High-Order Tensor Robust Principal Component Analysis with a Nonconvex Function Designed for Low-Rank Approximation

Authors: Hanxin Liu, Zhuolin Du, Yisheng Song

Date: 2026-02-10T18:02:18+00:00

Abstract

In recent years, tensor robust principal component analysis (TRPCA) has been widely applied in the field of image processing. However, the traditional tensor nuclear norm (TNN) can only be used for third-order tensors and imposes identical penalty weights on all singular values. To adaptively assign distinct weights to singular values of different magnitudes, this work proposes a tensor arctangent norm (TAN). This norm can more effectively distinguish the magnitudes of singular values by assigning smaller weights to large singular values and larger weights to small ones. Owing to the favorable derivative properties of the arctangent function, when its first-order Taylor expansion is employed for approximation, TAN can clearly delineate the boundary between large and small singular values. On this basis, the TAN-TRPCA model is developed. To extend the applicability of TAN-TRPCA to higher-order tensors, a tensor mode-(1×4) unfolding operation is introduced. Finally, extensive experiments on color image, hyperspectral image, and color video datasets demonstrate the superior performance of the proposed method.

Full Text

Preamble

High-Order Tensor Robust Principal Component Analysis with a Nonconvex Function Designed for Low-Rank Approximation

Hanxin Liu, Zhuolin Du, Yisheng Song
School of Mathematical Sciences, Chongqing Normal University, Chongqing, 401331, P.R. China

Abstract

In recent years, tensor robust principal component analysis (TRPCA) has been extensively applied in the field of image processing. The traditional tensor nuclear norm (TNN) can only be applied to third-order tensors, and identical penalty weights are imposed on all singular values. To adaptively assign distinct weights to singular values of different magnitudes, a tensor arctangent norm (TAN) is proposed in this work. This norm is capable of better distinguishing the magnitudes of singular values, as smaller weights are assigned to large singular values and larger weights to small ones. Benefiting from the excellent derivative properties of the arctangent function, when the first-order Taylor expansion of the arctangent function is employed for approximation, TAN can clearly delineate the boundaries between large and small singular values. Subsequently, the TAN-TRPCA model is proposed. To extend the applicability of TAN-TRPCA to higher-order tensors, a tensor mode-(1$×$4) unfolding operation is introduced. Finally, excellent experimental results are achieved on color images, hyperspectral images, and color video datasets.

Keywords: unfolding operation low-rank tensor, high-order tensor, tensor arctangent norm, tensor

1. Introduction

With the advancement of technology, high-dimensional data has increasingly been utilized in medicine, signal and image processing, and daily life [?]. Examples include color images [?], hyperspectral images [?], and color videos [?]. During acquisition, transmission, and reception, such high-dimensional data is often corrupted by noise and suffers from information loss, which is caused by equipment failures or human factors. Thus, the restoration of high-dimensional data is required.

As an important dimensionality reduction method, principal component analysis (PCA) can be applied to data contaminated by mild noise. However, PCA is sensitive to outliers and heavy noise, which are inevitable in real-world scenarios [?]. To address this issue, robust principal component analysis (RPCA) [?] was proposed to recover a low-rank matrix from observed data contaminated by sparse noise. The nuclear norm, which is adopted by RPCA as a convex approximation of the rank function, has been widely applied in image denoising [?], matrix completion [?], and image enhancement [?]. For three-dimensional tensors representing color images, RPCA can be applied to each frontal slice individually; however, this approach disrupts the inherent data structure of the tensors. Therefore, tensor robust principal component analysis (TRPCA) was developed to extend RPCA from matrices to tensors.

An observation tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ can be decomposed into a low-rank tensor $\mathcal{L}$ and a sparse tensor $\mathcal{E}$, as shown in Fig. 1. TRPCA is designed to recover $\mathcal{L}$ from $\mathcal{X}$.

[Figure 1: see original paper]

Unlike matrices, the rank of a tensor is defined differently across various decomposition schemes. The Tucker rank [?] is obtained via Tucker decomposition [?] and is defined as the vector formed by the ranks of the factor matrices derived from the decomposition. Since the minimization of the Tucker rank is proven to be NP-hard, the sum of nuclear norms (SNN) [?] is employed as a convex surrogate for the Tucker rank. A model of SNN-TRPCA [?] was established by Huang et al. as follows:

$$\min_{\mathcal{L},\mathcal{E}} \sum_{i=1}^{3} \lambda_i \|\mathbf{L}_{(i)}\|_* + \|\mathcal{E}\|_1, \quad \text{s.t. } \mathcal{X} = \mathcal{L} + \mathcal{E},$$

where $\|\mathbf{L}_{(i)}\|_*$ and $\|\mathcal{E}\|_1$ denote the nuclear norm of the mode-i matrix unfolding $\mathbf{L}_{(i)}$ of tensor $\mathcal{L}$ and the ℓ_1-norm of the tensor $\mathcal{E}$, respectively, with $\lambda_i > 0$. While SNN is designed to fully exploit the low-rank information across all dimensions of the tensor, it must be expanded along a specific dimension, which results in the destruction of the tensor' s inherent structural information. This limitation is addressed by tensor singular value decomposition (t-SVD) [?], which decomposes the tensor into block circulant matrices and is used to define the tensor average rank [?]. Compared with the Tucker rank, the tensor average rank is found to better preserve the structural information of the original tensor. Since the minimization of the tensor average rank is proven to be NP-hard, the tensor nuclear norm (TNN) was employed by Lu et al. [?] as a convex approximation, and the TNN-TRPCA model was subsequently introduced:

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

where $\|\mathcal{L}\|_T$ denotes the TNN of $\mathcal{L}$. When the TNN model is solved via the tensor singular value thresholding algorithm, singular values of varying magnitudes are subjected to identical shrinkage, which is inconsistent with practical scenarios. Large singular values are generally regarded as representing the texture information of the tensor, whereas small singular values are considered to correspond to the noise components embedded in the tensor. In image restoration tasks, it is usually required that more large singular values be retained, while greater shrinkage be applied to small singular values. Therefore, a heavier penalty should be imposed on small singular values, and a milder penalty on large ones.

To assign distinct weights to singular values of different magnitudes, the truncated nuclear norm for matrices [?] was extended to tensors by Xue et al. [?], and the truncated tensor nuclear norm was applied to the TRPCA model, whereby improved experimental results were achieved. In parallel, the weighted tensor nuclear norm (WTNN) [?] was proposed by Mu et al. based on the matrix weighted nuclear norm [?]. By imposing different penalty weights on distinct singular values via a weight vector, the WTNN is employed to achieve the goal of processing different singular values separately. Similarly, inspired by the matrix truncated Schatten-p norm [?], the tensor truncated Schatten-p norm was applied to the TRPCA model by Liu et al. [?]. This approach subtracts several large singular values and then computes the sum of the p-th powers of the remaining singular values, allowing the recovered tensor to preserve more texture structures. However, the weights associated with these norms must be preset manually. To enable automatic weighting of singular values based on their magnitudes, a logarithmic norm $\|\mathbf{X}\|_{\log_\varepsilon} = \sum_{n=1}^{N} \alpha_n \|\mathbf{X}_{(n)}\|_{\log_\varepsilon}$, where $\|\mathbf{X}_{(n)}\|_{\log_\varepsilon} = \sum_{j=1}^{r} \log(\sigma_j(\mathbf{X}_{(n)})+\varepsilon)$, was proposed by Xue et al. [?]. In this norm, the weight vector is determined by differentiating the logarithmic function.

Subsequently, a tensor adjustable logarithmic norm (TALN) of the form $\|\mathcal{A}\|_{\log} = \sum_{i=1}^{n_3} \sum_{j=1}^{r} g(\sigma_j(\bar{\mathcal{A}}^{(i)}))$, where $g(x) = \log(\theta x + 1)$, was proposed by Geng et al. [?]. This norm controls the shrinkage degree of singular values via the parameter θ and was then applied to the TRPCA model, leading to the development of the non-convex TRPCA (N-TRPCA) model.

All the aforementioned approximation methods for tensor low-rank nuclear norms, which are based on t-SVD decomposition, are limited to third-order tensors. To address higher-order tensors, an algebraic framework for high-order tensor t-SVD was developed by Qin et al. [?], where a high-order tensor is decomposed into two high-order orthogonal tensors and a high-order f-diagonal tensor. Based on this framework, higher-order tensors were preprocessed via orthogonal transformation by Lu et al. [?], leading to the proposal of the transformed tensor nuclear norm (TTNN). However, TTNN imposes extremely strict requirements on orthogonal transformations. To relax these constraints, the mode-(k_1, k_2) unfolding was proposed by Zheng et al. [?], an operation that expands a high-order tensor into a third-order tensor. Using this unfolding, the N-tube rank is defined, and its approximation is achieved via the weighted tensor nuclear norm. Building on the above research, a multi-mode tensor singular value decomposition was proposed by Feng et al. [?] for preprocessing high-order tensors. In the initial multi-mode tensor singular value decomposition, only the fast Fourier transform was applied. Subsequently, the discrete wavelet transform and additional orthogonal transforms were incorporated into the multi-mode tensor singular value decomposition by Feng et al. [?], leading to the development of multi-transform tensor decomposition (MTTD). In MTTD, the mode-n t-product operator and the mode-$(n, n+1)$ unfolding tensor were proposed. However, owing to the mode-$(n, n+1)$ unfolding tensor, MTTD can only operate on adjacent dimensions at a time, which significantly limits the algorithm’ s flexibility. To enhance this flexibility, tTucker decomposition and tTucker multirank were proposed by Li et al. [?] based on the mode-n t-product, with the rank function approximated via the Schatten-p norm. The three-dimensional tensors generated by these tensor expansion operations exhibit severe dimensional imbalances, with excessively large differences across dimensions. To mitigate this issue, tensor square decomposition was proposed by Zhang et al. [?], which decomposes high-order tensors into three-dimensional tensors where the first and second dimensions are approximately equal. Furthermore, an improved tensor nuclear norm was proposed by Liu et al. [?] to further alleviate the dimensional imbalance problem.

To summarize, significant progress has been achieved by existing models in addressing TRPCA tasks. However, a critical review of these models and methods reveals that several limitations still remain:

- The penalty intensity for singular values varies across different norms, and an issue is often observed where large singular values are subjected to excessive penalties, while small singular values are penalized insufficiently. For example, the nuclear norm tends to overpenalize large singular values, which leads to suboptimal recovery of image texture details. Therefore, a more appropriate norm is required to better delineate the boundaries between large and small singular values, such that the penalty intensity applied to different singular values can be controlled more effectively.
- A large number of tensor unfolding operations have been proposed to handle higher-order tensors, each with its own advantages and drawbacks. Thus, the development of a simple yet efficient tensor unfolding operation remains a challenging task.

To address the aforementioned challenges, the following contributions are presented in this paper:

- To resolve the issue of singular value weighting, a novel Tensor Arctangent Norm (TAN) is proposed. This norm is capable of adaptively assigning distinct penalty weights to singular values of different magnitudes and can better delineate the boundaries between large and small singular values. Compared with existing norms, TAN exhibits stronger discriminability in distinguishing singular value boundaries, thereby providing a new framework that can be extended to other image restoration tasks.
- To handle higher-order tensors, a more efficient and easy-to-implement tensor unfolding operation (Mode-(1×4) tensor unfolding) is proposed. This operation reduces the dimensional discrepancy of the expanded tensor, thereby mitigating the dimensional imbalance issue and enabling the expanded tensor to preserve more structural information.
- The TAN is incorporated into the TRPCA model, leading to the development of the TAN-TRPCA framework. The convergence of the corresponding algorithm is proven via theoretical analysis and experimental validation, and the superior performance of TAN-TRPCA is demonstrated through comparative experiments.

The rest of this article is structured as follows. Notations and relevant works on TRPCA are introduced in Section 2. The Mode-(1×4) tensor unfolding and the TAN-TRPCA framework are discussed in detail in Section 3. The convergence of the TAN-TRPCA algorithm is analyzed from both theoretical and numerical perspectives in Section 4. Comparative experiments are presented in Section 5.

Finally, conclusions are drawn in Section 6.

2. Notations and Related Works

This section introduces some notations used in our work, as well as some basic definitions and theorems of tensors.

N-order tensors are denoted as $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_N}$. Matrices are represented by boldface capital letters (e.g., $\mathbf{A} \in \mathbb{R}^{n_1 \times n_2}$), while scalars are denoted by lowercase letters (e.g., $a \in \mathbb{R}$). The tensor $\bar{\mathcal{A}}$ is defined as the Fourier-domain representation of $\mathcal{A}$, which is obtained by applying a Fast Fourier Transform (FFT) along the third dimension of $\mathcal{A}$.

Definition 1. (Block Diagonal Matrix) For $\bar{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its block diagonal matrix is defined as $\bar{\mathbf{A}} = \text{bdiag}(\bar{\mathcal{A}}) = \begin{bmatrix} \bar{\mathbf{A}}^{(1)} & & & \\ & \bar{\mathbf{A}}^{(2)} & & \\ & & \ddots & \\ & & & \bar{\mathbf{A}}^{(n_3)} \end{bmatrix} \in \mathbb{C}^{n_1 n_3 \times n_2 n_3}$, where $\bar{\mathbf{A}}^{(i)}$ represents the i-th frontal slice of $\bar{\mathcal{A}}$.

Definition 2. (Block Circulant Matrix) For $\bar{\mathcal{A}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, its block circulant matrix is $\text{bcirc}(\mathcal{A}) = \begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{A}^{(n_3)} & \cdots & \mathbf{A}^{(2)} \\ \mathbf{A}^{(2)} & \mathbf{A}^{(1)} & \cdots & \mathbf{A}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}^{(n_3)} & \mathbf{A}^{(n_3-1)} & \cdots & \mathbf{A}^{(1)} \end{bmatrix} \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$, where $\mathbf{A}^{(i)}$ represents the i-th frontal slice of $\mathcal{A}$. The block circulant matrix of $\mathcal{A}$ ($\text{bcirc}(\mathcal{A})$) can be transformed into a block diagonal matrix of $\mathcal{A}$ ($\bar{\mathbf{A}}$) through Fourier transformation:

$$(\mathbf{F}_n \otimes \mathbf{I}_{n_1}) \cdot \text{bcirc}(\mathcal{A}) \cdot (\mathbf{F}^{-1} \otimes \mathbf{I}_{n_2}) = \bar{\mathbf{A}},$$

where $\mathbf{F}_n \in \mathbb{C}^{n \times n}$ is the discrete Fourier transformation matrix, $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is the identity matrix, and $\otimes$ is the Kronecker product.

Definition 3. (T-Product) For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$, the t-product $\mathcal{A} * \mathcal{B}$ is defined as $\mathcal{A} * \mathcal{B} = \text{fold}(\text{bcirc}(\mathcal{A}) \cdot \text{unfold}(\mathcal{B}))$, where $\text{unfold}(\mathcal{B}) := \begin{bmatrix} \mathbf{B}^{(1)} \\ \mathbf{B}^{(2)} \\ \vdots \\ \mathbf{B}^{(n_3)} \end{bmatrix} \in \mathbb{R}^{n_1 n_3 \times n_2}$. Note that in the Fourier domain, the T-product is equivalent to matrix multiplication, i.e., $\mathcal{A} * \mathcal{B} = \bar{\mathcal{A}}\bar{\mathcal{B}}$.

Definition 4. (Conjugate Transpose) The conjugate transpose of tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the tensor $\mathcal{A}^\top \in \mathbb{R}^{n_2 \times n_1 \times n_3}$ obtained by performing conjugate transpose on each frontal slice in the Fourier domain.

Definition 5. (Identity Tensor) The identity tensor $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$ is the tensor whose first frontal slice is the $n \times n$ identity matrix, and all other frontal slices are zero matrices.

Definition 6. (Orthogonal Tensor) $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an orthogonal tensor if $\mathcal{A}^\top * \mathcal{A} = \mathcal{A} * \mathcal{A}^\top = \mathcal{I}$.

Definition 7. (T-SVD) For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tensor singular value decomposition (t-SVD) of $\mathcal{A}$ is described by $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$, where $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor, $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are two orthogonal tensors.

Definition 8. (Tensor Average Rank) For tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the tensor average rank of $\mathcal{A}$ is: $\text{rank}_a(\mathcal{A}) = \text{rank}(\text{bcirc}(\mathcal{A}))$.

Definition 9. (Tensor Nuclear Norm) The tensor nuclear norm of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is $\|\mathcal{A}\|_T = \|\text{bcirc}(\mathcal{A})\|_* = \sum_{i=1}^{n_3} \|\bar{\mathcal{A}}^{(i)}\|_* = \sum_{i=1}^{n_3} \sum_{j=1}^{\min(n_1,n_2)} \sigma_j(\bar{\mathcal{A}}^{(i)})$.

When performing low-rank tensor completion, the above definition of the tensor nuclear norm (TNN) is often utilized. As described in Definition 9, TNN serves as a low-rank approximation for processing third-order tensors. To handle higher-order tensors, certain decomposition methods have been defined, such as the following decomposition.

Definition 10. (Mode-(k, k + 1) unfolding) For tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_N}$, its mode-$(k, k+1)$ unfolding tensor is $\mathcal{A}_{(k,k+1)} \in \mathbb{R}^{n_k \times n_{k+1} \times \prod_{s \neq k,k+1} n_s}$, whose frontal slices are the lexicographic ordering of the mode-$(k, k+1)$ slices of $\mathcal{A}$.

Mode-$(k, k+1)$ unfolding expands the high-order tensor into the form of a third-order tensor, so that the methods used for handling third-order tensors can be applied to handle the high-order tensor as well. TNN processes different singular values with the same weights. To handle the singular values of different sizes differently, some weighted forms of norms have been defined, such as the following tensor adjustable logarithmic norm.

Definition 11. (Tensor Adjustable Logarithmic Norm (TALN)) For tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $r = \min(n_1, n_2)$, the tensor adjustable logarithmic norm of $\mathcal{A}$ is $\|\mathcal{A}\|_{\log} = \sum_{i=1}^{n_3} \sum_{j=1}^{r} g(\sigma_j(\bar{\mathcal{A}}^{(i)}))$, where $g(x) = \log(\theta x + 1)$ is a nonconvex function with adjustable positive parameter θ.

Lemma 1. Let $\mathcal{Y} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ be the t-SVD of $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$. $\mathcal{W} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is an f-diagonal tensor whose i-th frontal slice is $\text{diag}(w_i^1, w_i^2, \dots, w_i^r)$, where $r = \min(n_1, n_2)$ and $0 \leq w_i^j \leq 1$. For any $\tau > 0$, the optimization problem:

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

has the optimal solution $\mathcal{X}^* = \mathcal{U} * \mathcal{S}_{\tau\mathcal{W}} * \mathcal{V}^\top$, where $\mathcal{S}_{\tau\mathcal{W}} = \text{fold}((\bar{\mathcal{S}} - \tau\mathcal{W})_+, [], 3)$ is a weighted singular value contraction operator, $(\cdot)_+ := \max(\cdot, 0)$.

Lemma 2. For any $\tau > 0$, and matrix $\mathbf{C} \in \mathbb{R}^{m\times n}$, the optimal solution of $\min_{\mathbf{E}} \tau\|\mathbf{E}\|_1 + \frac{1}{2}\|\mathbf{E}-\mathbf{C}\|_F^2$ is $\mathbf{E}^* = \theta_\tau[\mathbf{C}_{ij}]$, where $\theta_\tau[\mathbf{C}_{ij}] := \mathrm{sgn}(\mathbf{C}_{ij})\max(|\mathbf{C}_{ij}|-\tau, 0)$, $\mathbf{C}_{ij}$ denotes the element of $\mathbf{C}$.

Lemma 3. Suppose $F : \mathbb{R}^{n_1\times n_2} \to \mathbb{R}$ is denoted as $F(\mathbf{X}) = f \circ \sigma(\mathbf{X})$, where f is differentiable, and let $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top$ be the SVD of $\mathbf{X} \in \mathbb{R}^{n_1\times n_2}$, $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$, $n = \min(n_1, n_2)$. The gradient of $F(\mathbf{X})$ at $\mathbf{X}$ is $\partial F(\mathbf{X}) = \mathbf{U}\mathrm{diag}(\theta)\mathbf{V}^\top$, where $\theta = \partial f(y)|_{y=\sigma(\mathbf{X})}$.

3. Proposed Methods

From Definitions 6 and 11, it can be observed that both the traditional TC model and the TRPCA model are limited to handling third-order tensors. However, higher-order tensor datasets (e.g., color videos) are widely encountered in practical scenarios. Thus, higher-order tensors are intended to be converted into third-order tensors for computation via tensor unfolding operations. Extensive studies on tensor unfolding have been conducted, as exemplified by Definition 10. To better preserve the structural characteristics of the original tensor when it is unfolded into a third-order tensor, a novel tensor unfolding method is proposed in this work.

3.1. Mode - (1 × 4) Tensor Unfolding

From Definitions 6 and 11, we can see that the traditional TC model and the TRPCA model can only handle third-order tensors. In practice, there exist higher-order tensor data, such as color videos. Therefore, we intend to convert the higher-order tensors into third-order tensors for calculation through tensor unfolding. There have been many studies on tensor unfolding, such as Definition 10. In order to better preserve the structural characteristics of the original tensor when it is unfolded into a third-order tensor, we propose a new tensor unfolding method.

Definition 12. (Mode - (1 × 4) tensor unfolding) For tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_N}$, its mode-$(1\times 4)$ unfolding tensor is $\mathcal{A}_{(1\times 4)} \in \mathbb{R}^{n_1 n_4\times n_2\times\prod_{s\neq 1,2,4} n_s}$. We define the corresponding operation as $\mathcal{A}_{(1\times 4)} = \mathrm{unfold}_{(1\times 4)}(\mathcal{A})$, and its inverse operation as $\mathcal{A} = \mathrm{fold}_{(1\times 4)}(\mathcal{A}_{(1\times 4)})$. The tensor average rank of $\mathcal{A}_{(1\times 4)}$ is $\mathrm{rank}_a(\mathcal{A}_{(1\times 4)}) = \frac{1}{\prod_{s\neq 1,2,4} n_s}\mathrm{rank}(\mathrm{bcirc}(\mathcal{A}_{(1\times 4)}))$.

For example, for a four-order tensor $\mathcal{X} \in \mathbb{R}^{2\times 3\times 3\times 2}$, its Mode-$(1\times 4)$ tensor unfolding is $\mathcal{X}_{(1\times 4)} \in \mathbb{R}^{4\times 3\times 3}$. The correspondence between its elements and the elements of the original tensor is as follows:

$$\mathcal{X}_{(1\times 4)}(:,:,1) = \begin{bmatrix} \mathcal{X}(1,1,1,1) & \mathcal{X}(1,1,1,2) & \mathcal{X}(2,1,1,1) & \mathcal{X}(2,1,1,2) \\ \mathcal{X}(1,1,2,1) & \mathcal{X}(1,1,2,2) & \mathcal{X}(2,1,2,1) & \mathcal{X}(2,1,2,2) \\ \mathcal{X}(1,1,3,1) & \mathcal{X}(1,1,3,2) & \mathcal{X}(2,1,3,1) & \mathcal{X}(2,1,3,2) \end{bmatrix}$$

$$\mathcal{X}_{(1\times4)}(:,:,2)=\begin{bmatrix}\mathcal{X}(1,2,1,1) & \mathcal{X}(1,2,1,2) & \mathcal{X}(2,2,1,1) & \mathcal{X}(2,2,1,2)\\ \mathcal{X}(1,2,2,1) & \mathcal{X}(1,2,2,2) & \mathcal{X}(2,2,2,1) & \mathcal{X}(2,2,2,2)\\ \mathcal{X}(1,2,3,1) & \mathcal{X}(1,2,3,2) & \mathcal{X}(2,2,3,1) & \mathcal{X}(2,2,3,2)\end{bmatrix}$$

$$\mathcal{X}_{(1\times4)}(:,:,3)=\begin{bmatrix}\mathcal{X}(1,3,1,1) & \mathcal{X}(1,3,1,2) & \mathcal{X}(2,3,1,1) & \mathcal{X}(2,3,1,2)\\ \mathcal{X}(1,3,2,1) & \mathcal{X}(1,3,2,2) & \mathcal{X}(2,3,2,1) & \mathcal{X}(2,3,2,2)\\ \mathcal{X}(1,3,3,1) & \mathcal{X}(1,3,3,2) & \mathcal{X}(2,3,3,1) & \mathcal{X}(2,3,3,2)\end{bmatrix}$$

The tensor average rank of $\mathcal{X}_{(1\times4)}$ is $\text{rank}_a(\mathcal{X}_{(1\times4)})=\frac{1}{3}\text{rank}(\text{bcirc}(\mathcal{X}_{(1\times4)}))$.

An example of applying the Mode-(1×4) tensor unfolding is shown in Fig. 2.

[Figure 2: see original paper]

3.2. TAN-TRPCA Model

In the traditional T-SVT method, when solving the problem of minimizing TNN, the same threshold τ is used to shrink different singular values. In real scenarios, there are significant differences among the singular values of tensors. For instance, large singular values in color images contain important texture information of the image, while small singular values typically represent noise. In order to preserve the important texture information of the tensor and remove the noise within it, we should shrink large singular values less and small singular values more. To achieve this goal, we propose the following tensor arctangent norm.

Definition 13. (Tensor Arctangent Norm (TAN)) For tensor $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times n_3}$, $r=\min(n_1,n_2)$, the tensor arctangent norm of $\mathcal{A}$ is $\|\mathcal{A}\|_{\text{atan}}=\sum_{i=1}^{n_3}\sum_{j=1}^{r}f(\sigma_j(\bar{\mathcal{A}}^{(i)}))$, where $f(x)=\theta\times\arctan(x)$ is the arctangent function.

When compared with the traditional tensor nuclear norm (TNN), the Arctangent Norm (ATN) is capable of adaptively assigning distinct penalty weights to singular values of different magnitudes. As demonstrated in Definition 11, the Tensor Adjustable Logarithmic Norm (TALN) is also designed to differentiate between singular values to preserve the structural characteristics of the tensor. However, owing to the differences between the function $g(x)=\log(\theta x+1)$ and $f(x)=\theta\times\arctan(x)$, superior discriminability in distinguishing the magnitudes of different singular values is achieved by ATN. Figure 3 presents the performance comparison of three representative functions: the function $f(x)=x$, the logarithmic function $f(x)=\log(\theta x+1)$ (TALN), and the proposed arctangent function $f(x)=\theta\times\arctan(x)$ (TAN).

As observed from Fig. 3, the proposed function exhibits a distinct curve, which clearly indicates that large and small singular values can be well differentiated by this function. As the magnitudes of singular values increase, a gradual decline in the values of the proposed function below those of the logarithmic function is observed, which demonstrates that the penalty intensity imposed on large singular values by the proposed function is progressively reduced.

[Figure 3: see original paper]

We propose a TRPCA model based on TAN (TAN-TRPCA):

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

where λ is a regularization parameter, $\|\cdot\|_{\text{atan}}$ is defined as in Definition 13, $\mathcal{X}$ is the observed tensor with noise, $\mathcal{L}$ is the low-rank tensor, and $\mathcal{E}$ is the sparse error tensor.

3.3. Optimization Algorithm of TAN-TRPCA

To solve Model (16), we introduce the ADMM algorithm. The augmented Lagrangian function of (16) is:

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

where $\mathcal{P}$ is a Lagrange multiplier and μ is a penalty parameter. Next, we will solve for $\mathcal{L}$, $\mathcal{E}$, $\mathcal{P}$, and μ through iteration.

Update $\mathcal{L}^k$: Fixing the parameters $\mathcal{E}^k$, $\mathcal{P}^k$, and μ^k, the iterative form of $\mathcal{L}^k$ is obtained as:

$$\begin{aligned}
\mathcal{L}^{k+1} &= \arg\min_{\mathcal{L}} \mathcal{L}(\mathcal{L},\mathcal{E}^k,\mathcal{P}^k,\mu^k) \\
&= \arg\min_{\mathcal{L}} \|\mathcal{L}\|_{\text{atan}} + \lambda\|\mathcal{E}^k\|_1 + \langle \mathcal{P}^k, \mathcal{L}+\mathcal{E}^k-\mathcal{X}\rangle + \frac{\mu^k}{2}\|\mathcal{L}+\mathcal{E}^k-\mathcal{X}\|_F^2 \\
&= \arg\min_{\mathcal{L}} \|\mathcal{L}\|_{\text{atan}} + \frac{\mu^k}{2}\left\|\mathcal{L}-\mathcal{X}-\mathcal{E}^k-\frac{\mathcal{P}^k}{\mu^k}\right\|_F^2.
\end{aligned}$$

Let $\mathcal{Q}^k = \mathcal{X} - \mathcal{E}^k - \frac{\mathcal{P}^k}{\mu^k}$. Substituting Definition 13 into Eq (18):

$$\mathcal{L}^{k+1} = \arg\min_{\mathcal{L}} \sum_{i=1}^{n_3}\sum_{j=1}^{r} f(\sigma_j(\bar{\mathcal{L}}^{(i)})) + \frac{\mu^k}{2}\|\mathcal{L}-\mathcal{Q}^k\|_F^2.$$

We denote the j-th singular value of $\bar{\mathcal{L}}^{(i)}$ as σ_{ij} (i.e., $\sigma_{ij} = \sigma_j(\bar{\mathcal{L}}^{(i)})$). When the number of iterations is k, $\sigma_{ij}^k = \sigma_j(\bar{\mathcal{L}}^{(i)})$ represents the j-th singular value of $\bar{\mathcal{L}}^{(i)}$ at the k-th iteration. When k is fixed, σ_{ij}^k becomes a constant. In order to assign different weights to different singular values, the function $f(\sigma_{ij})$ is approximated by its first-order Taylor expansion at σ_{ij}^k:

$$f(\sigma_{ij}) = f(\sigma_{ij}^k) + w_{ij}^k(\sigma_{ij} - \sigma_{ij}^k),$$

where $w_{ij}^k = \frac{\partial f(\sigma_{ij}^k)}{\partial \sigma_{ij}} = \frac{\theta}{(\sigma_{ij}^k)^2+1}$ is the weight assigned to σ_{ij}. From this, it can be seen that as the singular value decreases ($\sigma_{i1}^k \geq \sigma_{i2}^k \geq \cdots \geq \sigma_{ir}^k \geq 0$), its weight increases ($0 \leq w_{i1}^k \leq w_{i2}^k \leq \cdots \leq w_{ir}^k$). Let $\mathcal{W}^{(i)} = \mathrm{diag}(w_{i1}^k, w_{i2}^k, \dots, w_{ir}^k)$ be the i-th frontal slice of weight tensor $\mathcal{W}$. By using the weight tensor, we achieve the goal of shrinking large singular values less and small singular values more.

Then substituting Eq (20) into Eq (19), we have:

$$\mathcal{L}^{k+1} = \arg\min_{\mathcal{L}} \sum_{i=1}^{n_3}\sum_{j=1}^{r}\left(f(\sigma_{ij}^k) + w_{ij}^k(\sigma_{ij} - \sigma_{ij}^k)\right) + \frac{\mu^k}{2}\|\mathcal{L} - \mathcal{Q}^k\|_F^2 = \arg\min_{\mathcal{L}} \sum_{i=1}^{n_3}\sum_{j=1}^{r} w_{ij}^k\sigma_{ij} + \frac{\mu^k}{2}\|\mathcal{L} - \mathcal{Q}^k\|_F^2.$$

According to Lemma 1, we can get:

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

where $\mathcal{Q}^k = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top$ is the t-SVD of $\mathcal{Q}^k$.

Update $\mathcal{E}^k$: With parameters $\mathcal{L}^{k+1}$, $\mathcal{P}^k$, and μ^k fixed, $\mathcal{E}^k$ can be obtained by:

$$\begin{aligned}
\mathcal{E}^{k+1} &= \arg\min_{\mathcal{E}} \mathcal{L}(\mathcal{L}^{k+1}, \mathcal{E}, \mathcal{P}^k, \mu^k) \\
&= \arg\min_{\mathcal{E}} \|\mathcal{L}^{k+1}\|_T + \lambda\|\mathcal{E}\|_1 + \langle \mathcal{P}^k, \mathcal{L}^{k+1} + \mathcal{E} - \mathcal{X}\rangle + \frac{\mu^k}{2}\|\mathcal{L}^{k+1} + \mathcal{E} - \mathcal{X}\|_F^2 \\
&= \arg\min_{\mathcal{E}} \lambda\|\mathcal{E}\|_1 + \frac{\mu^k}{2}\left\|\mathcal{E} - \left(\mathcal{X} - \mathcal{L}^{k+1} - \frac{\mathcal{P}^k}{\mu^k}\right)\right\|_F^2.
\end{aligned}$$

From Lemma 2, we get:

$$\mathcal{E}^{k+1} = \theta_{\lambda/\mu^k}\left[\mathcal{X} - \mathcal{L}^{k+1} - \frac{\mathcal{P}^k}{\mu^k}\right].$$

Update $\mathcal{P}^k$: Similarly, holding $\mathcal{L}^{k+1}$, $\mathcal{E}^{k+1}$, and μ^k fixed, we can obtain:

$$\mathcal{P}^{k+1} = \mathcal{P}^k + \mu^k(\mathcal{L}^{k+1} + \mathcal{E}^{k+1} - \mathcal{X}).$$

Update μ^k:

$$\mu^{k+1} = \min(\rho^k \mu^k, \mu_{\max}) \quad (\rho^k = 0.99\rho^{k-1}).$$

Algorithm 1 shows the overall TAN-TRPCA. The details of the algorithm operation are shown in Fig. 4.

Algorithm 1 TAN-TRPCA

Input: Observed tensor $\mathcal{X}$, λ, μ_0, $\mu_{\max}$, $\rho_0 = 9.8$, ϵ, θ.

Output: Recovered tensor $\mathcal{L}$

1. Unfold tensor $\mathcal{X}$ as Definition 12.
2. While not converged do
   - Update $\mathcal{L}^{k+1}$ as Eq (22);
   - Update $\mathcal{E}^{k+1}$ as Eq (23);
   - Update $\mathcal{P}^{k+1}$ as Eq (25);
   - Update μ^{k+1} as Eq (26);
   - Check the convergence conditions $\|\mathcal{L}^{k+1} - \mathcal{L}^k\|_\infty \leq \epsilon$, $\|\mathcal{E}^{k+1} - \mathcal{E}^k\|_\infty \leq \epsilon$, $\|\mathcal{L}^{k+1} + \mathcal{E}^{k+1} - \mathcal{X}\|_\infty \leq \epsilon$;
3. End while

[Figure 4: see original paper]

4. Convergence Analysis

In this section, we rigorously verify the convergence of Algorithm 1 from both theoretical proof and experimental validation perspectives.

Theorem 1. The sequences of tensors $\mathcal{L}^k$, $\mathcal{E}^k$, and $\mathcal{P}^k$ generated by Algorithm 1 are bounded, and the $\{\mathcal{L}^*, \mathcal{E}^*, \mathcal{P}^*\}$ produced by Eq (17) satisfy the KKT conditions:

$$0 \in \partial\|\mathcal{L}\|_{\text{atan}} + \mathcal{P}^*, \quad \mathcal{L}^* + \mathcal{E}^* = \mathcal{X}, \quad 0 \in \partial\|\mathcal{E}\|_1 + \mathcal{P}^*.$$

Proof. First, we will prove that $\mathcal{L}^k$, $\mathcal{E}^k$, and $\mathcal{P}^k$ are bounded. From the first-order necessary condition of the sub-gradient of $\mathcal{E}^{k+1}$, it can be known that:

$$0 \in \partial_{\mathcal{E}} \mathcal{L}(\mathcal{L}^{k+1}, \mathcal{E}^{k+1}, \mathcal{P}^k, \mu^k) = \partial(\lambda\|\mathcal{E}^{k+1}\|_1) + \mathcal{P}^k + \mu^k(\mathcal{L}^{k+1} - \mathcal{X} + \mathcal{E}^{k+1}).$$

From the iterative form of $\mathcal{P}^{k+1} = \mathcal{P}^k + \mu^k(\mathcal{L}^{k+1} - \mathcal{X} + \mathcal{E}^{k+1})$, we have:

$$0 \in \partial(\lambda\|\mathcal{E}^{k+1}\|_1) + \mathcal{P}^{k+1}.$$

Since $\lambda\|\mathcal{E}^{k+1}\|_1$ is non-smooth when $\mathcal{E}^k_{ij} = 0$, we redefine the sub-gradient $[\partial\|\mathcal{E}\|_1]^k_{ij} = 0$ at $\mathcal{E}^k_{ij} = 0$. Then $0 \le \|\partial\|\mathcal{E}\|_1\|_F^2 \le n_1 n_2 n_3$, so $\partial(\lambda\|\mathcal{E}^{k+1}\|_1)$ is bounded. From Eq (28), $\mathcal{P}^k$ is bounded.

For Eq (17), we have:

$$\begin{aligned}\mathcal{L}(\mathcal{L}^k, \mathcal{E}^k, \mathcal{P}^k, \mu^k) - \mathcal{L}(\mathcal{L}^k, \mathcal{E}^k, \mathcal{P}^{k-1}, \mu^{k-1}) &= \langle \mathcal{P}^k - \mathcal{P}^{k-1}, \mathcal{L}^k - \mathcal{X} + \mathcal{E}^k \rangle + \frac{\mu^k - \mu^{k-1}}{2}\|\mathcal{L}^k - \mathcal{X} + \mathcal{E}^k\|_F^2 \\ &= \mathrm{tr}[(\mathcal{P}^k - \mathcal{P}^{k-1})(\mathcal{L}^k - \mathcal{X} + \mathcal{E}^k)] + \frac{\mu^k - \mu^{k-1}}{2}\|\mathcal{L}^k - \mathcal{X} + \mathcal{E}^k\|_F^2.\end{aligned}$$

From Eq (25), $\mathcal{P}^k - \mathcal{P}^{k-1} = \mu^{k-1}(\mathcal{L}^k - \mathcal{X} + \mathcal{E}^k)$. Substituting this into Eq (29), we get:

$$\mathcal{L}(\mathcal{L}^k, \mathcal{E}^k, \mathcal{P}^k, \mu^k) - \mathcal{L}(\mathcal{L}^k, \mathcal{E}^k, \mathcal{P}^{k-1}, \mu^{k-1}) = \frac{\mu^k + \mu^{k-1}}{2(\mu^{k-1})^2}\|\mathcal{P}^k - \mathcal{P}^{k-1}\|_F^2 > 0.$$

Therefore:

$$\mathcal{L}(\mathcal{L}^{k+1}, \mathcal{E}^{k+1}, \mathcal{P}^k, \mu^k) \le \mathcal{L}(\mathcal{L}^{k+1}, \mathcal{E}^k, \mathcal{P}^k, \mu^k) \le \mathcal{L}(\mathcal{L}^k, \mathcal{E}^k, \mathcal{P}^k, \mu^k) \le \mathcal{L}(\mathcal{L}^k, \mathcal{E}^k, \mathcal{P}^{k-1}, \mu^{k-1}) + \frac{\mu^k + \mu^{k-1}}{2(\mu^{k-1})^2}\|\mathcal{P}^k - \mathcal{P}^{k-}$$

Thus:

$$\mathcal{L}(\mathcal{L}^{k+1}, \mathcal{E}^{k+1}, \mathcal{P}^k, \mu^k) \le \mathcal{L}(\mathcal{L}^1, \mathcal{E}^1, \mathcal{P}^0, \mu^0) + \sum_{i=1}^{k} \frac{\mu^i + \mu^{i-1}}{2(\mu^{i-1})^2}\|\mathcal{P}^i - \mathcal{P}^{i-1}\|_F^2.$$

The above proves that $\mathcal{P}^k$ is bounded, so $\sum_{i=1}^{k} \frac{\mu^i+\mu^{i-1}}{2(\mu^{i-1})^2}\|\mathcal{P}^i - \mathcal{P}^{i-1}\|_F^2$ is also bounded. Therefore, $\mathcal{L}(\mathcal{L}^{k+1}, \mathcal{E}^{k+1}, \mathcal{P}^k, \mu^k)$ has an upper bound. It can be directly shown that:

$$\mathcal{L}(\mathcal{L}^{k+1}, \mathcal{E}^{k+1}, \mathcal{P}^k, \mu^k) + C = \|\mathcal{L}^{k+1}\|_{\log} + \lambda\|\mathcal{E}^{k+1}\|_1 + \|\mathcal{L}^{k+1} - \mathcal{X} + \mathcal{E}^{k+1}\|_F^2 + \|\mathcal{P}^k\|_F^2.$$

In Eq (34), $\mathcal{E}^k$, $\mathcal{P}^k$, and μ^k are bounded, so $\mathcal{L}^k$ is bounded.

Next, we will prove that the $\{\mathcal{L}^*, \mathcal{E}^*, \mathcal{P}^*\}$ produced by Eq (17) satisfy the KKT conditions $0 \in \partial\|\mathcal{L}\|_{\mathrm{atan}} + \mathcal{P}^*$, $\mathcal{L}^* + \mathcal{E}^* = \mathcal{X}$, and $0 \in \partial\|\mathcal{E}\|_1 + \mathcal{P}^*$. According to the Bolzano–Weierstrass theorem, we know that each infinite bounded sequence in $\mathbb{R}^n$ contains a convergent subsequence. Thus, the sequence $\{\mathcal{L}^k, \mathcal{E}^k, \mathcal{P}^k\}_{k=1}^{\infty}$ must have at least one accumulation point. Without loss of generality, we may assume the sequence itself converges to such a limit point, which we denote as $\{\mathcal{L}^*, \mathcal{E}^*, \mathcal{P}^*\}$. From Eq (30), we have:

$$\lim_{k\to\infty}(\mathcal{L}^{k+1} + \mathcal{E}^{k+1} - \mathcal{X}) = \lim_{k\to\infty}\frac{\mathcal{P}^{k+1} - \mathcal{P}^k}{\mu^k} = 0.$$

So the original feasibility condition $\mathcal{L}^* + \mathcal{E}^* = \mathcal{X}$ is achieved. $\mathcal{L}^{k+1}$ also satisfies the first-order necessary local optimality condition of Eq (18):

$$0 \in \partial\|\mathcal{L}^{k+1}\|_{\text{atan}} + \mathcal{P}^k + \mu^k(\mathcal{L}^{k+1} + \mathcal{E}^k - \mathcal{X}).$$

From Definition 13 and Lemma 3, for $i = 1, \dots, n_3$, we have:

$$\nabla\|\bar{\mathcal{L}}^{(i)}\|_{\text{atan}} = \bar{\mathbf{U}}^{(i)}\text{diag}\left(\frac{\theta}{(\sigma_j(\bar{\mathcal{L}}^{(i)}))^2 + 1}\right)\bar{\mathbf{V}}^{(i)\top},$$

so $\nabla\|\bar{\mathcal{L}}^{(i)}\|_{\text{atan}}$ is bounded. Thus $\bar{\mathcal{L}}$ is bounded. By restoring it to the original tensor domain, we obtain $\partial\|\mathcal{L}\|_{\text{atan}} = \nabla\|\mathcal{L}\|_{\text{atan}} = \partial\|\mathcal{L}\|_{\text{atan}} \times_3 \tilde{\mathbf{F}}_{n_3}^*$, so $\nabla\|\mathcal{L}\|_{\text{atan}}$ is bounded. From Eq (25), we get $\mu^k(\mathcal{L}^{k+1} - \mathcal{X}) = \mathcal{P}^{k+1} - \mathcal{P}^k - \mu^k\mathcal{E}^{k+1}$. Substituting this into Eq (36):

$$0 \in \partial\|\mathcal{L}^{k+1}\|_{\text{atan}} + \mathcal{P}^{k+1} - \mu^k(\mathcal{E}^{k+1} - \mathcal{E}^k).$$

Taking the limits on both sides yields $0 \in \partial\|\mathcal{L}\|_{\text{atan}} + \mathcal{P}^*$.

Similarly, for $\mathcal{E}^{k+1}$, its first-order necessary condition is:

$$0 \in \partial\|\mathcal{E}^{k+1}\|_1 + \mathcal{P}^k + \mu^k(\mathcal{L}^k + \mathcal{E}^{k+1} - \mathcal{X}).$$

Taking the limit for each term results in $0 \in \partial\|\mathcal{E}\|_1 + \mathcal{P}^*$. Thus, $(\mathcal{L}^*, \mathcal{E}^*, \mathcal{P}^*)$ satisfies the Karush-Kuhn-Tucker (KKT) conditions of the Lagrange function $\mathcal{L}(\mathcal{L}, \mathcal{E}, \mathcal{P}, \mu)$.

To better intuitively understand the convergence of the algorithm, we randomly selected 5 color images. The PSNR and error values generated during the iterative process are shown in Fig. 5. From Fig. 5, we can observe that after 100 iterations, the algorithm gradually becomes stable, which is in line with our theoretical analysis.

[Figure 5: see original paper]

5. Experimental Results

In this section, we first determined the optimal parameters of the algorithm through experiments. Then, we conducted comparative experiments on color images, hyperspectral videos, and color videos. By comparing the PSNR and SSIM values with the experimental results of SNN [?], TNN [?], N [?], TTSP [?], HTNN [?], and MTTD [?], we found that the algorithm in this paper has better experimental effects.

In order to make the running environment of all algorithms the same, all algorithms are implemented in the environment of MATLAB R2020a (Intel(R) Xeon(R) CPU E31245 @ 3.30GHz). The specific experimental results are described below.

[Figure 6: see original paper]

5.1. Parameter Selection

There are two parameters ρ and θ in the TAN-TRPCA. Parameter ρ is related to the iterative update of sub-problem μ. An excessively large ρ will cause μ to grow rapidly, thereby leading to missing the optimal point. On the other hand, an excessively small ρ will result in slow growth of μ, reducing the recovery effect of the algorithm. To determine the optimal parameter ρ, we conducted a comparative experiment on four randomly selected color images as shown in Fig. 6. From Fig. 6, we can see that the algorithm performs the best when $\rho = 9$. Therefore, in the subsequent experiments, we will set $\rho = 9$.

Additionally, the parameter θ is related to the singular values of the tensor to be processed. Smaller singular values require a larger θ, while larger singular values require a smaller θ. Determining the appropriate parameter θ can enhance the recovery effect of the algorithm. We discussed θ within the range of 1 to 10, and conducted experiments on 4 randomly selected images, as shown in Fig. 7.

[Figure 7: see original paper]

In Fig. 7, we observe that when $\theta = 2$, the parameter θ imposes an appropriate level of penalty on the singular values. In the subsequent experiments, we set $\theta = 2$.

5.2. Color Image Recovery

Color images were applied in our experiment as a third-order tensor. We randomly selected 100 color images from the color image database BSD500, which included landscapes, buildings, people, animals, and plants. Each image is a tensor $\mathcal{M} \in \mathbb{R}^{630\times 840\times 3}$. We added 10%, 20%, 30%, and 50% of random noise to the test images, meaning that 10%, 20%, 30%, and 50% of the pixel values in the images were randomly disrupted at different positions. On these 100 color images, our algorithm was compared with the results of SNN, TNN, N, TTSP, HTNN, and MTTD algorithms. The mean PSNR results and mean SSIM results are recorded in Table 1.

Due to the added noise damaging the pixels at random positions in the image, different experimental results would be obtained for the same picture. We conducted 20 experiments on each image, each time the obtained mean value is different, so the error ranges obtained are also recorded in Table 1. By observing Table 1, we can see that under different noise conditions, our algorithm achieves better mean PSNR and mean SSIM values.

To better understand the visual effects of different restoration methods on images, we randomly selected 8 experimental results of color images and presented them as shown in Fig. 8. The results shown in Fig. 8 are from top to bottom: 10%, 20%, 30%, and 50% of noise have been added respectively, and for each level of noise, two images have been selected. From left to right are the original image, the noisy image, SNN, TNN, N, TTSP, HTNN, MTTD, and TAN. Among the HTNN algorithm, the FFT is used. During the presentation, we highlighted some specific details and magnified them for closer inspection. Through this, we could observe that the image restoration effect using the high-order tensor decomposition algorithm has achieved a significant improvement.

[Figure 8: see original paper]

5.3. Color Video Restoration

Color video, as a high-order tensor, is used to verify the effectiveness of our algorithm. We downloaded six color videos from the database, namely 'akiyo' , 'bridge-close' , 'suzie' , 'carphone' , 'coastguard' , and 'flower' . We take the first 50 frames of each video to form a fourth-order tensor $\mathcal{M} \in \mathbb{R}^{840\times 630\times 3\times 50}$. The average values of PSNR and SSIM obtained by different algorithms are recorded in Table 2 as follows. Similarly, we ran each of the above six videos 20 times, and the average values obtained each time were different. Therefore, we provided a range of values for them. The SNN, TNN, N, and TTSP algorithms cannot process the tensor as a whole. Therefore, we will separately process each frame of the color image using these algorithms and record the results obtained.

We added 20% noise to the video 'carphone' and selected the 5th frame and the 20th frame for the experimental comparison results, which are shown in Fig. 9. The third and fourth lines show the experimental results of the 5th and 50th frames of the video 'suzie' . The noise we added to these frames was 30%.

[Figure 9: see original paper]

In Fig. 9, the actions of the characters in the 5th frame and the 50th frame of each video are different. By observing some details, we can notice that there is some blurriness in the recovery results of SNN and TNN. This is because these algorithms process each frame separately. In contrast, the algorithm that directly processes the fourth-order tensor has better experimental results.

5.4. Hyperspectral Video Recovery

In this experiment, the original hyperspectral video has a size of $480 \times 752 \times 33 \times 31$, containing 31 frames. Each frame has 33 bands from 400 nm to 720 nm wavelength with a 10 nm step. Upsampling is performed on each band of the original HSV, resulting in a smaller tensor $\mathcal{M} \in \mathbb{R}^{630\times840\times33\times31}$ that serves as the test data.

Just as we did with color videos, a comparison experiment has been conducted on hyperspectral videos. The PSNR and SSIM values of the experimental results are shown in Table 3.

Hyperspectral videos, unlike color images or color videos, do not allow each frame to be displayed separately. Our experimental subjects consist of 33 frames, each of which has multiple bands. The experimental results of the 10th band of the 5th frame, the 7th band of the 10th frame, and the 15th band of the 20th frame are shown in Fig. 10. In Fig. 10, from top to bottom, 10%, 20%, and 30% of noise were added respectively. From left to right, they are the original data, the observed data, SNN, TNN, N, TTSP, HTNN, MTTD, and TAN. Among them, ANN, TNN, N, and TTSP operate on each band of every frame of the hyperspectral video. It can be seen that the results obtained by these individual processing algorithms are somewhat ambiguous. However, the algorithm we proposed has better experimental results.

[Figure 10: see original paper]

6. Conclusion

In this paper, an arctangent (atan) norm based on the arctangent function is proposed. Compared with the traditional nuclear norm, this norm can be used to adaptively assign distinct penalty weights according to the magnitudes of singular values, with greater weights imposed on small singular values and smaller weights on large ones. As a result, texture information can be better preserved, and noise can be accurately removed during tensor data denoising. When compared with state-of-the-art weighted nuclear norms, the proposed atan norm can better delineate the boundaries between large and small singular values, thereby yielding improved experimental results. To extend the applicability of this method to higher-order tensors, a tensor unfolding operation is also proposed. This operation converts higher-order tensors into third-order tensors via a simple tensor transformation, allowing the algorithm to be naturally extended to higher-order cases and enabling the processing of higher-order tensor data.

In the future, the following three research directions will be prioritized: (1) A parameter θ that can be adaptively adjusted according to the magnitudes of different singular values will be introduced. This parameter is expected to be precisely estimated via deep learning-based algorithms, thereby enhancing the overall accuracy of the algorithm. (2) The TAN framework will be integrated with deep learning algorithms to expand its applicability, and the TAN method proposed in this study will be applied to a broader range of tensor processing tasks. (3) Advanced transformation or decomposition strategies will be developed to enable the processing of higher-order tensors.

Acknowledgments

This study was supported by the National Natural Science Foundation of P.R. China (Grant No. 12171064), Chongqing Postgraduate Research and Innovation Project (Grant No. CYB25249).

Declarations

Conflict of interest: The authors declare no conflicts of interest.

Authors' contributions: Two authors contributed equally to this work.

Availability of data and materials: This manuscript has no associated data or the data will not be deposited.

References

[1] L. Qi, Z. Luo. Tensor analysis: spectral theory and special tensors. Society for Industrial and Applied Mathematics, 2017.

[2] H. Wang, F. Zhang, J. Wang, et al. Generalized nonconvex approach for low-tubal-rank tensor recovery. IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(8): 3305-3319.

[3] Y. Tan, D. Hu, Y. Fu, et al. Generalized Nonconvex Low-Rank Approximation and Sparse Regularizer for Hyperspectral Image Denoising. IEEE Transactions on Geoscience and Remote Sensing, 2025, 63: 1-17.

[4] J. A. Bengua, H. N. Phien, H. D. Tuan, et al. Efficient Tensor Completion for Color Image and Video Recovery: Low-Rank Tensor Train, IEEE Transactions on Image Processing, 2017, 26(5): 2466-2479.

[5] H. Liu, Z. Fang, W. Lu. Noise level estimation based on eigenvalue learning. Multimedia Tools and Applications,