Computer Vision and Image Understanding 243 (2024) 104007

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Tensor robust PCA with nonconvex and nonlocal regularization

Xiaoyu Geng [a,b], Qiang Guo [a,b,*], Shuaixiong Hui [a,b], Ming Yang [c], Caiming Zhang [b,d]

[a] School of Computer Science and Technology, Shandong University of Finance and Economics, 250014, Jinan, China
[b] Shandong Provincial Key Laboratory of Digital Media Technology, 250014, Jinan, China
[c] College of Mathematical Sciences, Harbin Engineering University, 150001, Harbin, China
[d] School of Software, Shandong University, 250101, Jinan, China

ARTICLE INFO

Communicated by Xuelong Li

Keywords:
Low-rank property
Nonconvex surrogate
Nonlocal self-similarity
Tensor robust PCA

ABSTRACT

Tensor robust principal component analysis (TRPCA) is a classical way for low-rank tensor recovery, which minimizes the convex surrogate of tensor rank by shrinking each tensor singular value equally. However, for real-world visual data, large singular values represent more significant information than small singular values. In this paper, we propose a nonconvex TRPCA (N-TRPCA) model based on the tensor adjustable logarithmic norm. Unlike TRPCA, our N-TRPCA can adaptively shrink small singular values more and shrink large singular values less. In addition, TRPCA assumes that the whole data tensor is of low rank. This assumption is hardly satisfied in practice for natural visual data, restricting the capability of TRPCA to recover the edges and texture details from noisy images and videos. To this end, we integrate nonlocal self-similarity into N-TRPCA, and further develop a nonconvex and nonlocal TRPCA (NN-TRPCA) model. Specifically, similar nonlocal patches are grouped as a tensor and then each group tensor is recovered by our N-TRPCA. Since the patches in one group are highly correlated, all group tensors have strong low-rank property, leading to an improvement of recovery performance. Experimental results demonstrate that the proposed NN-TRPCA outperforms existing TRPCA methods in visual data recovery. The demo code is available at https://github.com/qguo2010/NN-TRPCA.

1. Introduction

As an important method for dimensionality reduction, principal component analysis (PCA) has received extensive attention in the fields of image processing and computer vision (Liang et al., 2023; Yayci et al., 2022; Wang et al., 2018). It is a powerful non-parametric tool to analyze the data corrupted with slight noise. Unfortunately, PCA is sensitive to outliers or large amounts of noise which are inevitably introduced into visual data during acquisition and transmission.

To alleviate this issue, robust PCA (RPCA) (Candès et al., 2011) was proposed to recover a low-rank matrix from its observation corrupted by sparse noise, in which the definition of matrix rank is unique. Since the matrix rank function is difficult to be minimized, RPCA adopts nuclear norm as its convex surrogate. Suppose that an observation matrix $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$ can be decomposed by $\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{E}$, where $\boldsymbol{L}$ is a low-rank matrix and $\boldsymbol{E}$ is a sparse matrix (noise), RPCA obtains $\boldsymbol{L}$ and $\boldsymbol{E}$ by solving the following minimization objective:

$$\min_{\boldsymbol{L},\boldsymbol{E}} \|\boldsymbol{L}\|_* + \lambda \|\boldsymbol{E}\|_1, \; s.t. \; \boldsymbol{X} = \boldsymbol{L} + \boldsymbol{E}, \tag{1}$$

where λ is a regularization parameter, $\|\boldsymbol{L}\|_*$ and $\|\boldsymbol{E}\|_1$ indicate the matrix nuclear norm of $\boldsymbol{L}$ and the ℓ_1-norm of $\boldsymbol{E}$, respectively. Problem (1) can be solved by the singular value thresholding (SVT) algorithm presented in Cai et al. (2010). Till now, RPCA and its extensions have plenty of applications, including image restoration/alignment (Oh et al., 2016), background foreground separation (Markowitz et al., 2022), and subspace clustering (Zhong and Pun, 2021). Nevertheless, RPCA can only deal with two-dimensional data. In real-world applications, high-dimensional data is growing explosively. Instead of matrix, tensor is the most appropriate representation of high-dimensional data. For instance, gray videos are three-dimensional tensors with row, column, and temporal modes, while color images are also three-dimensional tensors with size of $height \times width \times channel$. To handle these tensor data, one can apply the RPCA method on each frontal slice of tensors independently. But such a strategy will ignore the multidimensional structure information underlying the tensors. Therefore, it is natural to extend the RPCA from matrix domain to tensor domain.

Given an observed tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ that is a combination of a low-rank tensor $\boldsymbol{\mathcal{L}}$ and a sparse tensor $\boldsymbol{\mathcal{E}}$, i.e., $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{E}}$, tensor robust PCA (TRPCA) aims to estimate $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{E}}$ from tensor $\boldsymbol{\mathcal{X}}$. Unlike the matrix rank being unique, tensor rank has many definitions that are derived from different tensor decomposition methods. Tucker rank (Kolda and Bader, 2009) is induced for Tucker decomposition (Tucker, 1966), which is defined as a vector of the matrix rank unfolded along each mode of the given tensor. As minimizing Tucker rank is NP-hard, the sum of nuclear norm (SNN) (Liu et al., 2013) was presented as a

* Corresponding author at: School of Computer Science and Technology, Shandong University of Finance and Economics, 250014, Jinan, China.
E-mail address: guoqiang@sdufe.edu.cn (Q. Guo).

https://doi.org/10.1016/j.cviu.2024.104007
Received 11 July 2023; Received in revised form 14 February 2024; Accepted 25 March 2024
Available online 27 March 2024
1077-3142/© 2024 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

relaxation of the Tucker rank. Based on SNN, Huang et al. (2015) built a SNN-TRPCA model as follows,

$$\min_{\mathcal{L},\mathcal{E}} \sum_{i=1}^{n_3} \lambda_i \|\boldsymbol{L}^{\{i\}}\|_* + \|\mathcal{E}\|_1, \ s.t. \ \mathcal{X} = \mathcal{L} + \mathcal{E}, \tag{2}$$

where $\lambda_i > 0$ and $\boldsymbol{L}^{\{i\}}$ indicates the mode-i matricization (Kolda and Bader, 2009) of tensor $\mathcal{L}$. SNN-TRPCA exploits the low-rankness of tensor along each dimension. However, it is hard to set the weights λ_i due to the fact that the low-rankness of each dimension is usually different in real data. For example, the rank of a gray video along its temporal dimension is much lower than those along its spatial dimensions. Besides, the unfolding operation along one dimension could destroy the inherent structure information of tensors.

Recently, the tensor average rank (Lu et al., 2020) was defined by t-SVD (Kilmer and Martin, 2011), in which frontal slices of the tensor are arranged in a circulant way by using the block circulant matricization. As a result, this rank can preserve more structural information across frontal slices compared with Tucker rank. Since it is NP-hard to minimize the tensor average rank, the tensor nuclear norm (TNN) (Lu et al., 2020) is adopted as a surrogate of the tensor average rank. Based on TNN, a TNN-TRPCA model is introduced as follows:

$$\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_T + \lambda\|\mathcal{E}\|_1, \ s.t. \ \mathcal{X} = \mathcal{L} + \mathcal{E}, \tag{3}$$

where $\|\mathcal{L}\|_T$ denotes the TNN of $\mathcal{L}$ (See Definition 5 for details). Problem (3) can be efficiently solved by the alternating direction method of multipliers (ADMM) (Boyd et al., 2011), in which the tensor SVT (t-SVT) (Lu et al., 2020) is a key step. Mathematically, let $\mathcal{Y} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ be the t-SVD of $\mathcal{Y} \in \mathbb{R}^{n_1\times n_2\times n_3}$, for any $\tau > 0$, t-SVT is expressed as follows

$$\mathcal{D}_\tau(\mathcal{Y}) = \mathcal{U} * \mathcal{S}_\tau * \mathcal{V}^\top, \tag{4}$$

where

$$\mathcal{S}_\tau = \text{ifft}\big((\bar{\mathcal{S}} - \tau)_+, [\,], 3\big), \tag{5}$$

$\bar{\mathcal{S}}$ is the result of fast Fourier transform (FFT) on $\mathcal{S}$ along the 3rd dimension, and ifft is the inverse operator of FFT. It is easy to see that t-SVT shrinks each singular value equally according to the threshold τ. However, in practice, the tensor singular values often have different physical meanings. For a noisy color image, the large singular values usually correspond to the important information in the image, while the small singular values usually represent the noise. This motivates us to utilize different thresholds to shrink the small singular values more and the large ones less, so that the noise can be reduced precisely and the significant information can be preserved well. Besides, TNN-TRPCA has a potential limitation that it simply assumes the whole underlying tensor is of low rank. For visual data (e.g., natural images and videos), such an assumption is often difficult to be satisfied. Therefore, TNN-TRPCA cannot well recover the detail information in visual data, especially in data with complex structures.

For solving the above problems, we intend to propose a variant of TRPCA by shrinking tensor singular values differently and integrating nonlocal self-similarity. Specifically, to better preserve the important information of tensor data, a nonconvex TRPCA (N-TRPCA) model is built using tensor adjustable logarithmic norm as a nonconvex surrogate of the tensor average rank. It can apply adaptive thresholds for shrinking different tensor singular values. Then, the nonlocal self-similarity is further introduced into N-TRPCA to derive a nonconvex and nonlocal TRPCA (NN-TRPCA) model. By this way, our model can make full use of the structural redundancy of tensors to recover the detail information, resulting in remarkable performance improvements. In summary, our contributions are highlighted as follows:

- A nonconvex TRPCA (N-TRPCA) model under t-SVD framework is proposed for visual data recovery, which makes the large singular values shrink less and the small singular values shrink more simultaneously. Such a model can effectively preserve the important information in visual data.
- Beyond using the global low-rankness of tensors, nonlocal low-rank property is more crucial to fully utilize the structural redundancy in tensors. Thus, we further incorporate the nonlocal self-similarity into the N-TRPCA and then propose a nonconvex and nonlocal TRPCA model, named NN-TRPCA.
- To solve the proposed NN-TRPCA, we present an effective ADMM-based algorithm, by which the close-form solutions can be derived.
- We evaluate the efficacy of the N-TRPCA and NN-TRPCA in color image restoration and gray video restoration. Extensive experiment results confirm the superiority of our methods and show their competitive performance with the state-of-the-arts.

A preliminary conference version of this work was presented in Geng et al. (2021). We extend it both theoretically and experimentally. First, the recent works on the low-rank prior and the nonlocal prior of visual data are elaborated to provide theoretical basis for the proposed methods. Second, we offer a detailed and rigorous derivation for the closed-form solutions of our N-TRPCA algorithm. Third, we compare two methods of constructing group tensors and offer a new observation. Fourth, the proposed N-TRPCA and NN-TRPCA are evaluated in visual data restoration, where the data is corrupted with random noise. This restoration task is more challenging than denoising visual data corrupted by Gaussian white noise. Fifth, we analyze the influence of parameters in NN-TRPCA on our experiments. Finally, we discuss several possible future extensions of our NN-TRPCA.

The remainder of this paper is organized as follows. Some notations and preliminaries are given in Section 2. The recent works on the low-rankness and the nonlocal self-similarity are reviewed in Section 3. In Section 4, we propose the N-TRPCA and NN-TRPCA methods and design the corresponding optimization algorithms. Section 5 reports extensive experimental results. Finally, Section 6 draws a conclusion.

2. Preliminaries

For convenience of presentation, we first introduce some notations used in our work, and then list some basic definitions and theorems of the tensor algebra.

Third-order tensors are denoted as boldface calligraphic letters, e.g., $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, matrices are as boldface capital letters, e.g., $\boldsymbol{A} \in \mathbb{R}^{n_1\times n_2}$, vectors are as boldface lowercase letters, e.g., $\boldsymbol{a} \in \mathbb{R}^{n_1}$, and scalars are as lowercase letters, e.g., $a \in \mathbb{R}$. For a three-order tensor $\mathcal{A}$, we represent $\bar{\mathcal{A}}$ as the Fast Fourier Transform (FFT) of $\mathcal{A}$ along 3rd dimension using the Matlab command fft, i.e., $\bar{\mathcal{A}} = \text{fft}(\mathcal{A}, [\ ], 3)$, and achieve $\mathcal{A}$ by the inverse FFT, i.e., $\mathcal{A} = \text{ifft}(\bar{\mathcal{A}}, [\ ], 3)$. $\boldsymbol{A}^{(i)}$ and $\bar{\boldsymbol{A}}^{(i)}$ are respectively the ith frontal slice of $\mathcal{A}$ and $\bar{\mathcal{A}}$. Moreover, the inner product between $\mathcal{A}$ and $\mathcal{B}$ is represented as $\langle \mathcal{A}, \mathcal{B}\rangle = \sum_{i=1}^{n_3} \langle \boldsymbol{A}^{(i)}, \boldsymbol{B}^{(i)}\rangle$. The ℓ_1-norm, infinity norm, and Frobenius norm of $\mathcal{A}$ are defined as $\|\mathcal{A}\|_1 = \sum_{ijk} |a_{ijk}|$, $\|\mathcal{A}\|_\infty = \max_{ijk} |a_{ijk}|$, and $\|\mathcal{A}\|_F = \sqrt{\sum_{ijk} |a_{ijk}|^2}$, respectively, where a_{ijk} denotes the (i, j, k)th entry of $\mathcal{A}$. For a matrix $\boldsymbol{A}$, its nuclear norm is defined as the sum of singular values, i.e., $\|\boldsymbol{A}\|_* = \sum_i \sigma_i(\boldsymbol{A})$, where $\sigma_j(\cdot)$ is the ith largest singular value of $\boldsymbol{A}$.

Definition 1 (*Block Diagonal Matrix Lu et al., 2020*)**.** For $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, its block diagonal matrix is defined as

$$\bar{\boldsymbol{A}} = \text{bdiag}(\bar{\mathcal{A}}) = \begin{bmatrix} \bar{\boldsymbol{A}}^{(1)} & & & \\ & \bar{\boldsymbol{A}}^{(2)} & & \\ & & \ddots & \\ & & & \bar{\boldsymbol{A}}^{(n_3)} \end{bmatrix} \in \mathbb{C}^{n_1 n_3\times n_2 n_3}. \tag{6}$$

Definition 2 (*Block Circulant Matrix Lu et al., 2020*)**.** For $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, the block circulant matrix of $\mathcal{A}$ is

$$\text{bcirc}(\mathcal{A}) = \begin{bmatrix} \boldsymbol{A}^{(1)} & \boldsymbol{A}^{(n_3)} & \cdots & \boldsymbol{A}^{(2)} \\ \boldsymbol{A}^{(2)} & \boldsymbol{A}^{(1)} & \cdots & \boldsymbol{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}^{(n_3)} & \boldsymbol{A}^{(n_3-1)} & \cdots & \boldsymbol{A}^{(1)} \end{bmatrix} \in \mathbb{R}^{n_1 n_3\times n_2 n_3}. \tag{7}$$

Theorem 1 (*Diagonalization Lu et al., 2020*). *The block circulant matrix of $\mathcal{A}$ can be block diagonalized by the following equation:*

$$(\boldsymbol{F}_{n_3} \otimes \boldsymbol{I}_{n_1}) \cdot bcirc(\mathcal{A}) \cdot (\boldsymbol{F}_{n_3}^{-1} \otimes \boldsymbol{I}_{n_2}) = \bar{\mathbf{A}}, \tag{8}$$

where $\boldsymbol{F}_{n_3} \in \mathbb{C}^{n_3 \times n_3}$ is the discrete fourier transformation matrix, $\boldsymbol{I}_{n_1} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\boldsymbol{I}_{n_2} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are two identity matrices, $\otimes$ is the Kronecker product.

Definition 3 (*T-Product Kilmer and Martin, 2011*). Given $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times l \times n_3}$, the t-product $\mathcal{A} * \mathcal{B}$ is defined as a tensor with size $n_1 \times l \times n_3$,

$$\mathcal{A} * \mathcal{B} = \text{fold}\big(\text{bcirc}(\mathcal{A}) \cdot \text{unfold}(\mathcal{B})\big). \tag{9}$$

Using Theorem 1, the t-product can be transformed into the matrix multiplication in the Fourier domain, i.e., $\mathcal{A} * \mathcal{B} = \bar{\mathbf{A}}\bar{\mathbf{B}}$.

Theorem 2 (*T-SVD Kilmer and Martin, 2011*). *For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tensor singular value decomposition (t-SVD) of $\mathcal{A}$ is described by*

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^{\top}, \tag{10}$$

where $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor, $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are two orthogonal tensors.

Definition 4 (*Tensor Average Rank Lu et al., 2020*). Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tensor average rank of $\mathcal{A}$ is as:

$$\text{rank}_a(\mathcal{A}) = \frac{1}{n_3}\text{rank}\big(\text{bcirc}(\mathcal{A})\big). \tag{11}$$

Definition 5 (*Tensor Nuclear Norm (TNN) Lu et al., 2020*). Given a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tensor nuclear norm of $\mathcal{A}$ is depicted as

$$\begin{aligned}\|\mathcal{A}\|_T &= \frac{1}{n_3}\|\text{bcirc}(\mathcal{A})\|_* = \frac{1}{n_3}\sum_{i=1}^{n_3}\|\bar{\mathbf{A}}^{(i)}\|_* \\ &= \frac{1}{n_3}\sum_{i=1}^{n_3}\sum_{j=1}^{\min(n_1,n_2)}\sigma_j(\bar{\mathbf{A}}^{(i)}),\end{aligned} \tag{12}$$

which is a convex surrogate of the tensor average rank.

3. Related works

The most important issue of noise reduction from corrupted visual data is to fully use the underlying structure priors of the data. Both low-rank prior (Qin et al., 2022; Shi et al., 2021; Zhang et al., 2020; Guo et al., 2016) and nonlocal prior (Guo et al., 2018; Zhang et al., 2018; Dong et al., 2013, 2014) are two commonly-used structure priors for visual data recovery. In the following, we briefly review some related works.

3.1. Low-rank property

The low-rank prior indicates that the images have some structural redundancy, i.e., repeating structures. Numerous works (Zhang et al., 2020; Zarmehi et al., 2020; Guo et al., 2016) have employed this prior for the task of image recovery. Generally, they transform the image recovery into a matrix rank minimization problem. For instance, a gray image can be represented as a low-rank matrix. Similarly, a color image can be approximated by a low-rank matrix on RGB channels independently. Although the matrix rank is capable to characterize the global information in matrices, it is NP-hard to be minimized. Thus, many researchers have attempted to find an appropriate surrogate of the matrix rank. Candès and Recht (2009) originally used the matrix nuclear norm (MNN) as a convex surrogate of the matrix rank. In Candès and Tao (2010), it is theoretically proved that MNN is the best convex approximation of the matrix rank. Due to its convexity, the MNN minimization can be efficiently solved by the SVT algorithm (Cai et al., 2010) and has a global optimal solution. However, MNN ignores the difference between singular values. For visual data, the large singular values contain more important information than the small singular values. Thus, several variants of MNN were developed to treat the singular values in different manners. Matrix truncated nuclear norm (MTNN) (Hu et al., 2013) shrinks only the smallest singular values. Although MTNN achieves better performance than MNN, it ignores the fact that the large singular values contain a small amount of noise. To adaptively shrinks the singular values, Gu et al. (2017) proposed a matrix weighted nuclear norm (MWNN) and derived a weighted SVT (WSVT) algorithm for effectively minimizing the MWNN. By setting the weights to decrease as the singular value increases, the large singular values can be shrunk less, while the small singular values can be shrunk more. But it is troublesome to estimate a set of reasonable weights. Recently, by using logarithm function, a nonconvex approximation of the matrix rank (Nie et al., 2019) was proposed to adaptively estimate these weights. Since the estimated weights decrease as the singular values increase, this nonconvex surrogate can simultaneously increase the shrinkage on small singular values and reduce the shrinkage on large singular values.

One shortcoming of the matrix rank based methods is that they cannot preserve the correlations across frontal slices of multidimensional visual data, such as the correlations across RGB channels of color images or the correlations across frames of gray videos. Instead of matrix, tensor is an effective representation form for visual data without loss of its structural information. Hence, approximating visual data directly by low-rank tensors has gained significant popularity in recent years. Nevertheless, the definition of tensor rank is nonunique. The commonly-used definitions are CP rank (Kolda and Bader, 2009), Tucker rank (Kolda and Bader, 2009), and tensor average rank (Lu et al., 2020). CP rank is defined by the smallest number of tensor rank-one decomposition. But predefining CP rank is a challenge. As discussed in Section 1, the Tucker rank and tensor average rank are NP-hard to be minimized. Thus, SNN (Liu et al., 2013) and TNN (Lu et al., 2020) were used as convex approximations of the Tucker rank and tensor average rank, respectively. However, both SNN and TNN neglect the differences between tensor singular values. As a consequence, they cannot well preserve the important information of tensor data. In order to address this issue, several nonconvex surrogates of tensor rank (Xue et al., 2019; Xu et al., 2019a; Cai et al., 2019) were presented to treat the tensor singular values differently.

3.2. Nonlocal self-similarity

The nonlocal self-similarity of images is another important prior. Compared with global low-rankness, it can capture detailed structural redundancy, resulting in more accurate recovery results. Buades et al. (2005) firstly applied the nonlocal self-similarity to gray image denoising, and presented the nonlocal mean (NLM) filter. Concretely, NLM restores each pixel via nonlocal averaging of the pixels in its neighborhood, where the weights for a pixel reflect the similarity of other pixels with it. Another representative work is block matching and 3-D filtering (BM3D) (Dabov et al., 2007), which groups similar 2-D image patches as 3-D tensors and handles these tensors with sparse collaborative filtering. In essence, BM3D combines the sparsity and nonlocal self-similarity of images for image denoising. Furthermore, Dong et al. (2013) proposed nonlocally centralized sparse representation (NCSR) to learn the sparse coding of nonlocal redundancy in the images, and then utilized NCSR to deal with several image restoration tasks. In a recent decade, there has been a growing interest in using both nonlocal self-similarity and low-rankness of matrix in the field of image processing. Nonlocal low-rank regularization compressed sensing (NLR-CS) (Dong et al., 2014) introduces the nonlocal self-similarity into compressed sensing recovery. It groups similar 2-D image patches into a matrix and handles the group matrix by low-rank regularization. Due to the similar patches in one group having a strong correlation, each group matrix is

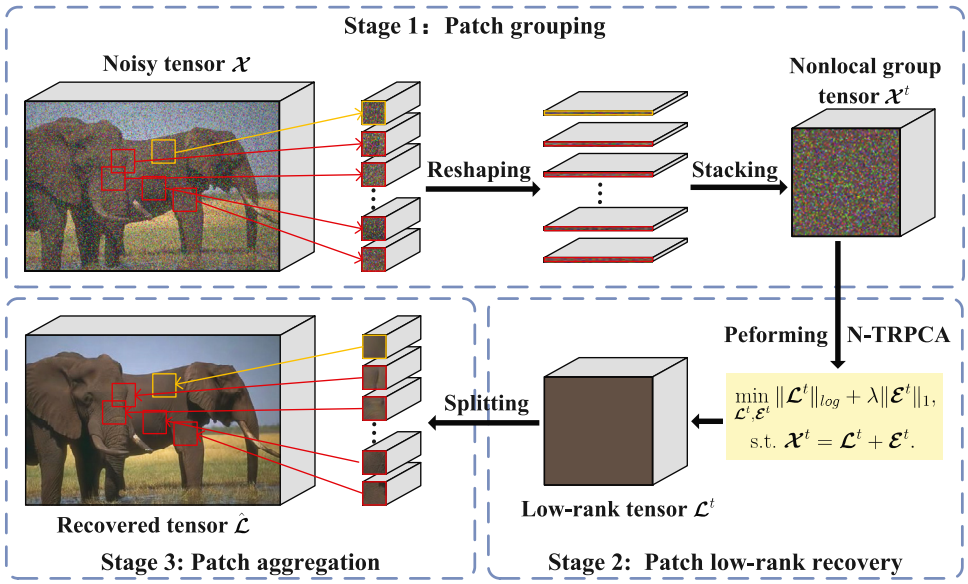

Fig. 1. Flowchart of the proposed NN-TRPCA by taking a group of similar patches as an example. The patch with yellow box is a reference patch, and the patches with red boxes are its similar patches. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

strongly low-rank. Similarly, the model presented in Guo et al. (2018) groups similar patches into a matrix, and then estimates each matrix by the low-rank approximation based on singular values truncation.

Recently, nonlocal low-rank matrix recovery has been extended to tensor domain. Unlike the matrix cases, nonlocal low-rank tensor recovery groups similar image patches as a tensor and treats the group tensors as the basic recovery units. For instance, a tensor-based compressed sensing recovery framework (NLR-TFA) (Zhang et al., 2018) was proposed by utilizing nonlocal self-similarity and low-CP-rank regularization. Since computing CP rank is NP-hard, NLR-TFA uses Jenrich's algorithm (Leurgans et al., 1993) to estimate CP rank. Additionally, nonlocal low-rank regularization-based tensor completion (NLRR-TC) (Xie et al., 2019) combines nonlocal prior and low-Tucker-rank constraint for hyperspectral image completion, which can simultaneously capture the spatial and spectral correlations of hyperspectral images. It is noteworthy that the tensor average rank can well describe the correlation across frontal slices of one tensor. Some works (Zhang et al., 2021; Qiu et al., 2021; Sun et al., 2020) integrated the nonlocal prior into low-rank tensor recovery based on the tensor average rank for visual data recovery.

Different from the above exiting works, our nonconvex model (N-TRPCA) utilizes a tensor adjustable logarithmic norm, which can flexibly control the shrinkage level to tensor singular values by an adjustable parameter θ. This is important because different tensor data have different tensor singular values. By adjusting the parameter θ, our N-TRPCA can efficiently recover different corrupted tensor data. Furthermore, to better recover the details in tensor data, we develop a nonlocal variant of N-TRPCA (i.e., NN-TRPCA). In the following section, we will elaborate the two proposed models.

4. Proposed methods

In this section, we first introduce a tensor adjustable logarithmic norm that is a nonconvex surrogate of the tensor average rank and use it to build a nonconvex TRPCA (N-TRPCA) model. Then an ADMM-based algorithm is developed to efficiently solve our N-TRPCA. Finally, the nonlocal self-similarity is integrated into N-TRPCA to derive the nonconvex and nonlocal TRPCA model, called NN-TRPCA. Fig. 1 illustrates the flowchart of the proposed NN-TRPCA method.

4.1. N-TRPCA model

By Eq. (5), we know that t-SVT shrinks each singular value by the same threshold τ in solving the TNN minimization. But in real scenarios, there exist great differences among tensor singular values. For instance, the large singular values of a noisy image usually deliver significant information, while the small singular values usually correspond the noise. Accordingly, the large singular values should be shrunk less, and the small singular values should be shrunk more. For this goal, we propose the following nonconvex surrogate of the tensor average rank.

Definition 6 (*Tensor Adjustable Logarithmic Norm (TALN)*)**.** Given any tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, $r = \min(n_1, n_2)$, the tensor adjustable logarithmic norm of $\mathcal{A}$ is defined as

$$\|\mathcal{A}\|_{log} = \frac{1}{n_3}\sum_{i=1}^{n_3}\sum_{j=1}^{r} g\left(\sigma_j(\bar{\mathbf{A}}^{(i)})\right), \tag{13}$$

where $g(x) = log(\theta x+1)$ is a nonconvex function with adjustable positive parameter θ.

One main advantage of TALN is that it better preserves the important information in tensor data than TNN. In Section 4.2, we will show that TALN can adaptively estimate weight for each singular value, and further show that the weight decreases as the singular value increases. According to these weights, different thresholds are used for shrinking the tensor singular values, and the smaller thresholds correspond to the larger singular values. As a result, TALN shrinks small singular values more and large singular values less. Another advantage of TALN is that it flexibly controls the shrinkage level of $g(x)$ to tensor singular values by the adjustable parameter θ. Fig. 2 provides a visual comparison of $g(x) = x$ used in TNN and $g(x) = log(\theta x + 1)$ used in our TALN with different θ. It can be observed that, unlike the behavior of $g(x) = x$, the nonconvex function $g(x) = log(\theta x + 1)$ can increase the shrinkage on small singular values and decrease the shrinkage on large singular values simultaneously. As θ increases, $g(x)$ increases the shrinkage on all singular values.

Based on TALN, we propose a N-TRPCA model, i.e.,

$$\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_{log} + \lambda\|\mathcal{E}\|_1, \ s.t.\ \mathcal{X} = \mathcal{L} + \mathcal{E}, \tag{14}$$

where λ is a regularization parameter.

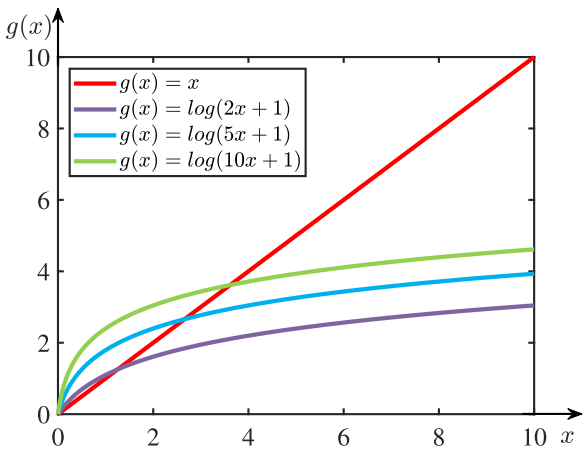

Fig. 2. Visual comparison of $g(x) = x$ and the noncovex function $g(x) = log(\theta x + 1)$ with different θ.

4.2. Optimization algorithm of N-TRPCA

As used in Zhang and Desrosiers (2018) and Yang et al. (2022), ADMM is an efficient approach to solve the optimization problem with multiple constraint terms. In the following, we present an ADMM-based algorithm for solving the model (14). The augmented Lagrangian function of (14) is

$$L(\mathcal{L},\mathcal{E},\mathcal{P},\mu) = \|\mathcal{L}\|_{log} + \lambda\|\mathcal{E}\|_1 + \langle \mathcal{P}, \mathcal{L}+\mathcal{E}-\mathcal{X}\rangle + \frac{\mu}{2}\|\mathcal{L}+\mathcal{E}-\mathcal{X}\|_F^2, \tag{15}$$

where $\mathcal{P}$ is a Lagrange multiplier and μ is a penalty parameter. $\mathcal{L}$ and $\mathcal{E}$ can be iteratively solved by minimizing function (15).

With parameters $\mathcal{E}_k$, $\mathcal{P}_k$, μ_k fixed, $\mathcal{L}_{k+1}$ can be obtained by

$$\mathcal{L}_{k+1} = \arg\min_{\mathcal{L}} \|\mathcal{L}\|_{log} + \frac{\mu_k}{2}\|\mathcal{L}-\mathcal{Q}_k\|_F^2, \tag{16}$$

where $\mathcal{Q}_k = \mathcal{X} - \mathcal{E}_k - \mu_k^{-1}\mathcal{P}_k$, subscript k indicates the kth iteration. Using Definition 6, Eq. (16) can be rewritten as

$$\mathcal{L}_{k+1} = \arg\min_{\mathcal{L}} \frac{1}{n_3}\sum_{i=1}^{n_3}\sum_{j=1}^{r} g\big(\sigma_j(\bar{\mathcal{L}}^{(i)})\big) + \frac{\mu_k}{2}\|\mathcal{L}-\mathcal{Q}_k\|_F^2, \tag{17}$$

where $r = \min(n_1, n_2)$. For simplicity of description, we denote $\sigma_j^i = \sigma_j(\bar{\mathcal{L}}^{(i)})$ and $\sigma_{j,k}^i = \sigma_{j,k}(\bar{\mathcal{L}}^{(i)})$. To estimate weights for the singular values σ_j^i, the function $g(\sigma_j^i)$ is approximated by its first-order Taylor expansion, i.e.,

$$g(\sigma_j^i) = g(\sigma_{j,k}^i) + w_{j,k}^i(\sigma_j^i - \sigma_{j,k}^i), \tag{18}$$

where $w_{j,k}^i = \partial g(\sigma_{j,k}^i) = \theta/(\theta\sigma_{j,k}^i + 1)$ is the derivative at point $\sigma_{j,k}^i$ that is equivalent to the weight assigned to σ_j^i. Since $\partial g(x)$ is monotonically decreasing and $\sigma_{1,k}^i \geq \sigma_{2,k}^i \geq \cdots \geq \sigma_{r,k}^i \geq 0$, the weights have the following property

$$0 \leq w_{1,k}^i \leq w_{2,k}^i \leq \cdots \leq w_{r,k}^i. \tag{19}$$

Using Eqs. (18), (17) is approximated as

$$\begin{aligned}\mathcal{L}_{k+1} &= \arg\min_{\mathcal{L}} \frac{1}{n_3}\sum_{i=1}^{n_3}\sum_{j=1}^{r}\big(g(\sigma_{j,k}^i) + w_{j,k}^i(\sigma_j^i - \sigma_{j,k}^i)\big) + \frac{\mu_k}{2}\|\mathcal{L}-\mathcal{Q}_k\|_F^2,\\ &= \arg\min_{\mathcal{L}} \frac{1}{n_3}\sum_{i=1}^{n_3}\sum_{j=1}^{r} w_{j,k}^i\sigma_j^i + \frac{\mu_k}{2}\|\mathcal{L}-\mathcal{Q}_k\|_F^2.\end{aligned} \tag{20}$$

Finally, the minimization problem (16) is converted to the problem (20). To solve Eq. (20), we use the following lemma and theorem.

Lemma 1 (Gu et al., 2017). *Let $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ be the SVD of $\boldsymbol{Y} \in \mathbb{R}^{n_1\times n_2}$, $r = \min(n_1, n_2)$, $\boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_r)$ satisfies $0 \leq \omega_1 \leq \omega_2 \leq \cdots \leq \omega_r$. For any $\tau > 0$, a global optimal solution of the following minimization objective*

$$\min_{\boldsymbol{X}} \tau\sum_{j=1}^{r}\omega_j\sigma_j(\boldsymbol{X}) + \frac{1}{2}\|\boldsymbol{X}-\boldsymbol{Y}\|_F^2 \tag{21}$$

is given by WSVT, i.e.,

$$\boldsymbol{X}^* = \boldsymbol{U}\boldsymbol{S}_{\tau\boldsymbol{\omega}}(\boldsymbol{\Sigma})\boldsymbol{V}^\top, \tag{22}$$

where $\boldsymbol{S}_{\tau\boldsymbol{\omega}}(\boldsymbol{\Sigma}) = \big(\boldsymbol{\Sigma} - \tau diag(\boldsymbol{\omega})\big)_+$, and $(x)_+$ represents the positive part of x, i.e., $(x)_+ = \max(x, 0)$.

Theorem 3. *Let $\mathcal{Y} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ be the t-SVD of $\mathcal{Y} \in \mathbb{R}^{n_1\times n_2\times n_3}$, $\mathcal{W} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is a f-diagonal tensor whose ith frontal slice is $diag(w_1^i, w_2^i, \ldots, w_r^i)$, where $r = \min(n_1, n_2)$ and $0 \leq \omega_1^i \leq \omega_2^i \leq \cdots \leq \omega_r^i$. For any $\tau > 0$, a global optimal solution of the following minimization objective*

$$\min_{\mathcal{X}} \tau\frac{1}{n_3}\sum_{i=1}^{n_3}\sum_{j=1}^{r}\omega_j^i\sigma_j(\bar{\boldsymbol{X}}^{(i)}) + \frac{1}{2}\|\mathcal{X}-\mathcal{Y}\|_F^2 \tag{23}$$

is given by the tensor WSVT (t-WSVT)

$$\mathcal{X}^* = \mathcal{U} * \mathcal{S}_{\tau\mathcal{W}} * \mathcal{V}^\top, \tag{24}$$

where $\mathcal{S}_{\tau\mathcal{W}} = ifft\big((\bar{\mathcal{S}} - \tau\mathcal{W})_+, [\,], 3\big)$.

Proof. In Fourier domain, the problem (23) is equivalent to

$$\min_{\mathcal{X}} \tau\frac{1}{n_3}\sum_{i=1}^{n_3}\sum_{j=1}^{r}\omega_j^i\sigma_j(\bar{\boldsymbol{X}}^{(i)}) + \frac{1}{2n_3}\|\bar{\boldsymbol{X}}-\bar{\boldsymbol{Y}}\|_F^2, \tag{25}$$

$$= \min_{\mathcal{X}} \frac{1}{n_3}\sum_{i=1}^{n_3}\Big(\tau\sum_{j=1}^{r}\omega_j^i\sigma_j(\bar{\boldsymbol{X}}^{(i)}) + \frac{1}{2}\Big\|\bar{\boldsymbol{X}}^{(i)} - \bar{\boldsymbol{Y}}^{(i)}\Big\|_F^2\Big). \tag{26}$$

In Eq. (26), the variables $\bar{\boldsymbol{X}}^{(i)}$ are independent. Then, the above problem can be divided into n_3 independent subproblems. By Lemma 1, we know that the global optimal solution of the ith ($i = 1, 2, \ldots, n_3$) subproblem is the ith frontal slice of $\bar{\mathcal{X}}^*$. Thus, $\mathcal{X}^*$ is the solution of the problem (26).

According to Eq. (19) and Theorem 3, the solution of the problem (20) is

$$\mathcal{L}_{k+1} = \mathcal{U} * \mathcal{S}_{\mu_k^{-1}\mathcal{W}} * \mathcal{V}^\top, \tag{27}$$

where $\mathcal{Q}_k = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ is the t-SVD of $\mathcal{Q}_k$, $\mathcal{W}$ is a f-diagonal tensor which the ith frontal slice is $diag(w_{1,k}^i, w_{2,k}^i, \ldots, w_{r,k}^i)$. According to the weight tensor $\mathcal{W}$, Eq. (27) utilizes small thresholds to shrink the large singular values and large thresholds to shrink the small singular values.

Similarly, holding $\mathcal{L}_{k+1}$, $\mathcal{P}_k$, μ_k fixed, $\mathcal{E}_{k+1}$ can be updated by

$$\mathcal{E}_{k+1} = \arg\min_{\mathcal{E}} \lambda\|\mathcal{E}\|_1 + \frac{\mu_k}{2}\|\mathcal{E}-\mathcal{H}_k\|_F^2, \tag{28}$$

where $\mathcal{H}_k = \mathcal{X} - \mathcal{L}_{k+1} - \mu_k^{-1}\mathcal{P}_k$. It has the following closed-form solution

$$\mathcal{E}_{k+1} = \mathrm{D}_{\lambda\cdot\mu_k^{-1}}(\mathcal{H}_k), \tag{29}$$

where $\mathrm{D}_\tau(x)$ is the soft thresholding operator (Tibshirani, 2011) defined as

$$\mathrm{D}_\tau(x) = \begin{cases} 0 & \text{if} \quad |x| \leq \tau, \\ \mathrm{sign}(x)(|x| - \tau) & \text{if} \quad |x| > \tau. \end{cases} \tag{30}$$

The whole optimization procedure for our N-TRPCA method is summarized in Algorithm 1. And the convergence proof of Algorithm 1 is provided in Appendix.

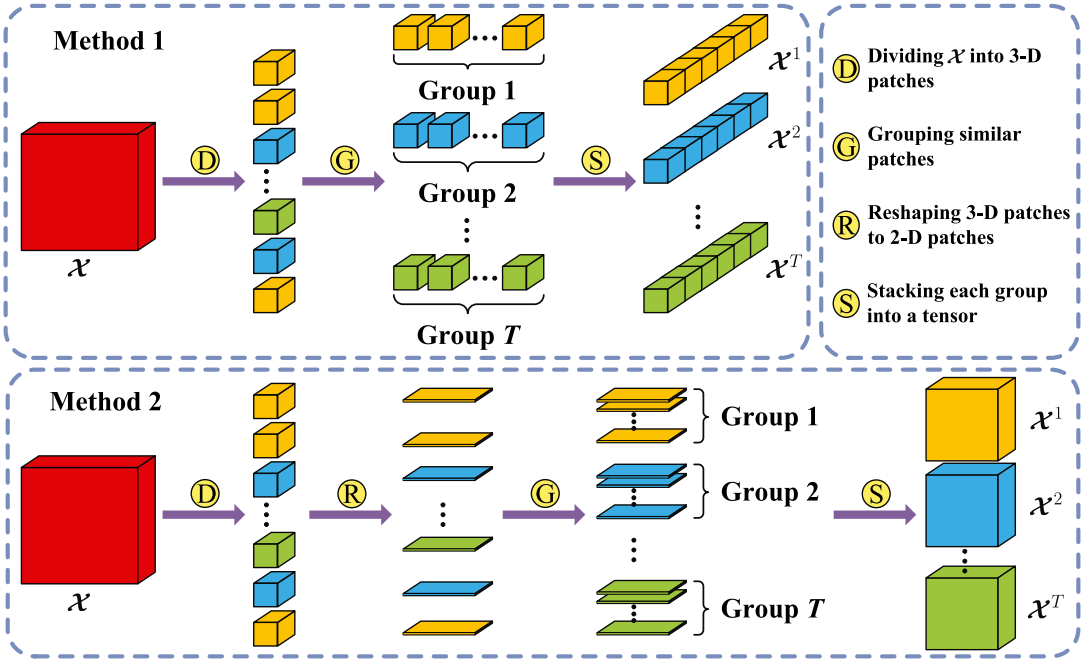

Fig. 3. An illustration of two methods to construct group tensors.

Table 1
Quantitative performance comparison of different methods on 20 images with noise rate 30%. The evaluation indices are PSNR, SSIM and FSIM.

Model	Parameter settings	PSNR	SSIM	FSIM
N-TRPCA	$\theta = 2$	27.78	0.826	0.905
NN-TRPCA (Method 1)	$\theta = 2, p = 10, m = 3$	29.35	0.876	0.917
NN-TRPCA (Method 2)	$\theta = 2, p = 10, m = 100$	**32.61**	**0.935**	**0.955**

Algorithm 1. N-TRPCA

Input: corrupted tensor $\mathcal{X}$.
Output: recovered tensor $\hat{\mathcal{L}}$.
Initialize: $\mathcal{L}_0 = \mathcal{E}_0 = \mathcal{P}_0 = 0$, λ, μ_0, $\mu_{\max}$, ρ, ϵ, θ.
while not converged **do**
1. Update $\mathcal{L}_{k+1}$ via (27);
2. Update $\mathcal{E}_{k+1}$ via (29);
3. Update $\mathcal{P}_{k+1}$ via $\mathcal{P}_{k+1} = \mathcal{P}_k + \mu_k(\mathcal{L}_{k+1} + \mathcal{E}_{k+1} - \mathcal{X})$;
4. Update μ_{k+1} via $\mu_{k+1} = \min(\rho\mu_k, \mu_{\max})$;
5. Check the convergence conditions
$\|\mathcal{L}_{k+1} - \mathcal{L}_k\|_\infty \le \epsilon$, $\|\mathcal{E}_{k+1} - \mathcal{E}_k\|_\infty \le \epsilon$,
$\|\mathcal{L}_{k+1} + \mathcal{E}_{k+1} - \mathcal{X}\|_\infty \le \epsilon$;
end while
6. Let $\hat{\mathcal{L}} = \mathcal{L}_{k+1}$ be the recovered tensor.

Algorithm 2. NN-TRPCA

Input: corrupted tensor $\mathcal{X}$.
Output: recovered tensor $\hat{\mathcal{L}}$.
1. $\{\Psi^t\}_{t=1}^T \leftarrow$ Divide the nonlocal similar patches of tensor $\mathcal{X}$ into T groups;
2. $\mathcal{X}^t \leftarrow$ Stack the similar patches in group Ψ^t as a tensor;
3. **for** $t = 1$ to T **do**
4. $\mathcal{L}^t \leftarrow$ Solve Eq. (31) on $\mathcal{X}^t$ via Algorithm 1;
5. **end for**
6. $\hat{\mathcal{L}} \leftarrow$ Output the recovered tensor by aggregating all $\mathcal{L}^t$ $(t = 1, 2, \dots, T)$ to the original position in $\mathcal{X}$.

4.3. NN-TRPCA model

As discussed in Section 3.2, nonlocal self-similarity implies that there is a lot of nonlocal structure redundancy in visual data. Thus, for each patch of visual data tensor, we search its nonlocal similar tensor patches as a group. Due to the high correlation between the target patch and its similar patches, the group tensor constructed by stacking these patches is strongly low-rank. Our N-TRPCA can be applied to each group tensor to estimate the values of corrupted pixels in the target patch by using the information in the similar patches, thereby improving the quality of the whole visual data. This inspires us to develop a nonlocal variant of N-TRPCA, i.e., NN-TRPCA. In the following, we describe the NN-TRPCA model. The procedure of NN-TRPCA is illustrated in Fig. 1, which mainly consists of three stages: tensor patch grouping, tensor patch low-rank recovery by N-TRPCA, and tensor patch aggregation.

Tensor patch grouping: Given a corrupted 3-D tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we divide $\mathcal{X}$ into overlapping tensor patches with the spatial size $p \times p$. Next we need to stack groups of nonlocal similar patches into tensors. Note that, the t-product induced tensor adjustable Logarithmic norm in our N-TRPCA is orientation dependent, which affects the restoration performance of group tensors constructed along different directions. Therefore, it is necessary to consider the construction of group tensors along different directions. Since the first and second dimensions are both spatial dimensions and are low-rank constrained by TALN in the same way, constructing group tensors along these two dimensions is the same. Hence, we only consider constructing group tensors along the first and third dimensions. See Fig. 3 for an intuitive illustration of these two construction methods. Method 1 constructs each group tensor by stacking similar patches along 3rd dimension. Specifically, for each 3-D patch, we search its $m-1$ similar patches based on the Euclidean metric. The group of each reference patch and its similar patches is indicated as $\Psi^t = \{\mathcal{Y}_i \in \mathbb{R}^{p \times p \times n_3}, i = 1, 2, \dots, m\}$, $(t = 1, 2, \dots, T)$. Here, T is the number of groups. At last, the group Ψ^t is stacked into a third-order tensor $\mathcal{X}^t \in \mathbb{R}^{p \times p \times mn_3}$. Different from the construction process above, Method 2 unfolds tensor patches to matrix patches and stacks similar patches along the first dimension. Specifically, we first reshape all 3-D patches to 2-D patches with size $p^2 \times n_3$. For each 2-D patch, we find its $m - 1$ similar patches. This

similar group is denoted as $\Psi^t = \{\boldsymbol{Y}_i \in \mathbb{R}^{p^2\times n_3}, i = 1,2,\dots,m\}$. The last step in Method 2 is to stack the group Ψ^t into a third-order tensor $\mathcal{X}^t \in \mathbb{R}^{m\times p^2\times n_3}$. Note that, $\mathcal{X}^t$ constructed by these two methods are both low-rank because the similar patches in each group have strong correlations.

But which construction method is more effective for tensor restoration? Intuitively, adopting Method 1 to construct the group tensors will achieve better restoration performance than using Method 2. The reason is that the unfolding operators in Method 2 will destroy the structural information of tensor patches, leading to a performance degradation of Method 2. To demonstrate this intuition, we conduct color image restoration experiments on 20 test images randomly selected from the Berkely Segmentation Dataset (Martin et al., 2001). 30% of the pixels in each image are set to random values in $[0,255]$, and the positions of corrupted pixels are unknown. The parameter settings and the average quantitative results are tabulated in Table 1. From this table, we can observe that whether Method 1 or Method 2 is adopted, NN-TRPCA outperforms N-TRPCA with respect to all the evaluation indices. This indicates that introducing the nonlocal prior is effective for color image restoration. But using Method 2 obtains better restoration results than using Method 1. This observation is contrary to the above intuition. We notice that whether adopting Method 1 or Method 2, NN-TRPCA obtains the best performance when the patch size is 10×10. Therefore, one reason is that the size of tensor patches is small, so the loss of structure information caused by unfolding operators in Method 2 can be negligible. More importantly, these two methods stack similar patches along different dimensions. Method 1 uses FFT to capture the similarity between patches while Method 2 uses SVD. Thus, the other reason is that SVD has greater ability to capture the correlation between similar patches than FFT. Therefore, in this work, for achieving better restoration performance, we choose Method 2 to form the group tensors.

Tensor patch low-rank recovery by N-TRPCA: For each group tensor, its underlying tensor is strongly low-rank due to the high correlation between similar patches, and the random noise in it is sparse. The group tensors conform to the assumption of N-TRPCA. Thus, they can be recovered by the N-TRPCA model, which is depicted to solve the following optimization problem

$$\begin{aligned} &\min_{\mathcal{L}^t,\mathcal{E}^t} \|\mathcal{L}^t\|_{log} + \lambda\|\mathcal{E}^t\|_1, \\ &s.t.\ \mathcal{X}^t = \mathcal{L}^t + \mathcal{E}^t,\ (t = 1,2,\dots,T). \end{aligned} \tag{31}$$

The low-rank tensors $\mathcal{L}^t$ are the recovered versions of the group tensors $\mathcal{X}^t$.

Tensor patch aggregation: At last, we aggregate all $\mathcal{L}^t$ back to their original positions in tensor $\mathcal{X}$ for obtaining the final recovered tensor $\hat{\mathcal{L}}$. Since one image patch may appear in several $\mathcal{L}^t$, we can obtain multiple estimates of this patch. Therefore, we average all estimates of this patch to obtain its recovered version. Then, the recovered patches are back to their positions in the original tensor for generating the final recovered tensor. For each pixel in overlapping regions of patches, we perform the similar operations to obtain the final estimate.

The overall procedure of the NN-TRPCA model is described in Algorithm 2. Although the convergence proof of Algorithm 2 is not given, each group tensor conforms to the assumption of N-TRPCA. Due to the convergence of N-TRPCA, each group tensor can be well restored, resulting in better restoration results by aggregating all restored group tensors.

4.4. Computational complexity

In this subsection, we further analyze the computational complexity of the proposed algorithms. Specifically, for Algorithm 1, the main cost at each iteration is the t-SVD calculation of the given input tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$. The per-iteration complexity is $\mathcal{O}\left(n_1n_2n_3\left(\log n_3 + \min\left\{n_1,n_2\right\}\right)\right)$. Let the number of iterations be k_1, the total computational complexity of the algorithm is $\mathcal{O}\left(k_1n_1n_2n_3\left(\log n_3 + \min\left\{n_1,n_2\right\}\right)\right)$. For Algorithm 2, its main computational cost is spent on applying Algorithm 1 to each nonlocal group tensor, and the running time of the patch grouping stage as well as the aggregation stage can be negligible. Suppose that the average number of iterations over all group tensors is k_2, the computational complexity on each group tensor with size of $m\times p^2\times n_3$ is $\mathcal{O}\left(k_2\ mp^2n_3\left(\log n_3 + \min\left\{m,p^2\right\}\right)\right)$. Thus, the overall computational complexity of Algorithm 2 with T group tensors is $\mathcal{O}\left(Tk_2\ mp^2n_3\left(\log n_3 + \min\left\{m,p^2\right\}\right)\right)$.

5. Experimental results

We evaluate the performance of NN-TRPCA in the task of color image and gray video recovery. The color images and gray videos can be considered as third-order tensors, and the recovery task is to estimate the clean visual tensors from their corrupted versions.

5.1. Experimental setup

For color image restoration, we randomly select 100 color images with size 321×481 from the popular Berkely Segmentation Dataset (Martin et al., 2001) as test images. These images include different natural scenes and objects, e.g., animals, plants, people, scenery, and buildings. We add random noise to the test images with different noise ratios (10%, 20%, and 30%). Note that, noise ratio refers to the proportion of pixels destroyed by noise among all the pixels of one image. Specifically, for each color image, the pixels of each rate are randomly set to random values in $[0,255]$, and the positions of corrupted pixels are unknown. All the three channels of a color image are corrupted in the same positions. This setting is more challenging than the noise of three channels in different positions. Besides, three video sequences from Scene Background Initialization Dataset (Maddalena and Petrosino, 2015) are employed for video restoration, including 'Hall & Monitor', 'Candela_m1.10', and 'CAVIAR1'. The frame sizes of these three videos are 352×240, 352×288, and 384×256, respectively. Owing to the computational limitation, we only use the first 30 frames of each video and resize each frame to a quarter of its original size. Similar to color images, we add random noise to each gray video with noise ratio 30%.

Our N-TRPCA and NN-TRPCA are compared with four state-of-the-art TRPCA methods, including SNN-TRPCA (Huang et al., 2015), KBR-TRPCA (Xie et al., 2018), TNN-TRPCA (Lu et al., 2020), and TPCPSF (Han et al., 2023). These comparison methods adopt different tensor ranks as low-rank constraints. SNN-TRPCA is based on the Tucker rank. KBR-TRPCA is a Kronecker-basis-representation based method that combines the Tucker rank and CP rank. TNN-TRPCA and TPCPSF are based on the tensor average rank. The proposed two methods and other competing methods are implemented in MATLAB R2021a and are run on a PC with an Intel Core i5-8500 3.00 GHz CPU and 16 GB RAM. The performance of different methods are evaluated by three quantitative picture quality indices (PQI), including peak signal-to-noise ratio (PSNR) (Lu et al., 2020), SSIM (Wang et al., 2004), and structure similarity (FSIM) (Zhang et al., 2011). PSNR and SSIM are two commonly-used PQIs in the task of image restoration. The former measures the similarity between the ground-truth and the restored image based on MSE, and the latter is to measure the structural consistency. Unlike SSIM, FSIM is more consistent with human eye perception by utilizing both phase congruency and image gradient magnitude. The higher the values of these three indices, the better the restoration results.

The parameters of all experiments are set as follows. $[\lambda_1,\lambda_2,\lambda_3]$ in SNN-TRPCA is empirically set to $[15,15,1.5]$ for color image restoration. This setting can enable SNN-TRPCA to perform well in most image cases. For video restoration, $[\lambda_1,\lambda_2,\lambda_3]$ are set differently since the three videos have different correlations along each mode. Specifically, $[\lambda_1,\lambda_2,\lambda_3]$ are set to $[12,12,17]$, $[13,13,20]$, and $[12,12,16]$ for 'Hall

Table 2
Quantitative performance comparison of different methods on 100 test images with different noise rates.

Method	10%			20%			30%		
	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM	PSNR	SSIM	FSIM
SNN-TRPCA	26.09	0.812	0.869	25.14	0.778	0.847	23.11	0.735	0.822
KBR-TRPCA	29.50	0.919	0.958	27.91	0.875	0.933	25.97	0.793	0.892
TNN-TRPCA	29.35	0.935	0.945	27.68	0.886	0.925	25.73	0.794	0.878
TPSCPSF	30.56	0.898	0.940	30.17	0.892	0.937	30.09	0.884	0.933
N-TRPCA	34.31	0.957	0.977	30.26	0.896	0.944	27.20	0.809	0.898
NN-TRPCA	**38.91**	**0.980**	**0.988**	**35.08**	**0.960**	**0.975**	**31.79**	**0.926**	**0.952**

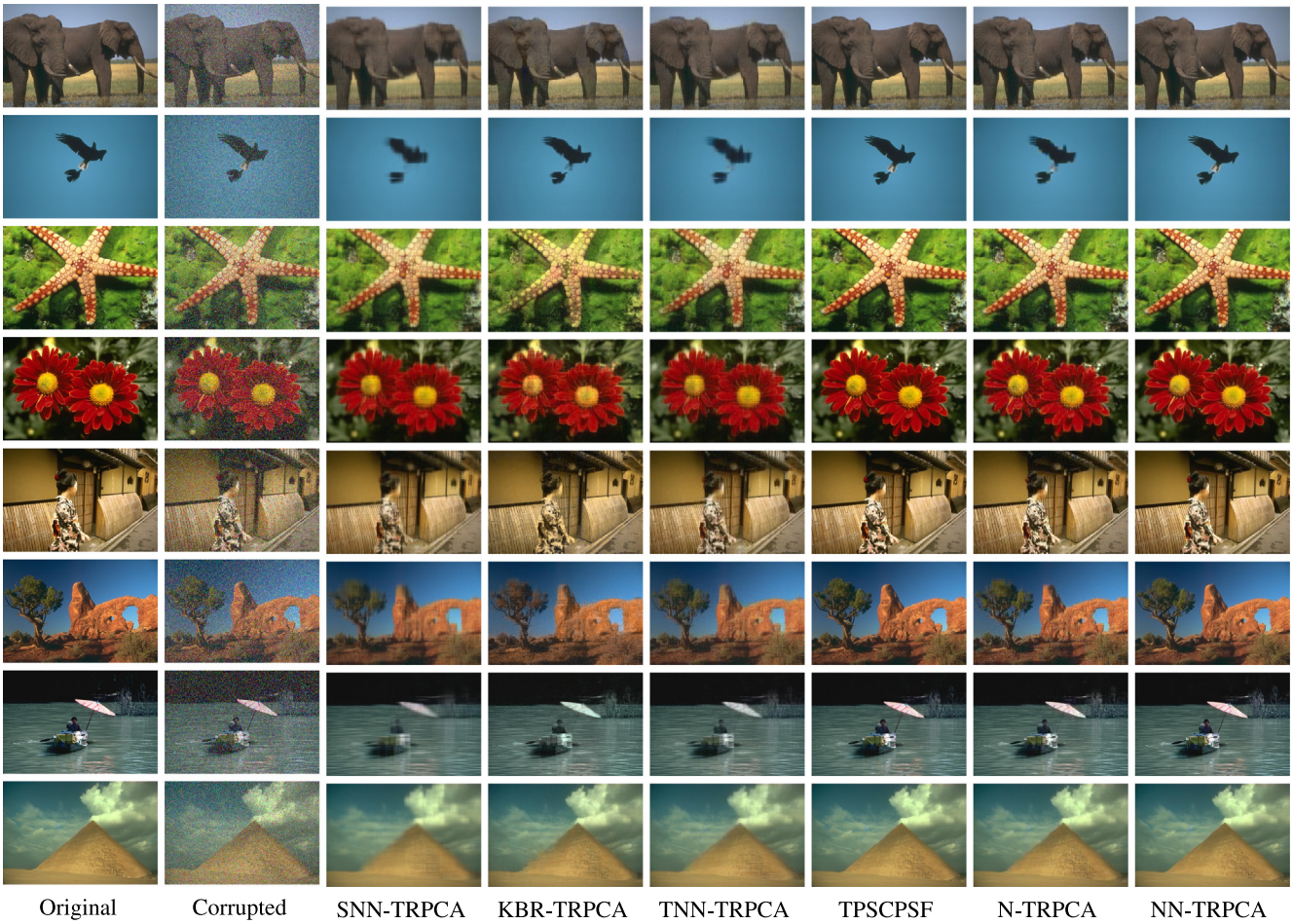

Fig. 4. Restoration results on eight example images. From top to bottom: 'Elephant', 'Bird', 'Starfish', 'Flower', 'Girl', 'Stone', 'Boat', and 'Pyramid'. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

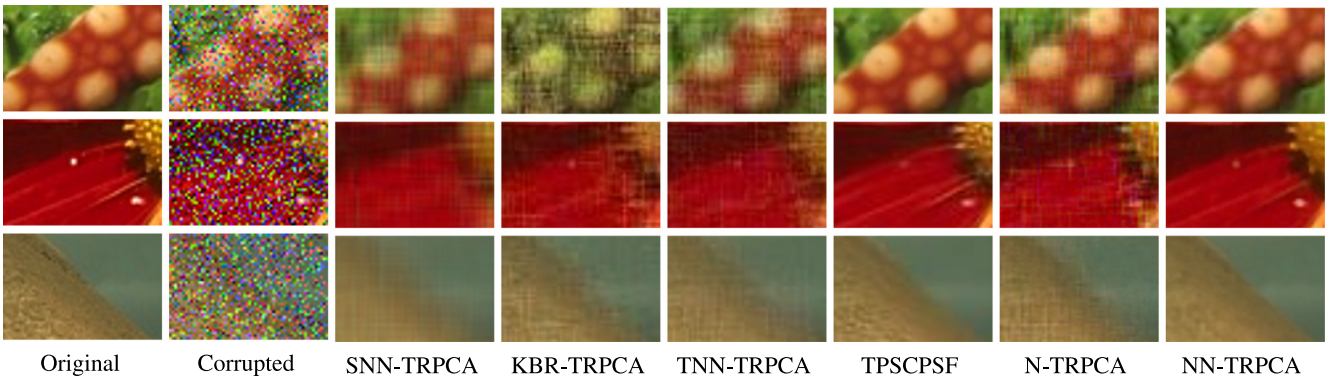

Fig. 5. Some zoomed-in restoration results of different methods on the color images 'Starfish', 'Flower', and 'Pyramid'. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Quantitative performance comparison of different methods on eight example images with noise rate 30%.

Image	Method	PSNR	SSIM	FSIM
Elephant	SNN-TRPCA	26.59	0.712	0.823
	KBR-TRPCA	29.01	0.793	0.904
	TNN-TRPCA	28.15	0.802	0.892
	TPSCPSF	31.83	0.860	0.928
	N-TRPCA	29.22	0.819	0.914
	NN-TRPCA	**33.40**	**0.920**	**0.952**
Bird	SNN-TRPCA	29.10	0.949	0.892
	KBR-TRPCA	32.15	0.962	0.919
	TNN-TRPCA	29.86	0.954	0.914
	TPSCPSF	40.83	0.989	0.982
	N-TRPCA	32.24	0.962	0.928
	NN-TRPCA	**41.62**	**0.988**	**0.931**
Starfish	SNN-TRPCA	22.17	0.669	0.845
	KBR-TRPCA	21.99	0.674	0.865
	TNN-TRPCA	23.59	0.686	0.868
	TPSCPSF	30.70	0.903	0.958
	N-TRPCA	25.68	0.735	0.889
	NN-TRPCA	**30.59**	**0.906**	**0.955**
Flower	SNN-TRPCA	24.34	0.721	0.864
	KBR-TRPCA	23.73	0.657	0.872
	TNN-TRPCA	24.44	0.689	0.887
	TPSCPSF	32.08	0.930	0.972
	N-TRPCA	26.31	0.734	0.900
	NN-TRPCA	**31.63**	**0.914**	**0.965**
Girl	SNN-TRPCA	21.71	0.756	0.810
	KBR-TRPCA	23.54	0.768	0.865
	TNN-TRPCA	23.61	0.785	0.861
	TPSCPSF	26.63	0.891	0.935
	N-TRPCA	25.83	0.846	0.896
	NN-TRPCA	**30.71**	**0.958**	**0.966**
Stone	SNN-TRPCA	26.41	0.733	0.793
	KBR-TRPCA	26.78	0.797	0.881
	TNN-TRPCA	27.27	0.802	0.879
	TPSCPSF	28.42	0.828	0.885
	N-TRPCA	29.18	0.824	0.892
	NN-TRPCA	**34.90**	**0.950**	**0.961**
Boat	SNN-TRPCA	23.67	0.831	0.821
	KBR-TRPCA	26.45	0.885	0.897
	TNN-TRPCA	25.65	0.881	0.878
	TPSCPSF	29.10	0.913	0.926
	N-TRPCA	28.01	0.899	0.910
	NN-TRPCA	**32.21**	**0.945**	**0.960**
Pyramid	SNN-TRPCA	26.82	0.754	0.738
	KBR-TRPCA	28.73	0.851	0.883
	TNN-TRPCA	28.67	0.845	0.843
	TPSCPSF	30.83	0.861	0.871
	N-TRPCA	29.73	0.839	0.862
	NN-TRPCA	**33.35**	**0.924**	**0.928**

& Monitor', 'Candela_m1.10', and 'CAVIAR1', respectively. For TNN-TRPCA and KBR-TRPCA, we follow the default parameters setting suggested by their authors. The parameter κ in TPSCPSF is set to 0.9 for color image restoration and 0.6 for video restoration. For N-TRPCA, we empirically set $\mu_0 = 1e-3$, $\mu_{max} = 1e10$, $\rho = 1.1$, $\epsilon = 1e-5$, and $\theta = 2$, respectively. As used in Lu et al. (2020), our parameter λ is set to $1/\sqrt{\max(n_1, n_2)n_3}$. For NN-TRPCA, the patch size p and the number of patches m are empirically set to 10 and 100.

5.2. Results on color image restoration

Table 2 displays the average quantitative results of competitive methods on test images with different noise rates $(10\%, 20\%, 30\%)$. It can be observed that KBR-TRPCA outperforms SNN-TRPCA evidently. The main reason is that, KBR-TRPCA combines the advantage of CP rank and Tucker rank. In addition, TNN-TRPCA is competitive with KBR-TRPCA and better than SNN-TRPCA. This is practically reasonable because TNN-TRPCA successfully captures the multidimensional structural information in tensors equipped with the tensor average rank. TPSCPSF and N-TRPCA are superior to SNN-TRPCA, TNN-TRPCA and KBR-TRPCA in all evaluation indices. This can be attributed to the factors: (1) TPSCPSF makes full use of side information and features; (2) N-TRPCA well preserves the significant information in tensors by shrinking the tensor singular values differently. More importantly, NN-TRPCA has yielded the best performance. This is due to the fact that NN-TRPCA takes full use of the structural redundancy in color images by introducing the nonlocal prior.

Table 4
Quantitative performance comparison of different algorithms on three videos.

Video	Method	PSNR	SSIM	FSIM
Hall & Monitor	SNN-TRPCA	26.08	0.911	0.941
	KBR-TRPCA	26.13	0.923	0.955
	TNN-TRPCA	31.43	0.973	0.988
	TPSCPSF	30.81	0.951	0.974
	N-TRPCA	34.28	0.981	0.992
	NN-TRPCA	**37.76**	**0.991**	**0.998**
Candela_m1.10	SNN-TRPCA	25.65	0.848	0.938
	KBR-TRPCA	26.78	0.910	0.967
	TNN-TRPCA	30.96	0.969	0.993
	TPSCPSF	35.03	0.976	0.988
	N-TRPCA	34.28	0.983	0.996
	NN-TRPCA	**38.74**	**0.993**	**0.997**
CAVIAR1	SNN-TRPCA	29.95	0.941	0.940
	KBR-TRPCA	31.02	0.951	0.946
	TNN-TRPCA	34.99	0.983	0.970
	TPSCPSF	35.49	0.975	0.976
	N-TRPCA	40.75	0.995	0.988
	NN-TRPCA	**42.25**	**0.996**	**0.992**

For more intuitive comparison, the restoration results of different methods on eight example images are shown in Fig. 4. One can observe that SNN-TRPCA produces serious block artifacts in all scenarios. This is because directly unfolding tensors along each mode will lose structural information in tensor data. The images restored by N-TRPCA also contain some artifacts like by TNN-TRPCA. Fortunately, by introducing the nonlocal self-similarity, NN-TRPCA removes the artifacts and yields the best visual effect. The visual comparisons are further illustrated in Fig. 5, which shows the zoomed-in restoration results of different methods on image 'Starfish', 'Flower' and 'Pyramid'. It is worth noting that NN-TRPCA can well restore the detail information of complex images, e.g., texture of starfish, water-drops on flowers, and the edge of pyramid. Futhermore, the quantitative results on these example images are recorded in Table 3, from which we make the following observations. Although N-TRPCA produces artifacts like TNN-TRPCA, N-TRPCA is far superior to TNN-TRPCA in terms of all the evaluation indices, which confirms the effectiveness of the tensor adjustable logarithmic norm to retain the important information in color images. Besides, our NN-TRPCA obtains the best evaluation indices over all competing methods by utilizing the nonlocal redundancy of natural color images. Especially for images with complex structures, such as 'Starfish', 'Flower', and 'Girl', NN-TRPCA still has a significant improvement compared with N-TRPCA. This is due to the fact that the nonlocal self-similarity is abundant in these complex images. By grouping nonlocal similar patches, the group tensor is strongly low-rank, which offers contribution for better restoration. In a word, the proposed NN-TRPCA method delivers the best recovery performance on both visual results and evaluation indices.

5.3. Results on gray video restoration

The three test videos capture a walking human but in three different scenes. Fig. 6 shows several frames of restoration results of different algorithms on the test videos with noise rate 30%, and Fig. 7 further gives some zoomed-in restoration results. From these results, we can

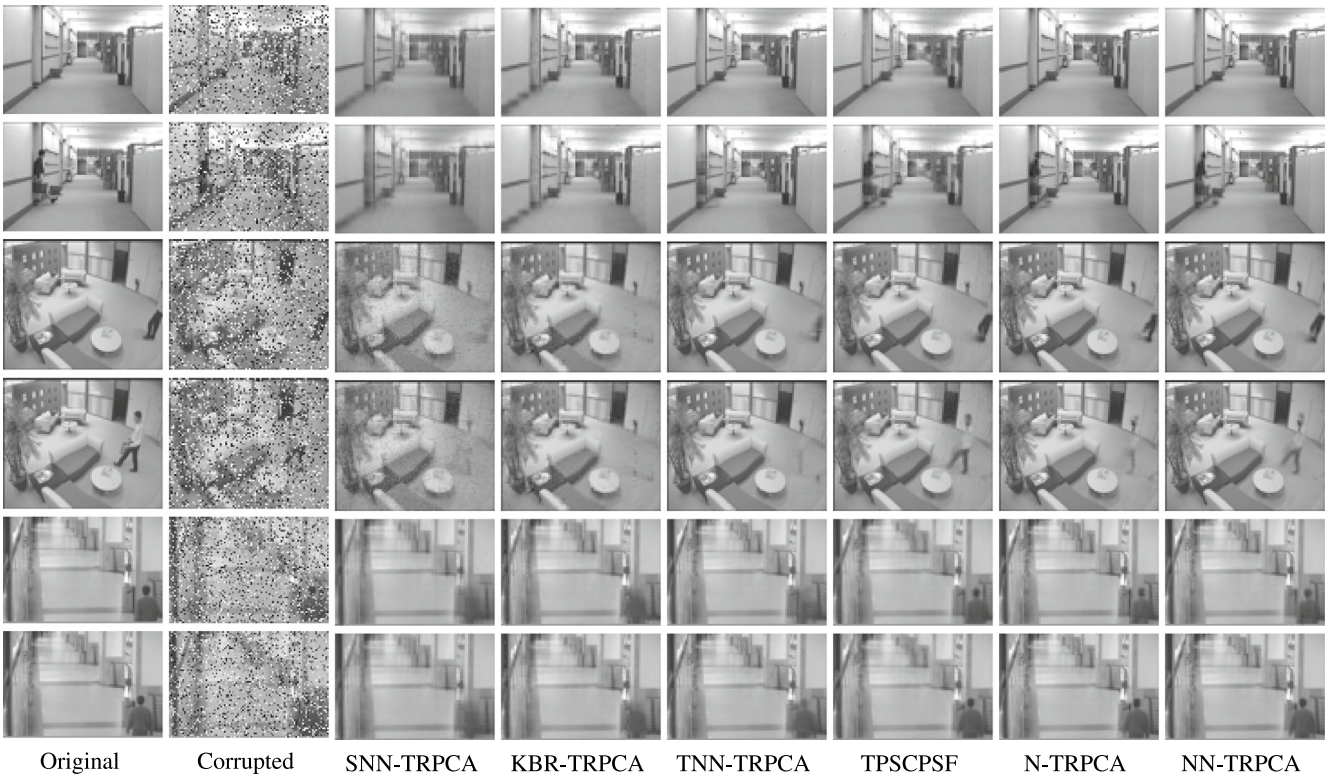

Fig. 6. Several frames of restoration results on three videos. From top to bottom: 'Hall & Monitor', 'Candela_m1.10', and 'CAVIAR1'.

observe that our N-TRPCA more clearly restores the important structure information of videos compared with TNN-TRPCA. The reason lies in that the tensor adjustable logarithmic norm used in N-TRPCA is capable to preserve the important information in videos by shrinking the large singular values less and the small ones more. It can be further found that, although both N-TRPCA and NN-TRPCA can recover the main structure of walking humans, NN-TRPCA restores the contour of the walking man more accurately. This indicates that the introduction of nonlocal self-similarity enables NN-TRPCA to recover more detail information in videos.

Meanwhile, as shown in Table 4, the proposed N-TRPCA and NN-TRPCA have yielded very competitive scores of evaluation indices. More specifically, our N-TRPCA can significantly outperform other competitive methods, e.g., N-TRPCA achieves 3.98 dB gain on PSNR beyond TNN-TRPCA on average of the three videos. This is due to the fact that N-TRPCA treats the tensor singular values differently. Furthermore, by integrating the nonlocal self-similarity, NN-TRPCA obtains performance improvements over N-TRPCA, i.e., averagely by 3.15 dB gain on PSNR, 0.007 on SSIM and 0.004 on FSIM. All these scores validate the superiority and effectiveness of our NN-TRPCA method.

5.4. Impacts of parameters

In NN-TRPCA, there are three algorithmic parameters, i.e., adjustable parameter θ, patch size p, and the number of patches in each group tensor m. The parameter θ controls the shrinkage level to tensor singular values. The larger θ indicates more shrinkage. To analyze the impact of θ, we fix other parameters and perform N-TRPCA and NN-TRPCA on the test images with different values $\theta = 1, 1.5, 2, \ldots, 5$. The average change curves of three PQIs with varying θ are plotted in Fig. 8. From them, we can see that the best restoration results are obtained at $\theta = 2$. Thus, the parameter θ is empirically set to 2.

The patch size p and the number of patches in each group tensor m are two important parameters to capture the nonlocal redundancy in images. When p is too small, it is hard to preserve the local structure of patches. When p is too large, there are too many details in each patch, resulting in the reduction of the similarity of patches in one group. Besides, a too small m could divide the closely similar patches into different groups and a too large m could classify other dissimilar patches into one similar group. Therefore, a too small m or a too large m will affect the accuracy of patch grouping, leading to a degraded restoration performance. To study the influence of parameters p and m, we run our NN-TRPCA algorithm on the test image 'Elephant' with different values of p and m. Fig. 9 displays the average change curves of three PQIs with varying value pairs (p, m). It can be found that NN-TRPCA with parameter pair $(p, m) = (10, 100)$ can yield the best restoration performance. Hence, in our experiments, p and m are fixed to 10 and 100, respectively.

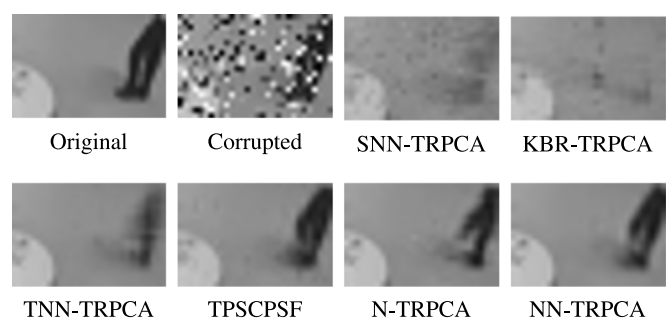

Fig. 7. Some zoomed-in restoration results on the video 'Hall & Monitor'.

5.5. Impacts of initial conditions

A challenge for nonconvex optimization problems is that they do not guarantee a unique solution. That is, different initial conditions may lead to varying results. To demonstrate that our N-TRPCA and

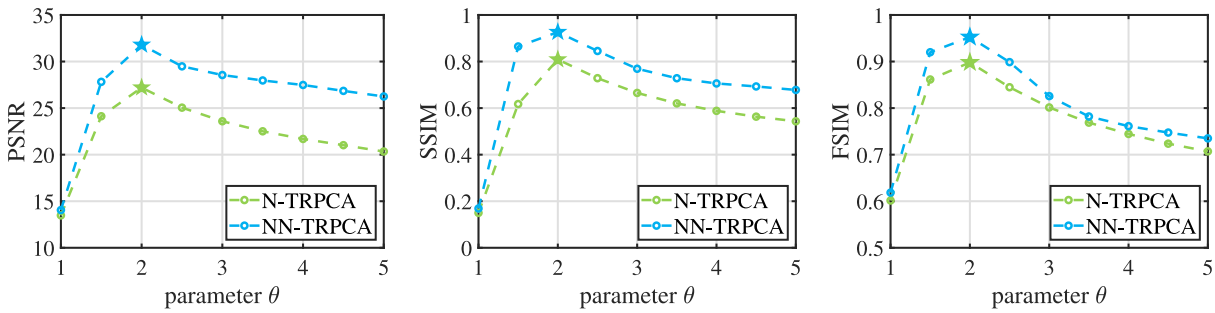

Fig. 8. Quantitative comparison of the effect of parameter θ on the proposed methods.

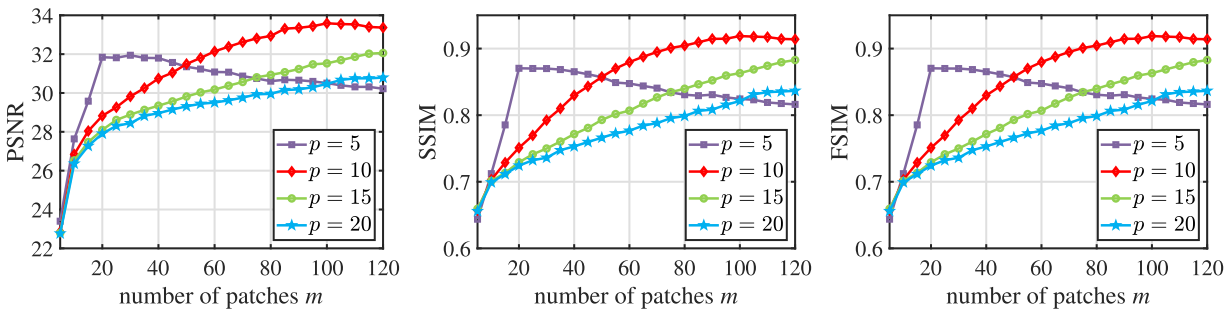

Fig. 9. Quantitative comparison of the effect of parameters p and m on NN-TRPCA.

Table 5
Average running times (in seconds) of different methods.

Task	SNN-TRPCA	KBR-TRPCA	TNN-TRPCA	TPSCPSF	N-TRPCA	NN-TRPCA
Image restoration	9.98	97.42	10.62	565.72	13.63	2679.98
Video restoration	2.23	2.87	2.40	489.94	4.04	1431.27

NN-TRPCA are insensitive to initial conditions, we conduct a set of experiments on color image restoration. Specifically, we fix other parameters and perform our two models with different initial tensors. The quantitative results are reported in Table 6. From them, we can observe that different initial tensors have almost no effect on the results. This demonstrates that both N-TRPCA and NN-TRPCA are not sensitive to initial conditions. Therefore, following Lu et al. (2020), the tensors $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{E}}$ in N-TRPCA and $\boldsymbol{\mathcal{L}}_t$ and $\boldsymbol{\mathcal{E}}_t$ $(t = 1, 2, \ldots, T)$ in NN-TRPCA are initialized as zero tensors.

Table 6
Quantitative performance comparison of our models on color image restoration with different initial tensors.

Model	Initial tensors	PSNR	SSIM	FSIM
N-TRPCA	$\boldsymbol{\mathcal{L}} = 0,\ \boldsymbol{\mathcal{E}} = 0$	27.15	0.816	0.901
	$\boldsymbol{\mathcal{L}} = 128,\ \boldsymbol{\mathcal{E}} = 128$	27.17	0.817	0.902
	$\boldsymbol{\mathcal{L}} = 255,\ \boldsymbol{\mathcal{E}} = 255$	27.17	0.817	0.901
NN-TRPCA	$\boldsymbol{\mathcal{L}}_t = 0,\ \boldsymbol{\mathcal{E}}_t = 0$	31.47	0.933	0.956
	$\boldsymbol{\mathcal{L}}_t = 128,\ \boldsymbol{\mathcal{E}}_t = 128$	31.43	0.933	0.955
	$\boldsymbol{\mathcal{L}}_t = 128,\ \boldsymbol{\mathcal{E}}_t = 128$	31.47	0.934	0.956

5.6. Computational cost

We also compare the computational cost of the proposed algorithms to those of three state-of-the-art TRPCA algorithms. Table 5 summarizes the average running times of different algorithms on the aforementioned tasks. From it, one can observe that the running time of our N-TRPCA is comparable to that of TNN-TRPCA. This is because the per-iteration complexity of N-TRPCA algorithm is $\mathcal{O}\left(n_1 n_2 n_3 \left(\log n_3 + \min\left\{n_1, n_2\right\}\right)\right)$, which is similar to that of TNN-TRPCA. Furthermore, we can observe that the computational cost of our NN-TRPCA is high expensive. The reason is that the number of group tensors T is large and the calculation of t-SVD for each group tensor is time-consuming. In fact, since the t-SVDs of T group tensors are calculated independently, a direct acceleration way is the parallel GPU implementation of the t-SVDs. Besides, as used in Simon and Zha (2000) and Guo et al. (2021), the Lanczos bidiagonalization process can be employed to accelerate the calculation of SVD by only considering the few dominant singular values.

6. Conclusion and future work

TRPCA, aiming to restore a low-rank tensor from it corrupted version, has attracted considerable interest in the fields of image and video processing. In this work, we firstly build the N-TRPCA based on the tensor logarithmic norm which can better retain the significant information of visual data. Furthermore, to fully utilize the structural redundancy in visual data, we propose the NN-TRPCA by introducing the nonlocal self-similarity. This model can well recover the edges and textures in images and videos. Meanwhile, an efficient algorithm based on ADMM is designed to solve the proposed models. Extensive experimental results confirm the effectiveness of our models compared with recent state-of-the-art TRPCA models.

For future work, there are two possible expansion directions. Firstly, this paper focuses on three-dimensional visual data due to the fact that the t-SVD and tensor average rank are defined on three-way tensors. Therefore, one can generalize the t-SVD and tensor average rank for higher-dimensional tensors, and further propose a higher-order version of the NN-TRPCA. Secondly, in order to obtain lower-rank tensors, some studies have attempted to replace FFT used in t-SVD with other invertible linear transform, such as framelet transform (Jiang et al., 2020), discrete cosine transform (Xu et al., 2019b). Inspired by these works, we will attempt to find a more suitable transform to capture the low-rankness of visual data, so as to achieve better performance of visual data restoration.

CRediT authorship contribution statement

Xiaoyu Geng: Methodology, Software, Writing – original draft. **Qiang Guo:** Conceptualization, Funding acquisition, Writing – review & editing. **Shuaixiong Hui:** Software, Validation. **Ming Yang:** Formal analysis. **Caiming Zhang:** Supervision.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 61873145.

Appendix

In this appendix, we provide the convergent analysis of Algorithm 1.

Lemma 2 (*Lewis and Sendov, 2005*). *Suppose $F:\mathbb{R}^{n_1\times n_2}\to\mathbb{R}$ is denoted as $F(\boldsymbol{X})=f\circ\sigma(\boldsymbol{X})$, f is differentiable, and let $\boldsymbol{X}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ be the SVD of $\boldsymbol{X}\in\mathbb{R}^{n_1\times n_2}$, $\boldsymbol{\Sigma}=diag(\sigma_1,\dots,\sigma_n)$, $n=\min(n_1,n_2)$. The gradient of $F(\boldsymbol{X})$ at $\boldsymbol{X}$ is*

$$\frac{\partial F(\boldsymbol{X})}{\partial \boldsymbol{X}}=\boldsymbol{U}diag(\theta)\boldsymbol{V}^T, \tag{32}$$

where $\theta=\frac{\partial f(y)}{\partial y}|_{y=\sigma(\boldsymbol{X})}$.

Theorem 4. *The tensors $\mathcal{L}_k$, $\mathcal{E}_k$, and $\mathcal{P}_k$ outputed by the proposed Algorithm 1 are bounded. And $\{\mathcal{L}^*,\mathcal{E}^*,\mathcal{P}^*\}$ is a KKT stationary point of the objective (15), satisfying the KKT conditions $\mathcal{P}^*\in\partial\|\mathcal{L}\|_{log}$, $\mathcal{L}^*+\mathcal{E}^*=\mathcal{X}$, $\mathcal{P}^*\in\partial\|\mathcal{E}\|_1$.*

Proof. $\mathcal{E}_{k+1}$ satisfies the first-order necessary local optimality condition of (28),

$$\begin{aligned}&0\in\partial_{\mathcal{E}}L\left(\mathcal{L}_{k+1},\mathcal{E}_{k+1},\mathcal{P}_k,\mu_k\right)\\&=\partial\left(\lambda\|\mathcal{E}_{k+1}\|_1\right)+\mathcal{P}_k+\mu_k\left(\mathcal{L}_{k+1}-\mathcal{X}+\mathcal{E}_{k+1}\right)\\&=\partial\left(\lambda\|\mathcal{E}_{k+1}\|_1\right)+\mathcal{P}_{k+1}.\end{aligned} \tag{33}$$

Due to $\|\mathcal{E}\|_1$ being non-smooth at $\mathcal{E}_{ijk}=0$, we redefine sub-gradient $\left[\partial\|\mathcal{E}\|_1\right]_{ijk}=0$ if $\mathcal{E}_{ijk}=0$. Then $0\le\|\partial\|\mathcal{E}\|_1\|_F^2\le n_1n_2n_3$, hence $\partial(\lambda\|\mathcal{E}_{k+1}\|_1)$ is bounded. Thus it is ready to see that $\mathcal{P}_k$ is bounded.

Considering the objective (15), we have

$$\begin{aligned}&L\left(\mathcal{L}_k,\mathcal{E}_k,\mathcal{P}_k,\mu_k\right)\\&=L\left(\mathcal{L}_k,\mathcal{E}_k,\mathcal{P}_{k-1},\mu_{k-1}\right)+\frac{\mu_k-\mu_{k-1}}{2}\|\mathcal{L}_k-\mathcal{X}+\mathcal{E}_k\|_F^2\\&+tr[(\mathcal{P}_k-\mathcal{P}_{k-1})(\mathcal{L}_k-\mathcal{X}+\mathcal{E}_k)],\\&=L\left(\mathcal{L}_k,\mathcal{E}_k,\mathcal{P}_{k-1},\mu_{k-1}\right)+\frac{\mu_k+\mu_{k-1}}{2(\mu_{k-1})^2}\|\mathcal{P}_k-\mathcal{P}_{k-1}\|_F^2.\end{aligned} \tag{34}$$

So,

$$\begin{aligned}&L\left(\mathcal{L}_{k+1},\mathcal{E}_{k+1},\mathcal{P}_k,\mu_k\right)\\&\le L(\mathcal{L}_{k+1},\mathcal{E}_k,\mathcal{P}_k,\mu_k)\le L(\mathcal{L}_k,\mathcal{E}_k,\mathcal{P}_k,\mu_k)\\&\le L\left(\mathcal{L}_k,\mathcal{E}_k,\mathcal{P}_{k-1},\mu_{k-1}\right)+\frac{\mu_k+\mu_{k-1}}{2(\mu_{k-1})^2}\|\mathcal{P}_k-\mathcal{P}_{k-1}\|_F^2.\end{aligned} \tag{35}$$

An easy induction gives

$$\begin{aligned}&L(\mathcal{L}_{k+1},\mathcal{E}_{k+1},\mathcal{P}_k,\mu_k)\\&\le L\left(\mathcal{L}_1,\mathcal{E}_1,\mathcal{P}_0,\mu_0\right)+\sum_{i=1}^{k}\frac{\mu_i+\mu_{i-1}}{2(\mu_{i-1})^2}\|\mathcal{P}_i-\mathcal{P}_{i-1}\|_F^2.\end{aligned} \tag{36}$$

Since $\frac{\mu_i+\mu_{i-1}}{2(\mu_{i-1})^2}\|\mathcal{P}_i-\mathcal{P}_{i-1}\|_F^2$ is bounded, it is not hard to know that $L\left(\mathcal{L}_{k+1},\mathcal{E}_{k+1},\mathcal{P}_k,\mu_k\right)$ is upper bounded. It is straightforward to show that

$$\begin{aligned}&L\left(\mathcal{L}_{k+1},\mathcal{E}_{k+1},\mathcal{P}_k,\mu_k\right)+\frac{1}{2\mu_k}\|\mathcal{P}_k\|_F^2\\&=\|\mathcal{L}_{k+1}\|_{log}+\lambda\|\mathcal{E}_{k+1}\|_1+\frac{\mu_k}{2}\|\mathcal{L}_{k+1}-\mathcal{X}+\mathcal{E}_{k+1}+\frac{\mathcal{P}_k}{\mu_k}\|_F^2.\end{aligned} \tag{37}$$

In the above equation, each term on the right-hand side is bounded, Therefore, $\mathcal{E}_{k+1}$ is bounded. And $\mathcal{L}_{k+1}$ is also bounded by the last term on the right-hand.

According to Bolzano–Weierstrass theorem, we know that each infinite bounded sequence in $\mathbb{R}^n$ has a convergent subsequence. There must be at least one accumulation point of the sequence $\{\mathcal{L}_k,\mathcal{E}_k,\mathcal{P}_k\}_{k=1}^{+\infty}$. We denote one of the points as $\{\mathcal{L}^*,\mathcal{E}^*,\mathcal{P}^*\}$, and we assume $\{\mathcal{L}_k,\mathcal{E}_k,\mathcal{P}_k\}_{k=1}^{+\infty}$ converge to $\{\mathcal{L}^*,\mathcal{E}^*,\mathcal{P}^*\}$ without loss of generality.

Since

$$(\mathcal{P}_{k+1}-\mathcal{P}_k)/\mu_k=\mathcal{L}_{k+1}+\mathcal{E}_{k+1}-\mathcal{X},$$

we can deduce that

$$\lim_{k\to\infty}(\mathcal{L}_{k+1}+\mathcal{E}_{k+1}-\mathcal{X})=\lim_{k\to\infty}(\mathcal{P}_{k+1}-\mathcal{P}_k)/\mu_k=0.$$

Then $\mathcal{L}^*+\mathcal{E}^*=\mathcal{X}$ is achieved.

$\mathcal{L}_{k+1}$ also satisfies the first-order necessary local optimality condition of (16)

$$0\in\partial\|\mathcal{L}_{k+1}\|_{log}+\mathcal{P}_k+\mu_k\left(\mathcal{L}_{k+1}+\mathcal{E}_k-\mathcal{X}\right).$$

From Lemma 2 and Definition 6, for $i=1,\dots,n_3$, we have

$$\nabla\|\bar{\mathcal{L}}^{(i)}\|_{log}=\bar{\boldsymbol{U}}^{(i)}\mathrm{diag}(\frac{\theta}{\theta\sigma_j(\bar{\mathcal{L}}^{(i)})+1})\bar{\boldsymbol{V}}^{(i)T},$$

and

$$\frac{\theta}{\theta\sigma_j(\bar{\mathcal{L}}^{(i)})+1}<\theta\Longrightarrow\nabla\|\bar{\mathcal{L}}^{(i)}\|_{log}$$

is bounded. Thus, $\frac{\partial\|\mathcal{L}\|_{log}}{\partial\bar{\mathcal{L}}}$ is bounded. In fact, $\bar{\mathcal{L}}$ is equivalent to Tucker product (see Kong et al. (2018)), $\bar{\mathcal{L}}=\mathcal{L}\times_3\widetilde{\mathbf{F}}_{n_3}$. From it and using the chain rule, we can obtain that

$$\nabla\|\mathcal{L}\|_{log}=\frac{\partial\|\mathcal{L}\|_{log}}{\partial\bar{\mathcal{L}}}\times_3\widetilde{\mathbf{F}}_{n_3}^*$$

is bounded. And it is easy to know

$$0\in\partial\|\mathcal{L}_{k+1}\|_{log}+\mathcal{P}_{k+1}-\mu_k\left(\mathcal{E}_{k+1}-\mathcal{E}_k\right).$$

Note that $\mu_k<\mu_{max}=10^{10}$ in Algorithm 1, then clearly $\lim_{k\to\infty}\mu_k\left(\mathcal{E}_{k+1}-\mathcal{E}_k\right)=0\Longrightarrow-\mathcal{P}^*\in\partial\|\mathcal{L}^*\|_{log}$.

Similarly, due to the fact that $\mathcal{E}_{k+1}$ is the minimum of the subproblem $L\left(\mathcal{L}_k,\mathcal{E},\mathcal{P}_k,\mu_k\right)$, we have

$$0\in\partial\|\mathcal{E}_{k+1}\|_1+\mathcal{P}_k+\mu_k\left(\mathcal{L}_k+\mathcal{E}_{k+1}-\mathcal{X}\right).$$

So $\lim_{k\to\infty}\mathcal{P}_{k+1}=\mathcal{P}^*\in\partial\|\mathcal{E}^*\|_1$. Thus $\left(\mathcal{L}^*,\mathcal{E}^*,\mathcal{P}^*\right)$ satisfies the Karush-Kuhn-Tuker (KKT) conditions of the Lagrange function $L(\mathcal{L},\mathcal{E},\mathcal{P},\mu)$. We are now in a position to complete the proof.

References

Boyd, S., Parikh, N., Chu, E., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. 3 (1), 1–122.

Buades, A., Coll, B., Morel, J.-M., 2005. A review of image denoising algorithms, with a new one. Multiscale Model. Simul. 4 (2), 490–530.

Cai, J.-F., Candès, E.J., Shen, Z., 2010. A singular value thresholding algorithm for matrix completion. SIAM J. Optim. 20 (4), 1956–1982.

Cai, S., Luo, Q., Yang, M., Li, W., Xiao, M., 2019. Tensor robust principal component analysis via non-convex low rank approximation. Appl. Sci. 9 (7), 1411.

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

Candès, E.J., Recht, B., 2009. Exact matrix completion via convex optimization. Found. Comput. Math. 9 (6), 717–772.

Candès, E.J., Tao, T., 2010. The power of convex relaxation: Near-optimal matrix completion. IEEE Trans. Inform. Theory 56 (5), 2053–2080.

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

Dong, W., Shi, G., Li, X., Ma, Y., Huang, F., 2014. Compressive sensing via nonlocal low-rank regularization. IEEE Trans. Image Process. 23 (8), 3618–3632.

Dong, W., Zhang, L., Shi, G., Li, X., 2013. Nonlocally centralized sparse representation for image restoration. IEEE Trans. Image Process. 22 (4), 1620–1630.

Geng, X., Guo, Q., Zhang, C., 2021. Color image denoising via tensor robust PCA with nonconvex and nonlocal regularization. In: Proceedings of ACM Multimedia Asia.

Gu, S., Xie, Q., Meng, D., Zuo, W., Feng, X., Zhang, L., 2017. Weighted nuclear norm minimization and its applications to low level vision. Int. J. Comput. Vis. 121 (2), 183–208.

Guo, Q., Gao, S., Zhang, X., Yin, Y., Zhang, C., 2018. Patch-based image inpainting via two-stage low rank approximation. IEEE Trans. Vis. Comput. Graphics 24 (6), 2023–2036.

Guo, Q., Zhang, Y., Qiu, S., Zhang, C., 2021. Accelerating patch-based low-rank image restoration using kd-forest and Lanczos approximation. Inform. Sci. 556, 177–193.

Guo, Q., Zhang, C., Zhang, Y., Liu, H., 2016. An efficient SVD-based method for image denoising. IEEE Trans. Circuits Syst. Video Technol. 26 (5), 868–880.

Han, Z., Zhang, S., Liu, Z., Wang, Y., Yao, J., Wang, Y., 2023. Tensor robust principal component analysis with side information: Models and applications. IEEE Trans. Circuits Syst. Video Technol. 33 (8), 3713–3725.

Hu, Y., Zhang, D., Ye, J., Li, X., He, X., 2013. Fast and accurate matrix completion via truncated nuclear norm regularization. IEEE Trans. Pattern Anal. Mach. Intell. 35 (9), 2117–2130.

Huang, B., Mu, C., Goldfarb, D., Wright, J., 2015. Provable models for robust low-rank tensor completion. Pac. J. Optim. 11 (2), 339–364.

Jiang, T.-X., Ng, M.K., Zhao, X.-L., Huang, T.-Z., 2020. Framelet representation of tensor nuclear norm for third-order tensor completion. IEEE Trans. Image Process. 29, 7233–7244.

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

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

Kong, H., Xie, X., Lin, Z., 2018. T-Schatten-p norm for low-rank tensor recovery. IEEE J. Sel. Top. Sign. Proces. 12 (6), 1405–1419.

Leurgans, S.E., Ross, R.T., Abel, R.B., 1993. A decomposition for three-way arrays. SIAM J. Matrix Anal. Appl. 14 (4), 1064–1083.

Lewis, A.S., Sendov, H.S., 2005. Nonsmooth analysis of singular values. Part I: Theory. Set-Valued Anal. 13, 213–241.

Liang, X., Tang, Z., Huang, Z., Zhang, X., Zhang, S., 2023. Efficient hashing method using 2D-2D PCA for image copy detection. IEEE Trans. Knowl. Data Eng. 35 (4), 3765–3778.

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

Lu, C., Feng, J., Chen, Y., Liu, W., Lin, Z., Yan, S., 2020. Tensor robust principal component analysis with a new tensor nuclear norm. IEEE Trans. Pattern Anal. Mach. Intell. 42 (4), 925–938.

Maddalena, L., Petrosino, A., 2015. Towards benchmarking scene background initialization. In: Proceedings of International Conference on Image Analysis and Processing. pp. 469–476.

Markowitz, S., Snyder, C., Eldar, Y.C., Do, M.N., 2022. Multimodal unrolled robust PCA for background foreground separation. IEEE Trans. Image Process. 31, 3553–3564.

Martin, D., Fowlkes, C., Tal, D., Malik, J., 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 416–423.

Nie, F., Hu, Z., Li, X., 2019. Matrix completion based on non-convex low-rank approximation. IEEE Trans. Image Process. 28 (5), 2378–2388.

Oh, T.-H., Tai, Y.-W., Bazin, J.-C., Kim, H., Kweon, I.S., 2016. Partial sum minimization of singular values in robust PCA: Algorithm and applications. IEEE Trans. Pattern Anal. Mach. Intell. 38 (4), 744–758.

Qin, W., Wang, H., Zhang, F., Wang, J., Luo, X., Huang, T., 2022. Low-rank high-order tensor completion with applications in visual data. IEEE Trans. Image Process. 31, 2433–2448.

Qiu, D., Bai, M., Ng, M.K., Zhang, X., 2021. Nonlocal robust tensor recovery with nonconvex regularization. Inverse Problems 37 (3), 035001.

Shi, M., Zhang, F., Wang, S., Zhang, C., Li, X., 2021. Detail preserving image denoising with patch-based structure similarity via sparse representation and SVD. Comput. Vis. Image Underst. 206, 103173.

Simon, H.D., Zha, H., 2000. Low-rank matrix approximation using the lanczos bidiagonalization process with applications. SIAM J. Sci. Comput. 21 (6), 2257–2274.

Sun, M., Zhao, L., Zheng, J., Xu, J., 2020. A nonlocal denoising framework based on tensor robust principal component analysis with ℓ_p norm. In: IEEE International Conference on Big Data. pp. 3333–3340.

Tibshirani, R., 2011. Regression shrinkage and selection via the lasso: A retrospective. J. R. Stat. Soc. 73 (3), 273–282.

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

Wang, Z., Bovik, A., Sheikh, H., Simoncelli, E., 2004. Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process. 13 (4), 600–612.

Wang, Q., Gao, Q., Gao, X., Nie, F., 2018. $\ell_{2,p}$-Norm based PCA for image recognition. IEEE Trans. Image Process. 27 (3), 1336–1346.

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

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

Xu, W., Zhao, X., Ji, T., Miao, J., Ma, T., Wang, S., Huang, T., 2019a. Laplace function based nonconvex surrogate for low-rank tensor completion. Signal Process., Image Commun. 73, 62–69.

Xu, W.-H., Zhao, X.-L., Ng, M., 2019b. A fast algorithm for cosine transform based tensor singular value decomposition. arXiv preprint arXiv:1902.03070.

Xue, J., Zhao, Y., Liao, W., Chan, J.C.-W., 2019. Nonconvex tensor rank minimization and its applications to tensor recovery. Inform. Sci. 503, 109–128.

Yang, M., Luo, Q., Li, W., Xiao, M., 2022. Nonconvex 3D array image data recovery and pattern recognition under tensor framework. Pattern Recognit. 122, 108311.

Yayci, Z.O., Dura, U., Kaya, Z.B., Cetin, A.E., Turkan, M., 2022. Microscale image enhancement via PCA and well-exposedness maps. In: Proceedings of IEEE International Conference on Image Processing. pp. 2092–2096.

Zarmehi, N., Amini, A., Marvasti, F., 2020. Low rank and sparse decomposition for image and video applications. IEEE Trans. Circuits Syst. Video Technol. 30 (7), 2046–2056.

Zhang, M., Desrosiers, C., 2018. Structure preserving image denoising based on low-rank reconstruction and gradient histograms. Comput. Vis. Image Underst. 171, 48–60.

Zhang, L., Song, L., Du, B., Zhang, Y., 2021. Nonlocal low-rank tensor completion for visual data. IEEE Trans. Cybern. 51 (2), 673–685.

Zhang, X., Yuan, X., Carin, L., 2018. Nonlocal low-rank tensor factor analysis for image restoration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8232–8241.

Zhang, L., Zhang, L., Mou, X., Zhang, D., 2011. FSIM: A feature similarity index for image quality assessment. IEEE Trans. Image Process. 20 (8), 2378–2386.

Zhang, X., Zheng, J., Wang, D., Zhao, L., 2020. Exemplar-based denoising: A unified low-rank recovery framework. IEEE Trans. Circuits Syst. Video Technol. 30 (8), 2538–2549.

Zhong, G., Pun, C.-M., 2021. RPCA-induced self-representation for subspace clustering. Neurocomputing 437, 249–260.