, B. Larue, An order-p tensor factorization with applications in imaging, SIAM J. Sci. Comput. 35 (1) (2013) A474–A490.

[23] E. Kernfeld, M. Kilmer, S. Aeron, Tensor–tensor products with invertible linear transforms, Linear Algebra Appl. 485 (2015) 545–570.

[24] O. Semerci, N. Hao, M.E. Kilmer, E.L. Miller, Tensor-based formulation and nuclear norm regularization for multienergy computed tomography, IEEE Trans. Image Process. 23 (4) (2014) 1678–1693.

[25] W. Hu, D. Tao, W. Zhang, Y. Xie, Y. Yang, The twist tensor nuclear norm for video completion, IEEE Trans. Neural Netw. Learn. Syst. 28 (12) (2017) 2961–2973.

[26] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, S. Yan, Tensor robust principal component analysis with a new tensor nuclear norm, IEEE Trans. Pattern Anal. Mach. Intell. (2019), doi:10.1109/TPAMI.2019.2891760.

[27] Z. Zhang, S. Aeron, Exact tensor completion using t-SVD, IEEE Trans. Signal Process. 65 (6) (2017) 1511–1526.

[28] G. Song, M.K. Ng, X. Zhang, Robust tensor completion by using transformed tensor SVD, (2019) arXiv:1907.01113v1.

[29] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.

[30] Q. Xie, Q. Zhao, D. Meng, Z. Xu, Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery, IEEE Trans. Pattern Anal. Mach. Intell. 40 (8) (2018) 1888–1902.

[31] X.-T. Li, X.-L. Zhao, T.-X. Jiang, Y.-B. Zheng, T.-Y. Ji, T.-Z. Huang, Low-rank tensor completion via combined non-local self-similarity and low-rank regularization, Neurocomputing 367 (2019) 1–12.

[32] T. Xie, S. Li, L. Fang, L. Liu, Tensor completion via nonlocal low-rank regularization, IEEE Trans. Cybern. 49 (6) (2019) 2344–2354.

[33] G.H. Golub, C.F. Van Loan, Matrix Computations, 4th ed., The Johns Hopkins University Press, Maryland, Baltimore, 2013.

[34] L. Chen, D. Sun, K.-C. Toh, An efficient inexact symmetric Gauss-Seidel based majorized ADMM for high-dimensional convex composite conic programming, Math. Program. 161 (1–2) (2017) 237–270.

[35] X. Li, D. Sun, K.-C. Toh, A Schur complement based semi-proximal ADMM for convex quadratic conic programming and extensions, Math. Program. 155 (1–2) (2016) 333–373.

[36] M. Bai, X. Zhang, G. Ni, C. Cui, An adaptive correction approach for tensor completion, SIAM J. Imaging Sci. 9 (3) (2016) 1298–1323.

[37] X. Zhang, M.K. Ng, A corrected tensor nuclear norm minimization method for noisy low-rank tensor completion, SIAM J. Imaging Sci. 12 (2) (2019) 1231–1273.

[38] Y. Xu, R. Hao, W. Yin, Z. Su, Parallel matrix factorization for low-rank tensor completion, Inverse Probl. Imaging 9 (2) (2015) 601–624.

[39] Y. Liu, F. Shang, L. Jiao, J. Cheng, H. Cheng, Trace norm regularized CANDECOMP/PARAFAC decomposition with missing data, IEEE Trans. Cybern. 45 (11) (2015) 2437–2448.

[40] D. Goldfarb, Z. Qin, Robust low-rank tensor recovery: models and algorithms, SIAM J. Matrix Anal. Appl. 35 (1) (2014) 225–253.

[41] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.

[42] I. Daubechies, Ten Lectures on Wavelets, PA, Philadelphia: SIAM, 1992.

Michael K. Ng received the B.Sc. degree in 1990 and the M.Phil. degree in 1992 from The University of Hong Kong, and the Ph.D. degree in 1995 from The Chinese University of Hong Kong. He is currently a Chair Professor in the Department of Mathematics, The University of Hong Kong, Hong Kong. His research interests include bioinformatics, image processing, scientific computing and data mining.

Xiongjun Zhang received the Ph.D. degree in 2017 from Hunan University, Changsha, China. He is currently an Assistant Professor in the School of Mathematics and Statistics, Central China Normal University, Wuhan, China. His research interests include image processing and tensor optimization.

Xi-Le Zhao received the M.S. and Ph.D. degrees both in applied mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009 and 2012, respectively. He is currently a Professor in the School of Mathematical Sciences, UESTC. His current research interests include image processing, computer vision, and machine learning.