

Tensor robust principal component analysis via dual l_p quasi-norm sparse constraints

Tinghe Yan ^{a,b}, Qiang Guo ^{a,b,*}

^a School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan 250014, China

^b Shandong Provincial Key Laboratory of Digital Media Technology, Jinan 250014, China



ARTICLE INFO

Keywords:

Tensor robust principal component analysis
Tensor singular value decomposition
Low-rank representation
Image recovery
Background modeling

ABSTRACT

As a pioneering method, tensor robust principal component analysis (TRPCA) can separate an underlying low-rank component and a sparse component from the original data by minimizing a convex objective function composed of tensor nuclear norm and l_1 -norm. However, it has two limitations. One is that tensor nuclear norm, as a constraint on the low-rank component, treats all singular values uniformly, ignoring the differences among singular values. In essence, this constraint is a sparse constraint that is imposed on singular values of the low-rank component by l_1 -norm. The other is that l_1 -norm is used as a constraint for the sparse component. However, l_1 -norm is a loose constraint, leading to the solutions of TRPCA deviating from the authentic ones. To alleviate these issues, we propose a TRPCA model called p -TRPCA that utilizes the l_p -norm to impose sparse constraints on both the singular values and the sparse component simultaneously. The l_p quasi-norm ($0 < p < 1$) is a tighter constraint than l_1 -norm, which enhances the low-rankness and sparsity of the proposed model. To solve p -TRPCA, we present an effective algorithm based on the alternating direction method of multipliers and also analyze its convergence. Extensive experiments are performed on data simulation, image recovery, and background modeling. Experimental results show that our p -TRPCA outperforms TRPCA and its variants.

1. Introduction

Principal component analysis (PCA) is a widely used technique for data dimensionality reduction, which extracts significant low-dimensional structures. In particular, it is very effective in handling sparse outliers. However, real-world data often suffer from gross corruptions that are not effectively handled by PCA, leading to the development of numerous robust PCA methods. While these methods are effective for 2-way data (i.e., matrices), they need to reshape data into matrices when dealing with multi-dimensional data. This reshaping breaks the intrinsic structure of the data, resulting in a degradation in performance. Three-dimensional data, such as videos, color images, and hyperspectral images, are commonly encountered in real-world applications. They are stored in three-dimensional arrays called third-order tensors [1]. Thus, some PCA methods that utilize tensor singular value decomposition (t-SVD) have been proposed to effectively handle third-order tensors.

The robust principal component analysis (RPCA) [2], a classic method, can recover the underlying low-rank structures of data matr-

ices corrupted by sparse noise or outliers. Given a data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, RPCA decomposes it into a low-rank matrix $\mathbf{L} \in \mathbb{R}^{m \times n}$ and a sparse matrix $\mathbf{S} \in \mathbb{R}^{m \times n}$, i.e., $\mathbf{X} = \mathbf{L} + \mathbf{S}$, by solving the following problem

$$\min_{\mathbf{L}, \mathbf{S}} \text{rank}(\mathbf{L}) + \lambda \|\mathbf{S}\|_0 \quad \text{s.t. } \mathbf{X} = \mathbf{L} + \mathbf{S}, \quad (1)$$

where $\text{rank}(\cdot)$ denotes the rank, and $\|\cdot\|_0$ is the l_0 -norm. Solving Eq. (1) is NP-hard due to the non-convexity and discontinuity of the rank function and l_0 -norm. The nuclear norm and l_1 -norm are typically employed as convex surrogates for them [3]. Consequently, RPCA is transformed into the following convex optimization problem

$$\min_{\mathbf{L}, \mathbf{S}} \|\mathbf{L}\|_* + \lambda \|\mathbf{S}\|_1 \quad \text{s.t. } \mathbf{X} = \mathbf{L} + \mathbf{S}, \quad (2)$$

where $\|\cdot\|_*$ and $\|\cdot\|_1$ represent the nuclear norm and l_1 -norm, respectively. Note that, in Eq. (1), $\text{rank}(\cdot)$ is essentially the number of non-zero singular values of a matrix, and $\|\cdot\|_0$ is the number of non-zero entries of a matrix. In this paper, we aim to improve the approximations of the rank and l_0 -norm by employing dual l_p -norm constraints. In fact, when $0 < p < 1$, the l_p -norm is a quasi-norm [4]. Note that following

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

[5,6], we misuse the l_p -norm for simplicity. Since the l_p quasi-norm is non-convex, how to effectively minimize it is an important problem. Several solving strategies have been developed, including proximal operators [7] and a two-block alternating direction method of multipliers [4]. In particular, a generalized soft-thresholding (GST) operator [5] was proposed for solving the l_p quasi-norm minimization problem with arbitrary values of p . Its effectiveness and efficiency have been fully verified [6,8]. Thus, this paper utilizes the GST operator to solve the l_p quasi-norm minimization problem.

When solving the objective function (2), RPCA exploits the singular value thresholding that all singular values are uniformly shrunk. However, treating all singular values equally makes RPCA inflexible. Although RPCA can be widely used for component decomposition, the nuclear norm and l_1 -norm used in the model are relatively relaxed. Some variants of RPCA have been proposed by applying different norm constraints on the low-rank component \mathbf{L} , including partial sum of singular values [9], weighted nuclear norm [10], weighted Schatten p -norm [6], and so on. The distinction between these variants arises from their utilization of different norm constraints, which makes their respective models closer to the low-rank assumption. In general, each singular value has a unique physical meaning. Moreover, large singular values represent significant information. To retain significant information, [9] employed a partial sum of singular values (PSSV) constraint that preserves the first j singular values unchanged and only shrinks the remaining singular values. Specifically, PSSV is defined as

$$\|\mathbf{L}\|_{p=j} = \sum_{i=j+1}^{\min(m,n)} \sigma_i(\mathbf{L}), \quad (3)$$

where $\sigma_i(\cdot)$ denotes the i -th largest singular value of a matrix. PSSV assumes that the first j large singular values are not encoded with irrelevant information. However, this assumption is not rigorous in practice [11]. For better modeling of practical applications, the weighted RPCA (WRPCA) [10] flexibly treats singular values by adopting the following weighted nuclear norm

$$\|\mathbf{L}\|_{\omega,*} = \sum_{i=1}^{\min(m,n)} \omega_i \cdot \sigma_i(\mathbf{L}), \quad (4)$$

where ω_i is the i -th element of the weight vector ω . Furthermore, the following weighted Schatten p -norm (WSN) [6] was proposed to model the low-rank component \mathbf{L} ,

$$\|\mathbf{L}\|_{\omega,S_p} = \left(\sum_{i=1}^{\min(m,n)} \omega_i \cdot (\sigma_i(\mathbf{L}))^p \right)^{\frac{1}{p}}. \quad (5)$$

Thanks to the better low-rank approximation of the component \mathbf{L} , the performance of WSN-RPCA is superior to WRPCA.

These robust variants of PCA have been widely used in various fields, such as background modeling [10,12], subspace clustering [13,14], inpainting [15,16], and denoising [17–19]. However, these methods are only applicable to handling matrix cases. When dealing with third-order tensors, their performance significantly degrades due to the information loss in the process of reshaping third-order tensors to matrices.

To address this problem, motivated by t-SVD, Lu et al. [20] presented TRPCA which extends from matrices to tensors. Suppose that there is an observed third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, which can be decomposed into a low-rank tensor $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and a sparse tensor $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, i.e., $\mathcal{X} = \mathcal{L} + \mathcal{S}$. Then TRPCA obtains the components \mathcal{L} and \mathcal{S} by solving the following minimization problem

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

In the above equation, $\|\mathcal{L}\|_*$ refers to the tensor nuclear norm of \mathcal{L} , which is defined as

$$\|\mathcal{L}\|_* = \frac{1}{n_3} \sum_{i=1}^{n_3} \|\overline{\mathbf{L}}^{(i)}\|_* = \frac{1}{n_3} \sum_{i=1}^{n_3} \sum_{j=1}^{n_{\min}} \sigma_j(\overline{\mathbf{L}}^{(i)}), \quad (7)$$

where $\overline{\mathbf{L}}^{(i)}$ is the i -th frontal slice of $\overline{\mathcal{L}} = \text{fft}(\mathcal{L}, [], 3)$, and $n_{\min} = \min(n_1, n_2)$. With the emergence of TRPCA, tensor-based methods have been further developed for background modeling [21,8], data denoising [22–24], and multi-view clustering [11,25].

Although TRPCA has shown impressive performance, it still has several limitations. When attempting to minimize the tensor nuclear norm, TRPCA utilizes the tensor singular value thresholding that shrinks all singular values uniformly. In practical scenarios, singular values carry different physical meanings, and there exists prior knowledge indicating that larger singular values are typically related to prominent information. The uniform shrinkage operation of TRPCA ignores the difference between singular values. To preserve prominent information, inspired by PSSV, Jiang et al. [26] proposed the partial sum of the tensor nuclear norm (PSTNN) which is described as

$$\|\mathcal{L}\|_{p=j} = \frac{1}{n_3} \sum_{i=1}^{n_3} \|\overline{\mathbf{L}}^{(i)}\|_{p=j} = \frac{1}{n_3} \sum_{i=1}^{n_3} \sum_{k=j+1}^{n_{\min}} \sigma_k(\overline{\mathbf{L}}^{(i)}). \quad (8)$$

Similar to PSSV, PSTNN keeps the first j singular values fixed and only shrinks others, which results in the first j prominent singular values with some irrelevant information.

To eliminate irrelevant information embedded in the prominent information, Gao et al. [8] proposed an enhanced TRPCA (ETRPCA) that reasonably utilizes the prior knowledge of singular values. ETRPCA models the low-rank component \mathcal{L} by the following weighted tensor Schatten p -norm (WTSN)

$$\|\mathcal{L}\|_{\omega, S_p} = \left(\frac{1}{n_3} \sum_{i=1}^{n_3} \|\overline{\mathbf{L}}^{(i)}\|_{\omega, S_p}^p \right)^{\frac{1}{p}} = \left(\frac{1}{n_3} \sum_{i=1}^{n_3} \sum_{j=1}^{n_{\min}} \omega_j \cdot \sigma_j(\overline{\mathbf{L}}^{(i)})^p \right)^{\frac{1}{p}}, \quad (9)$$

where $0 < p \leq 1$ and ω is a weight vector that satisfies $0 \leq \omega_1 \leq \omega_2 \leq \dots \leq \omega_{n_{\min}}$. As a result, the shrinkage operator derived by minimizing the WTSN constraint term shrinks less for large singular values and more for small singular values, which can effectively preserve the prominent information and eliminate irrelevant information.

In essence, the norm constraint on the low-rank component corresponds to a strategy of shrinking singular values. Furthermore, these norm constraints actually are equivalent to sparse constraints on singular values imposed by sparse norms. For example, the tensor nuclear norm constraint imposes a l_1 -norm constraint on singular values. Thus, by employing an appropriate sparse norm, singular values are shrunk reasonably, thereby enhancing the low-rank property of modeling. It is worth noting that the aforementioned methods only improve constraints enforced on the low-rank component while neglecting the improvement of the constraint on the sparse one. For these methods, the l_1 -norm provides a loose approximation as a sparse constraint. To alleviate this issue, several strategies were proposed to enhance the sparse constraint, e.g., l_p -norm [27], weighted l_1 -norm [21], and reweighted l_1 -norm [28]. Very recently, [21] presented a double auto-weighted TRPCA (DTRPCA), which utilizes the weighted l_1 -norm as a sparse constraint. Generally, the l_p -norm ($0 < p < 1$) is a tighter sparse constraint than the l_1 -norm [5]. Hence, we propose a TRPCA model called p -TRPCA, which constrains tensor singular values by the weighted l_p -norm and the sparse component by the l_p -norm.

The main contributions of our work are summarized as follows:

- We propose a new TRPCA model, i.e., p -TRPCA, based on dual l_p -norm sparse constraints. The weighted l_p -norm provides a flexible treatment of singular values as a sparse constraint, enhancing the low-rankness of p -TRPCA. Moreover, the l_p -norm makes the sparse component closer to the sparse assumption, which further has a beneficial impact on the low-rank approximation of our model.

Algorithm 1 T-SVD [29].

Input: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$.
 Compute $\bar{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3)$.
 Compute SVD on each frontal slice of $\bar{\mathcal{A}}$ by
for $i = 1, 2, \dots, n_3$ **do**
 $[\bar{\mathbf{U}}^{(i)}, \bar{\mathbf{D}}^{(i)}, \bar{\mathbf{V}}^{(i)}] = \text{SVD}(\bar{\mathbf{A}}^{(i)})$;
end for
 Compute $\mathcal{U} = \text{ifft}(\bar{\mathcal{U}}, [], 3)$, $\mathcal{D} = \text{ifft}(\bar{\mathcal{D}}, [], 3)$, and $\mathcal{V} = \text{ifft}(\bar{\mathcal{V}}, [], 3)$.
Output: T-SVD components \mathcal{U} , \mathcal{V} , and \mathcal{D} of \mathcal{A} .

- To solve p -TRPCA, we develop an efficient optimization algorithm with the alternating direction method of multipliers and analyze its convergence. Meanwhile, the relationships between p -TRPCA and eight popular PCA methods are also discussed. It is indicated that RPCA and its several variants can be regarded as special cases of our p -TRPCA.
- We conduct extensive experiments on data simulation, image recovery, and background modeling. Numerical simulation experiments verify that p -TRPCA obtains the best recovery performance among the competing methods. Furthermore, experimental results on image recovery and background modeling reveal that p -TRPCA can more effectively recover the corrupted images and model background compared with some existing methods.

The structure of this paper is as follows. Section 2 introduces some preliminaries. A detailed description of p -TRPCA is provided in Section 3. Section 4 first analyzes the convergence as well as the computational complexity of the proposed algorithm, and then discusses the relationships between p -TRPCA and some related works. The comparisons of p -TRPCA with various SOTA methods are presented in Section 5. Finally, Section 6 draws a conclusion.

2. Preliminaries

In this section, some symbols and definitions used throughout the paper are introduced. We denote third-order tensors by bold calligraphy letters, e.g., \mathcal{A} ; matrices by bold upper case letters, e.g., \mathbf{A} ; vectors by bold lower case letters, e.g., \mathbf{a} ; and scalars by lowercase letters, e.g., a . For a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, a_{ijk} is (i, j, k) -th element of \mathcal{A} , its i -th frontal slice is denoted succinctly as $\mathbf{A}^{(i)}$. Moreover, $\bar{\mathcal{A}}$ is denoted as the discrete Fast Fourier Transform (FFT) of \mathcal{A} along the third dimension, i.e., $\bar{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3)$, and \mathcal{A} is obtained by using the inverse FFT, i.e., $\mathcal{A} = \text{ifft}(\bar{\mathcal{A}}, [], 3)$.

Besides, for a matrix $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$, its l_p -norm and nuclear norm are defined as $\|\mathbf{A}\|_p = (\sum_{ij} |a_{ij}|^p)^{\frac{1}{p}}$ and $\|\mathbf{A}\|_* = \sum_i \sigma_i(\mathbf{A})$, respectively, where $\sigma_i(\mathbf{A})$ is the i -th largest singular value of \mathbf{A} . For a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we define $\|\mathcal{A}\|_1 = \sum_{ijk} |a_{ijk}|$, $\|\mathcal{A}\|_p = (\sum_{ijk} |a_{ijk}|^p)^{\frac{1}{p}}$, and $\|\mathcal{A}\|_F = \sqrt{\sum_{ijk} |a_{ijk}|^2}$.

Definition 1. (T-SVD) [29]. Let $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, then \mathcal{A} 's t-SVD is

$$\mathcal{A} = \mathcal{U} * \mathcal{D} * \mathcal{V}^T, \quad (10)$$

where $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors, \mathcal{V}^T denotes the conjugate transpose of \mathcal{V} , and $\mathcal{D} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is f-diagonal. The t-SVD components \mathcal{U} , \mathcal{V} , and \mathcal{D} of \mathcal{A} can be obtained by Algorithm 1.

Definition 2. (Block Diagonal Matrix) [20]. For a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the block diagonal matrix $\text{bdiag}(\bar{\mathcal{A}}) \in \mathbb{C}^{n_1 n_3 \times n_2 n_3}$ is

$$\text{bdiag}(\bar{\mathcal{A}}) = \begin{bmatrix} \bar{\mathbf{A}}^{(1)} & & & \\ & \bar{\mathbf{A}}^{(2)} & & \\ & & \ddots & \\ & & & \bar{\mathbf{A}}^{(n_3)} \end{bmatrix}. \quad (11)$$

3. The proposed p -TRPCA

3.1. Objective

TRPCA [20], as described in Eq. (6), is a pioneering tensor robust principal component analysis that can effectively separate a low-rank component \mathcal{L} and a sparse one \mathcal{S} from an observed tensor \mathcal{X} . It employs the tensor nuclear norm as a low-rank constraint. In fact, the tensor nuclear norm is equivalent to the l_1 -norm of singular values. Unlike TRPCA, PSTNN [26] utilizes the partial sum of the tensor nuclear norm, a special l_1 -norm of the truncated singular values, to constrain the low-rank component. Besides, ETRPCA [8] adopts the weighted tensor Schatten p -norm as a constraint to achieve satisfactory performance. Essentially, the weighted tensor Schatten p -norm constraint is a sparse constraint on tensor singular values by the weighted l_p -norm.

The three above models only focus on the low-rank approximation of the component \mathcal{L} while ignoring the sparse component \mathcal{S} . They all utilize the l_1 -norm as a substitution for l_0 -norm to constrain the sparse component. However, as discussed in Section 1, l_p -norm ($0 < p < 1$) can better model the sparse constraint than l_1 -norm, which obtains a tight substitution for l_0 -norm. Motivated by this, we propose the p -TRPCA that exploits the weighted l_p -norm as a sparse constraint of the singular values and the l_p -norm as a constraint on the sparse component. Therefore, the objective function of p -TRPCA is

$$\min_{\mathcal{L}, \mathcal{S}} \|\mathcal{L}\|_{\omega, S_p}^p + \lambda \|\mathcal{S}\|_p^p \quad \text{s.t. } \mathcal{X} = \mathcal{L} + \mathcal{S}, \quad (12)$$

where λ is a positive balanced parameter. This minimization problem is a non-convex optimization problem for the l_p -norm ($0 < p < 1$). As described in Eq. (9), $\|\mathcal{L}\|_{\omega, S_p}$ is actually the weighted l_p -norm of singular values. $\|\mathcal{S}\|_p$ denotes the l_p -norm of \mathcal{S} . From the perspective of the objective function, the main difference between p -TRPCA and ETRPCA is that the former employs the l_p -norm to constrain the sparse component, rather than the l_1 -norm. In fact, this difference affects not only the sparse component but also the low-rank component. The proposed method produces a more accurate sparse component \mathcal{S} through the strong regularization imposed by the l_p -norm, which in turn leads to a more accurate low-rank component \mathcal{L} since these two components are decomposed from the tensor \mathcal{X} .

3.2. Optimization

The alternating direction method of multipliers (ADMM) is an efficient method to solve convex and non-convex optimization problems [30–32]. Thus, in this paper, we adopt ADMM to solve Eq. (12). The augmented Lagrangian function of Eq. (12) is

$$\Gamma(\mathcal{L}, \mathcal{S}, \mathcal{Z}, \mu) = \|\mathcal{L}\|_{\omega, S_p}^p + \lambda \|\mathcal{S}\|_p^p + \langle \mathcal{Z}, \mathcal{X} - \mathcal{L} - \mathcal{S} \rangle + \frac{\mu}{2} \|\mathcal{X} - \mathcal{L} - \mathcal{S}\|_F^2, \quad (13)$$

where \mathcal{Z} is the Lagrange multiplier and $\mu > 0$ is a penalty parameter. According to the alternating minimization method [33], minimizing Eq. (13) can be transformed into several minimization subproblems. Specifically, the problem of minimizing Eq. (13) can be divided into the following three subproblems.

Solving \mathcal{L} -subproblem. In the $(k+1)$ -th step, when \mathcal{S}_k , \mathcal{Z}_k , and μ_k are fixed, \mathcal{L}_{k+1} can be obtained by solving

$$\arg \min_{\mathcal{L}} \|\mathcal{L}\|_{\omega, S_p}^p + \frac{\mu_k}{2} \|\mathcal{X} - \mathcal{L} - \mathcal{S}_k\|_F^2, \quad (14)$$

where $\mathcal{Y}_k = \mathcal{X} - \mathcal{S}_k + \mu_k^{-1} \mathcal{Z}_k$. Theorem 1 provides the solution to Eq. (14).

Theorem 1 ([8]). Let $\mathcal{Y} = \mathcal{U} * \mathcal{D} * \mathcal{V}^T$ be the t-SVD of $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mu > 0$, $n_{min} = \min(n_1, n_2)$, $0 \leq \omega_1 \leq \omega_2 \leq \dots \leq \omega_{n_{min}}$, $0 < p \leq 1$, for the following minimization problem

$$\arg \min_{\mathcal{L}} \|\mathcal{L}\|_{\omega, S_p}^p + \frac{\mu}{2} \|\mathcal{L} - \mathcal{Y}\|_F^2, \quad (15)$$

its global optimal solution is

$$\mathcal{L} = \mathcal{U} * \text{ifft} \left(\mathbf{P}_{\omega/\mu}(\bar{\mathcal{Y}}) \right) * \mathcal{V}^T, \quad (16)$$

where $\mathbf{P}_{\omega/\mu}(\cdot)$ is a thresholding operator. Specifically,

$$\begin{aligned} \mathbf{P}_{\omega/\mu}(\bar{\mathcal{Y}}^{(i)}) = \text{diag} \left(T_p^{GST}(\sigma_1(\bar{\mathcal{Y}}^{(i)}), \omega_1/\mu), T_p^{GST}(\sigma_2(\bar{\mathcal{Y}}^{(i)}), \omega_2/\mu), \dots, \right. \\ \left. T_p^{GST}(\sigma_{n_{min}}(\bar{\mathcal{Y}}^{(i)}), \omega_{n_{min}}/\mu) \right), \end{aligned} \quad (17)$$

where $\mathbf{P}_{\omega/\mu}(\bar{\mathcal{Y}}^{(i)})$ is the i -th frontal slice of $\mathbf{P}_{\omega/\mu}(\bar{\mathcal{Y}})$ and $T_p^{GST}(\cdot)$ is a generalized soft-thresholding (GST) operator [5], which is given by the following Theorem 2.

Theorem 2 ([5]). Given the scalars y , ω , and p ($0 < p \leq 1$), for the following minimization problem

$$\min_y \frac{1}{2}(l - y)^2 + \omega|y|^p, \quad (18)$$

its optimal solution is

$$T_p^{GST}(y, \omega) = \begin{cases} \text{sign}(y) S_p^{GST}(|y|, \omega), & |y| \in (\tau_p^{GST}(\omega), +\infty) \\ 0, & |y| \in [0, \tau_p^{GST}(\omega)] \end{cases}, \quad (19)$$

where $\tau_p^{GST}(\omega)$ is a specific threshold defined as

$$\tau_p^{GST}(\omega) = (2\omega(1-p))^{\frac{1}{2-p}} + \omega p(2\omega(1-p))^{\frac{p-1}{2-p}}, \quad (20)$$

and $S_p^{GST}(y, \omega)$ can be obtained by solving

$$S_p^{GST}(y, \omega) - y + \omega p(S_p^{GST}(y, \omega))^{p-1} = 0. \quad (21)$$

According to Theorem 1, the iterative solution of Eq. (14) is formulated as

$$\mathcal{L}_{k+1} = \mathcal{U}_k * \text{ifft} \left(\mathbf{P}_{\omega/\mu_k}(\bar{\mathcal{Y}}_k) \right) * \mathcal{V}_k^T. \quad (22)$$

As referred in [20], one can obtain the following formula

$$\text{conj}(\mathbf{P}_{\omega/\mu_k}(\bar{\mathcal{Y}}^{(i)})) = \mathbf{P}_{\omega/\mu_k}(\bar{\mathcal{Y}}^{(n_3-i+2)}), \quad i = 2, \dots, \lfloor \frac{n_3+1}{2} \rfloor, \quad (23)$$

where $\text{conj}(\cdot)$ denotes the complex conjugate. By utilizing Eq. (23), we can effectively reduce the number of computations when solving for Eq. (22). Algorithm 2 describes the iterative procedure of solving \mathcal{L} .

Solving \mathcal{S} -subproblem. When \mathcal{L}_{k+1} , \mathcal{Z}_k , and μ_k are fixed, \mathcal{S}_{k+1} can be derived by solving

$$\arg \min_{\mathcal{S}} \frac{1}{2} \|\mathcal{S} - \mathcal{N}_k\|_F^2 + \frac{\lambda}{\mu_k} \|\mathcal{S}\|_p^p, \quad (24)$$

where $\mathcal{N}_k = \mathcal{X} - \mathcal{L}_{k+1} + \mu_k^{-1} \mathcal{Z}_k$. In accordance with the property of Frobenius norm, Eq. (24) is equivalent to

$$\arg \min_{\mathcal{S}} \sum_{i=1}^{n_3} \left(\frac{1}{2} \|\mathbf{S}^{(i)} - \mathbf{N}_k^{(i)}\|_F^2 + \frac{\lambda}{\mu_k} \cdot \|\mathbf{S}^{(i)}\|_p^p \right), \quad (25)$$

where $\mathbf{S}^{(i)}$ and $\mathbf{N}_k^{(i)}$ are the i -th frontal slice of \mathcal{S} and \mathcal{N}_k , respectively. Due to each variable $\mathbf{S}^{(i)}$ being independent, the minimization problem of Eq. (25) can be decomposed into n_3 separate subproblems. Specifically, the i -th ($i = 1, 2, \dots, n_3$) subproblem is

Algorithm 2 Solving \mathcal{L} .

Input: Tensor data $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, weight vector ω .

Compute $\bar{\mathcal{Y}} = \text{fft}(\mathcal{Y}, [], 3)$.

Compute on each frontal slice of $\bar{\mathcal{Y}}$ by

for $i = 1, \dots, \lfloor \frac{n_3+1}{2} \rfloor$ do

$[\bar{\mathbf{U}}, \bar{\mathbf{D}}, \bar{\mathbf{V}}] = \text{SVD}(\bar{\mathcal{Y}}^{(i)})$;

$\mathbf{M} = \mathbf{P}_{\omega/\mu_k}(\bar{\mathbf{D}})$;

$\bar{\mathbf{F}}^{(i)} = \bar{\mathbf{U}} \cdot \mathbf{M} \cdot \bar{\mathbf{V}}^T$;

end for

for $i = \lfloor \frac{n_3+1}{2} \rfloor + 1, \dots, n_3$ do

$\bar{\mathbf{F}}^{(i)} = \text{conj}(\bar{\mathbf{F}}^{(n_3-i+2)})$;

end for

Compute $\mathcal{L} = \text{ifft}(\bar{\mathbf{F}}, [], 3)$.

Output: \mathcal{L} .

Algorithm 3 Solving p -TRPCA.

Input: Tensor data \mathcal{X} , weight vector ω , and parameter λ .

Initialize: $\mu_0 = 1e-4$, $\rho = 1.1$, $\mu_{max} = 1e10$, $\epsilon = 1e-8$, $\mathcal{L}_0 = \mathcal{S}_0 = \mathcal{Z}_0 = 0$.

while not converged do

Update \mathcal{L}_{k+1} by Algorithm 2;

Update \mathcal{S}_{k+1} by Eq. (29);

Update \mathcal{Z}_{k+1} by Eq. (30);

Update $\mu_{k+1} = \min(\rho \mu_k, \mu_{max})$;

Check the convergence conditions

$$\begin{aligned} \|\mathcal{L}_{k+1} - \mathcal{L}_k\|_F &\leq \epsilon, \quad \|\mathcal{S}_{k+1} - \mathcal{S}_k\|_F \leq \epsilon, \\ \|\mathcal{X} - \mathcal{L}_{k+1} - \mathcal{S}_{k+1}\|_F &\leq \epsilon. \end{aligned}$$

end while

Output: \mathcal{L} , \mathcal{S} .

$$\arg \min_{\mathbf{S}^{(i)}} \frac{1}{2} \|\mathbf{S}^{(i)} - \mathbf{N}_k^{(i)}\|_F^2 + \frac{\lambda}{\mu_k} \cdot \|\mathbf{S}^{(i)}\|_p^p, \quad (26)$$

which can be reformulated as

$$\arg \min_{\mathbf{S}^{(i)}} \sum_{j=1}^{n_1} \sum_{m=1}^{n_2} \left(\frac{1}{2} (s_{jm} - n_{jm})^2 + \frac{\lambda}{\mu_k} \cdot |s_{jm}|^p \right), \quad (27)$$

where s_{jm} and n_{jm} are (j, m) -th element of $\mathbf{S}^{(i)}$ and $\mathbf{N}_k^{(i)}$, respectively. Its optimal solution is given by Theorem 2 as follows

$$s_{jm} = \begin{cases} \text{sign}(n_{jm}) S_p^{GST}(|n_{jm}|, \frac{\lambda}{\mu_k}), & |n_{jm}| \in (\tau_p^{GST}(\frac{\lambda}{\mu_k}), +\infty) \\ 0, & |n_{jm}| \in [0, \tau_p^{GST}(\frac{\lambda}{\mu_k})] \end{cases}. \quad (28)$$

Thus,

$$s_{km} = T_p^{GST}(N_k, \frac{\lambda}{\mu_k}). \quad (29)$$

Solving \mathcal{Z} -subproblem. When \mathcal{L}_{k+1} , \mathcal{S}_{k+1} , and μ_k are fixed, the Lagrange multiplier \mathcal{Z}_{k+1} can be acquired by gradient ascent [33], i.e.,

$$\mathcal{Z}_{k+1} = \mathcal{Z}_k + \mu_k (\mathcal{X} - \mathcal{L}_{k+1} - \mathcal{S}_{k+1}). \quad (30)$$

Algorithm 3 summarizes the procedure for solving p -TRPCA.

4. Algorithm analysis and discussion

4.1. Convergence analysis

In the following, we analyze the convergence of Algorithm 3. The algorithm is guaranteed to converge to a locally optimal solution. Since the problem (12) is non-convex, there may exist other potentially better solutions. By restarting Algorithm 3 with different initial conditions, other potentially better solutions can be obtained. However, this way can incur a high computational burden. In this paper, we empirically select an initial condition (see Algorithm 3) to obtain a locally optimal solution. The experimental results also support its effectiveness.

Theorem 3. If the weight vector ω is nondecreasing, the sequences $\{\mathcal{L}_k\}$ and $\{\mathcal{S}_k\}$ meet the following three conditions

- C1. $\lim_{k \rightarrow \infty} \|\mathcal{X} - \mathcal{L}_{k+1} - \mathcal{S}_{k+1}\|_F = 0$.
- C2. $\lim_{k \rightarrow \infty} \|\mathcal{L}_{k+1} - \mathcal{L}_k\|_F = 0$.
- C3. $\lim_{k \rightarrow \infty} \|\mathcal{S}_{k+1} - \mathcal{S}_k\|_F = 0$.

Then Algorithm 3 is convergent.

Proof. The sequences $\{\mathcal{L}_k\}$, $\{\mathcal{S}_k\}$, and $\{\mathcal{Z}_k\}$ generated by Algorithm 3 are bounded, which is known by Theorems 4 and 5 in [8]. According to the Bolzano-Weierstrass Theorem [34], $\{\mathcal{Z}_k\}$ exists at least one limit point. Therefore, we have

$$\lim_{k \rightarrow \infty} \|\mathcal{Z}_{k+1} - \mathcal{Z}_k\|_F = 0. \quad (31)$$

From Eq. (30), the following equation holds,

$$\lim_{k \rightarrow \infty} \|\mathcal{X} - \mathcal{L}_{k+1} - \mathcal{S}_{k+1}\|_F = \lim_{k \rightarrow \infty} \frac{1}{\mu_k} \|\mathcal{Z}_{k+1} - \mathcal{Z}_k\|_F = 0. \quad (32)$$

Consequently, the condition C1 is proved.

For the \mathcal{L} -subproblem, we can obtain

$$\begin{aligned} & \lim_{k \rightarrow \infty} \|\mathcal{L}_{k+1} - \mathcal{L}_k\|_F \\ &= \lim_{k \rightarrow \infty} \|\mathcal{L}_{k+1} - \mathcal{X} + \mathcal{S}_k - \mu_k^{-1} \mathcal{Z}_k + \mu_k^{-1} \mathcal{Z}_k + \mu_{k-1}^{-1} (\mathcal{Z}_k - \mathcal{Z}_{k-1})\|_F \\ &= \lim_{k \rightarrow \infty} \|\mathcal{L}_{k+1} - \mathcal{Y}_k + \mu_k^{-1} \mathcal{Z}_k + \mu_{k-1}^{-1} (\mathcal{Z}_k - \mathcal{Z}_{k-1})\|_F. \end{aligned} \quad (33)$$

With $\mathcal{Y}_k = \mathcal{U}_k * \mathcal{D}_k * \mathcal{V}_k^T$, the low-rank component \mathcal{L}_{k+1} can be computed by Eq. (22). Thus, Eq. (33) can be reformulated as

$$\begin{aligned} & \lim_{k \rightarrow \infty} \|\mathcal{L}_{k+1} - \mathcal{L}_k\|_F \\ &= \lim_{k \rightarrow \infty} \|\mathcal{U}_k \cdot (\mathbf{P}_{\omega/\mu_k}(\mathcal{Y}_k) - (\mathcal{D}_k)) \cdot \mathcal{V}_k^T + \mu_k^{-1} \mathcal{Z}_k + \mu_{k-1}^{-1} (\mathcal{Z}_k - \mathcal{Z}_{k-1})\|_F \\ &\leq \lim_{k \rightarrow \infty} \left(\|\mathcal{U}_k \cdot (\mathbf{P}_{\omega/\mu_k}(\mathcal{Y}_k) - (\mathcal{D}_k)) \cdot \mathcal{V}_k^T\|_F + \|\mu_k^{-1} \mathcal{Z}_k + \mu_{k-1}^{-1} (\mathcal{Z}_k - \mathcal{Z}_{k-1})\|_F \right). \end{aligned} \quad (34)$$

According to Definition 2, for a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we have the following property

$$\|\mathcal{A}\|_F = \frac{1}{\sqrt{n_3}} \|\text{bdiag}(\overline{\mathcal{A}})\|_F. \quad (35)$$

Thus, Eq. (34) is equivalent to

$$\begin{aligned} & \lim_{k \rightarrow \infty} \|\mathcal{L}_{k+1} - \mathcal{L}_k\|_F \\ &\leq \lim_{k \rightarrow \infty} \left(\frac{1}{\sqrt{n_3}} \|\text{bdiag}(\overline{\mathcal{U}_k}) \cdot (\text{bdiag}(\mathbf{P}_{\omega/\mu_k}(\overline{\mathcal{Y}_k})) - \text{bdiag}(\overline{\mathcal{D}_k})) \cdot \text{bdiag}(\overline{\mathcal{V}_k})^T\|_F + \right. \\ &\quad \left. \|\mu_k^{-1} \mathcal{Z}_k + \mu_{k-1}^{-1} (\mathcal{Z}_k - \mathcal{Z}_{k-1})\|_F \right) \\ &\leq \lim_{k \rightarrow \infty} \left(\frac{1}{\sqrt{n_3}} \sqrt{\sum_{i=1}^{n_3} \sum_{j=1}^{n_{min}} J^2 \omega_j^2} + \|\mu_k^{-1} \mathcal{Z}_k + \mu_{k-1}^{-1} (\mathcal{Z}_k - \mathcal{Z}_{k-1})\|_F \right), \end{aligned} \quad (36)$$

where n_1 , n_2 , and n_3 are size of \mathcal{X} , $n_{min} = \min(n_1, n_2)$, and J is a positive integer that represents the number of iterations required to solve the minimization problem of Eq. (18). Obviously, as k approaches infinity, μ_k also approaches infinity [21]. Hence, the condition C2 holds, i.e.,

$$\lim_{k \rightarrow \infty} \|\mathcal{L}_{k+1} - \mathcal{L}_k\|_F = 0. \quad (37)$$

For the \mathcal{S} -subproblem, we have

$$\begin{aligned} & \lim_{k \rightarrow \infty} \|\mathcal{S}_{k+1} - \mathcal{S}_k\|_F \\ &= \lim_{k \rightarrow \infty} \|\mathcal{X} - \mathcal{L}_{k+1} - \mu_k^{-1} (\mathcal{Z}_{k+1} - \mathcal{Z}_k) - (\mathcal{X} - \mathcal{L}_k - \mu_{k-1}^{-1} (\mathcal{Z}_k - \mathcal{Z}_{k-1}))\|_F \\ &= \lim_{k \rightarrow \infty} \|\mathcal{L}_k - \mathcal{L}_{k+1} + \mu_{k-1}^{-1} (\mathcal{Z}_k - \mathcal{Z}_{k-1}) - \mu_k^{-1} (\mathcal{Z}_{k+1} - \mathcal{Z}_k)\|_F \\ &\leq \lim_{k \rightarrow \infty} \left(\|\mathcal{L}_{k+1} - \mathcal{L}_k\|_F + \mu_{k-1}^{-1} \|(\mathcal{Z}_k - \mathcal{Z}_{k-1})\|_F - \mu_k^{-1} \|(\mathcal{Z}_{k+1} - \mathcal{Z}_k)\|_F \right) \\ &= 0. \end{aligned} \quad (38)$$

Note that Eq. (38) equaling to 0 is obtained by substituting conditions C1 and C2 into the equation on the right-hand side of the inequality. Hence, the condition C3 is established. The convergence proof of Algorithm 3 is completed. \square

4.2. Computational complexity analysis

The per-iteration complexity of solving p -TRPCA mainly depends on updating \mathcal{S} and \mathcal{L} . It can be known from Eq. (29) that each update of \mathcal{S} needs to calculate $n_1 \cdot n_2 \cdot n_3 T_p^{GST}(\cdot)$ operations acting on scales. The computational cost of $T_p^{GST}(\cdot)$ is constant time. Hence, the computational complexity of updating \mathcal{S} is $\mathcal{O}(n_1 n_2 n_3)$. As shown in Algorithm 2, the update of \mathcal{L} mainly requires one fft operation, one ifft operation, and $\lceil \frac{n_3+1}{2} \rceil$ SVD operations, where the fft and ifft operations act on $n_1 \times n_2 \times n_3$ third-order tensors along the third dimension and the SVD operation acts on $n_1 \times n_2$ matrices. For ease of representation, we set $n_{max} = \max(n_1, n_2)$ and $n_{min} = \min(n_1, n_2)$. The cost of solving \mathcal{L} in Algorithm 2 is $\mathcal{O}(n_1 n_2 n_3 \log n_3 + n_{max} n_{min}^2 n_3)$. In conclusion, the computational complexity of each iteration in Algorithm 3 is $\mathcal{O}(n_1 n_2 n_3 + n_1 n_2 n_3 \log n_3 + n_{max} n_{min}^2 n_3)$.

4.3. Discussion

In this subsection, we discuss the relationships of p -TRPCA with several popular PCA variants, such as RPCA [2], PSSV [9], WRPCA [10], WSN-RPCA [6], TRPCA [20], PSTNN [26], WTRPCA [25], and ETRPCA [8]. Fig. 1 depicts these relationships. When certain parameters are fixed, p -TRPCA can degenerate into one of the aforementioned PCA methods. For convenient discussion, the parameter p ($0 < p \leq 1$) in the weighted l_p -norm that imposes a sparse constraint on singular values is denoted as p_1 . In the same way, the parameter p ($0 < p \leq 1$) in the l_p -norm, which serves as a constraint for the sparse component, is represented as p_2 .

p -TRPCA can be considered as an extension of ETRPCA. As discussed in Section 3, the weighted tensor Schatten p -norm is essentially the weighted l_p -norm of singular values. Therefore, p -TRPCA and ETRPCA are equivalent in terms of constraining the low-rank component. It is obvious that when $p_2 = 1$, the term $\|\mathcal{S}\|_p^p$ in the objective function (12) becomes $\|\mathcal{S}\|_1$. The generalized soft-thresholding operator $T_p^{GST}(\cdot)$ defined by Eq. (19) correspondingly reduces to the commonly used soft-thresholding operator for solving the l_1 -norm minimization. Specifically, by Theorem 2, if $p_2 = 1$, $\tau_1^{GST}(\omega)$ and S_1^{GST} are equal to ω and $y - \omega$, respectively. Thus, the generalized soft-thresholding operator is

$$T_1^{GST}(y, \omega) = \begin{cases} \text{sign}(y)(|y| - \omega), & |y| \in (\omega, +\infty) \\ 0, & |y| \in [0, \omega] \end{cases}, \quad (39)$$

which is changed to the soft-thresholding operator. As a consequence, Eq. (29) for updating \mathcal{S} is reformulated as

$$\mathcal{S}_{k+1} = T_1^{GST}(\mathcal{N}_k, \frac{\lambda}{\mu_k}) = T_{\frac{\lambda}{\mu_k}}(\mathcal{N}_k), \quad (40)$$

where $T_{\lambda/\mu_k}(\cdot)$ denotes the soft-thresholding operator. Hence, when $p_2 = 1$, p -TRPCA is viewed as ETRPCA.

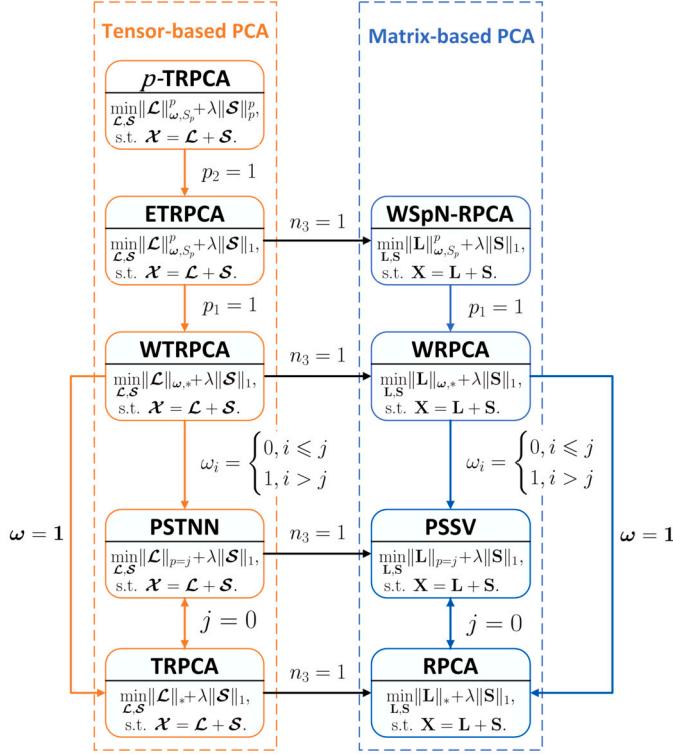


Fig. 1. The relationships of p -TRPCA with some representative works.

In addition, when $p_1 = 1$, ETRPCA reduces to WTRPCA. According to the above discussion, if $p_1 = 1$, the operator $\mathbf{P}_{\omega/\mu_k}(\cdot)$ used in Algorithm 2 can be rewritten as

$$\mathbf{P}_{\omega/\mu_k}(\bar{\mathbf{D}}) = \mathbf{T}_{\frac{\omega}{\mu_k}}(\bar{\mathbf{D}}), \quad (41)$$

where $\mathbf{T}_{\omega/\mu_k}(\cdot)$ is the singular value thresholding operation for minimizing the weighted tensor nuclear norm (WTNN) [25]. Correspondingly, the low-rank constraint is changed from WTSN to WTNN as

$$\|\mathcal{L}\|_{\omega,*} = \frac{1}{n_3} \sum_{m=1}^{n_3} \|\bar{\mathbf{L}}^{(m)}\|_{\omega,*} = \frac{1}{n_3} \sum_{m=1}^{n_3} \sum_{i=1}^{n_{min}} \omega_i \cdot \sigma_i(\bar{\mathbf{L}}^{(m)}), \quad (42)$$

where $n_{min} = \min(n_1, n_2)$. Hence, if $p_1 = 1$, ETRPCA degenerates into WTRPCA.

When the weight vector ω satisfies certain conditions, both TRPCA and PSTNN are special cases of WTRPCA. For example, WTRPCA reduces to PSTNN when ω satisfies the following condition

$$\omega_i = \begin{cases} 1, & i \leq j \\ 0, & i > j \end{cases}, \quad (43)$$

where j indicates the position of truncated singular values. Therefore, PSTNN is equivalent to TRPCA when the truncated position $j = 0$. This means WTRPCA degenerates into TRPCA when $\omega = 1$.

When $n_3 = 1$, third-order tensors \mathcal{L} and \mathcal{S} are matrices, i.e., \mathbf{L} and \mathbf{S} . Correspondingly, the tensor nuclear norm also becomes the matrix nuclear norm [20]. Hence, TRPCA can be viewed as a generalization of RPCA for tensors. Naturally, PSSV, WRPCA, and WSN-RPCA (matrix-based methods) are considered as matrix cases of PTSNN, WTRPCA, and ETRPCA (tensor-based methods), respectively. In summary, p -TRPCA is a generalized TRPCA, and some existing methods are its special cases.

5. Experiments

To evaluate the performance of various methods, p -TRPCA is compared to RPCA [2] and tensor-based methods including TRPCA [20],

Table 1

Singular values grouping for setting weight vector ω .

Category of Experiments	Group Settings
Data recovery	[1:5], [6:10], [11: n_{min}]
Phase transition	[1:50], [51:75], [76:100]
Image recovery	[1:10], [11:80], [81:321]
Background modeling	[1], [2], [3]

PSTNN [26], ETRPCA [8], nonconvex TRPCA (NTRPCA) [35], tensor correlated total variation (t-CTV) [36], and enhanced tensor low-rank representation (ETLRR) [37] on three tasks, i.e., data simulation, image recovery, and background modeling. Data simulation experiments aim to assess relative errors and the accuracy of estimating tubal rank. In order to explore the practical applications of p -TRPCA, we conduct extensive experiments on image recovery and background modeling. Note that the augmented Lagrange multipliers (ALM) algorithm [38] is used to implement RPCA. For the competing methods, we adopt the default values suggested by the authors. The parameter ω used in both ETRPCA and p -TRPCA is a weight vector with size n_{min} , whose elements are non-decreasing. As pointed out in [8], we divide it into three parts and assign a uniform weight value to each part. Table 1 shows the grouping of singular values corresponding to ω in different experiments. We perform all experiments on a computer equipped with an Intel Core i3-9100 3.60GHz CPU and 16 GB RAM. The demo code of our p -TRPCA is available at <http://github.com/qguo2010/p-TRPCA>.

5.1. Data simulation

To better quantify the performance of different methods, we perform two types of numerical experiments on the synthesized tensors. First, all competing methods are evaluated for different degrees of sparse errors. Then, we further investigate the performance of several methods for tensors with varying tubal rank and errors of different sparsity.

5.1.1. Data recovery from different degrees of sparse errors

We verify the performance of various methods in recovering the low-rank and sparse components from randomly generated tensors. Given two tensors $\mathcal{A} \in \mathbb{R}^{n \times r \times n}$ and $\mathcal{B} \in \mathbb{R}^{r \times n \times n}$ with elements obeying *i.i.d.* $N(0, 1/n)$ distribution, where n takes three different values: 100, 200, 300, we can obtain a low-rank tensor $\mathcal{L}_0 = \mathcal{A} * \mathcal{B}$ with tubal rank r [20]. A sparse tensor $\mathcal{S}_0 \in \mathbb{R}^{n \times n \times n}$ ($\|\mathcal{S}_0\|_0 = m$) is generated in random, where its non-zero elements obey independent Bernoulli distribution ± 1 . After that, we discuss two settings on different sparsity of \mathcal{S}_0 : (1) $rank_r(\mathcal{L}_0) = r = 0.1n$ and $\|\mathcal{S}_0\|_0 = m = 0.1n^2$. (2) $rank_r(\mathcal{L}_0) = r = 0.1n$ and $\|\mathcal{S}_0\|_0 = m = 0.2n^2$.

Tables 2 and 3 report the recovery results for tensors of different sizes in Settings (1) and (2), respectively. Here, we only discuss the tensors with size $n = 100$ recovered by different methods. It can be observed from Table 2 that TRPCA, ETRPCA, NTRPCA, ETLRR, and p -TRPCA provide the accurate tubal rank estimation, i.e., $r = 10$. But t-CTV and RPCA falsely estimate $r = 100$. It is apparent that RPCA performs worse than t-CTV in the other three metrics, except for r . Tensor-based methods have better recovery performance than RPCA as they utilize structural information of third-order tensors. In contrast, RPCA can only process matrices reshaped from third-order tensors, ignoring important structural information. Additionally, we observe that the estimated sparsity ($m = 102035$) of TRPCA for \mathcal{S}_0 is better than that of PSTNN ($m = 92939$) compared to the accurate value ($m = 100000$). Obviously, the assumption of PSTNN that the first j largest singular values only contain relevant information is unreasonable. ETRPCA outperforms TRPCA in estimating the sparsity m of \mathcal{S}_0 , thanks to the use of the prior knowledge on singular values [8]. Compared with other methods, p -TRPCA obtains accurate sparsity ($m = 100000$) and tubal

Table 2

Comparisons of random tensors with varying sizes recovered by different methods
(Setting (1): $\text{rank}_t(\mathcal{L}_0) = r = 0.1n$ and $\|\mathcal{S}_0\|_0 = m = 0.1n^3$).

n	r	m	Method	$\text{rank}_t(\hat{\mathcal{L}})$	$\ \hat{\mathcal{S}}\ _0$	$\frac{\ \hat{\mathcal{L}} - \mathcal{L}_0\ _F}{\ \mathcal{L}_0\ _F}$	$\frac{\ \hat{\mathcal{S}} - \mathcal{S}_0\ _F}{\ \mathcal{S}_0\ _F}$
100	10	1e5	RPCA	100	548400	6.978e-01	6.964e-03
			TRPCA	10	<u>102035</u>	2.532e-07	1.034e-09
			PSTNN	<u>20</u>	92939	3.203e+01	3.213e-01
			ETRPCA	10	<u>100000</u>	<u>8.611e-08</u>	<u>5.455e-10</u>
			NTRPCA	10	<u>100000</u>	2.214e-07	1.475e-09
			t-CTV	100	118872	7.230e-07	4.819e-09
			ETLRR	10	<u>100000</u>	5.120e-01	1.700e-03
			p -TRPCA	10	<u>100000</u>	<u>6.352e-08</u>	<u>7.759e-11</u>
			RPCA	200	4299523	7.022e-01	3.512e-03
200	20	8e5	TRPCA	20	<u>815839</u>	5.657e-07	1.065e-09
			PSTNN	<u>35</u>	739341	5.505e+01	2.755e-01
			ETRPCA	20	<u>800000</u>	<u>1.195e-07</u>	<u>9.847e-11</u>
			NTRPCA	20	<u>800000</u>	3.301e-07	1.105e-09
			t-CTV	200	952782	1.363e-06	4.569e-09
			ETLRR	20	<u>800000</u>	5.149e-01	8.497e-04
			p -TRPCA	20	<u>800000</u>	<u>3.592e-08</u>	<u>7.499e-11</u>
			RPCA	300	14247757	7.043e-01	2.345e-03
			TRPCA	30	<u>2754575</u>	5.658e-07	7.041e-10
300	30	27e5	PSTNN	<u>45</u>	2579386	6.700e+01	2.232e-01
			ETRPCA	30	<u>2700000</u>	<u>1.830e-07</u>	<u>8.168e-11</u>
			NTRPCA	30	<u>2700000</u>	5.275e-07	1.176e-09
			t-CTV	300	3029895	2.171e-06	4.761e-09
			ETLRR	30	<u>2700000</u>	5.136e-01	5.660e-04
			p -TRPCA	30	<u>2700000</u>	<u>6.580e-08</u>	<u>4.497e-11</u>

* The best result is highlighted in bold, and the second-best result is underlined.

Table 3

Comparisons of random tensors with varying sizes recovered by different methods
(Setting (2): $\text{rank}_t(\mathcal{L}_0) = r = 0.1n$ and $\|\mathcal{S}_0\|_0 = m = 0.2n^3$).

n	r	m	Method	$\text{rank}_t(\hat{\mathcal{L}})$	$\ \hat{\mathcal{S}}\ _0$	$\frac{\ \hat{\mathcal{L}} - \mathcal{L}_0\ _F}{\ \mathcal{L}_0\ _F}$	$\frac{\ \hat{\mathcal{S}} - \mathcal{S}_0\ _F}{\ \mathcal{S}_0\ _F}$
100	10	1e5	RPCA	100	549260	7.990e-01	5.660e-03
			TRPCA	10	<u>200008</u>	5.390e-07	2.925e-09
			PSTNN	<u>20</u>	179553	4.530e+01	3.197e-01
			ETRPCA	10	<u>200000</u>	<u>2.632e-07</u>	<u>1.548e-09</u>
			NTRPCA	10	<u>200000</u>	2.870e-07	1.555e-09
			t-CTV	100	259662	6.310e-01	4.468e-03
			ETLRR	10	<u>200000</u>	5.438e-01	1.787e-03
			p -TRPCA	10	<u>200000</u>	<u>1.378e-07</u>	<u>6.943e-10</u>
			RPCA	200	4293108	8.018e-01	2.824e-03
200	20	8e5	TRPCA	20	<u>1600241</u>	6.314e-07	1.697e-09
			PSTNN	<u>37</u>	1465795	8.248e+01	2.914e-01
			ETRPCA	20	<u>1600000</u>	<u>3.137e-07</u>	<u>7.854e-10</u>
			NTRPCA	20	<u>1600000</u>	9.168e-07	2.487e-09
			t-CTV	200	2071766	6.345e-01	2.247e-03
			ETLRR	20	<u>1600000</u>	5.430e-01	8.964e-04
			p -TRPCA	20	<u>1600000</u>	<u>1.901e-07</u>	<u>4.908e-10</u>
			RPCA	300	14188371	8.027e-01	1.887e-03
			TRPCA	30	<u>5401754</u>	7.790e-07	1.380e-09
300	30	27e5	PSTNN	<u>45</u>	5138579	9.494e+01	2.239e-01
			ETRPCA	30	<u>5400000</u>	<u>2.663e-07</u>	<u>4.233e-10</u>
			NTRPCA	30	<u>5400000</u>	9.379e-07	1.694e-09
			t-CTV	300	6972398	6.323e-01	1.491e-03
			ETLRR	30	<u>5400000</u>	5.426e-01	5.967e-04
			p -TRPCA	30	<u>5400000</u>	<u>1.444e-07</u>	<u>2.421e-10</u>

* The best result is highlighted in bold, and the second-best result is underlined.

rank ($r = 10$) with the smallest relative errors (i.e., $\|\hat{\mathcal{L}} - \mathcal{L}_0\|_F/\|\mathcal{L}_0\|_F$ and $\|\hat{\mathcal{S}} - \mathcal{S}_0\|_F/\|\mathcal{S}_0\|_F$). The reason for this superior performance is that p -TRPCA employs dual l_p -norm constraints on the low-rank component \mathcal{L}_0 and the sparse one \mathcal{S}_0 , which promotes the low-rankness and sparsity of the solutions. Table 3 presents similar observations. Moreover, as the degrees of errors increase, the recovery performance of all methods faces greater challenges. Accordingly, the relative errors of the recovery results listed in Table 3 are worse than the ones in Table 2.

5.1.2. Phase transition in tubal rank and sparsity

We further investigate the recovery performance of several competing methods when the tubal rank r of \mathcal{L}_0 and the sparsity m of \mathcal{S}_0 vary within a certain range. In this experiment, the size of the low-rank tensor \mathcal{L}_0 is changed to $n \times n \times n_3$ ($n = 100$ and $n_3 = 50$). Correspondingly, we need to generate two factor tensors $\mathcal{A} \in \mathbb{R}^{n \times r \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{r \times n \times n_3}$ to obtain $\mathcal{L}_0 \in \mathbb{R}^{n \times n \times n_3}$. For the sparse tensor $\mathcal{S}_0 \in \mathbb{R}^{n \times n \times n_3}$, its elements obey the Bernoulli distribution and are defined in the following two cases:

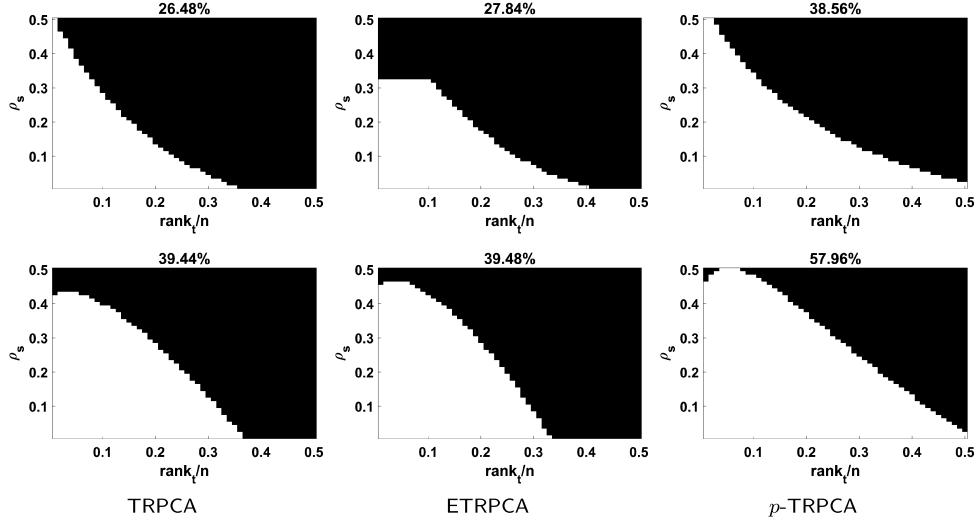


Fig. 2. Results of the recovered low-rank tensor $\hat{\mathcal{L}}$ under varying tubal ranks of \mathcal{L}_0 and sparsity of \mathcal{S}_0 . The first row shows \mathcal{S}_0 defined in *Case 1*, and the second row exhibits \mathcal{S}_0 defined in *Case 2*.

Case 1:

$$[\mathcal{S}_0]_{ijk} = \begin{cases} 1, & \text{w.p. } \rho_s/2 \\ 0, & \text{w.p. } 1 - \rho_s \\ -1, & \text{w.p. } \rho_s/2 \end{cases} \quad (44)$$

Case 2:

$$[\mathcal{S}_0]_{ijk} = \begin{cases} \text{sign}([\mathcal{L}_0]_{ijk}), & \text{w.p. } \rho_s \\ 0, & \text{w.p. } 1 - \rho_s \end{cases} \quad (45)$$

In the above equations, w.p. is the abbreviation for with probability in mathematics, which denotes the probability of the occurrence of an event, and ρ_s denotes the sparsity of \mathcal{S}_0 . We set the variation range of r/n and ρ_s to be between 0.01 and 0.5 with an increment of 0.01. For each pair of data generated by $(r/n, \rho_s)$, we consider the tensor $\hat{\mathcal{L}}$ to be successfully recovered if the relative error of $\hat{\mathcal{L}}$ is less than $1e-3$ [20].

For visual comparisons, Fig. 2 displays the results of the recovered tensor $\hat{\mathcal{L}}$ by TRPCA, ETRPCA, and p -TRPCA with varying tubal ranks r and sparsity $m = \rho_s \cdot n^2 \cdot n_3$, in which the white and black indicate recovery success and recovery failure, respectively. It can be clearly observed that each sub-figure has a large white area in the bottom left corner. This phenomenon means that when r and ρ_s are relatively low, TRPCA, ETRPCA, and p -TRPCA can obtain successful recovery results. In Fig. 2, three sub-figures in the second row have larger white areas than the ones in the first row. The reason is that these methods have more robust recovery performance for dealing with \mathcal{S}_0 as defined in *Case 2*. We can also see that p -TRPCA obtains the largest white area of 38.56% in *Case 1* and 57.96% in *Case 2*. Compared to the suboptimal ETRPCA, p -TRPCA achieves a significant improvement of 10.72% in *Case 1* and 18.48% in *Case 2*. It is attributed to the fact that l_p -norm as a constraint on the sparse component can effectively model the sparsity of \mathcal{S}_0 and produce a positive impact on modeling the low-rankness of $\hat{\mathcal{L}}$. As a result, $\hat{\mathcal{L}}$ can be recovered more effectively by p -TRPCA.

5.2. Image recovery with random noises

In this subsection, we assess the performance of all methods for recovering color images with random noises. Here, we choose fifty color images randomly from the Berkeley Segmentation dataset (BSD) [39] to test the performance of different methods. The size of color images is $481 \times 321 \times 3$ or $321 \times 481 \times 3$. We randomly corrupt θ percent of pixels (corrupted level $\theta = 10, 20$) in each color image by setting them to a random value between 0 and 255. Note that the corrupted

Table 4

Comparisons of the average values of different metrics for image recovery.

θ	Method	PSNR	FSIM	SSIM	ERGAS
10	RPCA	27.875	0.876	0.859	128.650
	TRPCA	31.484	0.946	0.946	84.816
	PSTNN	33.468	0.960	0.954	70.775
	ETRPCA	37.052	0.981	0.979	45.219
	NTRPCA	36.679	0.976	0.967	48.642
	t-CTV	36.362	0.980	0.979	50.269
20	ETLRR	33.241	0.951	0.945	70.248
	<i>p</i> -TRPCA	40.999	0.994	0.993	29.417
	RPCA	26.909	0.850	0.818	147.545
	TRPCA	29.601	0.916	0.905	106.037
	PSTNN	30.575	0.927	0.911	96.938
	ETRPCA	31.579	0.935	0.911	85.013
	NTRPCA	32.545	0.943	0.914	79.315
	t-CTV	34.221	0.966	0.959	64.588
	ETLRR	31.105	0.927	0.910	89.396
	<i>p</i> -TRPCA	33.860	0.964	0.955	66.253

* The best result is highlighted in bold, and the second-best result is underlined.

pixel positions are unknown. All competing methods are used to recover these corrupted images. All experimental results are evaluated on four quantitative metrics i.e., PSNR, FSIM [40], SSIM [41], and ERGAS [42]. ERGAS tends to be better recovery quality with lower values. Higher values for the remaining three metrics indicate better recovery performance. Besides, in p -TRPCA, we empirically set $\omega = [1, 1.1, 1.5]$, $p_1 = 0.8$, and $p_2 = 0.6$ for color images with $\theta = 10$.

Table 4 presents the average values of four metrics for image recovery. It is evident that RPCA is inferior to tensor-based methods, which again highlights the significance of utilizing structural information among three channels of color images. In addition, we also observed that PSTNN is superior to TRPCA, which means that TRPCA shrinks the first j singular values equally, damaging the significant information of color images. Meanwhile, both ETRPCA and p -TRPCA are stronger than NTRPCA, t-CTV, and ETLRR for recovering color images with $\theta = 10$. ETRPCA achieves second-best quantitative results as it not only preserves the significant information but also handles the random noise contained in the first j singular values. A comparison between ETRPCA and DTRPCA¹ was conducted by Wang et al. in [21]. As ob-

¹ At present, the source code of DTRPCA is not publicly available.



Fig. 3. Visual comparisons of recovery performance on three example images with $\theta = 10$. From top to bottom: Original images; Corrupted images; The images obtained by RPCA, TRPCA, PSTNN, ETRPCA, NTRPCA, t-CTV, ETLRR, and p -TRPCA, respectively. Odd columns present the recovered images. Even columns show the residual images between the original images and the recovered ones. For better visual comparison, a zoomed region is also provided for each image.

served in Table V of [21], DTRPCA improves by 7.29% in PSNR and 1.50% in SSIM compared to ETRPCA for the Berkeley Segmentation images with $\theta = 10$. In Table 4, p -TRPCA improves by 10.65% and 1.43% over ETRPCA in terms of PSNR and SSIM for the images in the same dataset. Therefore, using ETRPCA as a bridge, we can conclude that p -TRPCA is competitive to DTRPCA.

The visual comparisons of recovered results for color images with $\theta = 10$ are shown in Fig. 3. From the zoomed parts, we can clearly see that the images recovered by p -TRPCA preserve more details than other seven competing methods. Additionally, to better compare the edge information of the recovered images, the even columns in Fig. 3 also show the residual images between the original images and recovered images, in which p -TRPCA achieves the best visual quality with slightly visible edge residuals. As mentioned in [43], the signal leakage, as an effective metric, can measure the differences between the noisy images and their recovered images. Therefore, the residual images can also be used to evaluate the extent of signal leakage during the recovery process. In Fig. 3, the residual images of p -TRPCA show the fewest residuals compared to other methods. This means that our method has less signal leakage than other competitors. Fig. 4 compares the running times of these competing methods on test images. It visually indicates that t-CTV

requires the highest computational cost, followed by ETRPCA. p -TRPCA is more efficient than ETRPCA, thanks to Algorithm 2 which effectively reduces the number of SVD calculations.

By increasing θ from 10 to 20, we further explore the recovery performance of p -TRPCA. For color images with $\theta = 20$, we set $\omega = [1, 1.1, 1.5]$, $p_1 = 0.8$, and $p_2 = 0.85$. The quantitative results of various competing methods for the recovery of color images with $\theta = 20$ are listed in Table 4. Unlike $\theta = 10$, t-CTV performs slightly better than p -TRPCA when $\theta = 20$. This can be attributed to the regularization term combining low-rank and smooth prior, which exhibits strong robustness in condition of relatively high θ . p -TRPCA, obtaining the second-best quantitative metrics, promotes 4.04% over the third-best NTRPCA in terms of PSNR. In addition, Fig. 5 provides visual comparisons of recovery results for color images with $\theta = 20$. We observe that the images recovered by p -TRPCA and t-CTV visually outperform others due to effectively preserving some details and edges. There is almost no visual difference between the images recovered by p -TRPCA and t-CTV.

p -TRPCA has four critical parameters i.e., λ , ω , p_1 , and p_2 . We further examine the influence of parameters on the recovery performance of p -TRPCA. Similar to [20], the balance parameter λ is set to $\sqrt{\max(n_1, n_2) \cdot n_3}$. We set the weight vector ω consistent with the work

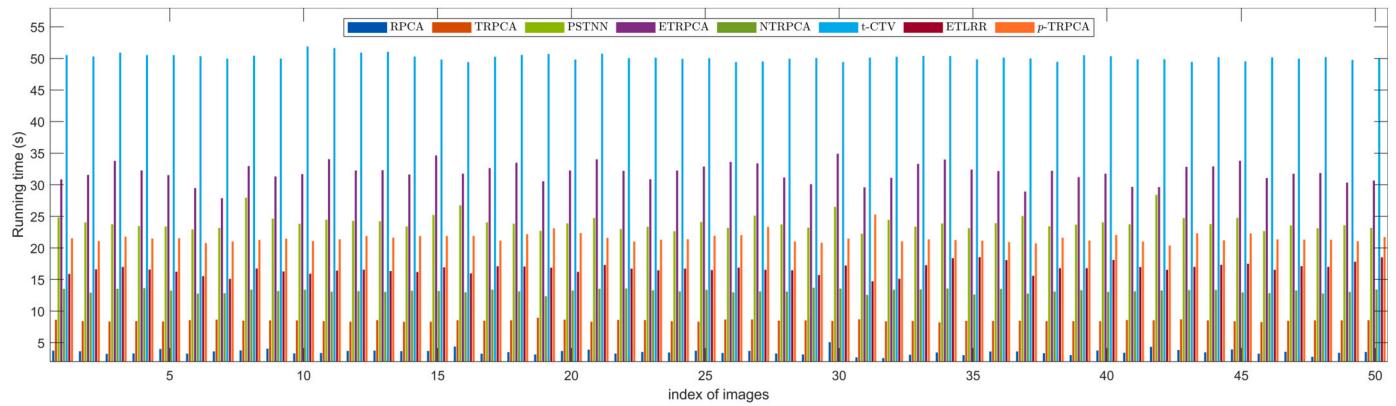


Fig. 4. Running times of RPCA, TRPCA, PSTNN, ETRPCA, NTRPCA, t-CTV, ETLRR, and *p*-TRPCA on fifty images with $\theta = 10$.



Fig. 5. Visual comparisons of recovery performance on three example images with $\theta = 20$. From top to bottom: Original images; Corrupted images; The images obtained by RPCA, TRPCA, PSTNN, ETRPCA, NTRPCA, t-CTV, ETLRR, and *p*-TRPCA, respectively. Odd columns present the recovered images. Even columns show the residual images between the original images and the recovered ones. For better visual comparison, a zoomed region is also provided for each image.

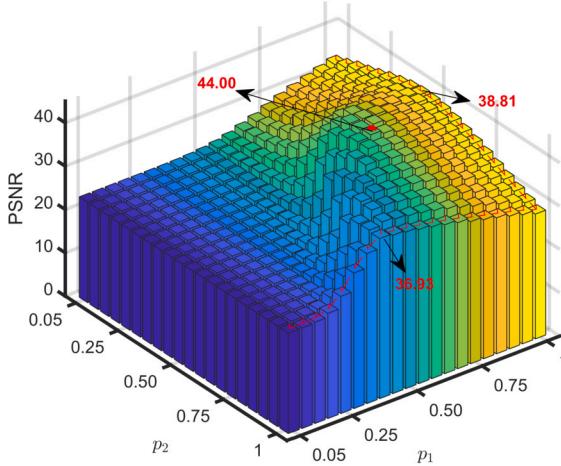


Fig. 6. Influence of varying p_1 and p_2 on PSNR.

[8]. For the subsequent experiments, we vary p_1 and p_2 from 0.05 to 1, employing a step size of 0.05. To evaluate the impact of these two parameters, we apply p -TRPCA with each (p_1, p_2) -pair as parameters to recover a test image with $\theta = 10$. Fig. 6 presents the recovered results (PSNR) on a test image for each pair (p_1, p_2) . As can be seen from it, when $p_2 = 1$, PSNR initially increases with the growth of p_1 and reaches its highest value of 36.93 when $p_1 = 0.4$. However, PSNR starts to drop as p_1 continues to increase beyond this value. We can also observe that if $p_1 = 1$, PSNR for different p_2 also has a similar changing trend. When $p_1 = 1$, p -TRPCA shows acceptable recovery performance with different p_2 . The highest PSNR value of 38.81 is achieved at $(1, 0.45)$. Fig. 6 also shows that PSNR takes the highest value 44.00 at $(0.65, 0.65)$. However, it is difficult to determine the optimal p_1 and p_2 values in practice. Fortunately, the recovery performance of p -TRPCA is not very sensitive to these two parameters p_1 and p_2 ranged in $[0.4, 1]$.

5.3. Background modeling for color image sequences

We further employ p -TRPCA to model the background of color image sequences, aiming to obtain background images (i.e., low-rank components) and foreground images (i.e., sparse components). The experiments of background modeling are conducted on three color image sequences from Scene Background Initialization database [44], including *CAVIAR1* ($96 \times 64 \times 3$, 610), *HighwayI* ($80 \times 60 \times 3$, 440), and *HumanBody2* ($80 \times 60 \times 3$, 740), where the numbers refer to the size and quantity of color images. For RPCA, each color image sequence ($w \times h \times 3, m$) is reshaped to a $(3wh) \times m$ matrix. For tensor-based methods, each color image sequence is reshaped to a $(wh) \times 3 \times m$ tensor. To assess the effect of background modeling, we adopt five quantitative metrics, including PSNR, AGE, pEPs, pCEPs [45], and CQM [46]. As the image sequence is composed of the background and foreground, the quality of background modeling is positively correlated with the accuracy of the foreground. The F1-score [47] is used to quantitatively evaluate the accuracy of the foregrounds, indirectly assessing the effectiveness of background modeling. Higher values of PSNR, CQM, and F1-score imply better background modeling, while the other three metrics are the opposite.

The odd rows in Fig. 7 display the modeling results in three color image sequences. We can observed that the background images extracted by p -TRPCA exhibit the most favorable visual outcomes compared to other methods. The background images extracted by p -TRPCA obtain the best visual effect among all methods. For both *CAVIAR1* and *HumanBody2* sequences, other methods exhibit more prominent ghost shadows in the background images than p -TRPCA. In particular, for *HighwayI* sequence, only the modeling result of p -TRPCA has no visible ghost shadow. The foreground images are shown in the even rows of Fig. 7. In

Table 5

Comparisons of the values of different metrics for background modeling.

Dataset	Method	PSNR	AGE	pEPs	pCEPs	CQM
<i>CAVIAR1</i>	RPCA	23.831	6.148	4.867	2.588	23.461
	TRPCA	28.955	3.900	3.695	1.746	28.477
	PSTNN	28.955	3.900	3.695	1.746	28.477
	ETRPCA	28.989	3.852	3.678	1.710	28.536
	<i>p</i> -TRPCA	30.501	<u>3.315</u>	<u>2.734</u>	<u>1.139</u>	29.972
<i>HighwayI</i>	RPCA	26.242	6.210	8.229	4.917	26.424
	TRPCA	34.914	2.974	0.083	0.000	34.851
	PSTNN	34.914	2.974	0.083	0.000	34.851
	ETRPCA	34.980	<u>2.954</u>	<u>0.083</u>	0.000	<u>34.921</u>
	<i>p</i> -TRPCA	39.446	<u>1.814</u>	<u>0.021</u>	0.000	<u>39.335</u>
<i>HumanBody2</i>	RPCA	25.087	5.547	7.292	3.938	24.873
	TRPCA	27.228	5.034	7.063	2.729	27.003
	PSTNN	27.228	5.034	7.063	2.729	27.003
	ETRPCA	27.373	<u>5.009</u>	<u>6.875</u>	<u>2.458</u>	27.170
	<i>p</i> -TRPCA	32.245	<u>3.387</u>	<u>2.708</u>	<u>0.604</u>	31.179
Average	RPCA	25.053	5.969	6.796	3.814	24.920
	TRPCA	30.366	3.969	3.614	1.490	30.110
	PSTNN	30.366	3.969	3.614	1.490	30.110
	ETRPCA	30.447	<u>3.938</u>	<u>3.546</u>	<u>1.389</u>	<u>30.209</u>
	<i>p</i> -TRPCA	34.064	<u>2.839</u>	<u>1.821</u>	<u>0.581</u>	33.675

* The best result is highlighted in bold, and the second-best result is underlined.

Table 6

Quantitative evaluation (F1-score) of the foregrounds.

Dataset	RPCA	TRPCA	PSTNN	ETRPCA	<i>p</i> -TRPCA
<i>CAVIAR1</i>	0.2392	0.5281	0.5281	<u>0.5292</u>	0.6889
<i>HighwayI</i>	0.3668	0.9949	0.9949	<u>0.9949</u>	0.9990
<i>HumanBody2</i>	0.1741	0.4172	0.4172	<u>0.4384</u>	0.8095
Average	0.2600	0.6467	0.6467	<u>0.6542</u>	0.8324

* The best result is highlighted in bold, and the second-best result is underlined.

the sixth row, it is observed that the foreground separated by p -TRPCA is very close to the ground truth.

Quantitatively, we calculate five metrics for background modeling, as listed in Table 5. It is seen that tensor-based methods consistently outperform RPCA on all evaluation metrics. In addition, an interesting observation is that the quantitative results of TRPCA are identical to those of PSTNN. This is because that PSTNN uniformly shrinks all singular values by setting the truncation position $j = 0$, making it equivalent to TRPCA. We also find that ETRPCA obtains slightly superior quantization values compared to TRPCA, which demonstrates the effectiveness of the prior knowledge of singular values. p -TRPCA outputs the best results for background modeling. Table 6 reports the F1-score for the foregrounds of different sequences. Among all the methods, our method has the highest F1-score on all three sequences, demonstrating that the foregrounds obtained by p -TRPCA are more accurate than other methods. This is due to the dual l_p -norm providing tight constraints for the low-rank and sparse components, which makes our method easier to classify pixels belonging to the low-rank component as the background and leads to better performance of background modeling.

6. Conclusion

In this paper, we propose the p -TRPCA for tensor robust principal component analysis. The proposed model utilizes the weighted l_p -norm to enforce a sparse constraint on the singular values of the low-rank component, while also employing the l_p -norm to impose a sparse constraint on the sparse component. The weighted l_p -norm of singular values, i.e., the weighted tensor Schatten p -norm, allocates weight values to singular values according to their importance and effectively models the low-rankness. The l_p -norm for a sparse constraint can better model the sparsity and further promote the low-rankness of our model.

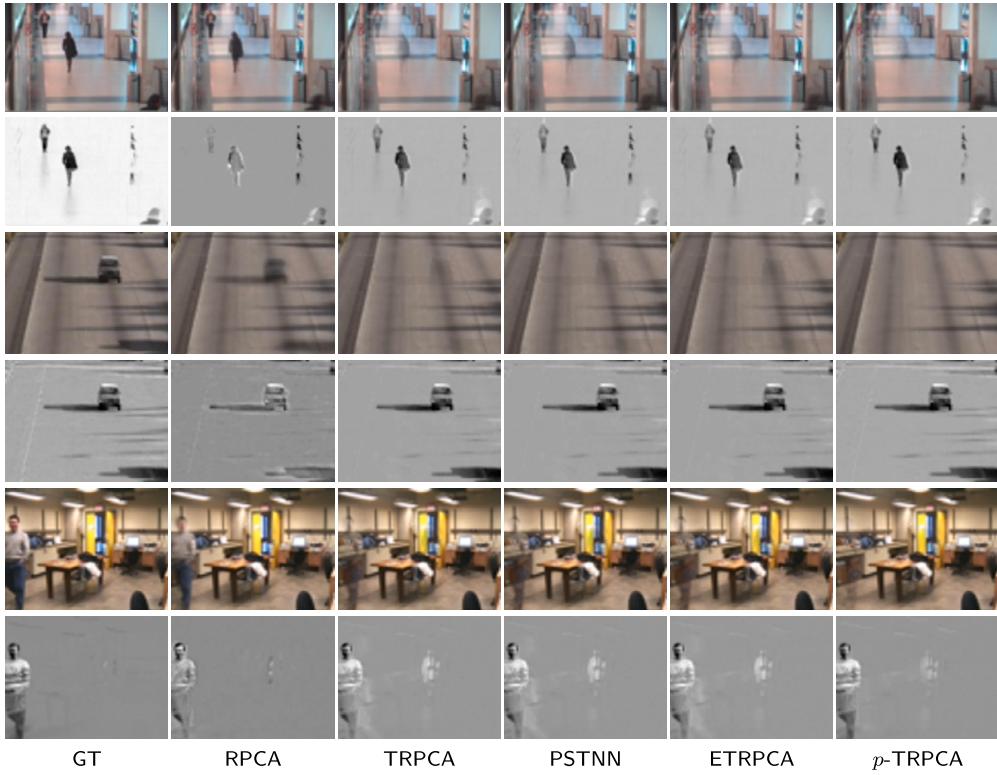


Fig. 7. Background modeling from three color image sequences. From left to right: Ground truth; images separated by RPCA, TRPCA, PSTNN, ETRPCA, and p -TRPCA, respectively. Odd rows correspond to the background images, and even rows show the foreground ones.

Hence, the solutions obtained by p -TRPCA based on dual l_p -norm sparse constraints are closer to the authentic ones. An efficient optimization algorithm based on ADMM is presented to solve the proposed model, and its convergence is analyzed. The relationships between our method and several popular methods are further discussed, indicating that our method can be considered as a generalization of these methods. The performance of the proposed p -TPRCA is validated by extensive experimental results on the tasks of data simulation, image recovery, and background modeling.

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

We would like to thank the authors of [20], [26], [8], [35], [36], [37], and [38] for sharing their codes. This work was supported by National Natural Science Foundation of China (61873145).

References

- [1] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, *SIAM Rev.* 51 (3) (2009) 455–500.
- [2] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis?, *J. ACM* 58 (3) (2011) 1–37.
- [3] N. Vaswani, T. Bouwmans, S. Javed, P. Narayanamurthy, Robust subspace learning: robust PCA, robust subspace tracking, and robust subspace recovery, *IEEE Signal Process. Mag.* 35 (4) (2018) 32–55.
- [4] O.M. Sleem, M.E. Ashour, N.S. Aybat, C.M. Lagoa, L_p quasi-norm minimization: algorithm and applications, *EURASIP J. Adv. Signal Process.* 2024 (1) (2024) 22.
- [5] W. Zuo, D. Meng, L. Zhang, X. Feng, D. Zhang, A generalized iterated shrinkage algorithm for non-convex sparse coding, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 217–224.
- [6] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, L. Zhang, Weighted Schatten p -norm minimization for image denoising and background subtraction, *IEEE Trans. Image Process.* 25 (10) (2016) 4842–4857.
- [7] C. O'Brien, M.D. Plumley, Inexact proximal operators for ℓ_p -quasinorm minimization, in: *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 4724–4728.
- [8] Q. Gao, P. Zhang, W. Xia, D. Xie, X. Gao, D. Tao, Enhanced tensor RPCA and its application, *IEEE Trans. Pattern Anal. Mach. Intell.* 43 (6) (2021) 2133–2140.
- [9] T.-H. Oh, Y.-W. Tai, J.-C. Bazin, H. Kim, I.S. Kweon, Partial sum minimization of singular values in robust PCA: algorithm and applications, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (4) (2016) 744–758.
- [10] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, L. Zhang, Weighted nuclear norm minimization and its applications to low level vision, *Int. J. Comput. Vis.* 121 (2) (2017) 183–208.
- [11] W. Xia, X. Zhang, Q. Gao, X. Shu, J. Han, X. Gao, Multiview subspace clustering by an enhanced tensor nuclear norm, *IEEE Trans. Cybern.* 52 (9) (2022) 8962–8975.
- [12] J. Wang, M. Wang, X. Hu, S. Yan, Visual data denoising with a unified Schatten- p norm and l_q norm regularized principal component pursuit, *Pattern Recognit.* 48 (10) (2015) 3135–3144.
- [13] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (1) (2013) 171–184.
- [14] T. Zhang, Z. Tang, Q. Liu, Robust subspace clustering via joint weighted Schatten- p norm and l_q norm minimization, *J. Electron. Imaging* 26 (3) (2017) 033021.
- [15] Q. Guo, S. Gao, X. Zhang, Y. Yin, C. Zhang, Patch-based image inpainting via two-stage low rank approximation, *IEEE Trans. Vis. Comput. Graph.* 24 (6) (2018) 2023–2036.
- [16] Q. Guo, Y. Zhang, S. Qiu, C. Zhang, Accelerating patch-based low-rank image restoration using kd-tree and Lanczos approximation, *Inf. Sci.* 556 (2021) 177–193.
- [17] Q. Guo, C. Zhang, Y. Zhang, H. Liu, An efficient SVD-based method for image denoising, *IEEE Trans. Circuits Syst. Video Technol.* 26 (5) (2016) 868–880.
- [18] X. Huang, B. Du, W. Liu, Multichannel color image denoising via weighted Schatten p -norm minimization, in: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 2020, pp. 637–644.
- [19] Z. Bekhtoui, K. Abed-Meraim, A. Meche, Robust adaptive algorithms for fast principal component analysis, *Digit. Signal Process.* 127 (2022) 103561.

- [20] 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.* 42 (4) (2020) 925–938.
- [21] Y. Wang, K.I. Kou, H. Chen, Y.Y. Tang, L. Li, Double auto-weighted tensor robust principal component analysis, *IEEE Trans. Image Process.* 32 (2023) 5114–5125.
- [22] J. Mafakheri, A.R. Kahoo, R. Anvari, M. Mohammadi, M. Radad, M.S. Monfared, Expand dimensional of seismic data and random noise attenuation using low-rank estimation, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 15 (2022) 2773–2781.
- [23] X. Zhang, Y. Zhang, Z. Liu, H. Wang, Three-dimensional seismic data reconstruction via partial sum of tensor nuclear norm minimization, *J. Geophys. Eng.* 20 (2) (2023) 376–386.
- [24] R. Anvari, M. Mohammadi, A.R. Kahoo, Enhancing 3-d seismic data using the t-svd and optimal shrinkage of singular value, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 12 (1) (2019) 382–388.
- [25] Q. Gao, W. Xia, Z. Wan, D. Xie, P. Zhang, Tensor-SVD based graph learning for multi-view subspace clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 3930–3937.
- [26] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm, *J. Comput. Appl. Math.* 372 (2020) 112680.
- [27] Q. Lyu, Z. Lin, Y. She, C. Zhang, A comparison of typical l_p minimization algorithms, *Neurocomputing* 119 (2013) 413–424.
- [28] E.J. Candès, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted l_1 minimization, *J. Fourier Anal. Appl.* 14 (5) (2008) 877–905.
- [29] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, S. Yan, Tensor robust principal component analysis: exact recovery of corrupted low-rank tensors via convex optimization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5249–5257.
- [30] X. Geng, Q. Guo, C. Zhang, Color image denoising via tensor robust PCA with non-convex and nonlocal regularization, in: Proceedings of ACM Multimedia Asia, 2022.
- [31] S. Mohaoui, A. Hakim, S. Raghay, Smooth tensor robust principal component analysis with application to color image recovery, *Digit. Signal Process.* 123 (2022) 103390.
- [32] H. Zhang, H. Fan, Y. Li, X. Liu, Y. Ye, X. Zhu, Two new low rank tensor completion methods based on sum nuclear norm, *Digit. Signal Process.* 135 (2023) 103949.
- [33] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.* 3 (1) (2011) 1–122.
- [34] J. Kimber Jr, Two extended Bolzano-Weierstrass theorems, *Am. Math. Mon.* (1965) 1007–1012.
- [35] X. Geng, Q. Guo, S. Hui, M. Yang, C. Zhang, Tensor robust pca with nonconvex and nonlocal regularization, *Comput. Vis. Image Underst.* (2024) 104007.
- [36] H. Wang, J. Peng, W. Qin, J. Wang, D. Meng, Guaranteed tensor recovery fused low-rankness and smoothness, *IEEE Trans. Pattern Anal. Mach. Intell.* (2023) 1–17.
- [37] S. Du, B. Liu, G. Shan, Y. Shi, W. Wang, Enhanced tensor low-rank representation for clustering and denoising, *Knowl.-Based Syst.* 243 (2022) 108468.
- [38] Z. Lin, R. Liu, Z. Su, Linearized Alternating Direction Method with Adaptive Penalty for Low-Rank Representation, *Advances in Neural Information Processing Systems*, vol. 24, 2011.
- [39] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proceedings of Eighth IEEE International Conference on Computer Vision, vol. 2, 2001, pp. 416–423.
- [40] L. Zhang, L. Zhang, X. Mou, D. Zhang, FSIM: a feature similarity index for image quality assessment, *IEEE Trans. Image Process.* 20 (8) (2011) 2378–2386.
- [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] L. Wald, Data Fusion: Definitions and Architectures: Fusion of Images of Different Spatial Resolutions, Presses des MINES, 2002.
- [43] R. Anvari, M. Mohammadi, J. Mafakheri, A. Roshandel Kahoo, M. Soleimani Monfared, S. Rashidi, A.H. Mohammed, Denoising of multidimensional seismic data in the physical domain by a new non-local self similarity method, *Earth Sci. Inform.* 16 (1) (2023) 1041–1060.
- [44] L. Maddalena, A. Petrosino, Towards benchmarking scene background initialization, in: New Trends in Image Analysis and Processing —ICIAP 2015 Workshops, 2015, pp. 469–476.
- [45] T. Bouwmans, L. Maddalena, A. Petrosino, Scene background initialization: a taxonomy, *Pattern Recognit. Lett.* 96 (2017) 3–11.
- [46] Y. Yalman, İ. Ertürk, A new color image quality measure based on yuv transformation and PSNR for human vision system, *Turk. J. Electr. Eng. Comput. Sci.* 21 (2) (2013) 603–612.
- [47] X. Zhou, C. Yang, W. Yu, Moving object detection by detecting contiguous outliers in the low-rank representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (3) (2013) 597–610.

Tinghe Yan received the B.S. degree from Shandong University of Finance and Economics, Jinan, China, in 2021. He is currently a postgraduate with Shandong Provincial Key Laboratory of Digital Media Technology, School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, China. His research interests include image restoration, low-rank representation, and clustering.

Qiang Guo received the B.S. degree from Shandong University of Technology, Zibo, China, in 2002, the M.S. and Ph.D. degrees from Shanghai University, Shanghai, China, in 2005 and 2010, respectively. From 2012 to 2015, he was a Post-Doctoral Fellow with Shandong University, Jinan, China. He is currently a professor with Shandong Provincial Key Laboratory of Digital Media Technology, School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan, China. His research interests include image restoration, sparse representation, and object detection. He is an IEEE member.