IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 26 December 2023, accepted 5 January 2024, date of publication 11 January 2024, date of current version 29 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3352597

RESEARCH ARTICLE

Orthogonal Tensor Recovery Based on Non-Convex Regularization and Rank Estimation

XIXIANG CHEN[1], JINGJING ZHENG[2], LI ZHAO[1], WEI JIANG[1], AND XIAOQIN ZHANG[1]

[1]Key Laboratory of Intelligent Informatics for Safety and Emergency of Zhejiang Province, Wenzhou University, Wenzhou 325035, China
[2]Department of Mathematics, The University of British Columbia, Vancouver, BC V6T 1Z4, Canada

Corresponding author: Li Zhao (lizhao@wzu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant U2033210 and Grant 62101387 and in part by the Zhejiang Provincial Natural Science Foundation under Grant LDT23F02024F02.

ABSTRACT In this paper, a method for orthogonal tensor recovery based on non-convex regularization and rank estimation (OTRN-RE) is proposed, which aims to accurately recover the low-rank and sparse components of the tensor. Specifically, a new low-rank tensor decomposition algorithm is designed, which can efficiently establish the equivalence between the rank of a large tensor before decomposition and the rank of the coefficient tensor after decomposition. The large tensor is decomposed into a small standard orthogonal tensor and another coefficient tensor, and a generalized non-convex regularization is used to inscribe the low rank of the coefficient tensor. Meanwhile, a new rank estimation strategy is developed to dynamically adjust the size of small orthogonal tensors and coefficient tensors. Experimental results on image denoising and salient object detection tasks confirm the state-of-the-art performance of the proposed method in terms of denoising capability and computational speed.

INDEX TERMS Orthogonal tensor recovery, non-convex regularization, tensor decomposition, low-rank recovery.

I. INTRODUCTION

With the rapid development of modern information technology, a large amount of multi-dimensional data have been generated, including hyperspectral data, color images, and video data. These real-world datasets are stored in multi-dimensional arrays known as tensors, enabling efficient computation and manipulation of the data. However, in practice, the tensor is often affected by many degradation factors such as noise pollution [1], [2], missing observations [3], [4], partial occlusion [5], [6], and misalignment [7]. Fortunately, recent studies [8], [9], [10], [11] have shown that high-dimensional tensor data such as image and video collections usually have low-rank or approximately low-rank properties and are widely used in computer vision [8], [9], [10], collaborative filtering [12], and data mining [13]. Consequently, the challenge of more accurately recovering low-rank tensor data from high-dimensional tensor data affected by various degradation factors is garnering increasing attention.

Recently, Kilmer et al. [14] introduced the concept of tubal rank that is based on the t-product and t-SVD [9], [15], [16], [17]. The tubal rank is defined as the number of nonzero singular tubes of the singular value tensor. To develop a convex surrogate for the tubal rank, Semerci et al. [18] formulated a new tensor nuclear norm (TNN). Zhang et al. [19] applied TNN to calculate the tubal rank derived from t-SVD and exploited it to obtain an approximation of the tensor rank. Later, Lu et al. [9] defined a new tensor rank (tensor average rank) based on TNN and established a tensor robust principal component analysis (TRPCA) model, which guarantees the corresponding TRPCA recovery and performs well in many tasks. To distinguish it from other tensor nuclear norms, the t-product-induced TNN proposed by Lu et al. is referred to as t-TNN in this paper.

In recent years, t-TNN has become a popular solution to the tensor recovery problem, but it has some limitations. Firstly, when handling large-scale tensor data, the computational

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy.

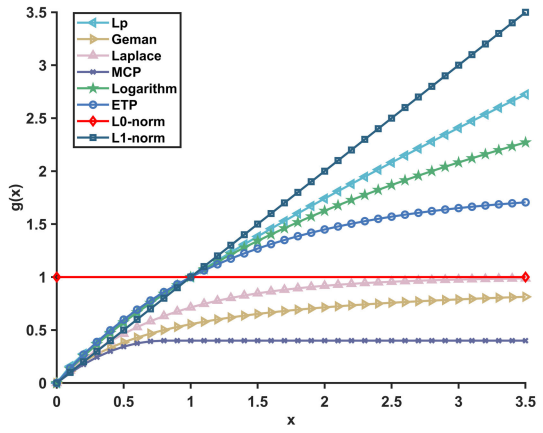


FIGURE 1. Some surrogate functions.

TABLE 1. Some surrogate functions of ℓ_0.

Name	$g(x), x \geq 0, \lambda > 0$
ℓ_p [22]	$\lambda x^p, 0 < p < 1$
Geman [23]	$\frac{\lambda x}{x+\gamma}$
Laplace [24]	$\lambda\left(1-\exp\left(-\frac{x}{\gamma}\right)\right)$
MCP [25]	$\lambda x - \frac{x^2}{2\gamma}$, if $x < \lambda\gamma$ $\frac{1}{2}\gamma\lambda^2$, if $x \geq \lambda\gamma$
Logarithm [26]	$\frac{\lambda}{\log(\gamma+1)}\log(\gamma x+1)$
ETP [27]	$\frac{\lambda}{1-\exp(-\gamma)}(1-\exp(-\gamma x))$

complexity of t-TNN increases dramatically. For example, solving the TRPCA problem with t-TNN leads to a computational complexity of $\mathcal{O}\left(n_1n_2n_3\log n_3 + n_{(1)}n_{(2)}^2n_3\right)$ per iteration. To solve this complex problem, Zhou et al. [20] introduced a tensor decomposition method (TCTF) based on the definition of tensor multi-rank and tubal rank. This method decomposes a large tensor into the product of two smaller tensors, thus reducing the computational cost of each iteration to $\mathcal{O}\left(r\left(n_1+n_2\right)n_3\log n_3 + rn_1n_2n_3\right)$. Nevertheless, it is essential to emphasize that TCTF primarily concentrates on tensor decomposition and abstains from employing the nuclear norm for rank approximation, potentially resulting in suboptimal tensor recovery performance. On the other hand, the low-rank matrix recovery algorithm proposed in [21] takes full advantage of the unitary invariance of the decomposed matrix, achieving a noteworthy improvement in computational speed with a complexity of $\mathcal{O}\left(r^2(n_1+n_2)\right)$. Nevertheless, with the increasing scale of data, the processing of data represented in tensor form becomes more crucial. Therefore, in practical applications, it is necessary to transform algorithms designed for matrix processing into algorithms capable of handling tensors.

Meanwhile, due to the loose approximation of the t-TNN tensor tubal rank, there is still a considerable gap with the minimization of the tensor tubal rank [28]. The latest comprehensive survey of sparse regularization [29] provides a systematic introduction to the fundamentals of general, convex, and non-convex optimization. Several popular non-convex penalty functions are defined and summarized in TABLE 1, and their visualizations are presented in Fig. 1. It can be observed that as the value of x becomes smaller, it tends to approach the ℓ_0 norm, and for a larger value of x, it tends to approach the ℓ_1 norm. Moreover, in certain cases, these penalty functions can contribute to a better rank approximation than ℓ_1 or ℓ_0 norms. Additionally, in [30], it is shown that non-convex regularization has a positive effect on smoothing convex images.

To overcome the above limitations, this paper proposes OTRN-RE, and the contributions of this work are threefold.

1) A new orthogonal tensor decomposition model is proposed for low-rank subspace learning. Different from the traditional tensor low-rank representation model, the tensor decomposition method we introduce enables the learning of low-rank subspaces more efficiently. The use of non-convex regularization to constrain the coefficient tensor in the model helps to recover the low-rank subspace more accurately.
2) A new rank estimation strategy has been designed, capable of seeking the most compact standard orthogonal small tensor that best represents the low-rank target tensor, thereby enhancing tensor recovery capabilities.
3) Extensive comparative experiments demonstrate that, when compared with other tensor recovery methods, our approach achieves relatively better performance in image denoising and salient object detection tasks, particularly in terms of computational speed.

The rest of this paper is organized as follows. Section II provides some notations. Section III gives the details of our proposed orthogonal tensor decomposition algorithm based on non-convex regularization and rank estimation. Section IV presents numerical experiments performed on real data and performs convergence analysis and parametric analysis. Finally, our work is summarized in Section V.

II. NOTATIONS AND RELATED WORK

A. NOTATIONS

This section introduces some of the notations used in this paper. For brevity, the notations are listed in TABLE 2. The Discrete Fourier transform (DFT) and the Inverse Discrete Fourier Transform (inverse DFT) play a crucial role in the tensor-tensor product (t-product). In addition, unfold $(\cdot)$, fold $(\cdot)$, bcirc $(\cdot)$, and bdiag $(\cdot)$ are defined as

$$\text{unfold}(\mathcal{A}) = \begin{pmatrix} \boldsymbol{A}_{(1)} \\ \boldsymbol{A}_{(2)} \\ \vdots \\ \boldsymbol{A}_{(n_3)} \end{pmatrix} \in \mathbb{R}^{n_1n_3\times n_2},$$

$$\text{fold}(\text{unfold}(\mathcal{A})) = \mathcal{A},$$

$$\text{bcirc}(\mathcal{A}) = \begin{pmatrix} \boldsymbol{A}_{(1)} & \boldsymbol{A}_{(n_3)} & \cdots & \boldsymbol{A}_{(2)} \\ \boldsymbol{A}_{(2)} & \boldsymbol{A}_{(1)} & \cdots & \boldsymbol{A}_{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{A}_{(n_3)} & \boldsymbol{A}_{(n_3-1)} & \cdots & \boldsymbol{A}_{(1)} \end{pmatrix},$$

$$\text{bdiag}(\mathcal{A}) = \begin{pmatrix} \boldsymbol{A}_{(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{A}_{(2)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{A}_{(n_3)} \end{pmatrix}.$$

TABLE 2. Some notations.

Notations	Descriptions	Notations	Descriptions
$\mathbb{R}$	real field	$\mathbb{C}$	complex field
t	scalar	$\boldsymbol{t}$	vector
$\boldsymbol{T}$	matrix	$\mathcal{T}$	tensor
$\mathcal{T}$	true tensor	$\mathcal{T}^{rec}$	recovered tensor
$\mathcal{T}(i,j,k)$	(i,j,k)-th entry	$\|\mathcal{T}\|_1$	$\sum_{ijk}\|\mathcal{T}_{ijk}\|$
$\mathcal{T}(i,j,:)$	(i,j)-th tube	$\|\mathcal{T}\|_F$	$\sqrt{\sum_{ijk}\mathcal{T}_{ijk}^2}$
$\mathcal{T}_{(k)}$	k-th frontal slice	$\|\mathcal{T}\|_\infty$	$\max_{ijk}\|\mathcal{T}_{ijk}\|$
$\langle\mathcal{A},\mathcal{B}\rangle$	$\sum_{i=1}^{n_3}\langle\boldsymbol{A}^{(i)},\boldsymbol{B}^{(i)}\rangle$	fft(ifft)	DFT(inverse DFT)
$\overline{\mathcal{A}}$	fft$(\mathcal{A},[\,],3)$	$\mathcal{A}$	ifft$(\overline{\mathcal{A}},[\,],3)$

Definition 1 (t-product). *Let* $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$ *and* $\mathcal{B} \in \mathbb{R}^{n_2\times l\times n_3}$. *The t-product* $\mathcal{A} * \mathcal{B} = \mathcal{C} \in \mathbb{R}^{n_1\times l\times n_3}$.

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

B. RELATED WORK

A key issue in low-rank tensor recovery is how to properly define the rank of the tensor. There are three widely recognized types of tensor ranks: CANDECOMP/PARAFAC (CP) rank [11], [31], Tucker rank [8], [10], [32], [33], [34], and Tensor tubal rank [9], [15], [16], [17].

1) CP RANK

The CP rank (CANDECOMP/PARAFAC rank) of a tensor is the minimum number of rank-1 tensors that need to approximate the original tensor, i.e.

$$\text{rank}_{\text{cp}}(\mathcal{X}) = \min\left\{r \mid \mathcal{X} = \sum_{i=1}^{r} \boldsymbol{a}_1^{(i)} \otimes \boldsymbol{a}_2^{(i)} \otimes \cdots \otimes \boldsymbol{a}_k^{(i)}\right\}, \quad (1)$$

where the symbol $\otimes$ denotes the outer product, and $\boldsymbol{a}_j^{(i)} \in \mathbb{R}^{n_j}(\forall(i,j))$ is a vector. However, the rank minimization problem for CP decomposition is usually considered an NP-hard problem [35].

2) TUCKER RANK

The Tucker rank of a tensor is defined based on the tensor's unfolding matrices. For a k-way tensor $\mathcal{X}$, the Tucker rank is a vector defined as

$$\text{rank}_{\text{tc}}(\mathcal{X}) = \left(\text{rank}\left(\boldsymbol{X}_{(1)}\right), \cdots, \text{rank}\left(\boldsymbol{X}_{(i)}\right), \cdots, \text{rank}\left(\boldsymbol{X}_{(k)}\right)\right), \quad (2)$$

where $\mathcal{X}_{(i)}$ is its mode-i matricization. Since the sum of the ranks of different unfolding matrices can lead to an NP-hard optimization problem, [36] proposed to use the sum of nuclear norms (SNN) to improve solution efficiency. SNN is denoted as $\sum_{i=1}^{k}\left\|\mathcal{X}_{(i)}\right\|_*$. However, the SNN is not a convex envelope of $\sum_{i=1}^{k}\text{rank}\left(\mathcal{X}_{(i)}\right)$ [37] but its overlapping regularization. To address these issues, a new tensor nuclear norm based on the t-product is proposed, which is discussed in detail below.

3) TENSOR TUBAL RANK

Recently, tensor average rank based on t-product has been proposed. Lu et al. [9] established a tensor robust principal component analysis (TRPCA) model that aims to recover the low tensor average rank component $\mathcal{L}_0$ and sparse component $\mathcal{E}_0$ from noisy observations $\mathcal{X} = \mathcal{L}_0 + \mathcal{E}_0 \in \mathbb{R}^{n_1\times n_2\times n_3}$ by convex optimization

$$\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_* + \lambda\|\mathcal{E}\|_1, \text{ s.t. } \mathcal{X} = \mathcal{L} + \mathcal{E}, \quad (3)$$

where $\|\mathcal{L}\|_*$ denotes the t-product-induced TNN (t-TNN), $\|\mathcal{E}\|_1$ denotes the ℓ_1-norm (the sum of the absolute values of all the entries in $\mathcal{E}$), and λ is a balancing parameter. Based on the TRPCA model, Zheng et al. [38] found that the slice permutations of tensors have a significant influence on the results of tensor recovery. They developed an algorithm that can more accurately exploit the low-dimensional subspace structures to improve the accuracy of recovery.

Although t-TNN performs well on tensor recovery, recently, some studies [28], [34], [39], [40] have shown that the non-convex tensor tubal rank surrogate method can achieve better approximation than t-TNN. Notably, the generalized tensor singular value thresholding (GTSVT) method, as proposed by Zhang et al. [34], achieves well rank approximation, improving the accuracy of tensor recovery. Additionally, researchers propose various tensor decomposition methods to improve algorithm speed, such as Zhou et al. [20] factorization method using Fourier domain tensor nuclear norm for third-order tensor completion.

The rank k of a decomposed tensor dictates its dimension and impacts algorithmic complexity. To enhance computational efficiency and improve tensor recovery, a dedicated rank estimation algorithm is essential. For instance, Xu et al. [41] proposed the TMac algorithm, offering two rank estimation methods. A choice between rank-increasing or rank-decreasing algorithms must be made beforehand. Shi et al. [42] introduced RTPCA-RE, a robust tensor PCA algorithm that estimates the rank from corrupted observations, subsequently utilizing it for improved low-rank recovery. However, its computational speed is relatively slow, emphasizing the urgent need to seek a faster and more efficient rank estimation method.

III. PROPOSED ALGORITHM SCHEME

In this section, the proposed method OTRN-RE is introduced in detail, and a new rank estimation strategy is given to improve the tensor recovery.

A. ORTHOGONAL TENSOR DECOMPOSITION ALGORITHM BASED ON NON-CONVEX OPTIMIZATION

To handle large-scale tensor data, this paper refers to the decomposition method of TCTF [20]: a large-size tensor

is decomposed into the product of two small-size tensors. Based on TCTF, this paper performs forced orthogonality on one of the decomposed small tensors. Then, the large-size tensor $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is decomposed into a k-dimensional orthogonal small tensor $\mathcal{U} \in \mathbb{R}^{n_1 \times k \times n_3}$ (i.e., $\mathcal{U}^T * \mathcal{U} = \mathcal{I}$) and a coefficient small tensor $\mathcal{V} \in \mathbb{R}^{k \times n_2 \times n_3}$. The details of the decomposition are illustrated in Fig. 2, and the formula is as follows:

$$\mathcal{L} = \mathcal{U} * \mathcal{V}, \text{ s.t. } \mathcal{U}^T * \mathcal{U} = \mathcal{I}. \tag{4}$$

Due to the unitary invariance of the nuclear norm [43], we have

$$\|\mathcal{L}\|_* = \|\mathcal{U} * \mathcal{V}\|_* = \|\mathcal{V}\|_*. \tag{5}$$

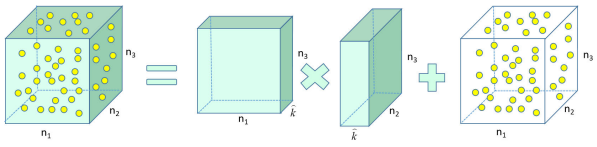


FIGURE 2. A description of the observation model in OTRN-RE. The size of the orthogonal small tensor is $n_1 \times \hat{k} \times n_3$, and the size of the coefficient small tensor is $\hat{k} \times n_2 \times n_3$, where $\hat{k}$ represents the optimal decomposition dimension k.

Based on the conclusion derived from (5), this paper explores the resolution of the tubal rank minimization problem for the coefficient small tensor $\mathcal{V}$. Considering the limitation of the ℓ_0 norm in practical applications, this paper introduces a variety of non-convex surrogates, and the minimization problem of (3) becomes

$$\begin{aligned} &\min_{\mathcal{U},\mathcal{V},\mathcal{E}} \|\mathcal{V}\|_*^g + \lambda \|\mathcal{E}\|_g, \\ &\text{s.t.} \quad \mathcal{X} = \mathcal{U} * \mathcal{V} + \mathcal{E}, \\ &\qquad \mathcal{U}^T * \mathcal{U} = \mathcal{I}, \end{aligned} \tag{6}$$

where $\|\mathcal{V}\|_*^g = \sum_{i=1}^{r} g(\sigma_i(\mathcal{V}))$, $r = min(n_1, n_2)$, $\|\mathcal{E}\|_g = \sum_{i_1,i_2,i_3} g(|\mathcal{E}(i_1, i_2, i_3)|)$, and $g : \mathbb{R}^+ \longrightarrow \mathbb{R}^+$ is an increasing function. Note that all the surrogate functions of ℓ_0 listed in Fig. 1 satisfy this condition. The ℓ_1 norm as a special case also holds here, and it's worth mentioning that the low-rank and sparse functions g can be different. Meanwhile, the computational complexity is greatly reduced as the large tensor tubal rank minimization problem is equivalent to the coefficient small tensor tubal rank minimization problem.

The k is introduced to denote the size of the decomposition tensor, and it can be used as a controllable upper bound on the $\mathcal{L}$ rank because

$$\text{rank}(\mathcal{L}) = \text{rank}(\mathcal{U} * \mathcal{V}) \leq \text{rank}(\mathcal{U}) = \hat{k}, \tag{7}$$

where $\hat{k}$ denotes the best choice of k, which also represents the optimal rank. In this paper, a new rank estimation algorithm is proposed, which can quickly search for the optimal rank $\hat{k}$. See section III-C for details.

This paper is dedicated to the analysis of three-dimensional tensors. Notably, a recent work [44] innovatively used t-SVD to accomplish the higher-order tensor complementation problem with missing values. In the future, we plan to further exploit (6) to solve higher-dimensional tensor recovery challenges.

B. ALTERNATING DIRECTION METHOD OF MULTIPLIERS (ADMM) ALGORITHM

In this section, the Alternating Direction Method (ADMM) is employed to solve (6), and the following augmented Lagrange function is obtained:

$$\begin{aligned} \mathcal{L}_\mu(\mathcal{U}, \mathcal{V}, \mathcal{E}, \mathcal{Y}) = &\|\mathcal{V}\|_*^g + \lambda \|\mathcal{E}\|_g + \langle \mathcal{X} - \mathcal{U} * \mathcal{V} - \mathcal{E}, \mathcal{Y} \rangle \\ &+ \frac{\mu}{2} \|\mathcal{X} - \mathcal{U} * \mathcal{V} - \mathcal{E}\|_F^2, \end{aligned} \tag{8}$$

where $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the Lagrangian multiplier tensor. According to the framework of ADMM, (8) can be solved by the following iterations.

Step 1 Set a set $\mathbb{P}$ composed of $\mathcal{U}$, and the tensors in the set satisfy $\mathcal{U}^T * \mathcal{U} = \mathcal{I}$. Given $\mathcal{V}^k$, $\mathcal{E}^k$ and $\mathcal{Y}^k$, update $\mathcal{U}^{k+1}$ by

$$\begin{aligned} \mathcal{U}^{k+1} &= \arg\min_{\mathcal{U} \in \mathbb{P}} \mathcal{L}_\mu \left(\mathcal{U}, \mathcal{V}^k, \mathcal{E}^k, \mathcal{Y}^k \right) \\ &= \arg\min_{\mathcal{U} \in \mathbb{P}} \frac{\mu^k}{2} \left\| \mathcal{X} - \mathcal{U} * \mathcal{V}^k - \mathcal{E}^k + \frac{\mathcal{Y}^k}{\mu^k} \right\|_F^2. \end{aligned} \tag{9}$$

Assume the following singular value decomposition

$$\left(\mathcal{X} - \mathcal{E}^k + \frac{\mathcal{Y}^k}{\mu^k} \right) * (\mathcal{V}^k)^T = \mathcal{U}_S * \mathcal{D}_S * \mathcal{V}_S^T. \tag{10}$$

Here, it is assumed that $\mathcal{U}_S \in \mathbb{R}^{n_1 \times k \times n_3}$, $\mathcal{D}_S$, and $\mathcal{V}_S \in \mathbb{R}^{k \times k \times n_3}$, where $k \leq n_1$ is the number of principal components. According to the Reduced Rank Procrustes Theorem (RRPT) [45], the solution to (9) becomes $\mathcal{U}^{k+1} = \mathcal{U}_S * \mathcal{V}_S^T$.

Step 2 Given $\mathcal{U}^{k+1}$, $\mathcal{E}^k$ and $\mathcal{Y}^k$, update $\mathcal{V}^{k+1}$ by

$$\begin{aligned} \mathcal{V}^{k+1} &= \arg\min \mathcal{L}_\mu \left(\mathcal{U}^{k+1}, \mathcal{V}, \mathcal{E}^k, \mathcal{Y}^k \right) \\ &= \arg\min \frac{\mu^k}{2} \left\| \mathcal{X} - \mathcal{U}^{k+1} * \mathcal{V} - \mathcal{E}^k + \frac{\mathcal{Y}^k}{\mu^k} \right\|_F^2 + \|\mathcal{V}\|_*^g \\ &= \arg\min \frac{\mu^k}{2} \left\| \mathcal{V} - (\mathcal{U}^{k+1})^T * (\mathcal{X} - \mathcal{E}^k + \frac{\mathcal{Y}^k}{\mu^k}) \right\|_F^2 \\ &\quad + \|\mathcal{V}\|_*^g. \end{aligned} \tag{11}$$

To make the problem (11) concise, this paper uses η to represent $\frac{1}{\mu^k}$ and $\mathcal{H}$ to represent $(\mathcal{U}^{k+1})^T * (\mathcal{X} - \mathcal{E}^k + \frac{\mathcal{Y}^k}{\mu^k})$. Then, the minimization problem (11) becomes:

$$\mathcal{V}^{k+1} = \arg\min \frac{1}{2} \|\mathcal{H} - \mathcal{V}\|_F^2 + \eta \|\mathcal{V}\|_*^g. \tag{12}$$

Solving this subproblem is one of the difficulties in this paper, and it is solved here by using the GTSVT proposed by Zhang et al. [34].

Step 3 Given $\mathcal{U}^{k+1}$, $\mathcal{V}^{k+1}$ and $\mathcal{Y}^k$, update $\mathcal{E}^{k+1}$ by

$$\begin{aligned}\mathcal{E}^{k+1} &= \arg\min \mathcal{L}_\mu \left(\mathcal{U}^{k+1}, \mathcal{V}^{k+1}, \mathcal{E}, \mathcal{Y}^k\right) \\ &= \mathcal{T}_g\{\mathcal{X} - \mathcal{U}^{k+1} * \mathcal{V}^{k+1} + \frac{\mathcal{Y}^k}{\mu^k}, \frac{\lambda}{\mu^k}\}. \end{aligned} \tag{13}$$

Step 4 Given $\mathcal{U}^{k+1}$, $\mathcal{V}^{k+1}$ and $\mathcal{E}^{k+1}$, update $\mathcal{Y}^{k+1}$ and μ^{k+1} by

$$\mathcal{Y}^{k+1} = \mathcal{Y}^k + \mu^k(\mathcal{X} - \mathcal{U}^{k+1}\mathcal{V}^{k+1} - \mathcal{E}^{k+1}), \tag{14}$$

$$\mu^{k+1} = \min\left(\rho\mu^k, \mu_{\max}\right). \tag{15}$$

where, $\rho = 1.3$ represents the ADMM algorithm's growth step size, and $\mu_{\max} = 1 \times 10^{10}$ is a constant parameter.

C. RANK-ESTIMATION

The optimal decomposition dimension, denoted as $\hat{k}$, plays a crucial role in low-rank tensor decomposition because it represents the optimal rank and can greatly improve tensor recovery. Furthermore, given that our algorithm has a complexity of $\mathcal{O}\left(\hat{k}\,(n_1+n_2)\,n_3\log n_3 + \hat{k}n_1n_2n_3\right)$, it is imperative to develop a method that is both faster and more accurate for estimating the tensor rank. In OTRN-RE, the rank-increasing algorithm is combined with the rank-decreasing algorithm to dynamically adjust the value of k. Specifically, our rank estimation algorithm determines whether to update the rank k or not in the iterative process by judging specific conditions. If ***condition*1** is satisfied, the rank-decreasing algorithm is applied; if ***condition*2** is satisfied, the rank-increasing algorithm is applied; otherwise, continue updating until convergence and determine the best rank $\hat{k}$. The algorithm is described in detail below.

1) Rank-decreasing scheme execution ***condition*1**:
   - The rank $k_{(n)}$ of the low-rank tensor $\mathcal{L}_{(n)}$ is larger than the lower bound of the rank estimate.
   - For the coefficient tensor $\mathcal{V}$. Calculate the singular values of $\mathcal{V}_{(n)}^T\mathcal{V}_{(n)}$ first to obtain the rank of the low-rank tensor $\mathcal{L}$ under different slices and sort by $\lambda_{(n)}^1 \geq \lambda_{(n)}^2 \geq \ldots \geq \lambda_{(n)}^{k_{(n)}}$. Then, calculate the quotients $\bar{\lambda}_{(n)}^i = \lambda_{(n)}^i/\lambda_{(n)}^{i+1}, i = 1, \ldots, k_{(n)} - 1$. Suppose

$$\hat{k}_{(n)} = \underset{1\leq i\leq k_{(n)}-1}{\operatorname{argmax}} \bar{\lambda}_{(n)}^i,$$

   and the following conditions are satisfied:

$$\text{gap}_{(n)} = \frac{(k_{(n)}-1)\,\bar{\lambda}_{\hat{k}_{(n)}}}{\sum_{i\neq\hat{k}_{(n)}}\bar{\lambda}_{(n)}^i} \geq 10.$$

   If ***condition*1** is true, then execute:
   - Reduce $k_{(n)}$ to $\hat{k}_{(n)}$. Then, assume that the SVD of $\mathcal{U}_{(n)}\mathcal{V}_{(n)}$ is $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. Update $\mathcal{U}_{(n)}$ to $\boldsymbol{U}_{(\hat{k}_{(n)})}$ and $\mathcal{V}_{(n)}$ to $\boldsymbol{\Sigma}_{(\hat{k}_{(n)})}(\boldsymbol{V}_{(\hat{k}_{(n)})})^T$, where $\boldsymbol{U}_{(\hat{k}_{(n)})}$ is a submatrix of $\boldsymbol{U}$ containing $\hat{k}_{(n)}$ columns corresponding to the largest $\hat{k}_{(n)}$ singular values, and $\boldsymbol{\Sigma}_{(\hat{k}_{(n)})}$ and $\boldsymbol{V}_{(\hat{k}_{(n)})}$ are obtained accordingly.

2) Rank-increasing scheme execution ***condition*2**:
   - The rank $k_{(n)}$ of the low-rank tensor $\mathcal{L}_{(n)}$ is smaller than the upper bound of rank estimate.
   - The relative error is satisfied as follows [41]:

$$tol = \frac{\|\mathcal{L}^{rec} - \mathcal{L}\|_F}{\|\mathcal{L}\|_F} \leq 10^{-2},$$

   where $\mathcal{L}^{\text{rec}}$ represents the recovered tensor.

   If ***condition*2** is true, then execute:
   - Increase $k_{(n)}$ to $\min\left(k_{(n)} + \Delta k_{(n)}, k_{(n)}^{\max}\right)$. Here, $\Delta k_{(n)}$ is a given positive integer, and $k_{(n)}^{\max}$ is an upper bound of the rank estimate. Augment $\boldsymbol{U}_{(n)} \leftarrow [\boldsymbol{U}_{(n)}, \hat{\boldsymbol{Q}}]$ where $\hat{\boldsymbol{Q}}$ is a randomly generated orthogonal column of $\Delta k_{(n)}$ and is orthogonal to $\boldsymbol{U}_{(n)}$. Then, update $\boldsymbol{V}_{(n)}^{k+1} \leftarrow \left[\boldsymbol{V}_{(n)}^{k+1}, \boldsymbol{0}\right]$, where $\boldsymbol{0}$ is an $\boldsymbol{I}_{(n)} \times \Delta k_{(n)}$ zero matrix.[1]

The rank-decreasing scheme is efficient in identifying the low-rank tensor as the gap is usually large and easy to identify, and only one rank adjustment is required to determine the true rank. However, for an approximate low-rank tensor, the existence of a large gap is not guaranteed, and if the gap is small, the rank can be easily overestimated. Therefore, involving a rank descent scheme and a rank ascent scheme in our rank estimation scheme will contribute to fast convergence in successive update iterations (see Section IV-D for convergence analysis). The full ALM algorithm for OTRN-RE is presented in Algorithm 1.

Algorithm 1 OTRN-RE

Input: Data tensor $\mathcal{X}$, ρ, λ, p, tol, initialize k^0;
for k=0,1, … **do**
  $\mathcal{U}^{k+1} = \text{RRPT}\left\{(\mathcal{X} - \mathcal{E}^k + \mathcal{Y}^k/\mu^k)(\mathcal{V}^k)^T\right\}$
  $\mathcal{V}^{k+1} = \text{GTSTV}_g\left\{(\mathcal{U}^{k+1})^T(\mathcal{X} - \mathcal{E}^k + \frac{\mathcal{Y}^k}{\mu^k})\right\}$
  $\mathcal{E}^{k+1} = \mathcal{T}_g\{\mathcal{X} - \mathcal{U}^{k+1} * \mathcal{V}^{k+1} + \frac{\mathcal{Y}^k}{\mu^k}, \frac{\lambda}{\mu^k}\}$
  $\mathcal{Y}^{k+1} = \mathcal{Y}^k + \mu^k(\mathcal{X} - \mathcal{U}^{k+1} * \mathcal{V}^{k+1} - \mathcal{E}^{k+1})$
  $\mu^{k+1} = \min\left(\rho\mu^k, \mu_{\max}\right)$
  if *stopping criterion tol* $\leq$ *1.00E-05* **then**
    Break;
  end
  for n=1,… n_3 **do**
    if ***condition*1** *satisfied* **then**
      Rank-decreasing;
    end
    if ***condition*2** *satisfied* **then**
      Rank-increasing;
    end
  end
end

[1] Since the variables are updated in the order of $\boldsymbol{U}$ and $\boldsymbol{V}$, appending any matrix of an appropriate size after $\boldsymbol{U}$ and $\boldsymbol{V}$ does not affect the result.

TABLE 3. The parameter settings for algorithms RTPCA-RE, TRPCA, SNN, ETRPCA, WSTNN, and OTRN-RE.

algorithm	ETRPCA [40]	RTPCA-RE [42]	TRPCA [9]	WSTNN [46]
μ	1.00E-04	1.00E-04	1.00E-04	2.00E-04
ρ	1.1	1.05	1.1	1.2
tol	1.00E-08	1.00E-05	1.00E-05	1.00E-05
max-iter	500	500	500	500
p	0.9	\	\	\
λ	$1/\sqrt{3\max(n_1,n_2)}$	$1/\sqrt{\max(n_1,n_2)n_3}$	$1/\sqrt{3\max(n_1,n_2)}$	tuning parameter setting in original
initial rank k^0	\	round(0.5*min(n_1,n_2))	\	\
algorithm	SNN [36](BSD)	SNN [36](DOTA)	SNN [36](video)	OTRN-RE (our)
μ	1.00E-04	1.00E-04	1.00E-04	1.00E-04
ρ	1.1	1.1	1.1	1.3
tol	1.00E-08	1.00E-08	1.00E-08	1.00E-05
max_iter	500	500	500	500
p	\	\	\	0.98
λ	[15,15,1.5]	[20,20,20]	[200,2,20]	$1/\sqrt{\max(n_1,n_2)n_3}$
initial rank k^0	\	\	\	round(0.3*min(n_1,n_2))

IV. EXPERIMENTS

This section comprehensively evaluates the performance of our proposed algorithm through numerous experiments. Our algorithm was compared with other state-of-the-art techniques, including RTPCA-RE [42], TRPCA [9], SNN [36], ETRPCA [40], and WSTNN [46] in terms of both image denoising and salient object detection. For RTPCA-RE, the source code was provided by the authors. For SNN and TRPCA, the source code was obtained from the LibADMM toolbox[2]. For WSTNN[3] and ETRPCA[4], the source code was obtained from the author's GitHub. The parameter settings suggested by Lu et al. [9] were adopted for SNN, and TABLE 3 shows the detailed parameter settings for each method. In selecting a non-convex surrogate function, I draw inspiration from [28], [34], and [39] and employ the prevalent ℓ_p norm for the surrogate, thereby enhancing the generalizability of the non-convex function.

The platform used for this experiment is MATLAB(2021a) running on a Windows 10 PC equipped with a 3.00 GHz Intel i7-9700 CPU and 8GB memory. Our code is open access.[5]

A. ZERO-MEAN GAUSSIAN-IMPULSE MIXED NOISE: IMAGE RECOVERY

In this section, the proposed model is applied to image recovery since clean color images can be approximated by a low-rank tensor. The first 50 color images from the Berkeley segmentation dataset (BSD)[6] of size $321 \times 481 \times 3$ or $481 \times 321 \times 3$ and the first 10 color images of the large-scale DOTA dataset[7] of size $1024 \times 1024 \times 3$ are used in the experiment. If the color image $\mathcal{L}_0$ is contaminated with a mixture of zero-mean Gaussian noise $\mathcal{Z}_0$ and impulse noise $\mathcal{S}_0$, the noise can be elminated from the noised color images $\mathcal{X} = \mathcal{L}_0 + \mathcal{Z}_0 + \mathcal{S}_0$ by all six methods (including RTPCA-RE [42], TRPCA [9], SNN [36], ETRPCA [40], WSTNN [46], and OTRN-RE). In this experiment, (δ, c) is set to $(\delta, c) = \{(10, 10\%), (10, 20\%), (10, 30\%), (15, 10\%), (15, 20\%), (15, 30\%), (25, 10\%), (25, 20\%), (25, 30\%)\}$, where δ represents the standard deviation of the zero-mean Gaussian noise, and c represents the density of the impulse noise. Peak Signal-to-Noise Ratio (PSNR), which is defined as

$$\text{PSNR} = 10\log_{10}\left(\frac{\|\mathcal{X}\|_\infty^2}{\frac{1}{n_1n_2n_3}\|\hat{\mathcal{X}} - \mathcal{X}\|_F^2}\right),$$

is adopted to evaluate recovery performance. Meanwhile, the algorithm's running time is exploited to evaluate the recovery speed. The highest PSNR values and the lowest time are marked in bold. For all methods, the stopping criteria are

$$tol = \frac{\|\mathcal{L}^{rec} - \mathcal{L}\|_F}{\|\mathcal{L}\|_F}.$$

1) BSD IMAGE DENOISING

TABLE 4 presents the PSNR values and running time of different methods when the Berkeley segmentation dataset is contaminated by zero-mean Gaussian-impulse mixed noise. The visual quality of all methods is reported in Fig. 3, 4 and 5. From these results, the observations can be obtained. First, compared with the other five methods, OTRN-RE achieves the best denoising performance and the lowest running time in most cases. Specifically, for $(\delta = 15, c = 20\%)$ in TABLE 4, OTRN-RE outperforms the five other comparison methods by at least 0.4 dB in the average PSNR, and its running speed is 1.55 times that of TRPCA.

2) LARGE-SCALE DOTA IMAGE DENOISING

TABLE 5 and 6 present the PSNR values and running time of various methods on the Berkeley segmentation dataset contaminated with zero-mean Gaussian-impulse mixed noise. Fig. 6 shows the visual quality of all methods. Based on these results, several observations can be made. First, the PSNR values of the proposed method (OTRN-RE) and the other five methods show that OTRN-RE achieves the best denoising

[2]https://github.com/canyilu/LibADMM-toolbox
[3]https://github.com/YuBangZheng/code_WSTNN
[4]https://github.com/xdweixia/TPAMI2020_ETRPCA
[5]https://github.com/15058795765/Tensor-recovery
[6]https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/
[7]https://captain-whu.github.io/DOTA/dataset.html

TABLE 4. The PSNR values and running time of various methods for denoising BSD images contaminated by mixed noise with different (δ, c).

δ	c	RTPCA-RE [42]		TRPCA [9]		SNN [36]		ETRPCA [40]		WSTNN [46]		OTRN-RE(our)	
		PSNR	time(s)	PSNR	time(s)	PSNR	time(s)	PSNR	time(s)	PSNR	time(s)	PSNR	time(s)
10	10	26.95	69.35	27.20	7.33	26.35	20.47	**27.23**	22.14	25.78	6.01	26.88	**4.33**
	20	25.41	72.72	25.52	7.25	24.90	20.35	25.47	22.06	24.86	6.13	**25.62**	**4.31**
	30	22.89	80.26	22.88	7.17	22.53	20.23	22.48	22.12	23.26	5.54	**23.57**	**4.34**
15	10	25.62	73.79	25.72	6.75	25.07	19.93	25.73	21.32	25.06	5.95	**25.83**	**4.31**
	20	24.01	76.69	24.04	6.72	23.56	19.84	23.88	21.32	24.10	6.02	**24.51**	**4.33**
	30	21.55	78.36	21.50	6.68	21.21	19.71	20.94	21.36	22.35	5.65	**22.46**	**4.33**
25	10	23.24	78.43	23.22	6.66	22.71	19.51	23.08	21.33	23.76	6.22	**24.00**	**4.36**
	20	21.66	78.27	21.60	6.65	21.20	19.24	21.23	21.35	22.58	6.39	**22.60**	**4.45**
	30	19.40	78.26	19.30	6.64	19.04	19.40	18.60	21.39	**20.65**	5.90	20.54	**4.46**
avg		23.42	76.24	23.44	6.87	22.95	19.85	23.18	21.06	23.60	5.98	**24.00**	**4.36**

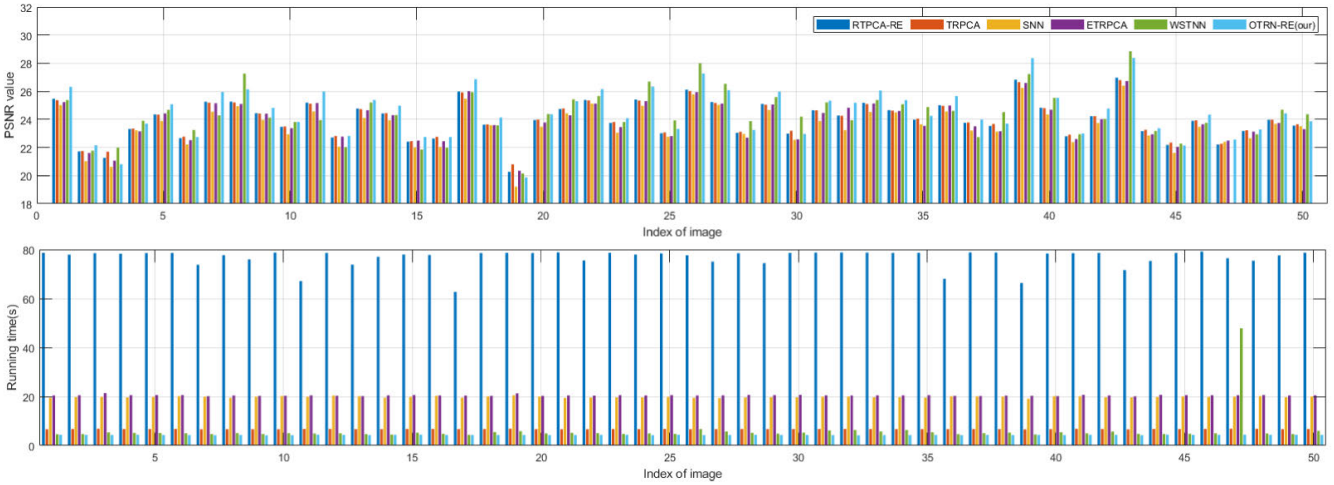


FIGURE 3. The PSNR values and running time of various methods on BSD image data when $\delta = 15$, $c = 20\%$.

TABLE 5. The PSNR values of various methods for denoising large-size DOTA images contaminated by mixed noise with different (δ, c).

	(δ,c)=(10,10%)						(δ,c)=(10,20%)						(δ,c)=(10,30%)					
images	RTPCA-RE [42]	TRPCA [9]	SNN [36]	ETRPCA [40]	WSTNN [46]	OTRN-RE	RTPCA-RE [42]	TRPCA [9]	SNN [36]	ETRPCA [40]	WSTNN [46]	OTRN-RE	RTPCA-RE [42]	TRPCA [9]	SNN [36]	ETRPCA [40]	WSTNN [46]	OTRN-RE
1	27.79	28.12	26.79	27.99	25.65	**28.20**	25.82	25.87	24.04	26.27	25.65	**26.48**	22.55	22.32	19.70	23.37	22.81	**23.83**
2	23.31	**24.51**	23.44	23.68	21.57	23.39	21.91	**22.54**	21.18	22.20	21.57	22.09	19.57	19.72	17.60	19.99	**20.42**	20.26
3	27.42	27.52	26.30	27.61	26.83	**27.91**	25.11	25.04	23.43	25.70	**26.83**	25.89	21.78	21.51	19.21	22.75	22.69	**23.11**
4	27.17	**27.80**	26.49	27.40	23.61	27.30	25.14	25.35	23.46	25.55	23.61	**25.66**	21.87	21.71	19.02	22.56	22.64	**23.01**
5	29.11	29.18	27.65	29.44	25.54	**29.63**	26.88	26.76	24.76	27.54	25.54	**27.69**	23.37	23.04	20.28	24.42	23.08	**24.82**
6	28.01	28.20	26.89	28.18	26.43	**28.45**	26.00	25.99	24.25	26.52	26.43	**26.71**	22.79	22.52	19.99	23.72	23.58	**24.06**
7	29.09	29.12	27.63	29.38	26.97	**29.63**	26.97	26.82	24.95	27.63	26.97	**27.82**	23.59	23.23	20.61	24.62	23.73	**25.06**
8	28.03	**28.35**	27.11	28.20	25.02	28.37	25.97	26.01	24.20	26.42	25.02	**26.59**	22.68	22.44	19.81	23.50	**24.09**	23.92
9	29.35	29.16	27.37	**30.40**	26.16	30.29	27.38	27.07	25.02	**28.61**	26.16	28.59	24.00	23.56	20.75	25.46	21.92	**25.81**
10	28.34	28.49	27.04	28.53	24.17	**28.84**	26.35	26.29	24.37	26.85	24.17	**27.10**	23.02	22.74	19.95	23.94	21.82	**24.35**
avg	27.76	28.05	26.67	28.08	25.20	**28.20**	25.75	25.77	23.97	26.33	25.20	**26.46**	22.52	22.28	19.69	23.43	22.68	**23.82**
	(δ,c)=(15,10%)						(δ,c)=(15,20%)						(δ,c)=(15,30%)					
images	RTPCA-RE [42]	TRPCA [9]	SNN [36]	ETRPCA [40]	WSTNN [46]	OTRN-RE	RTPCA-RE [42]	TRPCA [9]	SNN [36]	ETRPCA [40]	WSTNN [46]	OTRN-RE	RTPCA-RE [42]	TRPCA [9]	SNN [36]	ETRPCA [40]	WSTNN [46]	OTRN-RE
1	25.98	26.00	24.48	26.38	25.02	**26.60**	24.04	23.89	22.03	24.61	24.29	**24.92**	21.05	20.75	18.38	21.68	22.00	**22.41**
2	22.63	**23.39**	22.33	22.88	21.32	22.82	21.12	21.48	20.07	21.39	21.21	**21.51**	18.78	18.80	16.83	19.10	**20.03**	19.58
3	25.38	25.26	23.73	25.90	25.93	**26.12**	23.23	23.01	21.24	23.95	**24.60**	24.24	20.19	19.86	17.68	20.95	**21.85**	21.65
4	25.51	25.71	24.25	25.83	23.23	**25.99**	23.53	23.50	21.62	24.00	23.33	**24.29**	20.54	20.30	17.90	21.07	**22.00**	21.76
5	26.85	26.69	24.96	27.50	25.00	**27.66**	24.76	24.48	22.46	25.53	24.45	**25.84**	21.60	21.23	18.73	22.41	22.18	**23.14**
6	26.18	26.13	24.59	26.65	25.81	**26.86**	24.20	24.02	22.16	24.86	**25.17**	25.14	21.19	20.88	18.55	21.94	**22.74**	22.61
7	26.90	26.71	25.00	27.55	26.29	**27.75**	24.87	24.57	22.60	25.65	25.31	**25.96**	21.77	21.38	18.98	22.56	22.81	**23.35**
8	26.17	26.17	24.66	26.56	24.55	**26.78**	24.17	24.01	22.12	24.74	24.75	**25.04**	21.15	20.85	18.46	21.78	**23.36**	22.56
9	26.97	26.62	24.64	**28.23**	25.40	28.27	24.96	24.55	22.41	26.18	23.88	**26.44**	21.82	21.35	18.80	22.88	20.92	**23.74**
10	26.41	26.33	24.70	26.91	23.77	**27.16**	24.42	24.21	22.24	25.09	23.25	**25.43**	21.37	21.02	18.52	22.10	21.07	**22.82**
avg	25.90	25.90	24.33	26.44	24.63	**26.60**	23.93	23.77	21.90	24.60	24.02	**24.88**	20.95	20.64	18.28	21.65	21.90	**22.36**
	(δ,c)=(25,10%)						(δ,c)=(25,20%)						(δ,c)=(25,30%)					
images	RTPCA-RE [42]	TRPCA [9]	SNN [36]	ETRPCA [40]	WSTNN [46]	OTRN-RE	RTPCA-RE [42]	TRPCA [9]	SNN [36]	ETRPCA [40]	WSTNN [46]	OTRN-RE	RTPCA-RE [42]	TRPCA [9]	SNN [36]	ETRPCA [40]	WSTNN [46]	OTRN-RE
1	23.11	22.83	21.01	23.65	23.67	**24.12**	21.30	20.95	18.98	21.77	**22.53**	22.51	18.72	18.35	16.19	18.99	**20.56**	20.22
2	21.04	21.28	19.98	21.27	20.71	**21.52**	19.49	19.52	17.96	19.72	**20.37**	20.17	17.31	17.16	15.32	17.42	**19.21**	18.24
3	22.30	21.95	20.06	22.96	**24.26**	23.51	20.39	20.00	18.03	20.95	**22.73**	21.75	17.77	17.38	15.29	18.11	**20.38**	19.26
4	22.77	22.60	20.86	23.19	22.48	**23.61**	20.95	20.69	18.75	21.35	**22.15**	22.00	18.41	18.08	15.95	18.61	**20.79**	19.71
5	23.60	23.21	21.23	24.32	23.66	**24.86**	21.69	21.27	19.18	22.30	22.56	**23.10**	19.03	18.61	16.38	19.37	20.60	**20.63**
6	23.26	22.95	21.10	23.89	**24.37**	24.34	21.40	21.04	19.06	21.95	**23.30**	22.67	18.80	18.41	16.27	19.13	**21.21**	20.32
7	23.68	23.27	21.26	24.40	24.89	**25.03**	21.79	21.35	19.28	22.40	**23.47**	23.27	19.15	18.71	16.51	19.48	**21.26**	20.75
8	23.24	22.94	21.08	23.78	23.69	**24.27**	21.39	21.03	19.03	21.86	**23.51**	22.63	18.81	18.41	16.26	19.07	**22.05**	20.26
9	23.52	23.01	20.85	24.56	23.45	**25.18**	21.60	21.09	18.89	22.39	21.53	**23.30**	18.89	18.40	16.11	19.33	19.21	**20.60**
10	23.44	23.13	21.28	24.07	22.55	**24.54**	21.54	21.18	19.16	22.10	21.49	**22.83**	18.91	18.52	16.31	19.23	19.62	**20.44**
avg	23.00	22.72	20.87	23.61	23.37	**24.10**	21.15	20.81	18.83	21.68	22.36	**22.42**	18.58	18.20	16.06	18.87	**20.49**	20.04

performance and the lowest running time for large-size color images in most cases. Especially, for ($\delta = 25$, $c = 10\%$) in TABLE 5, OTRN-RE outperforms the other five methods by at least 0.5 dB in terms of the average PSNR, and its running speed is 1.65 times that of TRPCA. Besides, compared to the TRPCA method, our OTRN-RE method runs not only faster but also performs better in processing large-size images (combined with TABLE 4 and 6).

Experiments on the BSD and DOTA datasets indicate that WSTNN performs poorly relative to OTRN-RE in performing the low noise tensor recovery task. For example, in TABLE 5 ($\delta = 10, c = 10$), the average PSNR of

TABLE 6. The running time of various methods for denoising large-size DOTA images contaminated by mixed noise with different (δ, c).

δ	c	RTPCA-RE [42]	TRPCA [9]	SNN [36]	ETRPCA [40]	WSTNN [46]	OTRN-RE(our)
10	10	1272.52	99.69	185.74	344.33	**53.14**	55.78
	20	1250.17	98.97	178.38	342.64	56.05	**55.75**
	30	1271.37	97.88	175.34	340.19	59.32	**56.15**
15	10	1228.26	94.04	172.69	377.58	**54.40**	55.81
	20	1228.59	93.76	167.46	376.04	57.27	**56.01**
	30	1228.76	93.38	165.59	374.23	59.33	**56.25**
25	10	1242.04	93.09	162.73	374.10	57.39	**56.20**
	20	1213.23	92.85	166.11	371.36	60.47	**56.45**
	30	1213.74	92.87	169.06	366.11	61.22	**56.50**
avg		1238.74	95.17	171.46	362.95	57.62	**56.10**

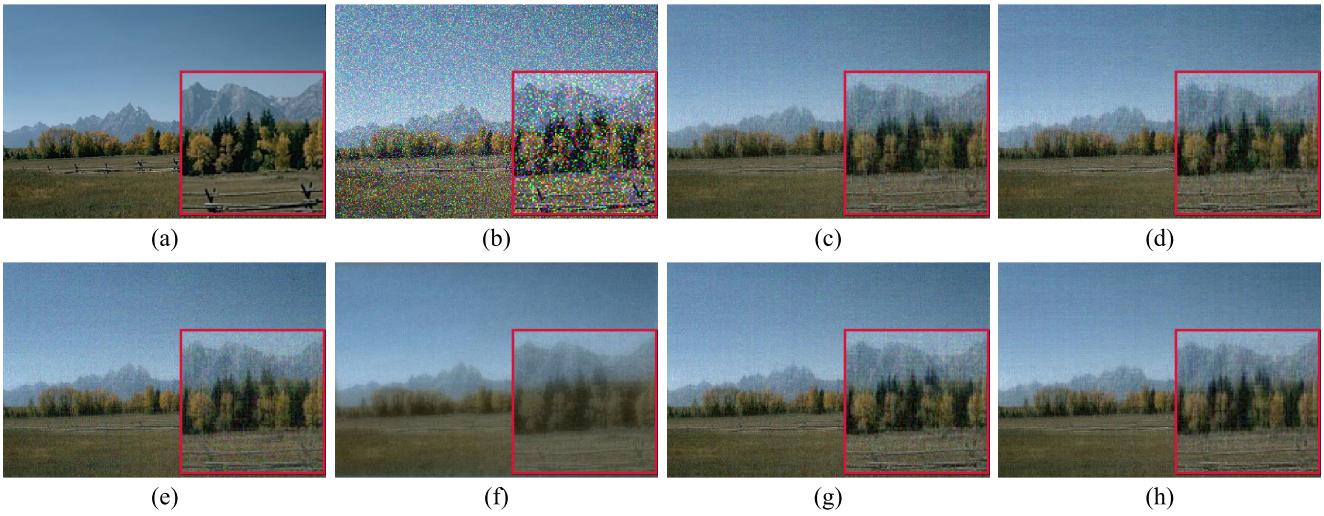


FIGURE 4. Denoised results on "Mountain", $\delta = 15$, $c = 20\%$. (a) Original image (b) Noised image (c) TRPCA [9]. (d) ETRPCA [40]. (e) SNN [36]. (f) WSTNN [46]. (g) RTPCA-RE [42]. (h) OTRN-RE.

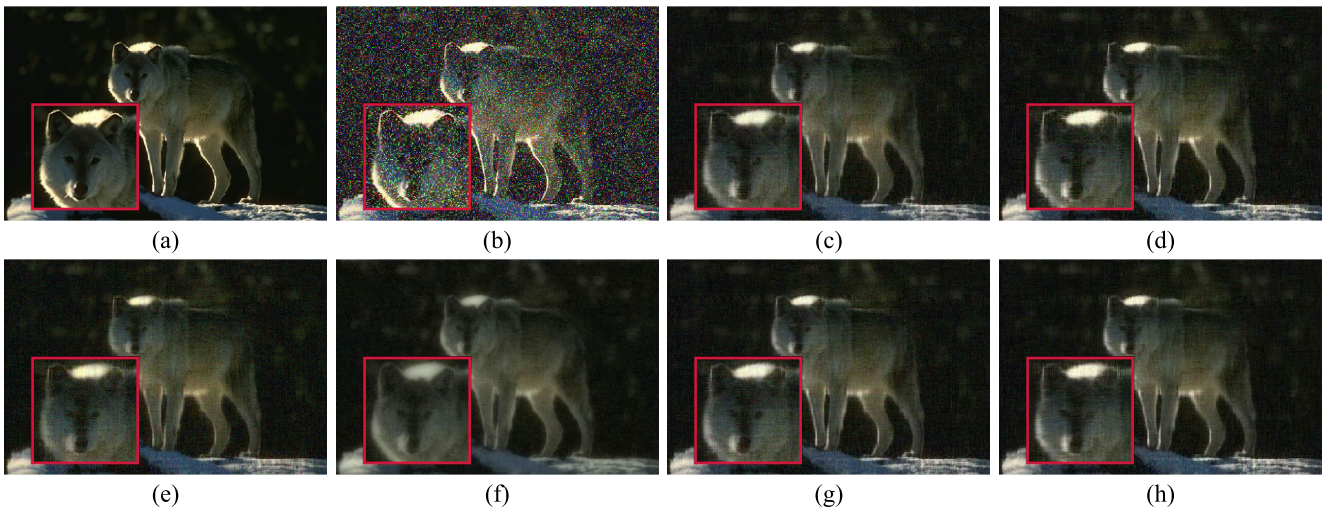


FIGURE 5. Denoised results on "Wolf", $\delta = 15$, $c = 20\%$. (a) Original image (b) Noised image (c) TRPCA [9]. (d) ETRPCA [40]. (e) SNN [36]. (f) WSTNN [46]. (g) RTPCA-RE [42]. (h) OTRN-RE.

WSTNN is lower than that of OTRN-RE by 3 dB. However, WSTNN obtains a good result when dealing with high noise. For example, in TABLE 6 ($\delta = 25, c = 30$), the average PSNR of WSTNN is 0.4 dB higher than that of OTRN-RE. This is because WSTNN proposes a new tensor rank, which has the advantage of high flexibility for correlations in different modes and is conducive to dealing with high noise. In addition, the performance of RTPCA-RE using the rank

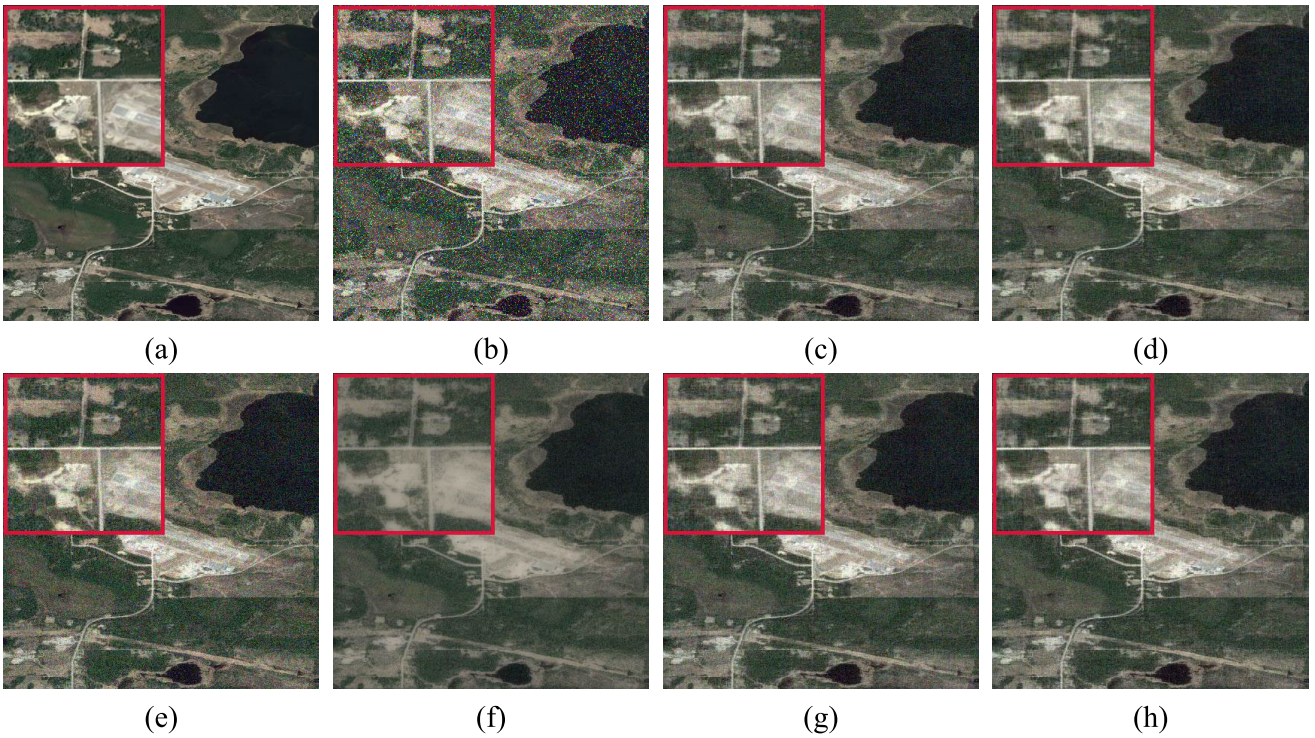


FIGURE 6. Denoised results on "DOTA", $\delta = 25$, $c = 10\%$. (a) Original image (b) Noised image (c) TRPCA [9]. (d) ETRPCA [40]. (e) SNN [36]. (f) WSTNN [46]. (g) RTPCA-RE [42]. (h) OTRN-RE.

estimation algorithm was observed. In TABLE 6 ($\delta = 25, c = 30$), the average PSNR of RTPCA-RE is 1.5 dB lower than that of OTRN-RE, while the average running time of RTPCA-RE is 24.5 times that of OTRN-RE. This is because the RTPCA-RE algorithm consumes much time in using the rank estimation algorithm to identify the most suitable rank size before performing tensor recovery. Particularly, RTPCA-RE becomes slower when dealing with large tensors.

Compared with the other five algorithms, our OTRN-RE method performs better in tensor recovery, especially in processing contaminated large-size images, with an obvious speed advantage. This is attributed to the proposed novel tensor decomposition algorithm, which can significantly reduce the computational complexity and speed up the tensor recovery. Meanwhile, this paper combines a generalized non-convex framework to achieve better rank approximation results and recover the tensor more efficiently. Moreover, this paper proposes a new rank estimation algorithm, which can find the optimal rank without affecting the speed of the algorithm and enhances the recovery of the tensor.

B. PARAMETER ANALYSIS

This section presents experimental results that demonstrate the relationship between the PSNR values of the proposed OTRN-RE algorithm and the penalty parameter p. When $\delta = 10, c = 30\%$, the denoising results of our OTRN-RE algorithm for images of the Berkeley segmentation dataset with different settings of the penalty parameter p are shown in Fig. 8. By setting the penalty parameter p between 0.9 and 1, the denoising algorithm achieves a high PSNR value with a good denoising effect. In particular, the algorithm achieves the best denoising effect when p is set to 0.98, and this setting is adopted for all subsequent experiments.

C. SALIENT OBJECT DETECTION FOR VIDEOS

In this section, the salient object detection problem is resolved, which involves separating foreground objects from the background in a video. The background frames in the video are highly correlated and can be modeled as a low-rank tensor, whereas the moving foreground objects occupy only a small portion of the image pixels and can be modeled as sparse errors. In this experiment, various methods were employed to implement foreground extraction, including RTPCA-RE [42], TRPCA [9], SNN [36], WSTNN [46], ETRPCA [40], and OTRN-RE. The experimental data were three color videos from the ChangeDetection.net (CDNet) dataset 2014[8] [47], including highway, skating, and busStation. Between 900 and 1000 frames were taken from each video as experimental data, where each video can be represented as an $n_1 \times n_2 \times 101 \times 3$ tensor, where 101 is the number of frames, and 3 is the number of color channels. The resolution of each frame is $n_1 \times n_2$. The frames in each color channel were stacked as a column vector of size $N_1 \times 1$, where $N_1 = n_1 \times n_2$, and the video was reshaped into a tensor of size $N_1 \times N_2 \times 3$, where N_2 is the number of frames. Then, the similarity between the estimated foreground regions and the ground truth, or the Intersection over Union (IoU), i.e., $S(A, B)=\frac{A \cap B}{A \cup B}$, was adopted to measure the performance

[8]http://changedetection.net/

TABLE 7. Salient object detection.

video	RTPCA-RE [42]		TRPCA [9]		SNN [36]		WSTNN [46]		ETRPCA [40]		OTRN-RE	
	S	time(s)	S	time(s)	S	time(s)	S	time(s)	S	time(s)	S	time(s)
highway	0.3723	2512.31	0.3683	205.09	0.374	1392.99	0.2196	404.90	0.3666	681.62	**0.3765**	**87.01**
skating	0.5372	37185.8	0.5338	523.07	0.5385	9542.95	0.3026	384.24	0.5332	1957.31	**0.5398**	**213.69**
busStation	0.3517	2858.33	0.3474	234.33	**0.3521**	1351.15	0.2058	496.75	0.3462	803.97	0.35	**92.14**
Average	0.4204	14185.48	0.4165	320.83	0.4215	4095.69	0.2427	428.63	0.4153	1147.63	**0.4221**	**130.95**

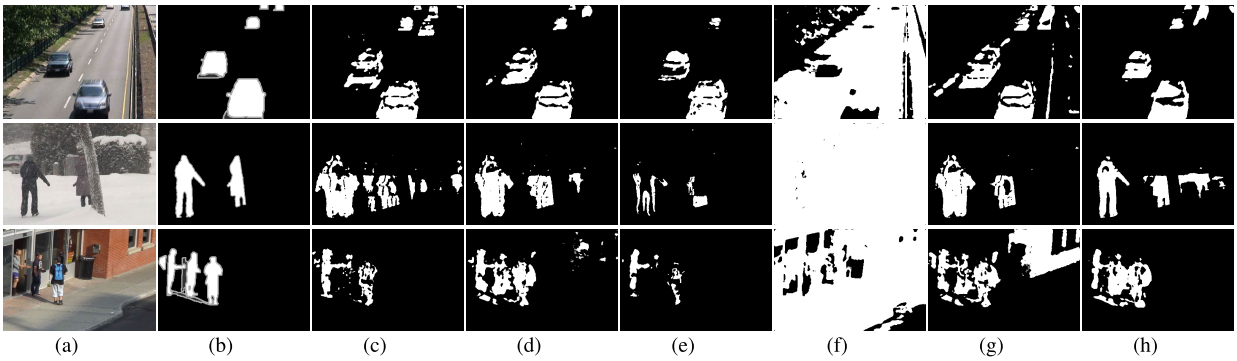


FIGURE 7. Some salient object detection results on the CDNet dataset. Row 1 shows a sample of a highway; row 2 shows a sample of skating; row 3 shows a sample of busStation. (a) Original; (b) Ground Truth; (c) RTPCA-RE [42]. (d) TRPCA [9]. (e) SNN [36]. (f) WSTNN [46] (g) ETRPCA [40]. (h) OTRN-RE.

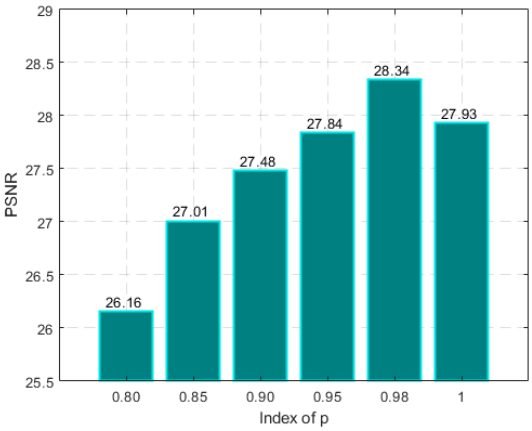


FIGURE 8. The results of OTRN-RE in denoising analysis of BSD image data under different settings of p.

of different methods in background modeling. TABLE 7 shows the IoU and time of different methods for foreground extraction on the CDNet dataset. Fig. 7 presents the visual quality of foreground extraction for all methods, where the highest IoU and the lowest time values are marked in bold. Through experiments, it was found that our proposed method (OTRN-RE) performs the best in foreground extraction while having the fastest running speed. Meanwhile, the following observations can be obtained from the experimental results.

The performance of the SNN is unsatisfactory on the skating and busStation datasets due to its inability to extract enough foreground information. This is because the SNN considers the low rank of each mode and does not perform well in tight convex relaxation rank optimization, thus failing to obtain sufficient foreground information. On the video datasets of highway and busStation, ETRPCA captures too much foreground information, such as highway edge lines and bus stop walls. This is attributed to the fact that ETRPCA uses a weighted Schatten p-norm to minimize the tubal rank, but the rank approximation is excessive, leading to much unnecessary foreground information. Besides, WSTNN retains a large amount of foreground information in all video datasets and demonstrates very poor foreground extraction capability. This is because WSTNN exploits mode-k tensor matriculation and uses a weighted sum of the tensor nuclear norm for tight convex relaxation rank optimization. Although this helps to handle tensor information in different modes, it retains too many tensor singularities and results in excessive foreground information extraction.

Compared to other methods, our proposed method (OTRN-RE) combines a tensor decomposition algorithm with a generalized non-convex tubal rank minimization framework to better approximate the rank and quickly retain more complete and accurate foreground information, so it performs better in foreground extraction tasks.

D. CONVERGENCE ANALYSIS

According to the currently available information, our model involves non-convex surrogate and orthogonal constraints, making it difficult to prove the convergence of Eq. (6) in theory. Therefore, to prove the convergence of the model, experiments were conducted to compare the convergence of all methods by recovering images of the contaminated Berkeley segmentation dataset. Fig. 9 presents the convergence analysis results of RTPCA-RE [42], TRPCA [9], SNN [36], WSTNN [46], ETRPCA [40], and our proposed OTRN-RE algorithm with different values of p. Through the experimental results ERROR $= \frac{|\mathcal{L}^{rec}-\mathcal{L}|_F}{|\mathcal{L}|_F}$, the following observations can be made:

1) OTRN-RE achieves faster convergence than most comparative methods.

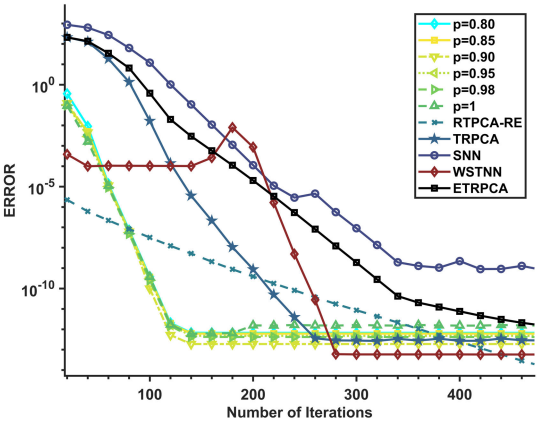


FIGURE 9. The convergence analysis results of OTRN-RE under different p options with RTPCA-RE, TRPCA, SNN, WSTNN, and ETRPCA for denoising on BSD image data.

2) The convergence of OTRN-RE shows robustness to the parameter p.
3) RTPCA-RE and WSTNN exhibit lower error values. The is because RTPCA-RE spends a considerable amount of time determining the optimal rank size, resulting in a lower initial error. Meanwhile, WSTNN uses the weighted sum of tensor nuclear norms to minimize the defined rank, which can quickly reduce the error. However, the convergence ability of WSTNN is limited, and after a significant number of iterations, there is an abrupt convergence and a sharp drop in PSNR values.

These observations show that our algorithm has better convergence properties than other methods.

V. CONCLUSION AND FUTURE WORK

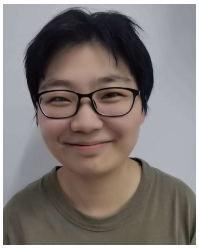
In this work, a new orthogonal tensor decomposition model is proposed for low-rank subspace learning. This model is designed to handle large-scale tensor data and solve the problem of high computational complexity. It is found that if the large tensor is decomposed into the product of a standard orthogonal small tensor and a small coefficient tensor since the decomposed small orthogonal tensor has orthogonal unitary invariance, the rank of the large tensor before decomposition is equivalent to that of the coefficient tensor after decomposition. To enhance the accuracy of recovering low-rank subspaces, we incorporate non-convex regularization to constrain the coefficient tensor within the model. Additionally, a new rank estimation algorithm is proposed, which uses the estimated optimal rank as an upper bound on the dimension of the decomposed small tensor to dynamically adjust the size of the orthogonal small tensor and the coefficient tensor. It has been experimentally verified that our method OTRN-RE not only achieves better performance but also is faster than most advanced tensor recovery methods. Currently, a large amount of data contain tensors of more than three orders, such as color videos. Therefore, in future work, we will continue to handle higher-order tensors, which is a significant direction for intensive research.

REFERENCES

[1] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.
[2] Y. Xie, S. Gu, Y. Liu, W. Zuo, W. Zhang, and L. Zhang, "Weighted Schatten p-norm minimization for image denoising and background subtraction," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4842–4857, Oct. 2016.
[3] M. A. Davenport and J. Romberg, "An overview of low-rank matrix recovery from incomplete observations," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 608–622, Jun. 2016.
[4] W.-J. Zeng and H. C. So, "Outlier-robust matrix completion via ℓ_p-minimization," *IEEE Trans. Signal Process.*, vol. 66, no. 5, pp. 1125–1140, Mar. 2017.
[5] J. Yang, L. Luo, J. Qian, Y. Tai, F. Zhang, and Y. Xu, "Nuclear norm based matrix regression with applications to face recognition with occlusion and illumination changes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 156–171, Jan. 2017.
[6] H. Yong, D. Meng, W. Zuo, and L. Zhang, "Robust online matrix factorization for dynamic background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 7, pp. 1726–1740, Jul. 2018.
[7] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.
[8] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.
[9] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 925–938, Apr. 2020.
[10] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, Feb. 2011, Art. no. 025010.
[11] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, Aug. 2009.
[12] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proc. 4th ACM Conf. Recommender Syst.*, Sep. 2010, pp. 79–86.
[13] M. Mørup, "Applications of tensor (multiway array) factorizations and decompositions in data mining," *WIREs Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 24–40, Jan. 2011.
[14] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, Jan. 2013.
[15] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2017.
[16] Y. Xie, D. Tao, W. Zhang, Y. Liu, L. Zhang, and Y. Qu, "On unifying multi-view self-representations for clustering by tensor multi-rank minimization," *Int. J. Comput. Vis.*, vol. 126, no. 11, pp. 1157–1179, Nov. 2018.
[17] F. Zhang, J. Wang, W. Wang, and C. Xu, "Low-tubal-rank plus sparse tensor recovery with prior subspace information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3492–3507, Oct. 2021.
[18] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller, "Tensor-based formulation and nuclear norm regularization for multienergy computed tomography," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1678–1693, Apr. 2014.
[19] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-SVD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3842–3849.
[20] P. Zhou, C. Lu, Z. Lin, and C. Zhang, "Tensor factorization for low-rank tensor completion," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1152–1163, Mar. 2018.
[21] X. Shu, F. Porikli, and N. Ahuja, "Robust orthonormal subspace learning: Efficient recovery of corrupted low-rank matrices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3874–3881.
[22] L. E. Frank and J. H. Friedman, "A statistical view of some chemometrics regression tools," *Technometrics*, vol. 35, no. 2, pp. 109–135, May 1993.

[23] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Image Process.*, vol. 4, no. 7, pp. 932–946, Jul. 1995.
[24] J. Trzasko and A. Manduca, "Highly undersampled magnetic resonance image reconstruction via homotopic ℓ_0-minimization," *IEEE Trans. Med. Imag.*, vol. 28, no. 1, pp. 106–121, Jan. 2009.
[25] C.-H. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *Ann. Statist.*, vol. 38, no. 2, pp. 894–942, Apr. 2010.
[26] J. H. Friedman, "Fast sparse regression and classification," *Int. J. Forecasting*, vol. 28, no. 3, pp. 722–738, Jul. 2012.
[27] C. Gao, N. Wang, Q. Yu, and Z. Zhang, "A feasible nonconvex relaxation approach to feature selection," in *Proc. AAAI Conf. Artif. Intell.*, 2011, vol. 25, no. 1, pp. 356–361.
[28] H. Wang, F. Zhang, J. Wang, T. Huang, J. Huang, and X. Liu, "Generalized nonconvex approach for low-tubal-rank tensor recovery," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3305–3319, Aug. 2022.
[29] Q. Li, "A comprehensive survey of sparse regularization: Fundamental, state-of-the-art methodologies and applications on fault diagnosis," *Expert Syst. Appl.*, vol. 229, Nov. 2023, Art. no. 120517.
[30] G. Zhu, X. Lv, L. Jiang, X. Sun, and B. Fang, "Nonconvex regularization for convex image smoothing," *Signal Process.*, vol. 205, Apr. 2023, Art. no. 108862.
[31] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an 'explanatory' multimodal factor analysis," Univ. Microfilms, UCLA, Ann Arbor, MI, USA, Working Papers 10,085, pp. 1–84, 1970, vol. 16.
[32] R. Tomioka, K. Hayashi, and H. Kashima, "Estimation of low-rank tensors via convex optimization," 2010, *arXiv:1010.0789*.
[33] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. A. K. Suykens, "Learning with tensors: A framework based on convex optimization and spectral regularization," *Mach. Learn.*, vol. 94, no. 3, pp. 303–351, Mar. 2014.
[34] X. Zhang, J. Zheng, L. Zhao, Z. Zhou, and Z. Lin, "Tensor recovery with weighted tensor average rank," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 1, pp. 1142–1156, 2024.
[35] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *J. ACM*, vol. 60, no. 6, pp. 1–39, Nov. 2013.
[36] H. Kasai and B. Mishra, "Low-rank tensor completion: A Riemannian manifold preconditioning approach," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1012–1021.
[37] B. Romera-Paredes and M. Pontil, "A new convex relaxation for tensor completion," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–12.
[38] J. Zheng, X. Zhang, W. Wang, and X. Jiang, "Handling slice permutations variability in tensor recovery," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 3, pp. 3499–3507.
[39] C. Lu, J. Tang, S. Yan, and Z. Lin, "Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 829–839, Feb. 2016.
[40] Q. Gao, P. Zhang, W. Xia, D. Xie, X. Gao, and D. Tao, "Enhanced tensor RPCA and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 2133–2140, Jun. 2021.
[41] Y. Xu, R. Hao, W. Yin, and Z. Su, "Parallel matrix factorization for low-rank tensor completion," 2013, *arXiv:1312.1254*.
[42] Q. Shi, Y.-M. Cheung, and J. Lou, "Robust tensor SVD and recovery with rank estimation," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10667–10682, Oct. 2022.
[43] G. Papamakarios, "Robust low-rank modelling on matrices and tensors," Ph.D. dissertation, M.Sc. thesis, Dept. Comput., Imperial College, London, U.K., 2014.
[44] W. Qin, H. Wang, F. Zhang, J. Wang, X. Luo, and T. Huang, "Low-rank high-order tensor completion with applications in visual data," *IEEE Trans. Image Process.*, vol. 31, pp. 2433–2448, 2022.
[45] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *J. Comput. Graph. Statist.*, vol. 15, no. 2, pp. 265–286, Jun. 2006.
[46] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-Y. Ji, and T.-H. Ma, "Tensor N-tubal rank and its convex relaxation for low-rank tensor recovery," *Inf. Sci.*, vol. 532, pp. 170–189, Sep. 2020.
[47] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 393–400.

XIXIANG CHEN received the B.S. degree in computer science and technology from the Binjiang College, Zhejiang Chinese Medical University, in 2019. He is currently pursuing the degree in computer science and technology with the School of Computer Science and Artificial Intelligence, Wenzhou University, China. His research interests include low-rank recovery and machine learning.



JINGJING ZHENG received the B.A. degree in art and design from the Wuchang Institute of Technology, Wuhan, China, in 2015, the M.S. degree in applied mathematics from the College of Mathematics and Physics, Wenzhou University, Wenzhou, China, in 2020, and the Ph.D. degree in computer science from Memorial University, St. John's, NL, Canada, in 2023. She is currently pursuing the Ph.D. degree with the Department of Mathematics, The University of British Columbia, Vancouver, BC, Canada. Her current research interests include pattern recognition and machine learning.

LI ZHAO received the B.Sc. degree in automation and the M.Eng. degree in control theory and control engineering from Central South University, China, in 2005 and 2008, respectively. She is currently an Assistant Researcher with Wenzhou University. Her research interests include pattern recognition, computer vision, and machine learning.

WEI JIANG received the B.S. and Ph.D. degrees from the School of Computer and Communication Engineering, University of Science and Technology Beijing, in 2004 and 2012, respectively. He became a Faculty Member with the School of Mathematics, Liaoning Normal University, Dalian, China, in July 2004, where he is currently a Professor. He has worked at the College of Computer and Artificial Intelligence, Wenzhou University, Wenzhou, China. His research interests include computer vision and machine learning.

XIAOQIN ZHANG received the B.Sc. degree in electronic information science and technology from Central South University, China, in 2005, and the Ph.D. degree in pattern recognition and intelligent system from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China, in 2010. He is currently a Professor with Wenzhou University, China. He has authored or coauthored more than 100 papers in international and national journals and international conferences. His research interests include pattern recognition, computer vision, and machine learning.