# Fast Tensor Robust Principal Component Analysis with Estimated Multi-rank and Riemannian Optimization

Qile Zhu[1], Shiqian Wu[1,2*], Shun Fang[1], Qi Wu[3], Shoulie Xie[4], Sos Agaian[5]

[1]School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan, 430000, China.
[2]Institute of Advanced Displays and Imaging, Henan Academy of Sciences, Zhengzhou, 450000, China.
[3]School of Information Management, Jiangxi University of Finance and Economics, Nanchang, 330000, China.
[4]Institute for Infocomm Research (I$^2$R), Agency for Science, Technology and Research (A*STAR), Singapore, 138632, Singapore.
[5]College of Staten Island and the Graduate Center, The City University of New York, New York, 10314, USA.

*Corresponding author(s). E-mail(s): shiqian.wu@wust.edu.cn;
Contributing authors: zhuqile@wust.edu.cn; fangshun97@wust.edu.cn; wuqi@jxufe.edu.cn; slxie@i2r.a-star.edu.sg; sosagaian@yahoo.com;

**Abstract**

Motivated by the fact that tensor robust principal component analysis (TRPCA) and its variants donot utilize actual rank value, limiting the recovery performance, and their computational costs are always monumental for large-scale tensor recovery, a fast TRPCA is proposed to recover the low-rank and sparse tensors by estimating multi-rank vector and adopting Riemannian optimization strategy in this paper. Specifically, a fast multi-rank estimation of low-rank tensor is proposed by modifying the Gershgorin disk theorem-based matrix rank estimation. An innovative TRPCA with Estimated Multi-Rank (TRPCA-EMR) is proposed to eliminate hyperparameter tuning by imposing strict multi-rank equality constraints. Additionally, Riemannian optimization is employed to project each frontal slice of the tensor in Fourier domain onto a low multi-rank manifold in efficiently coping with tensor Singular Value Decomposition (t-SVD) and reduce computational complexity. Experimental results demonstrate TRPCA-EMR's superior efficiency and effectiveness compared to existing methods, confirming its potential for practical applications and reassuring its reliability.

**Keywords:** Tensor robust principal component analysis. Tensor singular value decomposition. Tensor multi-rank. Riemannian optimization. Background subtraction

# 1 Introduction

As an effective way to recover low-rank and sparse matrices, robust principal component analysis (RPCA) [1] has been gained wide applications, such as background subtraction [2], infrared target tracking [3], and face recognition [4]. A major shortcoming of RPCA is that it cannot well handle high-dimensional data, such as hyperspectral images and videos. Since RPCA performs on matrices, the underlying multidimensional data have to be transformed into a matrix to recover the low-rank data, which ignores information embedded in multidimensional structure. For example, it has been noted in [5] that both the spectral correlation and the spatial correlation in multispectral images are always strong. Tensor, as a high-order generalization of matrices, is a natural way to represent multidimensional arrays to retain the intrinsic structures. In this study, we focus on the third-order tensors, but the results can be extended to higher-order tensors.

Unlike matrices, tensor ranks have multiple definitions. Generally, there are two common and fundamental definitions: CP-rank and Tucker-rank. Several robust CP-/Tucker- based approaches have been introduced to recover low-rank tensors [6–8]. Chen et al. [6] described a generalized weighted low-rank tensor factorization (GWLRTF) and extended its applicability by integrating it with CP and Tucker factorization. Bahr et al. [7] proposed a robust Kronecker component analysis (RKCA), which combined RPCA with the ideas from sparse dictionary learning. The performances of these methods are limited by some drawbacks of CP and Tucker models [9]. For example, CP-rank is NP-hard [10], and the Tucker-based methods unfold a tensor into matrices, which destroys the intrinsic structure of the tensor [11].

Tensor singular value decomposition (t-SVD) was proposed in [12] to treat third-order tensors, based on this decomposition, multi-rank and tubal rank are defined to characterize the intrinsic low-rank structure well [11]. As a convex relaxation of tubal-rank, the tensor nuclear norm (TNN) based on t-SVD was presented in [13]. Inspired by the work in [13], Lu et al. [14] introduced tensor robust principal component analysis (TRPCA) to extend RPCA to the tensor case for recovering the low-rank and sparse tensors. It has been proved in [15] that TRPCA theoretically guarantee the exact recovery of low-rank tensor. In [16], Zhou et al. proposed an outlier-TPCA (OR-TPCA) by introducing $\ell_{2,1}$-norm, which achieves good outlier detection, clustering, semi-supervised and supervised learning. By defining a new TNN which extends TNN with core matrix, Liu et al. [17] developed an improved TPCA (IRT-PCA). Based on TRPCA, Zhang et al. [18] proposed a generalized non-convex method with an objective function that effectively approximates the tensor rank function and $l_0$-norm. It is highlighted that these TNN-based TRPCA methods cannot ensure the recovered low rank is equal to that of the target tensor, which results in performance degradation.

Several TRPCA methods with target multi-rank or tubal-rank have been presented to address the above issue. Jiang et al. [19] proposed a RPCA using partial sum of TNN

(PSTNN-RPCA), which does not shrink the first $r$ (the target tubal-rank value) largest singular values, and limits its capability to deal with practical problems [20]. To address this issue, an enhanced TRPCA (ETRPCA) was introduced via weighted tensor Schatten $p$-norm minimization [20], considering the salience of singular value differences of tensorial data. ETRPCA has a good performance with an appropriate set of weights of p-norm, whereas the weights are heuristically chosen. Assuming that the tensor multi-rank is known, Qiu et al. [21] studied nonconvex tensor robust PCA and proposed an alternating projection algorithm EAPT, which utilizes the tangent space of low-rank tensor to reduce computational complexity. It should be highlighted that these methods perform well when the tensor multi-rank or tubal-rank is known. This leads to a question: can we estimate the multi-rank of the target low-rank tensor?

Most of the current methods for estimating tensor rank revolve around CP- and Tucker-rank [22–24], and adopt regularization to gradually approximate the rank, rather than using the estimated rank as a priori knowledge of the optimization process. By using the estimated tubal-rank, a robust tensor principal component analysis with rank estimate (RTPCA-RE) was described to improve the performance of tensor recovery [9]. However, the tubal rank cannot comprehensively capture the difference in low- and high-frequency information in transformed tensors [21]. In addition, RTPCA-RE iteratively obtained the tubal-rank estimation by incorporating the TNN regularization into the optimization model, which results in increased computational complexity for large-scale tensors.

If the rank of a matrix is known, the computational cost of SVD can be significantly reduced by projecting the matrix onto a low-rank manifold [25]. Therefore Riemannian optimization is employed for fast matrix completion and low-rank matrix recovery [25, 26]. Recently, Riemannian optimization has been used in tensor processing. Cai et al. [27] proposed a fast Riemannian algorithm to simultaneously estimate both the low-rank and sparse tensors in a generalized framework. Luo and Zhang [28] introduced an efficient Riemannian Gauss-Newton (RGN) method to estimate the low Tucker rank tensor. Despite these advancements, to our knowledge, Riemannian optimization has not yet been applied to t-SVD in the context of TRPCA methods.

In this paper, A fast Gerschgorin disk based multi-rank estimation (GDME) of the low-rank tensor is proposed first , which is inspired by the great success of Gerschgorin disk theorem in matrix rank estimation [2, 29]. GDME is characterized by providing real multi-rank information quickly and accurately before the optimization process. Subsequently, we introduce a novel tensor RPCA with the estimated multi-rank (TRPCA-EMR), which leverages the ascertained tensor rank information to enhance the tensor recovery. Different from existing TRPCA methods, the TNN regularization is replaced with strict equality constraints based on the estimated tensor multi-rank, which avoids adjustment of the hyper-parameters in existing TRPCA models. A truncated singular value solution in low-rank tensor recovery sub-problem is obtained owing to the strict low-rank constraints. Inspired by the successful applications of Riemannian optimization on matrix completion and low-rank matrix recovery, a novel algorithm by employing Riemannian optimization to deal with t-SVD is described. Different from the bounded multi-rank constraint used in [21], the low-rank space with estimated multi-rank is smooth manifold in our optimization model. The key idea is to project each frontal slice of a tensor in Fourier domain onto a low multi-rank manifold space, which significantly reduces the computational complexity of t-SVD.

In a nutshell, the main contributions of this paper are:

1) ***GDME:*** Provides fast, accurate multi-rank information, improving tensor recovery performance.
2) ***TRPCA-EMR:*** Replaces TNN regularization with strict equality constraints, eliminating the need for manual hyperparameter adjustment.
3) ***Riemannian Optimization Algorithm:*** Efficiently handles t-SVD, reducing computational complexity.
4) Extensive experiments show that TRPCA-EMR outperforms existing methods in video background subtraction and hyperspectral image denoising, demonstrating its superior effectiveness and efficiency across diverse datasets.

# 2 Fast TRPCA with Estimated Multi-rank and Riemannian Optimization

In the following, a scalar is denoted by a lowercase letter, e.g., $x$. A vector is denoted by a boldface lowercase letter, e.g., $\mathbf{x}$ (the $i$th entry of vector $\mathbf{x}$ is denoted by $\mathbf{x}_i$ ); a matrix is written as a boldface capital letter, e.g., $\mathbf{X}$ ; tensors are denoted by calligraphic letters, e.g., $\boldsymbol{\mathcal{X}}$. Given a Riemannian manifold $\boldsymbol{\mathcal{M}}$ and a nonempty set $S \subset \boldsymbol{\mathcal{M}}$, $P_S(q)$ denotes the projection of point $q$ onto the set $S$ as follows

$$P_S(q) = \{p \in S \mid \|q - p\| = \inf_{s \in S} \|q - s\|\} \tag{1}$$

For a third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we denote the $i$th frontal slice of $\boldsymbol{\mathcal{X}}$ by $\boldsymbol{\mathcal{X}}(:,:,i)$ or $\boldsymbol{\mathcal{X}}^{(i)}$. The inner product between $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ in $\mathbb{R}^{\mathrm{n_1 \times n_2 \times n_3}}$ is defined as $\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle = \sum_1^{n_3} \langle \boldsymbol{\mathcal{X}}^{(i)}, \boldsymbol{\mathcal{Y}}^{(i)} \rangle$. The Frobenius norm of $\boldsymbol{\mathcal{X}}$ is calculated by $\|\boldsymbol{\mathcal{X}}\|_F = \sqrt{\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle}$. Besides, $\hat{\boldsymbol{\mathcal{X}}} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ is the discrete Fourier transform (DFT) of $\boldsymbol{\mathcal{X}}$ along the 3rd dimension and can be computed by $\hat{\boldsymbol{\mathcal{X}}} = \mathrm{fft}(\boldsymbol{\mathcal{X}}, [], 3)$. The inverse DFT is computed by $\boldsymbol{\mathcal{X}} = \mathrm{ifft}(\hat{\boldsymbol{\mathcal{X}}}, [], 3)$.

## 2.1 Multi-rank Estimation of a Low-rank Tensor

For a third-ordor tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the multi-rank is defined as a vector $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_{n_3}) \in \mathbb{R}^{n_3}$ [12], and the ith entry is given by

$$\mathbf{r}_i = \mathrm{rank}(\hat{\boldsymbol{\mathcal{X}}}^{(i)}) \tag{2}$$

where $\hat{\boldsymbol{\mathcal{X}}}^{(i)} \in \mathbb{R}^{n_1 \times n_2}$ is the $i$th frontal slice of $\boldsymbol{\mathcal{X}}$ in the Fourier domain. It means that the tensor multi-rank estimation problem can be converted into a set of matrix rank estimates in the Fourier domain. Since the size of tensor data is often large, TPCA is used to reduce the complexity of multi-rank estimation. In order to preserve the frequency information in the tensors, we reduce only the first two dimensions of $\boldsymbol{\mathcal{X}}$. When $n_1 <= n_2$, the TPCA can be implemented by a series of traditional PCAs as follows

$$\hat{\boldsymbol{\mathcal{T}}}^{(i)} = \mathrm{PCA}(\hat{\boldsymbol{\mathcal{X}}}^{(i)} (\hat{\boldsymbol{\mathcal{X}}}^{(i)})^{\mathrm{T}}, d) \tag{3}$$

where $\mathcal{T} \in \mathbb{R}^{n_1 \times d \times n_3}$ is the low-dimensional tensor after dimensionality reduction, and where $d$ represents the number of selected principal components, which is set to $\mathrm{round}(0.1n_1)$ in our method. Similarly, Eq. (3) can be written as $\hat{\mathcal{T}}^{(i)} = \mathrm{PCA}((\hat{\mathcal{X}}^{(i)})^{\mathrm{T}}\hat{\mathcal{X}}^{(i)}, \mathrm{round}(0.1n_2))$ if $n_1 > n_2$.

The covariance tensor $\mathcal{C} \in \mathbb{R}^{d \times d \times n_3}$ of the low-dimensional tensor $\mathcal{T}$ with multi-rank $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_{n_3}) \in \mathbb{R}^{n_3}$ can be defined as:

$$\hat{\mathcal{C}}^{(i)} = (\hat{\mathcal{T}}^{(i)})^{\mathrm{T}}\hat{\mathcal{T}}^{(i)} \tag{4}$$

Then the eigenvalue decomposition of $\hat{\mathcal{C}}^{(i)}$ is given by

$$\hat{\mathcal{C}}^{(i)} = \mathbf{U}_{\hat{\mathcal{C}}^{(i)}}\mathbf{\Sigma}_{\hat{\mathcal{C}}^{(i)}}\mathbf{U}_{\hat{\mathcal{C}}^{(i)}}^{\mathrm{H}} \tag{5}$$

where $\mathbf{U}_{\hat{\mathcal{C}}^{(i)}} = [u_1, u_2, \cdots, u_d]$ is the eigenvector matrix, and $\mathbf{\Sigma}_{\hat{\mathcal{C}}^{(i)}} = diag(\sigma_1, \sigma_2, \cdots, \sigma_d)$ represents the eigenvalue matrix. According to the Gerschgorin disk theorem [30], $\hat{\mathcal{C}}^{(i)}$ can be transformed as follows

$$\hat{\mathcal{C}}^{(i)} = \begin{bmatrix} c_{11} & \cdots & c_{1d} \\ \vdots & \ddots & \vdots \\ c_{d1} & \cdots & c_{dd} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_1^i & \mathbf{C}^i \\ (\mathbf{C}^i)^{\mathbf{H}} & c_{dd} \end{bmatrix} \tag{6}$$

where $\mathbf{C}_1^i \in \mathbb{R}^{(d-1)\times(d-1)}$ is the square matrix by the first $(d-1)$ rows and the first $(d-1)$ columns of $\mathbf{C}_1^i$, and $\mathbf{C}^i = [c_{1d}, c_{2d}, \cdots, c_{(d-1)d}]^{\mathrm{T}}$. Then the eigenvalue decomposition on $\mathbf{C}_1^i$ is obtained as:

$$\mathbf{C}_1^i = \mathbf{U}_{\mathbf{C}_1^i}\mathbf{\Sigma}_{\mathbf{C}_1^i}\mathbf{U}_{\mathbf{C}_1^i}^{\mathrm{H}} \tag{7}$$

where $\mathbf{U}_{\mathbf{C}_1^i} = [u'_1, u'_2, \cdots, u'_{k-1}]$ is an $(d-1) \times (d-1)$ unitary matrix composed of the eigenvectors of $\mathbf{C}_{\mathcal{T}\mathbf{1}}$, and $\mathbf{\Sigma}_{\mathbf{C}_1^i} = diag(\sigma'_1, \sigma'_2, \cdots, \sigma'_{k-1})$. Similar to Eq. (6), a unitary transformed matrix $\mathbf{U}'^i \in \mathbb{R}^{d\times d}(\mathbf{U}'^i(\mathbf{U}'^i)^{\mathbf{H}} = \mathbf{I})$ is defined as:

$$\mathbf{U}'^i = \begin{bmatrix} \mathbf{U}_{\mathbf{C}_1^i} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{8}$$

The transformed covariance tensor $\mathcal{C}_T$ is obtained by:

$$\begin{aligned} \hat{\mathcal{C}}_T^{(i)} &= (\mathbf{U}'^i)^{\mathbf{H}}\hat{\mathcal{C}}^{(i)}\mathbf{U}'^i \\ &= \begin{bmatrix} \mathbf{U}_{\mathbf{C}_1^i}^{\mathrm{H}}\mathbf{C}_1^i\mathbf{U}_{\mathbf{C}_1^i} & \mathbf{U}_{\mathbf{C}_1^i}^{\mathrm{H}}\mathbf{C}^i \\ (\mathbf{C}^i)^{\mathrm{H}}\mathbf{U}_{\mathbf{C}_1^i} & c_{dd} \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{diag}(\sigma'_1, \sigma'_2, ..., \sigma'_{d-1}) & (\rho_1, \rho_2, ..., \rho_{d-1})^{\mathrm{T}} \\ (\rho_1^*, \rho_2^*, ..., \rho_{d-1}^*) & c_{dd} \end{bmatrix} \end{aligned} \tag{9}$$

where $\rho_l = {u'_l}^{\mathrm{H}} \mathbf{C}^i, (l = 1, ..., d-1)$. The eigenvalues of $\hat{\mathcal{C}}_T^{(i)}$ can be estimated using the Gerschgorin disk theorem [30]. Then, the radii of the first $(d-1)$ Gerschgorin disk can be written as:

$$\mathrm{rad}_l = |\rho_l| = |{u'_l}^{\mathrm{H}} \mathbf{C}^i| \tag{10}$$

where the radius $\mathrm{rad}_l$ of the $l$th Gerschgorin disk depends on the size of ${u'_l}^{\mathrm{H}} \mathbf{C}^i$.

To further improve the accuracy of the rank estimation, the radii of the Gerschgorin disk are shrank in [2]. Then a new diagonal matrix $\mathbf{\Sigma}_T^i = \mathrm{diag}(\sigma'_1, \sigma'_2, \cdots, \sigma'_{d-1}, \sigma'_d)$ is constructed, where $\sigma'_d = \sqrt{\sum_{l=1}^{d-1} {\sigma'_l}^2}$. Thus, the new transformed tensor $\mathcal{C}_{T\mathbf{\Sigma}_T}$ is constructed as follows

$$\begin{aligned} \hat{\mathcal{C}}_{T\mathbf{\Sigma}_T}^{(i)} &= \mathbf{\Sigma}_T^i \hat{\mathcal{C}}_T^{(i)} (\mathbf{\Sigma}_T^i)^{-1} \\ &= \begin{bmatrix} \mathrm{diag}(\sigma'_1, \sigma'_2, ..., \sigma'_{d-1}) & (\frac{\sigma'_1}{\sigma'_d}\rho_1, \frac{\sigma'_2}{\sigma'_d}\rho_2, ..., \frac{\sigma'_{d-1}}{\sigma'_d}\rho_{d-1})^{\mathrm{T}} \\ (\frac{\sigma'_1}{\sigma'_d}\rho_1^*, \frac{\sigma'_2}{\sigma'_d}\rho_2^*, ..., \frac{\sigma'_{d-1}}{\sigma'_d}\rho_{d-1}^*) & c_{dd} \end{bmatrix} \end{aligned} \tag{11}$$

$\mathcal{C}_{T\mathbf{\Sigma}_T}$ and $\mathcal{C}_T$ are similar tensors and have the same eigenvalues on Fourier domain. In Eq. (11), the radii undergo varying degrees of compression. The estimated multi-rank is subsequently determined using an enhanced heuristic decision rule:

$$\mathrm{GDME}^i(t) = \frac{1}{\sigma'_d}[|\sigma'_t|\mathrm{rad}_t - \frac{H_{\hat{\mathcal{T}}^{(l)}}(t)}{d-1}\sum_{l=1}^{d-1}|\sigma'_i|\mathrm{rad}_l] \tag{12}$$

where $t = 1, 2, \cdots, d-2$, and $0 < H_{\hat{\mathcal{T}}^{(i)}}(t) < 1$ is the adjustment factor:

$$H_{\hat{\mathcal{T}}^{(i)}}(t) = \frac{2|\sigma'_{t+1}|}{\sqrt{\sum_{l=t}^{d-1} {\sigma'_l}^2}} \tag{13}$$

As a result, the estimated multi-rank $\mathbf{r}_i = t-1$ if the first negative value of (12) is reached at $t$.

## 2.2 TRPCA with Estimated Multi-rank

Given a tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , it is shown in [15] that the low-rank tensor $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and the sparse tensor $\mathcal{E} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ can be recovered by solving the following convex optimization

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

where $\|\mathcal{L}\|_*$ is the TNN of tensor $\mathcal{L}$, which are defined as the average of the nuclear norm of all the frontal slices of $\hat{\mathcal{L}}$ [31], $\|\mathcal{E}\|_1$ denotes the $l_1$-norm, i.e., sum of absolute values of entries in $\mathcal{E}$), and $\lambda$ is a parameter balancing the rankness and sparsity. It should be noted that the low-rank implies low tubal-rank (the largest one of multi-ranks).

In this study, the multi-rank of the target low-rank tensor has been estimated by our proposed GDME. Hence, a estimated multi-rank is used to tighten the constraints of problem

(14). The TRPCA, which consists of recovering the low-rank tensor and sparse tensor, can be formulated as the following optimization:

$$\min_{\boldsymbol{\mathcal{L}},\boldsymbol{\mathcal{E}}} \|\boldsymbol{\mathcal{E}}\|_1, \text{ s.t. } \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{E}}, \mathrm{rank}(\hat{\boldsymbol{\mathcal{L}}}^{(i)}) = \mathbf{r}_i \tag{15}$$

where $\mathbf{r} = (\mathbf{r}_1, \dots, \mathbf{r}_{n_3}) \in \mathbb{R}^{n_3}$ is the estimated multi-rank of the target tensor.

In this study, the alternating direction method of multipliers (ADMM) is used to solve the optimization (15). It should be noted that the rank function is kept as a constraint term because it is non-convex and discontinuous. The augmented Lagrangian function of Eq. (15) can be written as

$$\begin{gathered} \min_{\boldsymbol{\mathcal{L}},\boldsymbol{\mathcal{E}},\boldsymbol{\mathcal{Y}}} \|\boldsymbol{\mathcal{E}}\|_1 + \langle \boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}} - \boldsymbol{\mathcal{E}} \rangle + \tfrac{\mu}{2} \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}} - \boldsymbol{\mathcal{E}}\|_F^2 \\ \text{s.t. } \mathrm{rank}(\widehat{\boldsymbol{\mathcal{L}}}^{(i)}) = \mathbf{r}_i \end{gathered} \tag{16}$$

where $\langle \cdot, \cdot \rangle$ represents the tensor inner product, $\| \cdot \|_F$ is the Frobenius norm, $\mu$ is a positive penalty scalar, and $\boldsymbol{\mathcal{Y}}$ is the Lagrangian multiplier. According to ADMM, the three variables of the problem (16) need to be solved separately in each iteration. Thus problem (16) is divided into the following three sub-problems:

$\boldsymbol{\mathcal{E}}$ sub-problem: While $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{Y}}$ are fixed, Eq. (16) is equal to the following problem:

$$\boldsymbol{\mathcal{E}}^* = \min_{\boldsymbol{\mathcal{E}}} \|\boldsymbol{\mathcal{E}}\|_1 + \frac{\mu}{2} \left\| \boldsymbol{\mathcal{E}} - (\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}} + \mu^{-1}\boldsymbol{\mathcal{Y}}) \right\|_F^2 \tag{17}$$

A soft-thresholding operator [32] is introduced as follows to solve sub-problem (17)

$$\mathrm{prox}_\tau(x) = \begin{cases} x - \tau & \text{if} \quad x > \tau \\ 0 & \text{if} \quad |x| \le \tau \\ x + \tau & \text{if} \quad x < -\tau \end{cases} \tag{18}$$

where $x \in \mathbb{R}$ and $\tau > 0$. $\boldsymbol{\mathcal{E}}$ can be updated by the soft-thresholding operator: $\mathrm{prox}_{1/\mu}(\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}} + \mu^{-1}\boldsymbol{\mathcal{Y}})$ [33].

$\boldsymbol{\mathcal{L}}$ sub-problem: While $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\mathcal{Y}}$ are fixed, Eq. (16) is equal to the following problem:

$$\begin{gathered} \boldsymbol{\mathcal{L}}^* = \min_{\boldsymbol{\mathcal{L}}} \tfrac{1}{2} \|\boldsymbol{\mathcal{L}} - (\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{E}} + \mu^{-1}\boldsymbol{\mathcal{Y}})\|_F^2 \\ \text{s.t. } \mathrm{rank}(\widehat{\boldsymbol{\mathcal{L}}}^{(i)}) = \mathbf{r}_i \end{gathered} \tag{19}$$

In TRPCA methods, the $\boldsymbol{\mathcal{L}}$ sub-problem is usually solved via tensor singular value thresholding (t-SVT) [15], which applies a soft-thresholding rule in t-SVD. However, we adopt truncation strategy in t-SVD due to the equality constraints $\mathrm{rank}(\widehat{\boldsymbol{\mathcal{L}}}^{(i)}) = \mathbf{r}_i$. Denoting $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{E}} + \mu^{-1}\boldsymbol{\mathcal{Y}}$, the t-SVD of $\boldsymbol{\mathcal{Z}}$ is obtained by computing the matrix SVD in the Fourier domain [12]:

$$[\hat{\boldsymbol{\mathcal{U}}}^{(i)}, \hat{\boldsymbol{\mathcal{S}}}^{(i)}, \hat{\boldsymbol{\mathcal{V}}}^{(i)}] = \mathrm{SVD}(\hat{\boldsymbol{\mathcal{Z}}}^{(i)}) \tag{20}$$

where $\hat{\boldsymbol{\mathcal{U}}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\hat{\boldsymbol{\mathcal{V}}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ are orthogonal tensors. $\hat{\boldsymbol{\mathcal{S}}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor, then the optimization (19) is directly solved by the truncation singular value decomposition as follows

$$\hat{\mathcal{L}}^{(i)} = H_{\mathbf{r}_i}(\hat{\mathcal{Z}}^{(i)}) = \hat{\mathcal{U}}_{\mathbf{r}_i}^{(i)} \mathrm{diag}(\sigma_{\mathbf{r}_i}(\hat{\mathcal{Z}}^{(i)}))(\hat{\mathcal{V}}_{\mathbf{r}_i}^{(i)})^{\mathrm{T}} \tag{21}$$

where $\hat{\mathcal{U}}_{\mathbf{r}_i}^{(i)} = \hat{\mathcal{U}}^{(i)}(:, 1 : \mathbf{r}_i)$, $\hat{\mathcal{V}}_{\mathbf{r}_i}^{(i)} = \hat{\mathcal{V}}^{(i)}(:, 1 : \mathbf{r}_i)$, and $\sigma_{\mathbf{r}_i}(\hat{\mathcal{Z}}^{(i)})$ is the first $\mathbf{r}_i$ singular values of $\hat{\mathcal{Z}}^{(i)}$.

In this way, we avoid the case of falsely separating the sparse information into the low-rank tensor in a scenario where sparse data are distributed nonuniformly while retaining the low-rank structure of the low-rank tensor. Therefore, the low-rank tensor $\mathcal{L}$ can be recovered more accurately by the proposed TRPCA-EMR model in Eq. (15).

## 2.3 Riemannian Optimization for TRPCA with Estimated Multi-rank

It is complex to solve t-SVD of the tensor with a large size, and which is a major problem in TRPCA or its related algorithms. To reduce the complexity of t-SVD, Riemannian optimization is introduced to solve the $\mathcal{L}$ sub-problem (19).

The $i$th frontal slice of $\hat{\mathcal{L}}$ is in a low-rank space. Thus, TRPCA-EMR model based on Riemannian manifold is as follows

$$\min_{\mathcal{L},\mathcal{E}} \|\mathcal{E}\|_1, \text{ s.t. } \mathcal{X} = \mathcal{L} + \mathcal{E}, \widehat{\mathcal{L}}^{(i)} \in \mathcal{M}_{\mathbf{r}_i} \tag{22}$$

where $\mathcal{M}_{\mathbf{r}_i}$ is an $(n_1 + n_2 - \mathbf{r}_i)\mathbf{r}_i$-dimensional smooth manifold which is embedded in $\mathbb{R}^{n_1 \times n_2}$, and satisfies

$$\mathcal{M}_{\mathbf{r}_i} = \{\widehat{\mathcal{L}}^{(i)} \in \mathbb{R}^{n_1 \times n_2}, \mathrm{rank}(\hat{\mathcal{L}}^{(i)}) = \mathbf{r}_i\} \tag{23}$$

The $\mathcal{L}$ sub-problem (19) is solved via Riemannian manifold as follows

$$\mathcal{L}^* = \min_{\mathcal{M}_{\mathbf{r}_i}} \frac{1}{2}\|\mathcal{L} - \mathcal{Z}\|_F^2 \tag{24}$$

In the $k$th iteration of low-tensor frontal slice $\hat{\mathcal{L}}_k^{(i)}$, we obtain the $(n_1+n_2-\mathbf{r}_i)\mathbf{r}_i$-dimensional tangent subspace composed of the left and right singular matrix $(\mathbf{U}_k^i, \mathbf{V}_k^i)$ of $\hat{\mathcal{L}}_k^{(i)}$ as

$$T_{\widehat{\mathcal{L}}_k^{(i)}} = \{\mathbf{A}{\mathbf{V}_k^i}^{\mathrm{T}} + \mathbf{U}_k^i\mathbf{B}^{\mathrm{T}} : \mathbf{A} \in \mathbb{R}^{n_1 \times \mathbf{r}_i}, \mathbf{B} \in \mathbb{R}^{n_2 \times \mathbf{r}_i}\} \tag{25}$$

There are two steps in solving Eq. (24): we first project $\widehat{\mathcal{Z}}_{k+1}^{(i)}$ onto the low-dimensional subspace $T_{\widehat{\mathcal{L}}^{(i)}}$, then shrink the projection on $\mathcal{M}_{\mathbf{r}_i}$ to get $\widehat{\mathcal{L}}_{k+1}^{(i)}$.

Assuming that $\mathbf{P}_{\mathbf{U}_k^i} = \mathbf{U}_k^i{\mathbf{U}_k^i}^{\mathrm{T}}$ and its orthogonal complement $\mathbf{P}_{\mathbf{U}_k^i}^{\perp} = \mathbf{I} - \mathbf{P}_{\mathbf{U}_k^i}$, $\mathbf{P}_{\mathbf{V}_k^i} = \mathbf{V}_k^i{\mathbf{V}_k^i}^{\mathrm{T}}$ and its orthogonal complement $\mathbf{P}_{\mathbf{V}_k^i}^{\perp} = \mathbf{I} - \mathbf{P}_{\mathbf{V}_k^i}$, where $\mathbf{I}$ is an identity matrix, the

$\mathcal{L}$ sub-problem (24) can be solved by the following equation:

$$\begin{aligned}\widehat{\mathcal{L}}_{k+1}^{(i)} = H_{\mathbf{r}_i}(P_{T_{\widehat{\mathcal{L}}_k^{(i)}}}(\widehat{\mathcal{Z}}_{k+1}^{(i)})) = H_{\mathbf{r}_i}(\mathbf{P}_{\mathbf{U}_k^i}\widehat{\mathcal{Z}}_{k+1}^{(i)}\mathbf{P}_{\mathbf{V}_k^i} \\ +\mathbf{P}_{\mathbf{U}_k^i}^{\perp}\widehat{\mathcal{Z}}_{k+1}^{(i)}\mathbf{P}_{\mathbf{V}_k^i} + \mathbf{P}_{\mathbf{U}_k^i}\widehat{\mathcal{Z}}_{k+1}^{(i)}\mathbf{P}_{\mathbf{V}_k^i}^{\perp})\end{aligned} \tag{26}$$

Let $\mathbf{C}_1 = {\mathbf{U}_k^i}^{\mathrm{T}}\widehat{\mathcal{Z}}_{k+1}^{(i)}(\mathbf{I} - \mathbf{V}_k^i{\mathbf{V}_k^i}^{\mathrm{T}})$ and its QR decompose $\mathbf{C}_1 = \mathbf{R}_1^{\mathrm{T}}\mathbf{Q}_1^{\mathrm{T}}$, $\mathbf{C}_2 = (\mathbf{I} - \mathbf{U}_k^i{\mathbf{U}_k^i}^{\mathrm{T}})\widehat{\mathcal{Z}}^{(i)}\mathbf{V}$ and its QR decompose $\mathbf{C}_2 = \mathbf{Q}_2\mathbf{R}_2$. The project formulation in (26) can be represented as

$$\begin{aligned}&P_{T_{\widehat{\mathcal{L}}_k^{(i)}}}(\widehat{\mathcal{Z}}_{k+1}^{(i)}) \\ &= \mathbf{U}_k^i{\mathbf{U}_k^i}^{\mathrm{T}}\widehat{\mathcal{Z}}_{k+1}^{(i)}\mathbf{V}_k^i{\mathbf{V}_k^i}^{\mathrm{T}} + \mathbf{U}_k^i\mathbf{R}_1^{\mathrm{T}}\mathbf{Q}_1^{\mathrm{T}} + \mathbf{Q}_2\mathbf{R}_2{\mathbf{V}_k^i}^{\mathrm{T}} \\ &= \begin{bmatrix}\mathbf{U}_k^i & \mathbf{Q}_2\end{bmatrix}\begin{bmatrix}{\mathbf{U}_k^i}^{\mathrm{T}}\widehat{\mathcal{Z}}_{k+1}^{(i)}\mathbf{V}_k^i & \mathbf{R}_1^{\mathrm{T}} \\ \mathbf{R}_2 & 0\end{bmatrix}\begin{bmatrix}{\mathbf{V}_k^i}^{\mathrm{T}} \\ \mathbf{Q}_1^{\mathrm{T}}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{U}_k^i & \mathbf{Q}_2\end{bmatrix}\mathbf{D}\begin{bmatrix}{\mathbf{V}_k^i}^{\mathrm{T}} \\ \mathbf{Q}_1^{\mathrm{T}}\end{bmatrix}\end{aligned} \tag{27}$$

where $\begin{bmatrix}\mathbf{U}_k^i & \mathbf{Q}_2\end{bmatrix}$ and $\begin{bmatrix}{\mathbf{V}_k^i}^{\mathrm{T}} \\ \mathbf{Q}_1^{\mathrm{T}}\end{bmatrix}$ are unitary matrices. The shrinkage of $P_{T_{\widehat{\mathcal{L}}_k^{(i)}}}(\widehat{\mathcal{Z}}_{k+1}^{(i)})$ can be solved by the SVD of $\mathbf{D}$, which is an $2\mathbf{r}_i \times 2\mathbf{r}_i$ matrix. Therefore, the complexity of SVD of $\widehat{\mathcal{Z}}^{(i)}$ is reduced from $\mathrm{O}(\min(n_1, n_2)^3)$ to $\mathrm{O}(\mathbf{r}_i)^3$. If $\mathbf{r}_i \ll \min(n_1, n_2)$, the TRPCA-EMR combined with Riemannian manifold can greatly accelerate the operation of the algorithm. The overall procedure is summarized in Algorithm 1.

**Remark 1** *It should be noted that the proposed Algorithm 1 follows the framework of ADMM [34]. However, the optimization (16) is a nonconvex problem. Although mathematical proof of the convergence is challenging, the following empirical claim is provided.*

**Claim 1** *The sequences $\mathcal{L}_k$ and $\mathcal{E}_k$ generated by Algorithm 1 satisfy the following:*

$$C_{\mathcal{L}} = \lim_{k\to\infty} \|\mathcal{L}_{k+1} - \mathcal{L}_k\|_F = 0 \tag{28}$$

$$C_{\mathcal{E}} = \lim_{k\to\infty} \|\mathcal{E}_{k+1} - \mathcal{E}_k\|_F = 0 \tag{29}$$

$$C_{\mathcal{X}} = \lim_{k\to\infty} \|\mathcal{X} - \mathcal{L}_{k+1} - \mathcal{E}_{k+1}\|_F = 0 \tag{30}$$

*Claim 1 has been proven by the experiment shown in Section 3.1.*

# 3 Experiments

In this section, we present our experimental results of our TRPCA with estimated tensor multi-rank (TRPCA-EMR) in detail. We compare the proposed methods with competitive tensor robust PCA algorithms (TRPCA [15], IRTPCA [17], OR-TPCA [16], ETRPCA [20], RTPCA-RE [9]). We set all parameters of each algorithm as recommended values in all experiments, and all the simulations are conducted on a laptop with Windows 10, Intel(R) Core (TM) i5-9300H CPU (8 cores at 2.4 GHz) and 16 GB RAM, and run in MATLAB R2016a.

**Algorithm 1** TRPCA-EMR

**Input:** $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\boldsymbol{\mathcal{L}}_0 = \boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{E}}_0 = \boldsymbol{\mathcal{Y}}_0 = \mathbf{0} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathbf{r}_i$ is estimated by Eq. (12), $\rho = 1.1$, $\mu_0 = 1e-4$, $\mu_{max} = 1e10$, $\epsilon = 1e-8$.

**while** not converged **do**
  compute $\boldsymbol{\mathcal{E}}_{k+1} = \text{prox}_{1/\mu_k}(\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}}_k + \mu_k^{-1}\boldsymbol{\mathcal{Y}}_k)$,
  compute $\boldsymbol{\mathcal{Z}}_{k+1} = \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{E}}_{k+1} + \mu_k^{-1}\boldsymbol{\mathcal{Y}}_k$.
  **for** $i = 1$ to $n_3$ **do**
    compute $\widehat{\boldsymbol{\mathcal{L}}}_{k+1}^{(i)}$ by Eq. (26).
  **end for**
  compute $\boldsymbol{\mathcal{Y}}_{k+1} = \boldsymbol{\mathcal{Y}}_k + \mu_k(\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}}_{k+1} - \boldsymbol{\mathcal{E}}_{k+1})$,
  update $\mu_{k+1} = \min(\rho\mu_k, \mu_{max})$.
  check the convergence conditions:
  $\|\boldsymbol{\mathcal{L}}_{k+1} - \boldsymbol{\mathcal{L}}_k\|_\infty \leq \epsilon$, $\|\boldsymbol{\mathcal{E}}_{k+1} - \boldsymbol{\mathcal{E}}_k\|_\infty \leq \epsilon$,
  $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}}_{k+1} - \boldsymbol{\mathcal{E}}_{k+1}\|_\infty \leq \epsilon$.
**end while**

**Output:** $\boldsymbol{\mathcal{L}}$,$\boldsymbol{\mathcal{E}}$

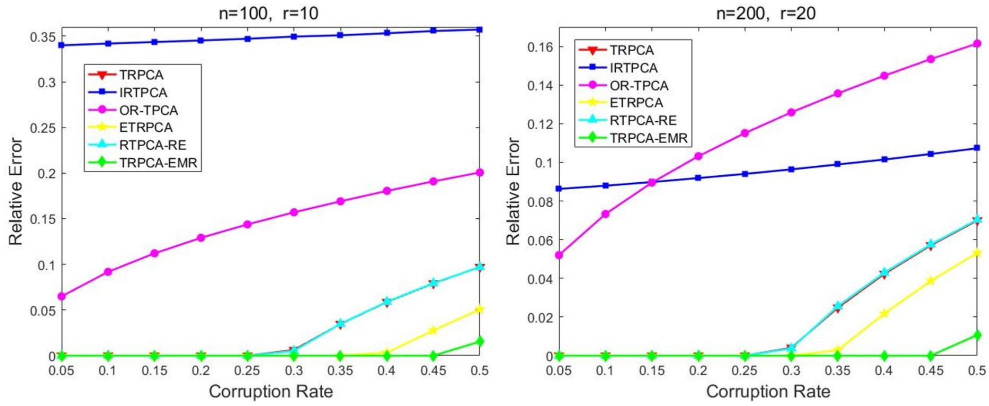


**Fig. 1** Relative errors $E_{\boldsymbol{\mathcal{L}}}$ for various algorithms with different corruption rates.

## 3.1 Synthetic Datasets

In this subsection, we test the algorithms on synthetic data. A tensor $\boldsymbol{\mathcal{L}}_\mathbf{0} \in \mathbb{R}^{n \times n \times n}$ with multi-rank $[r, r, \cdots, r]^T$ is generated by sampling two tensors, $\boldsymbol{\mathcal{P}} \in \mathbb{R}^{n \times r \times n}$ and $\boldsymbol{\mathcal{Q}} \in \mathbb{R}^{r \times n \times n}$ with entries belonging to normal distribution $\mathcal{N}(0, 1/n)$, namely, $\boldsymbol{\mathcal{L}}_\mathbf{0} = \boldsymbol{\mathcal{P}} * \boldsymbol{\mathcal{Q}}$. The ground truth tensor $\boldsymbol{\mathcal{L}}_\mathbf{0}$ is corrupted by sparse tensor $\boldsymbol{\mathcal{E}}_\mathbf{0} \in \mathbb{R}^{n \times n \times n}$, which has $p \times (m \times n)$ non-zero elements. The positions of non-zero elements in $\boldsymbol{\mathcal{E}}_\mathbf{0}$ are randomly selected, and their values are generated from a Gaussian distribution $\mathcal{N}(0, 1/n)$. In this experiment, we define $p$ as the corruption rate of synthetic data and set it from 0.1 to 0.5. In addition, we set $r = 0.1n$ and $n = 100, 200$. Denote the solution as $\boldsymbol{\mathcal{L}}_*$ and $\boldsymbol{\mathcal{E}}_*$ in a certain algorithm, we evaluate the relative errors $E_{\boldsymbol{\mathcal{L}}} = \|\boldsymbol{\mathcal{L}}_* - \boldsymbol{\mathcal{L}}_\mathbf{0}\|_F / \|\boldsymbol{\mathcal{L}}_\mathbf{0}\|_F$ and $E_{\boldsymbol{\mathcal{E}}} = \|\boldsymbol{\mathcal{E}}_* - \boldsymbol{\mathcal{E}}_\mathbf{0}\|_F / \|\boldsymbol{\mathcal{E}}_\mathbf{0}\|_F$ under different settings and present the results in Fig. 1, Table 1 and Table 2. Furthermore, for methods with

**Table 1** Comparison of relative errors and running times (s) for various algorithms with different corruption rates ($n = 100, r = 10$).

| Methods | TRPCA [15] | | | IRTPCA [17] | | | OR-TPCA [16] | | | ETRPCA [20] | | | RTPCA-RE [9] | | | | | TRPCA-EMR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | $E_{\mathcal{L}}$ | $E_{\mathcal{E}}$ | Time | $E_{\mathcal{L}}$ | $E_{\mathcal{E}}$ | Time | $E_{\mathcal{L}}$ | $E_{\mathcal{E}}$ | Time | $E_{\mathcal{L}}$ | $E_{\mathcal{E}}$ | Time | $E_{\mathcal{L}}$ | $E_{\mathcal{E}}$ | Time | TR | TimeRE | $E_{\mathcal{L}}$ | $E_{\mathcal{E}}$ | Time | $\mathrm{MR}_i$ | TimeRE |
| $p = 0.05$ | 2.13e-10 | 1.68e-09 | 20.7 | 3.40e-01 | 9.52e-01 | 23.6 | 6.49e-02 | 9.27e-01 | 34.0 | 1.67e-10 | 1.35e-09 | 46.9 | 1.18e-10 | 1.90e-09 | 55.1 | 10 | 14.5 | **6.03e-12** | **1.53e-10** | **18.1** | 10 | 1.5 |
| $p = 0.1$ | 2.43e-10 | 1.57e-09 | 24.0 | 3.42e-01 | 9.50e-01 | 23.9 | 9.18e-02 | 9.23e-01 | 33.8 | 1.16e-10 | 7.33e-10 | 49.6 | 2.72e-10 | 3.05e-09 | 61.6 | 10 | 17.8 | **4.34e-11** | **6.94e-10** | **18.5** | 10 | 1.6 |
| $p = 0.15$ | 3.19e-10 | 1.79e-09 | 23.9 | 3.44e-01 | 9.49e-01 | 24.0 | 1.12e-01 | 9.20e-01 | 32.8 | 1.42e-10 | 8.19e-10 | 49.6 | 4.10e-10 | 3.64e-09 | 63.0 | 10 | 20.9 | **6.20e-11** | **7.18e-10** | **18.5** | 10 | 1.6 |
| $p = 0.2$ | 3.92e-10 | 2.03e-09 | 24.3 | 3.45e-01 | 9.47e-01 | 25.0 | 1.29e-01 | 9.17e-01 | 33.4 | 2.37e-10 | 1.25e-09 | 52.0 | 1.20e-10 | **9.25e-10** | 78.3 | 10 | 24.6 | **1.05e-10** | 1.01e-09 | **18.8** | 10 | 1.6 |
| $p = 0.25$ | 3.10e-10 | 1.53e-09 | 25.4 | 3.47e-01 | 9.46e-01 | 24.9 | 1.44e-01 | 9.13e-01 | 33.3 | 2.95e-10 | 1.45e-09 | 53.5 | 4.23e-10 | 2.82e-09 | 94.4 | 11 | 43.3 | **1.65e-10** | **1.36e-09** | **17.7** | 10 | 1.5 |
| $p = 0.3$ | 6.20e-03 | 3.59e-02 | 27.0 | 3.49e-01 | 9.45e-01 | 24.3 | 1.57e-01 | 9.10e-01 | 32.2 | 8.41e-10 | 3.82e-09 | 52.8 | 5.23e-03 | 3.01e-02 | 183.1 | 50 | 40.3 | **1.12e-10** | **8.14e-10** | **18.3** | 10 | 1.6 |
| $p = 0.35$ | 3.47e-02 | 1.86e-01 | 29.0 | 3.51e-01 | 9.44e-01 | 24.7 | 1.69e-01 | 9.07e-01 | 32.6 | 3.15e-10 | 1.43e-09 | 56.3 | 3.48e-02 | 1.86e-01 | 82.0 | 50 | 41.1 | **2.55e-10** | **1.61e-09** | **18.4** | 10 | 1.6 |
| $p = 0.4$ | 5.88e-02 | 2.95e-01 | 29.5 | 3.53e-01 | 9.43e-01 | 25.7 | 1.80e-01 | 9.05e-01 | 33.6 | 3.35e-03 | 1.68e-02 | 69.9 | 5.90e-02 | 2.94e-01 | 86.3 | 50 | 46.2 | **3.52e-10** | **2.00e-09** | **21.1** | 10 | 1.5 |
| $p = 0.45$ | 7.92e-02 | 3.74e-01 | 31.2 | 3.56e-01 | 9.41e-01 | 27.2 | 1.91e-01 | 9.03e-01 | 34.9 | 2.76e-02 | 1.30e-01 | 73.3 | 7.94e-02 | 3.74e-01 | 92.0 | 50 | 51.9 | **1.14e-09** | **5.90e-09** | **22.6** | 10 | 1.6 |
| $p = 0.5$ | 9.71e-02 | 4.36e-01 | 29.7 | 3.57e-01 | 9.41e-01 | 26.4 | 2.01e-01 | 9.01e-01 | 32.7 | 5.07e-02 | 2.27e-01 | 67.1 | 9.70e-02 | 4.33e-01 | 99.0 | 50 | 59.5 | **1.54e-02** | **6.93e-02** | **23.7** | 10 | 1.6 |

**Table 2** Comparison of relative errors and running times (s) for various algorithms with different corruption rates ($n = 200, r = 20$).

| Methods | TRPCA [15] | | | IRTPCA [17] | | | OR-TPCA [16] | | | ETRPCA [20] | | | RTPCA-RE [9] | | | | | TRPCA-EMR | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Data | $E_{\mathcal{L}}$ | $E_{\mathcal{E}}$ | Time | $E_{\mathcal{L}}$ | $E_{\mathcal{E}}$ | Time | $E_{\mathcal{L}}$ | $E_{\mathcal{E}}$ | Time | $E_{\mathcal{L}}$ | $E_{\mathcal{E}}$ | Time | $E_{\mathcal{L}}$ | $E_{\mathcal{E}}$ | Time | TR | TimeRE | $E_{\mathcal{L}}$ | $E_{\mathcal{E}}$ | Time | $\mathrm{MR}_i$ | TimeRE |
| $p = 0.05$ | 1.06e-11 | 2.73e-10 | 182.5 | 8.63e-02 | 8.13e-01 | 156.7 | 5.20e-02 | 1.04e+00 | 233.9 | 1.06e-11 | 2.74e-10 | 417.5 | 1.39e-11 | 3.74e-10 | 586.0 | 20 | 70.3 | **2.57e-13** | **1.11e-11** | **128.8** | 20 | 6 |
| $p = 0.1$ | 4.30e-12 | 2.67e-11 | 242.3 | 8.79e-02 | 8.13e-01 | 197.1 | 7.33e-02 | 1.04e+00 | 277.5 | 4.23e-12 | 2.82e-11 | 676.7 | 1.66e-11 | 3.34e-10 | 603.1 | 20 | 90.4 | **7.00e-12** | **1.93e-10** | **172.4** | 20 | 5.9 |
| $p = 0.15$ | 3.84e-12 | 1.94e-11 | 313.1 | 8.98e-02 | 8.13e-01 | 265.6 | 8.95e-02 | 1.03e+00 | 423.0 | 2.56e-12 | 1.52e-11 | 657.5 | 1.61e-11 | 2.44e-10 | 637.0 | 20 | 110.6 | **1.43e-12** | **2.39e-11** | **173.3** | 20 | 5.9 |
| $p = 0.2$ | 1.31e-11 | 1.54e-10 | 226.6 | 9.19e-02 | 8.12e-01 | 184.9 | 1.03e-01 | 1.03e+00 | 260.1 | 4.53e-12 | 3.29e-11 | 483.5 | 2.27e-11 | 3.09e-10 | 688.9 | 20 | 138.3 | **9.77e-13** | **1.72e-11** | **145.9** | 20 | 6 |
| $p = 0.25$ | 4.89e-12 | 3.14e-11 | 220.9 | 9.40e-02 | 8.12e-01 | 183.5 | 1.15e-01 | 1.03e+00 | 245.9 | 4.61e-12 | 2.90e-11 | 458.9 | 1.42e-11 | 1.43e-10 | 747.3 | 20 | 174.7 | **1.71e-12** | **2.18e-11** | **143.0** | 20 | 5.9 |
| $p = 0.3$ | 4.11e-03 | 3.35e-02 | 196.1 | 9.64e-02 | 8.12e-01 | 182.8 | 1.26e-01 | 1.03e+00 | 242.3 | 6.40e-12 | 4.07e-11 | 472.5 | 3.78e-03 | 3.08e-02 | 1380.4 | 100 | 300.5 | **2.58e-12** | **2.70e-11** | **142.4** | 20 | 6 |
| $p = 0.35$ | 2.49e-02 | 1.88e-01 | 194.0 | 9.89e-02 | 8.13e-01 | 184.0 | 1.36e-01 | 1.03e+00 | 257.8 | 2.81e-03 | 2.13e-02 | 499.0 | 2.56e-02 | 1.93e-01 | 889.7 | 100 | 338.0 | **1.48e-11** | **1.46e-10** | **157.4** | 20 | 5.9 |
| $p = 0.4$ | 4.24e-02 | 3.00e-01 | 218.5 | 1.01e-01 | 8.13e-01 | 206.0 | 1.45e-01 | 1.02e+00 | 273.1 | 2.17e-02 | 1.54e-01 | 492.5 | 4.29e-02 | 3.03e-01 | 924.3 | 100 | 357.7 | **1.31e-11** | **1.05e-10** | **160.1** | 20 | 5.9 |
| $p = 0.45$ | 5.71e-02 | 3.81e-01 | 184.4 | 1.04e-01 | 8.13e-01 | 173.6 | 1.53e-01 | 1.02e+00 | 229.2 | 3.87e-02 | 2.58e-01 | 412.7 | 5.75e-02 | 3.83e-01 | 973.8 | 100 | 389.0 | **6.11e-11** | **4.45e-10** | **133.8** | 20 | 5.9 |
| $p = 0.5$ | 7.00e-02 | 4.43e-01 | 184.1 | 1.07e-01 | 8.14e-01 | 172.8 | 1.61e-01 | 1.02e+00 | 227.9 | 5.31e-02 | 3.36e-01 | 415.7 | 7.04e-02 | 4.44e-01 | 1042.5 | 100 | 426.1 | **1.07e-02** | **6.79e-02** | **141.6** | 20 | 5.9 |

rank estimation, namely RTPCA-RE and the proposed TRPCA-EMR, we introduce additional metrics in Tables 1 and 2: the estimated rank values (denoted as TR and $\mathrm{MR}_i$) and the time taken for rank estimation (denoted as TimeRE).

From Fig. 1, it can be seen that the relative errors of the proposed TRPCA-EMR are significantly smaller than others except that it is very close to TRPCA, RTPCA-RE, and ETRPCA in some cases. It should be noted that the performance of all the algorithms gradually decreases with an increasing corruption rate under the same rank and size, however, our method decreases more slowly, which means that TRPCA-EMR is superior when the low-rank information is perturbed very seriously.

We further evaluate the performance of our proposed TRPCA-EMR in Tables 1 and 2. When the corruption rate is low ($p \leq 0.25$), TRPCA, ETRPCA, RTPCA-RE, and TRPCA-EMR can effectively separate low-rank and sparse tensors. In this scenario, the proposed TRPCA-EMR achieves only a slight improvement in accuracy compared with the other methods. However, at higher corruption rates ($0.3 \leq p \leq 0.45$), only our proposed TRPCA-EMR can accurately recover low-rank tensors. This is attributed to the fact that, in contrast to the tubal-rank estimation in RTPCA-RE, our proposed GDME ensures an accurate estimation of multi-rank MR even under high corruption rates. Simultaneously, the speed of the proposed GDME is nearly unaffected by the corruption rate, making it faster and more stable than RTPCA-RE. Moreover, the proposed TRPCA-EMR has the shortest running time among all the algorithms across all the different settings, indicating that Riemannian optimization significantly reduces the computational cost of t-SVD.

We subsequently assess the convergence of TRPCA-EMR under the following two cases: Case 1) $r = 0.1n$, $n = 100$, and the corruption rate $p = (0.1, 0, 2, 0.3, 0.4, 0.5)$; Case 2) $n = 100$, $p = 0.1$, and $r = (0.01n, 0.05n, 0.1n, 0.15n, 0.2n)$. For each of the ten settings across the two cases, we conducted 10 simulations and recorded the algorithm's iteration count when $max(C_{\mathcal{L}}, C_{\mathcal{E}}, C_{\mathcal{Y}}) < e - 8$ ($C_{\mathcal{L}}$, $C_{\mathcal{E}}$ and $C_{\mathcal{Y}}$ are defined in Claim 1 in Section 1), as depicted in Fig. 2. Additionally, Fig. 3 illustrates the convergence of our proposed TRPCA-EMR in a single simulation.

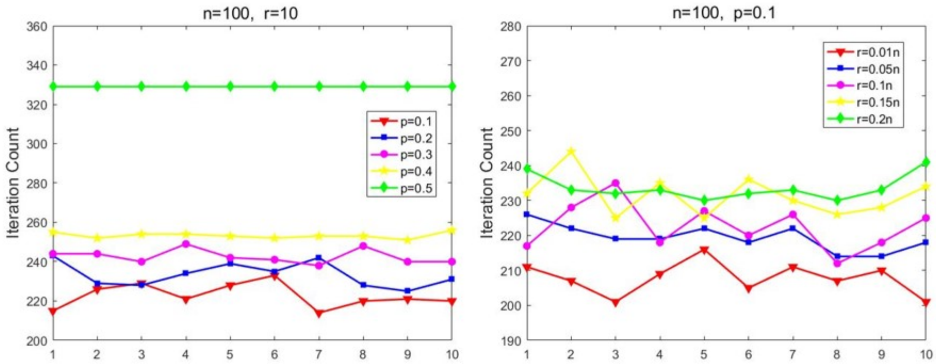


**Fig. 2** Convergence iteration count achieved by TRPCA-EMR under different settings.

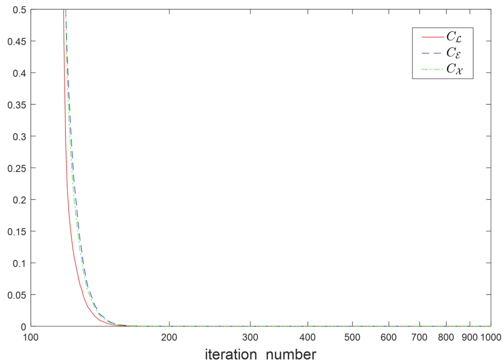


**Fig. 3** The values of $C_{\boldsymbol{\mathcal{L}}}$, $C_{\boldsymbol{\mathcal{E}}}$ and $C_{\boldsymbol{\mathcal{X}}}$ with the number of iterations. Assume that $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n\times n\times n}$ with multi-rank $[r, r, \cdots, r]^T$, $n = 100$, $r = 10$, and the corruption rate p=0.1.

## 3.2 Real Datasets

In this subsection, we evaluate the proposed TRPCA-EMR method on three real-world applications: background subtraction of video, dynamic background subtraction of video and hyperspectral image denoising, and compare the performance with those by TRPCA, IRTPCA, OR-TPCA, ETRPCA, RTPCA-RE methods.

### 3.2.1 Background subtraction of video

The background subtraction aims to separate the moving foreground objects from the static background. Our experiment use the Highway1 dataset [35, 36], which features a relatively static background with fast-moving cars as the dynamic foreground. The size of each frame of the Highway dataset is $320 \times 240$, which is reduced to $160 \times 120$ to save time. The dataset, consisting of 439 frames, is reshaped into a $19200 \times 439 \times 3$ tensor. We apply each algorithm to decompose the tensor into low-rank parts, which represent the static background

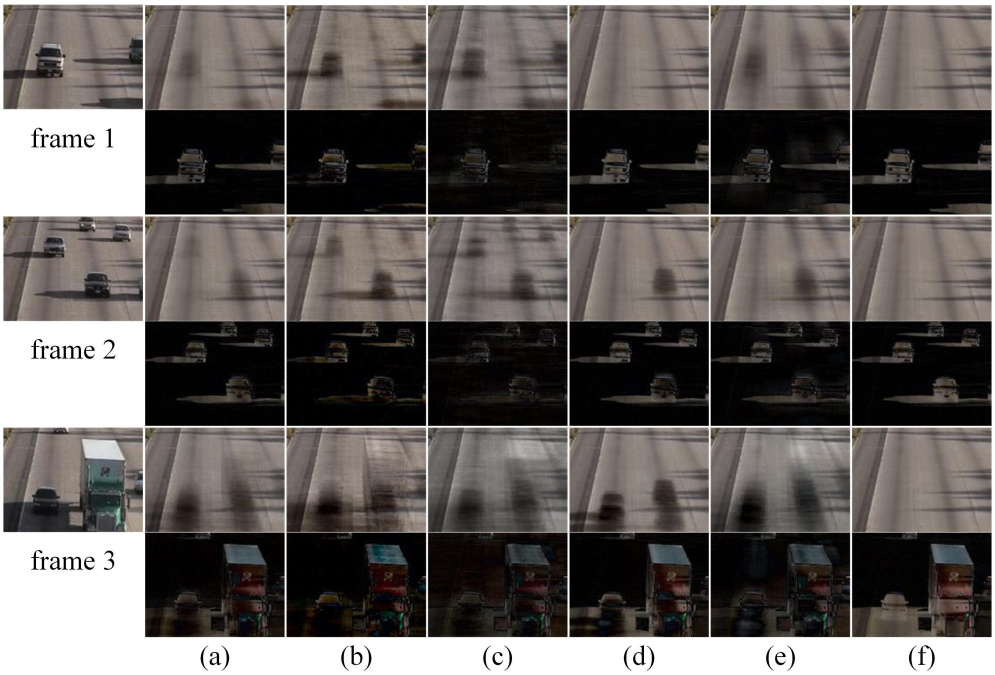


**Fig. 4** Video background subtraction: the first column corresponds to three frames from the video. The second to last columns are the separated background and foreground of TRPCA, IRTPCA, OR-TPCA, ETRPCA, RTPCA-RE and TRPCA-EMR respectively.

of the video, and sparse parts, which represent the moving objects in the video. The results are shown in Fig. 4. It can be seen from Fig. 4 (b, c) that both IRTPCA and OR-TPCA cannot completely separate the foreground from the background. Fig. 4 (a, e, f) shows that without an adjustment of the balance parameter, TRPCA and RTPCA-RE have poorer performances than our method. While RTPCA-RE successfully separated the foreground and background in frame 1, it failed in frames 2 and 3. Actually, almost all the methods have poorer performance on frames that have more complex foregrounds, such as frame 3. However, it can be seen from Fig. 4 (f) that our proposed TRPCA-EMR has stable and good results on all the frames, which benefits from the exact multi-rank estimate of the low-rank tensor obtained by our proposed GDME.

We then evaluate TRPCA-EMR on seven other video datasets [35–37] CAVIAR1 ($128 \times 192$, 610), CAVIAR2 ($128 \times 192$, 460), HallAndMonitor ($120 \times 176$, 296), HumanBody2 ($120 \times 160$, 740), IBMtest2 ($240 \times 320$, 90), CaVignal ($136 \times 200$, 113) and Blurred ($200 \times 296$, 106) where the numbers denote frame size and frame number. A comparison of the peak signal to noise ratio (PSNR) values and running times obtained by the 6 methods on all 8 video background subtraction datasets is shown in Fig. 5. It can be observed that the differences in PSNR values between our proposed TRPCA-EMR and other methods on the HallAndMonitor dataset are not significant. This is attributed to the fact that the foreground objects in this dataset move much slower than those in the other datasets. However, the accuracy of

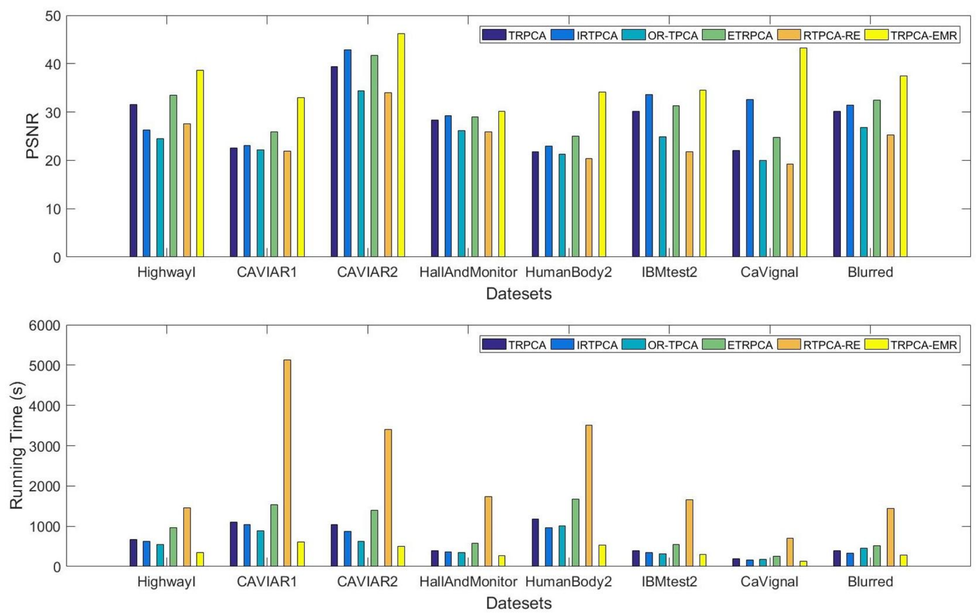


**Fig. 5** Comparison of the PSNR values (top) and running times (bottom) obtained by TRPCA, IRTPCA, OR-TPCA, ETRPCA, RTPCA-RE and TRPCA-EMR on 8 video background subtraction datasets.

(a)$165 \times 252 \times 598$ (b)$165 \times 598 \times 252$

**Fig. 6** The reshape scheme for dynamic background subtraction.

TRPCA-EMR is significantly higher than that of the methods on the other datasets. The high performance of TRPCA-EMR results from the strict low multi-rank constraints in Eq. (15), which preserve the maximum singular value group that contains most of the low-rank information. In addition, our method has the lowest running time of all the methods on all datasets, which proves that Riemannian optimization greatly reduces the computational complexity of TRPCA-EMR.

### 3.2.2 Dynamic background subtraction of video

In background subtraction, many existing algorithms work well when the backgrounds are stationary, but are not successful in a real dynamic background [38, 39]. For example, the background of a natural video may not be entirely static if it contains varying elements such

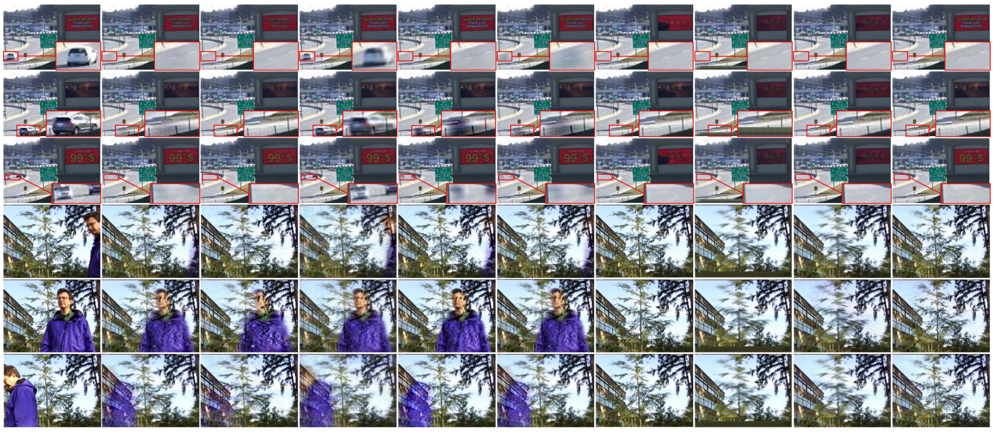


**Fig. 7** Dynamic background subtraction of video: the first column corresponds to six frames with dynamic background from the original video. The second to last columns are the separated backgrounds by TRPCA, IRTPCA, OR-TPCA, ETRPCA, RTPCA-RE, AE-NE, G-LBM, our TRPCA-FMR and TRPCA-EMR, respectively.

as swaying leaves, rainfall, and changes in illumination. In this experiment, we evaluated the proposed TRPCA-EMR on two chanllenging datasets with dynamic backgrounds [37, 40]: advertisementBoard ($165 \times 252, 598$) containing moving vehicles and periodically changing billboards, and WavingTrees ($120\times160, 287$) featuring moving individuals and swaying trees. Fig. 6 illustrates the tensor reshape strategy employed in this experiment. Taking the "advertisementBoard" dataset in Fig. 6 (a) as an example, we reconstructed it into a $165 \times 598 \times 252$ tensor shown in Fig. 6 (b). Owing to the periodic changes in advertisements, the multi-rank value of the frontal slice in the Fourier domain containing advertisement pixels is greater than 1. The tensor rank estimated by our proposed GDME method could accurately capture the differential information reflecting the dynamic and static portions of the background. To further demonstrate the effectiveness of the proposed GDME method on our TRPCA-EMR, we developed another TRPCA with fixed multi-rank (TRPCA-FMR) as follows

$$\min_{\boldsymbol{\mathcal{L}},\boldsymbol{\mathcal{E}}} \|\boldsymbol{\mathcal{E}}\|_1, \text{ s.t. } \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{E}}, \text{rank}\left(\hat{\boldsymbol{\mathcal{L}}}^{(i)}\right) = (1, ..., 1) \tag{31}$$

In Eq. (31), the tensor multi-rank is assumed to be known and each element is equal to 1, i.e., the background is static. In this experiment, in addition to the TRPCA methods, we also consider two deep learning methods AE-NE [41] and G-LBM [42] as comparative baselines, which were recently proposed as advanced dynamic background subtraction methods. It is observed from Table 3 that the deep learning-based approaches are very fast to separate the foreground, but the training stages need a large data and much time. On the other hand, the model-based TRPCA methods handle the dynamic background subtraction directly without training stage, and take more time than the data-based learning methods. It is highlighted that, among the model-based TRPCA methods, our proposed method is the fastest. TRPCA-FMR requires less running time than TRPCA-EMR because the rank is known prior without to estimation. Fig. 7 demonstrates the separation performances of different methods. It can be seen

**Table 3** Comparison of running time (s) on dynamic background subtraction datasets.

| Methods | TRPCA [15] | IRTPCA [17] | OR-TPCA [16] | ETRPCA [20] | RTPCA-RE [9] | AE-NE [41] | G-LBM [42] | TRPCA -FMR | TRPCA -EMR |
|---|---|---|---|---|---|---|---|---|---|
| advertisementBoard | 1814.1 | 1944.7 | 2751.6 | 4157.7 | 5334.2 | 68.9 | **47.1** | 1034.7 | 1297.5 |
| WavingTrees | 491.7 | 460.0 | 950.5 | 332.9 | 2135.0 | 676.1 | **12.7** | 226.0 | 266.0 |

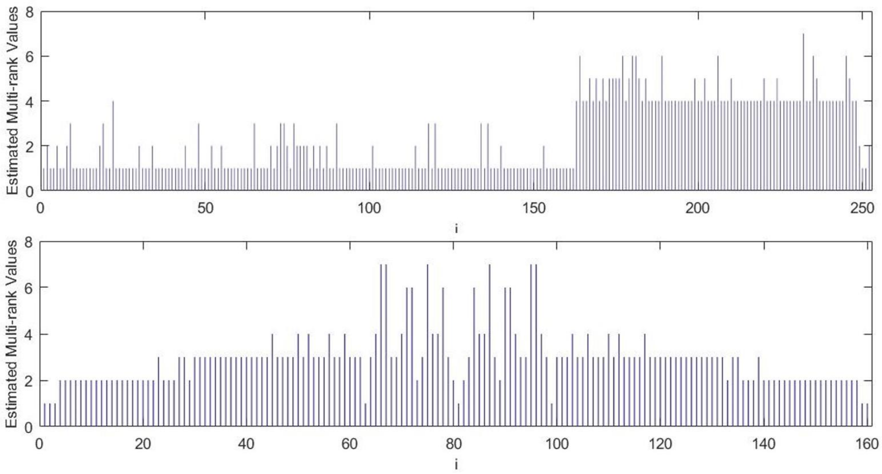


**Fig. 8** The estimated multi-rank on the advertisementBoard dataset (top) and WavingTrees (bottom).

that the deep learning methods AE-NE and G-LBM both accurately remove the foreground on both datasets, but AE-NE fails to identify the varying billboards as dynamic backgrounds when processing the advertisementBoard dataset, resulting in significant missing of the billboards. Moreover, the deep learning-based methods heavily rely on the quality and quantity of the training data. In our examples, the number of underlying datasets is insufficient to train a good deep neural network model. This leads to significant blurring of the dynamic background across both datasets for G-LBM. On the other hand, the first five TRPCA methods (Fig. 7 (b-f)) struggle to separate the foreground in most frames effectively. IRTPCA successfully separates the foreground in the advertisementBoard dataset but significantly blurs dynamically changing billboards in the background. TRPCA-FMR removes the foreground but compromises the dynamic characteristics of the background. In contrast, our proposed TRPCA-EMR effectively removed the foreground and preserved the dynamic background, owing to the accurate tensor rank estimation by GDME, which is shown in Fig. 8. In light of underlying small datasets, we only compare the performance of the model-based TRPCA methods in the following section.

### 3.2.3 Hyperspectral image denoising

Hyperspectral image denoising is a challenging task because of the variations in noise intensity across the different spectral bands. We evaluated the denoising performance of our proposed TRPCA-EMR method on the classic Indian Pines dataset [43]. The data size is $145 \times 145$ pixels with 220 bands, which are reshaped to a $145 \times 220 \times 145$ tensor. The denoising results of bands 3, 110, and 218 and the combination results of the three bands are

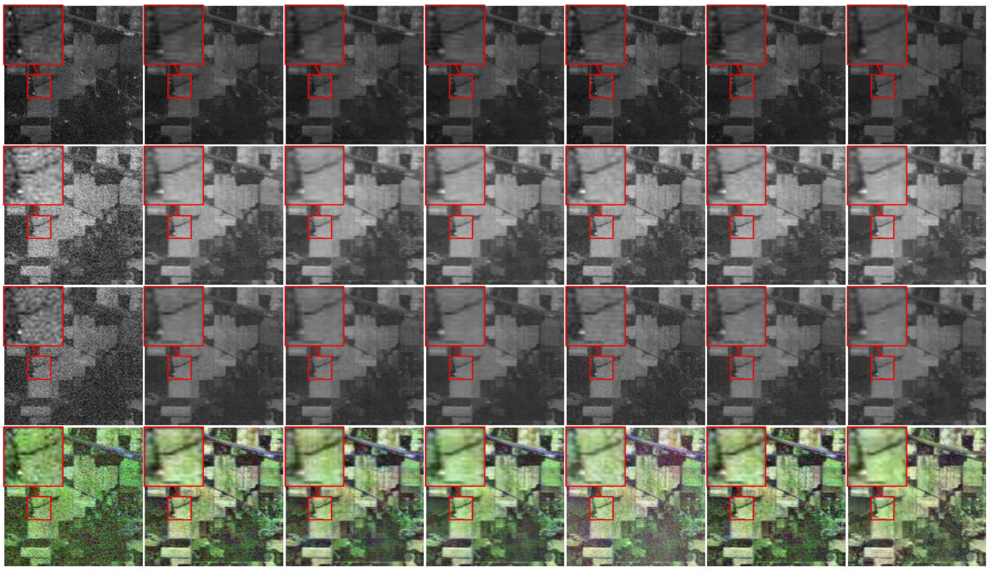


**Fig. 9** Hyperspectral image denoising: the first column corresponds to three noise bands (from top to bottom are bands 3, 110, and 218) and one RGB image (combination of band 3, 110, and 218) from the original noise hyperspectral image. The second to last columns are the denoising results of TRPCA, IRTPCA, OR-TPCA, ETRPCA, RTPCA-RE and TRPCA-EMR, respectively.

**Table 4** Comparison of the running time (s) on hyperspectral image denoising.

| Methods | TRPCA [15] | IRTPCA [17] | OR-TPCA [16] | ETRPCA [20] | RTPCA-RE [9] | TRPCA-EMR |
|---|---|---|---|---|---|---|
| Time(s) | 79.0 | 103.1 | 134.7 | 193.2 | 444.2 | **59.0** |

shown in Fig. 9 respectively. The experimental results show that our proposed TRPCA-EMR outperforms other existing denoising methods across various spectral bands. It effectively removes noise and preserves textures and edges without excessive smoothing. The running times of the different methods are shown in Table 4 and our proposed TRPCA-EMR is the fastest.

# 4 Conclusions

This paper presents a fast tensor robust principal component analysis with estimated multi-rank (TRPCA-EMR) to recover the low-rank and sparse tensor. The proposed TRPCA-EMR estimates the tenor multi-rank using modified Gershgorin disks and replaces the TNN with strict equality constraints based on the estimated tensor multi-rank. Such processing ensures the low-rank information can be reserved in each iteration. On the other hand, Riemannian optimization is introduced to project each frontal slice of a tensor in the Fourier domain onto a low multi-rank manifold space, significantly reducing the computational cost of t-SVD. The experimental results on synthetic data demonstrate that our method is superior when low-rank information is perturbed very seriously. The real data results indicate that the proposed TRPCA-EMR can more accurately separate low-rank and sparse components than five competitive TRPCA algorithms. The developed method can be applied in real-world applications, including signal processing, face recognition, color image and video data recovery, and hyperspectral data recovery.

## Author contribution

**Qile Zhu:** Conceptualization, Methodology, Validation, Software, Writing - original draft. **Shiqian Wu:** Formal analysis, Validation, Writing - review & editing. **Shun Fang:** Writing - review & editing. **Qi Wu:** Writing - review & editing. **Shoulie Xie:** Conceptualization, Methodology, Writing - review & editing. **Sos Agaian:** Supervision, writing - review & editing.

# References


[1] Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? Journal of the ACM (JACM) **58**(3), 1–37 (2011)

[2] Xu, Z., He, R., Xie, S., Wu, S.: Adaptive rank estimate in robust principal component analysis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6577–6586 (2021)

[3] Ma, C., Wan, M., Xu, Y., Ren, K., Qian, W., Chen, Q., Gu, G.: Infrared target tracking based on proximal robust principal component analysis method. Applied Intelligence **52**(3), 2785–2800 (2022)

[4] Xue, N., Deng, J., Cheng, S., Panagakis, Y., Zafeiriou, S.: Side information for face completion: a robust PCA approach. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(10), 2349–2364 (2019)

[5] Wang, P., Wang, L., Leung, H., Zhang, G.: Super-resolution mapping based on spatial-spectral correlation for spectral imagery. IEEE Transactions on Geoscience and Remote Sensing **59**(3), 2256–2268 (2021)

[6] Han, Z., Wang, Y., Zhao, Q., Meng, D., Lin, L., Tang, Y., *et al.*: A generalized model for robust tensor factorization with noise modeling by mixture of Gaussians. IEEE Transactions on Neural Networks and Learning Systems **29**(11), 5380–5393 (2018)

[7] Bahri, M., Panagakis, Y., Zafeiriou, S.: Robust Kronecker component analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **41**(10), 2365–2379 (2018)

[8] Zhou, Y., Lu, H., Cheung, Y.-M.: Probabilistic rank-one tensor analysis with concurrent regularizations. IEEE Transactions on Cybernetics **51**(7), 3496–3509 (2019)

[9] Shi, Q., Cheung, Y.-M., Lou, J.: Robust tensor SVD and recovery with rank estimation. IEEE Transactions on Cybernetics **52**(10), 10667–10682 (2021)

[10] Hillar, C.J., Lim, L.-H.: Most tensor problems are NP-hard. Journal of the ACM (JACM) **60**(6), 1–39 (2013)

[11] Zhou, P., Lu, C., Lin, Z., Zhang, C.: Tensor factorization for low-rank tensor completion. IEEE Transactions on Image Processing **27**(3), 1152–1163 (2017)

[12] Kilmer, M.E., Martin, C.D.: Factorization strategies for third-order tensors. Linear Algebra and its Applications **435**(3), 641–658 (2011)

[13] Semerci, O., Hao, N., Kilmer, M.E., Miller, E.L.: Tensor-based formulation and nuclear norm regularization for multienergy computed tomography. IEEE Transactions on Image Processing **23**(4), 1678–1693 (2014)

[14] Lu, C., Feng, J., Chen, Y., Liu, W., Lin, Z., Yan, S.: 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, pp. 5249–5257 (2016)

[15] Lu, C., Feng, J., Chen, Y., Liu, W., Lin, Z., Yan, S.: Tensor robust principal component analysis with a new tensor nuclear norm. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(4), 925–938 (2019)

[16] Zhou, P., Feng, J.: Outlier-robust tensor PCA. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2263–2271 (2017)

[17] Liu, Y., Chen, L., Zhu, C.: Improved robust tensor principal component analysis via low-rank core matrix. IEEE Journal of Selected Topics in Signal Processing **12**(6), 1378–1389 (2018)

[18] Zhang, F., Wang, H., Qin, W., Zhao, X., Wang, J.: Generalized nonconvex regularization for tensor rpca and its applications in visual inpainting. Applied Intelligence **53**(20), 23124–23146 (2023)

[19] Jiang, T.-X., Huang, T.-Z., Zhao, X.-L., Deng, L.-J.: Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm. Journal of Computational and Applied Mathematics **372**, 112680 (2020)

[20] Gao, Q., Zhang, P., Xia, W., Xie, D., Gao, X., Tao, D.: Enhanced tensor RPCA and its application. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(6), 2133–2140 (2020)

[21] Qiu, H., Wang, Y., Tang, S., Meng, D., Yao, Q.: Fast and provable nonconvex tensor RPCA. In: International Conference on Machine Learning, pp. 18211–18249 (2022)

[22] Xue, J., Zhao, Y., Liao, W., Chan, J.C.-W.: Nonconvex tensor rank minimization and its

applications to tensor recovery. Information Sciences **503**, 109–128 (2019)

[23] Imbiriba, T., Borsoi, R.A., Bermudez, J.C.M.: Low-rank tensor modeling for hyperspectral unmixing accounting for spectral variability. IEEE Transactions on Geoscience and Remote Sensing **58**(3), 1833–1842 (2020)

[24] Yokota, T., Lee, N., Cichocki, A.: Robust multilinear tensor rank estimation using higher order singular value decomposition and information criteria. IEEE Transactions on Signal Processing **65**(5), 1196–1206 (2017)

[25] Vandereycken, B.: Low-rank matrix completion by Riemannian optimization. SIAM Journal on Optimization **23**(2), 1214–1236 (2013)

[26] Wei, K., Cai, J.-F., Chan, T.F., Leung, S.: Guarantees of Riemannian optimization for low rank matrix recovery. SIAM Journal on Matrix Analysis and Applications **37**(3), 1198–1222 (2016)

[27] Cai, J.-F., Li, J., Xia, D.: Generalized low-rank plus sparse tensor estimation by fast Riemannian optimization. Journal of the American Statistical Association **118**(544), 2588–2604 (2023)

[28] Luo, Y., Zhang, A.R.: Low-rank tensor estimation via Riemannian Gauss-Newton: statistical optimality and second-order convergence. Journal of Machine Learning Research **24**(1), 18274–18321 (2024)

[29] Fang, S., Xu, Z., Wu, S., Xie, S.: Efficient robust principal component analysis via block Krylov iteration and cur decomposition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1348–1357 (2023)

[30] Wu, H.-T., Yang, J.-F., Chen, F.-K.: Source number estimators using transformed gerschgorin radii. IEEE transactions on signal processing **43**(6), 1325–1333 (1995)

[31] Zhang, Z., Ely, G., Aeron, S., Hao, N., Kilmer, M.: Novel methods for multilinear data completion and de-noising based on tensor-svd. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3842–3849 (2014)

[32] Osher, S., Mao, Y., Dong, B., Yin, W.: Fast linearized Bregman iteration for compressive sensing and sparse denoising. Communications in Mathematical Sciences **8**(1), 93–111 (2010)

[33] Hale, E.T., Yin, W., Zhang, Y.: Fixed-point continuation for $l_1$-minimization: Methodology and convergence. SIAM Journal on Optimization **19**(3), 1107–1130 (2008)

[34] Lu, C., Feng, J., Yan, S., Lin, Z.: A unified alternating direction method of multipliers by majorization minimization. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(3), 527–541 (2017)

[35] Maddalena, L., Petrosino, A.: Towards benchmarking scene background initialization.

In: New Trends in Image Analysis and Processing–ICIAP 2015 Workshops: ICIAP 2015 International Workshops, BioFor, CTMR, RHEUMA, ISCA, MADiMa, SBMI, and QoEM, Genoa, Italy, September 7-8, 2015, Proceedings 18, pp. 469–476 (2015)

[36] Bouwmans, T., Maddalena, L., Petrosino, A.: Scene background initialization: A taxonomy. Pattern Recognition Letters **96**, 3–11 (2017)

[37] Jodoin, P.-M., Maddalena, L., Petrosino, A., Wang, Y.: Extensive benchmark and survey of modeling methods for scene background initialization. IEEE Transactions on Image Processing **26**(11), 5244–5256 (2017)

[38] Cao, X., Yang, L., Guo, X.: Total variation regularized RPCA for irregularly moving object detection under dynamic background. IEEE Transactions on Cybernetics **46**(4), 1014–1027 (2015)

[39] Sanches, S.R., Oliveira, C., Sementille, A.C., Freire, V.: Challenging situations for background subtraction algorithms. Applied Intelligence **49**, 1771–1784 (2019)

[40] Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: Proceedings of the Seventh IEEE International Conference on Computer Vision, pp. 255–261 (1999)

[41] Sauvalle, B., La Fortelle, A.: Autoencoder-based background reconstruction and foreground segmentation with background noise estimation. In: IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3243–3254 (2023)

[42] Rezaei, B., Farnoosh, A., Ostadabbas, S.: G-LBM: Generative low-dimensional background model estimation from video sequences. In: European Conference on Computer Vision, pp. 293–310 (2020)

[43] Baumgardner, M.F., Biehl, L.L., Landgrebe, D.A.: 220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3. Purdue University Research Repository **10**(7), 991 (2015)