Received: 27 August 2022 | Revised: 11 November 2022 | Accepted: 23 December 2022

DOI: 10.1002/nla.2489

RESEARCH ARTICLE

WILEY

Nonconvex optimization for third-order tensor completion under wavelet transform

Quan Yu | **Minru Bai**

School of Mathematics, Hunan University, Changsha, China

Correspondence
Minru Bai, School of Mathematics, Hunan University, Changsha 410082, China.
Email: minru-bai@hnu.edu.cn

Funding information
Hunan Provincial Key Laboratory of Intelligent Information Processing and Applied Mathematics; National Natural Science Foundation of China, Grant/Award Numbers: 11971159, 12071399

Abstract

The main aim of this paper is to develop a nonconvex optimization model for third-order tensor completion under wavelet transform. On the one hand, through wavelet transform of frontal slices, we divide a large tensor data into a main part tensor and three detail part tensors, and the elements of these four tensors are about a quarter of the original tensors. Solving these four small tensors can not only improve the operation efficiency, but also better restore the original tensor data. On the other hand, by using concave correction term, we are able to correct for low rank of tubal nuclear norm (TNN) data fidelity term and sparsity of l_1-norm data fidelity term. We prove that the proposed algorithm can converge to some critical point. Experimental results on image, magnetic resonance imaging and video inpainting tasks clearly demonstrate the superior performance and efficiency of our developed method over state-of-the-arts including the TNN and other methods.

KEYWORDS

difference of convex functions, low rank, sparse, tensor completion, wavelet transform

1 | INTRODUCTION

Tensor completion, which recovers missing elements based on the known data, has received extensive research and increasing attentions, such as image/video inpainting,[1-7] medical image processing,[8,9] high altitude aerial image inpainting,[10] hyperspectral data recovery,[11,12] and internet traffic recovery.[13-15] Generally, the low rank tensor completion (LRTC) problem can be addressed by the rank minimization problem:

$$\min_{\mathcal{X}} \operatorname{rank}(\mathcal{X}), \quad \text{s.t.} \quad P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M}), \tag{1}$$

where $\mathcal{M}$ is an observed incomplete tensor and Ω is the index set corresponding to the observed entries of $\mathcal{M}$, and $P_\Omega(\cdot)$ is the sampling operator that remains the elements in Ω while making the others to be zeros.

However, unlike the matrix rank, there is no unique definition of the tensor rank. Among the known tensor ranks, Tucker rank and CANDECOMP/PARAFAC (CP) rank are the most widely used, and they correspond to Tucker decomposition[16] and CP decomposition[17,18] of tensors, respectively. It is NP-hard to compute CP rank of a tensor.[19-21] However, the Tucker rank is based on the matrix rank and thus computable. Therefore, the LRTC problem is mostly based on Tucker rank model. For example, Liu et al.[22] proposed the sum of nuclear norms of unfolding matrices of a tensor to approximate the Tucker rank minimization for tensor completion. However, unfolding a tensor as a matrix would destroy its original multiway structure, resulting in vital information being lost and decreasing performance.[21,23] Note that the rows and columns of the expanded matrix are very different, which is very unfavorable for matrix restoration. Xu et al.[24] introduced a alternating proximal gradient method for sparse nonnegative Tucker decomposition with missing values. However, this model does not take advantage of the low rank structure of the factor matrix and needs to predict the rank of the tensor in advance.

In a recent paper, a new decomposition method for third-order tensors called tensor singular value decomposition (t-SVD)[25,26] has been proposed. This method decomposes a tensor into the product of two orthogonal tensors and one f-diagonal tensor (see Section 2 for details). With the help of the t-SVD framework, the tensor multi-rank and tubal rank were proposed by Kilmer et al.[27] As the t-SVD is based on an operator theoretic interpretation of third-order tensors as linear operators on the space of oriented matrices, the tubal rank and multi-rank of the tensor describe the inherent low rank structure of the tensor without the loss of information inherent in matricization.[27,28] Then, Semerci et al.[29] developed a new tensor nuclear norm called TNN. Based on TNN, Zhang et al.[25] studied the tensor completion problem. From the definition of TNN, it can be seen that it is essentially the l_1-norm of all front slice singular value vectors of the tensor after Fourier transform. However, statisticians have long known that the l_1-norm penalty yields biased estimators and cannot achieve the best estimation performance.[30] In other words, TNN and l_1-norm will produce biased estimators. Growing evidence supports the use of nonconvex sparse (low rank) formulations to improve model fidelity and generalization.[31-33] For example, Jiang et al.[34] proposed partial sum of TNN for tensor recovery. Zhao et al.[12] proposed concave smooth correction of l_1-norm which is continuous, sparsity promoting and unbiasedness. Numerical tests showed that the concave smooth correction of l_1-norm improves the sparsity of the data fidelity term greatly. Thus, we add concave correction term to both TNN and l_1-norm.

However, TNN and other corresponding nonconvex TNN require computing t-SVD, which can take a lot of time when the data scale is large. On the other hand, model (1) does not solve the main part and the detail part of the multi-dimensional visual data separately, which leads to the fact that the detail part is easily lost. In order to take full advantage of the intrinsic structure in multi-dimensional visual data and improve the computational efficiency, we propose a novel tensor completion model based low rank and sparse representation under wavelet transform, which can characterize the internal structure of the main part and the detail part of the data very well. More precisely, the proposed wavelet transform of frontal slices involves two steps:

- In the first step, we use a single-level discrete two-dimensional wavelet transform to transform each frontal slice $X^{(k)} := \mathcal{X}(:,:,k),\ k = 1, \dots, n_3$ of tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ into four elements also known as subbands namely $A^{(k)}$, $H^{(k)}$, $V^{(k)}$, and $D^{(k)}$ with size $\lceil n_1/2 \rceil \times \lceil n_2/2 \rceil$. The approximation of the original image $X^{(k)}$ is known as the $A^{(k)}$ subband. The remaining three subbands are known as details, which represent components of wavelet coefficients, and are referred to as horizontal details, vertical details, and diagonal details, respectively. Mathematically, it is expressed as

$$WX^{(k)} = \left\{A^{(k)}, H^{(k)}, V^{(k)}, D^{(k)}\right\},$$

  where W is wavelet transform.
- In the second step, we construct four new tensors $\mathcal{A}$, $\mathcal{H}$, $\mathcal{V}$, $\mathcal{D}$ such that $\mathcal{A}(:,:,k) := A^{(k)}$, $\mathcal{H}(:,:,k) := H^{(k)}$, $\mathcal{V}(:,:,k) := V^{(k)}$ and $\mathcal{D}(:,:,k) := D^{(k)}$ for all $k = 1, \dots, n_3$.

Figure 1 shows wavelet transform of the color image "House." Figure 2 shows the comparison of the proportion of singular values of $\mathcal{X}$, $\mathcal{A}$, $\mathcal{H}$, $\mathcal{V}$, and the distributions of the pixel values of $\mathcal{D}$. From Figure 2, we can observe that $\mathcal{A}$, $\mathcal{H}$, and $\mathcal{V}$ are low rank and $\mathcal{D}$ is sparse.

Based on the above analysis, we introduce a nonconvex optimization model for tensor completion under wavelet transform:

$$\begin{aligned} \min_{\mathcal{X}} \quad & \lambda_A \|\mathcal{A}\|_{\circledast,\theta_1} + \lambda_H \|\mathcal{H}\|_{\circledast,\theta_1} + \lambda_V \|\mathcal{V}\|_{\circledast,\theta_1} + \lambda_D \left(\|\mathcal{D}\|_1 - \Psi_{\theta_2}(\mathcal{D})\right), \\ \text{s.t.} \quad & P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M}), \quad W\mathcal{X} = \{\mathcal{A}, \mathcal{H}, \mathcal{V}, \mathcal{D}\}, \end{aligned} \tag{2}$$

where regularization parameter λ_A, λ_H, λ_V, λ_D control the trade off. $\|\mathcal{C}\|_{\circledast,\theta} = \|\mathcal{C}\|_* - Q_\theta(\mathcal{C})$, $Q_\theta(\mathcal{C}) = \frac{1}{n_3}\sum_{k=1}^{n_3} \Psi_\theta\left(\sigma\left(\overline{C}^{(k)}\right)\right)$, the definition of $\|\mathcal{C}\|_*$ in Definition 8, Ψ_θ is a convex and continuous differential function which is defined as

$$\Psi_\theta(\mathcal{C}) = \sum_{l=1}^{\dim(\mathcal{C})} \psi_\theta\left(\operatorname{vec}(\mathcal{C})_l\right),$$

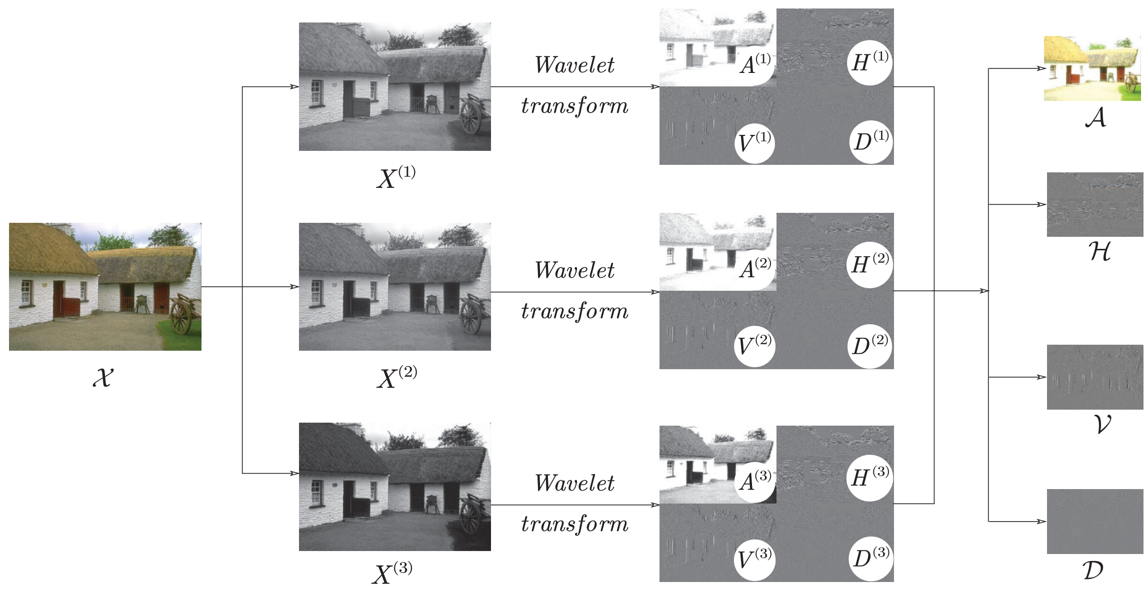

FIGURE 1 Wavelet transform of the color image "House." For better visualization, we add 0.5 to the pixel in $\mathcal{H}$, $\mathcal{V}$, and $\mathcal{D}$

where ψ_θ is a convex and continuous differentiable function defined as

$$\psi_\theta(x) := \begin{cases} \frac{x^2}{2\theta}, & |x| \le \theta, \\ |x| - \frac{\theta}{2}, & |x| > \theta. \end{cases}$$

The proposed minimization model (2) can be obtained via the difference of convex functions (DC) algorithm[35] with a theoretical convergence guarantee. We conduct numerical experiments on various types of visual data and the results verify that our method outperforms the compared methods.

In summary, our main contributions include:

(1) Through wavelet transform, we transform the solution of one tensor with size $n_1 \times n_2 \times n_3$ into four tensors with size $\lceil n_1/2 \rceil \times \lceil n_2/2 \rceil \times n_3$, which greatly improves the speed of the algorithm.
(2) By wavelet transform, we solve the main part and the detail part of tensors separately to facilitate better mining of their data features for a better recovery.
(3) By using concave correction term, we are able to correct for low rank of TNN data fidelity term and sparsity of l_1-norm data fidelity term.
(4) We prove that the proposed DC algorithm can converge to some critical point. The outperformance of our method in experimental results further corroborates the usage of wavelet transform.

The outline of this paper is given as follows. We recall the basic tensor notations in Section 2. In Section 3, we give the main results, including the proposed model, algorithm and the convergence analysis of algorithm. Extensive simulation results are reported in Section 4. Section 5 briefly concludes our study.

2 | NOTATIONS AND PRELIMINARIES

This section recalls some basic knowledge on tensors. We first give the basic notations and then present the tubal rank and t-SVD. We state them here in detail for the readers' convenience.

2.1 | Notations

For a positive integer n, $[n] := \{1, 2, \ldots, n\}$. Scalars, vectors and matrices are denoted as lowercase letters $(a, b, c, \ldots)$, boldface lowercase letters $(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \ldots)$ and uppercase letters $(A, B, C, \ldots)$, respectively. Third-order tensors are denoted as

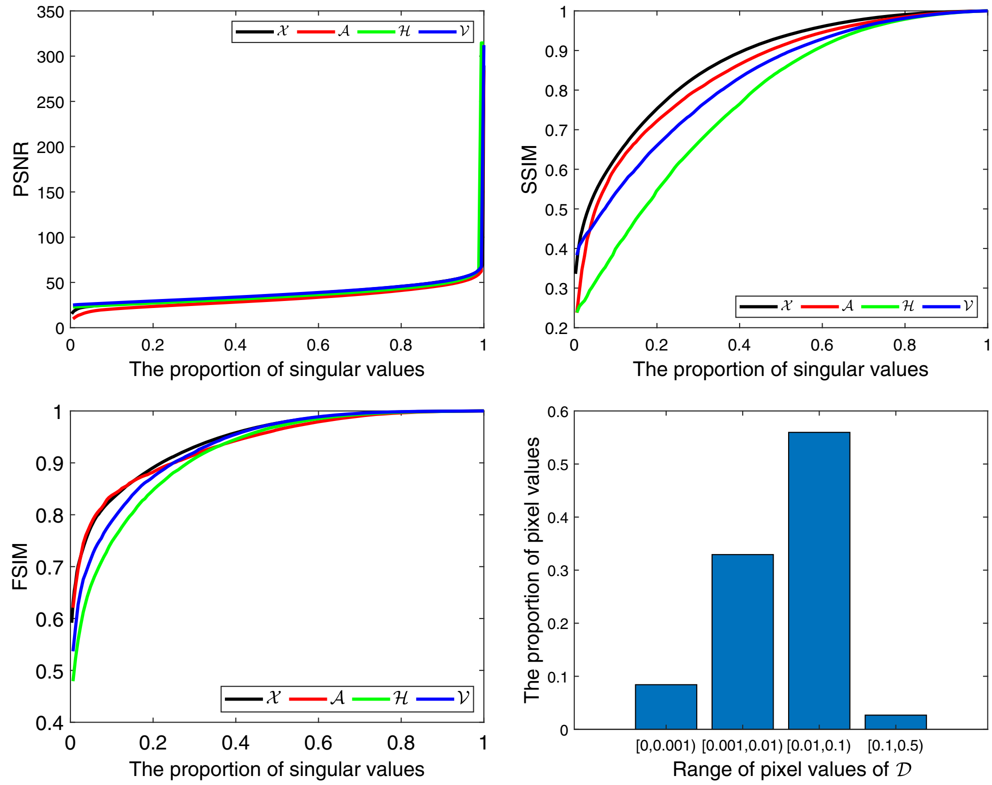

FIGURE 2 Compares the accuracy of tensor singular value decomposition of $\mathcal{X}$, $\mathcal{A}$, $\mathcal{H}$, $\mathcal{V}$ with respect to the change of proportion kept of singular values, and the distribution of pixel values of $\mathcal{D}$

calligraphic letters ($\mathcal{A}, \mathcal{B}, \mathcal{C}, \dots$). For a third-order tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we use the Matlab notations $\mathcal{A}(:,:,k)$ to denote its kth frontal slice, denoted by $A^{(k)}$ for all $k \in [n_3]$. The inner product of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the sum of products of their entries, that is,

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \mathcal{A}_{ijk} \mathcal{B}_{ijk}.$$

The Frobenius norm is $\|\mathcal{A}\| = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$.

2.2 | *T*-product, tubal rank and t-SVD

Discrete Fourier Transformation (DFT) plays a key role in tensor-tensor product (t-product). For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, let $\overline{\mathcal{A}} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ be the result of DFT of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ along the third dimension. Specifically, let $F_{n_3} = [\boldsymbol{f_1}, \dots, \boldsymbol{f_{n_3}}] \in \mathbb{C}^{n_3 \times n_3}$, where

$$\boldsymbol{f_i} = \left[\omega^{0\times(i-1)}; \omega^{1\times(i-1)}; \dots; \omega^{(n_3-1)\times(i-1)}\right] \in \mathbb{C}^{n_3},$$

with $\omega = e^{-\frac{2\pi \mathfrak{b}}{n_3}}$ and $\mathfrak{b} = \sqrt{-1}$. Then $\overline{\mathcal{A}}(i,j,:) = F_{n_3}\mathcal{A}(i,j,:)$, which can be computed by Matlab command "$\overline{\mathcal{A}} = fft(\mathcal{A}, [\,], 3)$." Furthermore, $\mathcal{A}$ can be computed by $\overline{\mathcal{A}}$ with the inverse DFT $\mathcal{A} = ifft(\overline{\mathcal{A}}, [\,], 3)$.

Lemma 1. *(Reference 36) Given any real vector $\boldsymbol{v} \in \mathbb{R}^{n_3}$, the associated $\overline{\boldsymbol{v}} = F_{n_3}\boldsymbol{v} \in \mathbb{C}^{n_3}$ satisfies*

$$\overline{v}_1 \in \mathbb{R} \text{ and } \operatorname{conj}\left(\overline{v}_i\right) = \overline{v}_{n_3-i+2},\ i = 2, \dots, \left\lfloor \frac{n_3+1}{2} \right\rfloor.$$

By using Lemma 1, the frontal slices of $\overline{\mathcal{A}}$ have the following properties:

$$\begin{cases} \bar{A}^{(1)} \in \mathbb{R}^{n_1 \times n_2}, \\ \operatorname{conj}\left(\bar{A}^{(i)}\right) = \bar{A}^{(n_3 - i + 2)},\ i = 2, \ldots, \left\lfloor \frac{n_3+1}{2} \right\rfloor. \end{cases} \tag{3}$$

For $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we define matrix $\bar{A} \in \mathbb{C}^{n_1 n_3 \times n_2 n_3}$ as

$$\bar{A} = \operatorname{bdiag}(\overline{\mathcal{A}}) = \begin{bmatrix} \bar{A}^{(1)} & & & \\ & \bar{A}^{(2)} & & \\ & & \ddots & \\ & & & \bar{A}^{(n_3)} \end{bmatrix}. \tag{4}$$

Here, bdiag(·) is an operator which maps the tensor $\overline{\mathcal{A}}$ to the block diagonal matrix $\bar{A}$. The block circulant matrix $\operatorname{bcirc}(\mathcal{A}) \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$ of $\mathcal{A}$ is defined as

$$\operatorname{bcirc}(\mathcal{A}) = \begin{bmatrix} A^{(1)} & A^{(n_3)} & \cdots & A^{(2)} \\ A^{(2)} & A^{(1)} & \cdots & A^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ A^{(n_3)} & A^{(n_3-1)} & \cdots & A^{(1)} \end{bmatrix}.$$

Based on these notations, the t-product is presented as follows.

Definition 1. (T-product)[28] For $\mathcal{A} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{r \times n_2 \times n_3}$, define

$$\mathcal{A} * \mathcal{B} := \operatorname{fold}\left(\operatorname{bcirc}(\mathcal{A}) \cdot \operatorname{unfold}(\mathcal{B})\right) \in \mathbb{R}^{n_1 \times n_2 \times n_3}.$$

Here

$$\operatorname{unfold}(\mathcal{B}) = \left[B^{(1)}; B^{(2)}; \ldots; B^{(n_3)}\right],$$

and its inverse operator "fold" is defined by

$$\operatorname{fold}(\operatorname{unfold}(\mathcal{B})) = \mathcal{B}.$$

We will now present the definition of tubal rank. Before then, we need to introduce some other concepts.

Definition 2. (F-diagonal tensor)[28] If each of a tensor's frontal slices is a diagonal matrix, the tensor is denoted f-diagonal.

Definition 3. (Conjugate transpose)[28] The conjugate transpose of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, denoted as $\mathcal{A}^*$, is the tensor obtained by conjugate transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through n_3.

Definition 4. (Identity tensor)[28] The identity tensor $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$ is a tensor with the identity matrix as its first frontal slice and all other frontal slices being zeros.

Definition 5. (Orthogonal tensor)[28] A tensor $\mathcal{P} \in \mathbb{R}^{n \times n \times n_3}$ is orthogonal if it fulfills the condition $\mathcal{P}^* * \mathcal{P} = \mathcal{P} * \mathcal{P}^* = \mathcal{I}$.

Definition 6. (T-SVD)[28] A tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ can be factored as

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

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

Tensor multi-rank, tubal rank and TNN are now introduced.

Definition 7. (Tensor multi-rank and tubal rank)[27] For tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, let $r_k = rank\left(\bar{A}^{(k)}\right)$ for all $k \in [n_3]$. Then multi-rank of $\mathcal{A}$ is defined as $rank_m(\mathcal{A}) = (r_1, \ldots, r_{n_3})$. The tensor tubal rank is defined as $rank_t(\mathcal{A}) = \max\{r_k | k \in [n_3]\}$.

Definition 8. (TNN)[37] The TNN of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, denoted as $\|\mathcal{A}\|_*$, is defined as the sum of the singular values of all frontal slices of $\overline{\mathcal{A}}$, that is, $\|\mathcal{A}\|_* = \frac{1}{n_3}\sum_{k=1}^{n_3} \|\bar{A}^{(k)}\|_*$.

3 | TENSOR COMPLETION UNDER WAVELET TRANSFORM AND CONCAVE SMOOTH CORRECTION

This section is divided into four parts. Section 3.1 gives a nonconvex optimization model based wavelet transform for tensor completion. Section 3.2 provides the corresponding DC algorithm and Section 3.3 gives its complexity analysis. Section 3.4 provides convergence details of the algorithm.

3.1 | The proposed model under wavelet transform and concave smooth correction

Based on the above analysis, we introduce a nonconvex optimization model for tensor completion under wavelet transform and concave smooth correction:

$$\begin{array}{cc} \min\limits_{\mathcal{A},\mathcal{H},\mathcal{V},\mathcal{D},\mathcal{X}} & \lambda_A \|\mathcal{A}\|_{\circledast,\theta_1} + \lambda_H \|\mathcal{H}\|_{\circledast,\theta_1} + \lambda_V \|\mathcal{V}\|_{\circledast,\theta_1} + \lambda_D \left(\|\mathcal{D}\|_1 - \Psi_{\theta_2}(\mathcal{D})\right), \\ \text{s.t.} & P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M}), \quad W\mathcal{X} = \{\mathcal{A},\mathcal{H},\mathcal{V},\mathcal{D}\}. \end{array} \tag{5}$$

By penalizing the constraint $P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M})$ and $W\mathcal{X} = \{\mathcal{A},\mathcal{H},\mathcal{V},\mathcal{D}\}$, we get the following problem

$$\begin{aligned} \min L(\mathcal{A},\mathcal{H},\mathcal{V},\mathcal{D},\mathcal{X}) :=& \, l_{\mathbb{S}}(\mathcal{X}) + \lambda_A \|\mathcal{A}\|_{\circledast,\theta_1} + \lambda_H \|\mathcal{H}\|_{\circledast,\theta_1} + \lambda_V \|\mathcal{V}\|_{\circledast,\theta_1} + \lambda_D \left(\|\mathcal{D}\|_1 - \Psi_{\theta_2}(\mathcal{D})\right) \\ &+ \frac{\beta}{2}\|W\mathcal{X} - \{\mathcal{A},\mathcal{H},\mathcal{V},\mathcal{D}\}\|^2, \end{aligned} \tag{6}$$

where $\beta > 0$ is a penalty parameter and $l_{\mathbb{S}}(\mathcal{X})$ is an indicator function defined as

$$l_{\mathbb{S}}(\mathcal{X}) = \begin{cases} 0, & \mathcal{X} \in \mathbb{S}, \\ +\infty, & \text{otherwise}, \end{cases}$$

with $\mathbb{S} := \{\mathcal{X} | P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M})\}$. It is well known that an optimal solution of (6) approaches an optimal solution of (5) as $\beta \to +\infty$.

3.2 | The DC algorithm

We develop a DC algorithm to solve the model (6). Via linearizing the part of $Q_{\theta_1}(\mathcal{C})$ at $\mathcal{C}^k$ and $\Psi_{\theta_2}(\mathcal{D})$ at $\mathcal{D}^k$, the DC algorithm generates the next iterate $\mathcal{W}^{k+1} := \left(\mathcal{A}^{k+1},\mathcal{H}^{k+1},\mathcal{V}^{k+1},\mathcal{D}^{k+1},\mathcal{X}^{k+1}\right)$ by solving the problem

$$\begin{aligned} \min L\left(\mathcal{W};\mathcal{W}^k\right) :=& \, l_{\mathbb{S}}(\mathcal{X}) + \lambda_A U_{\theta_1}\left(\mathcal{A};\mathcal{A}^k\right) + \lambda_H U_{\theta_1}\left(\mathcal{H};\mathcal{H}^k\right) + \lambda_V U_{\theta_1}\left(\mathcal{V};\mathcal{V}^k\right) \\ &+ \lambda_D\left(\|\mathcal{D}\|_1 - \Psi_{\theta_2}(\mathcal{D}^k) - \left\langle \nabla\Psi_{\theta_2}(\mathcal{D}^k), \mathcal{D} - \mathcal{D}^k \right\rangle\right) \\ &+ \frac{\beta}{2}\|W\mathcal{X} - \{\mathcal{A},\mathcal{H},\mathcal{V},\mathcal{D}\}\|^2 + \frac{\gamma}{2}\left\|\mathcal{W} - \mathcal{W}^k\right\|^2, \end{aligned} \tag{7}$$

where $U_{\theta_1}\left(\mathcal{C};\mathcal{C}^k\right) = \|\mathcal{C}\|_* - Q_{\theta_1}\left(\mathcal{C}^k\right) - \left\langle \nabla Q_{\theta_1}\left(\mathcal{C}^k\right), \mathcal{C} - \mathcal{C}^k \right\rangle$, $\|\mathcal{W}\| = \sqrt{\|\mathcal{A}\|^2 + \|\mathcal{H}\|^2 + \|\mathcal{V}\|^2 + \|\mathcal{D}\|^2 + \|\mathcal{X}\|^2}$ and $\gamma > 0$. Then, $\mathcal{A},\ \mathcal{H},\ \mathcal{V},\ \mathcal{D}$, and $\mathcal{X}$ are alternately updated as

$$\begin{cases} \mathcal{A}^{k+1} = \arg\min\limits_{\mathcal{A}} L\left(\mathcal{A},\mathcal{H}^k,\mathcal{V}^k,\mathcal{D}^k,\mathcal{X}^k;\mathcal{W}^k\right), \\ \mathcal{H}^{k+1} = \arg\min\limits_{\mathcal{H}} L\left(\mathcal{A}^{k+1},\mathcal{H},\mathcal{V}^k,\mathcal{D}^k,\mathcal{X}^k;\mathcal{W}^k\right), \\ \mathcal{V}^{k+1} = \arg\min\limits_{\mathcal{V}} L\left(\mathcal{A}^{k+1},\mathcal{H}^{k+1},\mathcal{V},\mathcal{D}^k,\mathcal{X}^k;\mathcal{W}^k\right), \\ \mathcal{D}^{k+1} = \arg\min\limits_{\mathcal{D}} L\left(\mathcal{A}^{k+1},\mathcal{H}^{k+1},\mathcal{V}^{k+1},\mathcal{D},\mathcal{X}^k;\mathcal{W}^k\right), \\ \mathcal{X}^{k+1} = \arg\min\limits_{\mathcal{X}} L\left(\mathcal{A}^{k+1},\mathcal{H}^{k+1},\mathcal{V}^{k+1},\mathcal{D}^{k+1},\mathcal{X};\mathcal{W}^k\right). \end{cases} \tag{8}$$

Below, we give the details of updating each minimizing subproblem.

Step 1, the $\mathcal{A}$-subproblem at the kth iteration is

$$\mathcal{A}^{k+1} = \arg\min_{\mathcal{A}} \lambda_A \left(\|\mathcal{A}\|_* - \left\langle \nabla Q_{\theta_1}\left(\mathcal{A}^k\right), \mathcal{A} \right\rangle \right) + \frac{\beta}{2}\left\|\mathcal{A} - \hat{\mathcal{A}}^k\right\|^2 + \frac{\gamma}{2}\left\|\mathcal{A} - \mathcal{A}^k\right\|^2, \tag{9}$$

where $W\mathcal{X}^k := \left\{ \hat{\mathcal{A}}^k, \hat{\mathcal{H}}^k, \hat{\mathcal{V}}^k, \hat{\mathcal{D}}^k \right\}$. A closed-form solution of (9) can be obtained by a tensor singular value shrinkage (t-SVT) operator,[37] that is,

$$\mathcal{A}^{k+1} = \text{t} - \text{SVT}_{\frac{\lambda_A}{\beta+\gamma}} \left(\frac{\beta \widehat{\mathcal{A}}^k + \gamma \mathcal{A}^k + \lambda_A \nabla Q_{\theta_1}\left(\mathcal{A}^k\right)}{\beta + \gamma} \right). \tag{10}$$

Step 2, similar to the case discussed in $\mathcal{A}$-subproblem, updating $\mathcal{H}$ and $\mathcal{V}$ by

$$\mathcal{H}^{k+1} = \text{t-SVT}_{\frac{\lambda_H}{\beta+\gamma}} \left(\frac{\beta \widehat{\mathcal{H}}^k + \gamma \mathcal{H}^k + \lambda_H \nabla Q_{\theta_1}\left(\mathcal{H}^k\right)}{\beta + \gamma} \right), \tag{11}$$

$$\mathcal{V}^{k+1} = \text{t-SVT}_{\frac{\lambda_V}{\beta+\gamma}} \left(\frac{\beta \widehat{\mathcal{V}}^k + \gamma \mathcal{V}^k + \lambda_V \nabla Q_{\theta_1}\left(\mathcal{V}^k\right)}{\beta + \gamma} \right). \tag{12}$$

Step 3, by soft-thresholding operator, we can update $\mathcal{D}$ by

$$\mathcal{D}^{k+1} = \text{sgn}\left(\mathcal{G}^k\right) \circ \max\left\{ |\mathcal{G}^k| - \frac{\lambda_D}{\beta+\gamma}, 0 \right\}, \tag{13}$$

where $\mathcal{G}^k = \frac{1}{\beta+\gamma}\left(\beta \hat{\mathcal{D}}^k + \gamma \mathcal{D}^k + \lambda_D \nabla \Psi_{\theta_2}(\mathcal{D}^k)\right)$.

Step 4, updating $\mathcal{X}$ by

$$\mathcal{X}^{k+1} = P_{\Omega^C}\left(\frac{\beta W^{-1}\left\{\mathcal{A}^{k+1}, \mathcal{H}^{k+1}, \mathcal{V}^{k+1}, \mathcal{D}^{k+1}\right\} + \gamma \mathcal{X}^k}{\beta + \gamma} \right) + P_{\Omega}\left(\mathcal{M}\right). \tag{14}$$

Finally, our algorithm is summarized in Algorithm 1.

Algorithm 1. Tensor completion under wavelet transform (WTTC)

Require: The tensor data $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the observed set Ω and parameter β, γ.
  while not converge **do**
    Step 1. Update $\mathcal{A}^{k+1}$ by (10).
    Step 2. Update $\mathcal{H}^{k+1}$ by (11).
    Step 3. Update $\mathcal{V}^{k+1}$ by (12).
    Step 4. Update $\mathcal{D}^{k+1}$ by (13).
    Step 5. Update $\mathcal{X}^{k+1}$ by (14).
    Let $k := k + 1$ and go to **Step 1**.
  end while
Ensure: $\mathcal{X}^{k+1}$.

3.3 | Complexity analysis

Let $\tilde{n}_1 = \lceil n_1/2 \rceil$ and $\tilde{n}_2 = \lceil n_2/2 \rceil$. Computing $\mathcal{A}$, $\mathcal{H}$, and $\mathcal{V}$ cost $O(\tilde{n}_1\tilde{n}_2 n_3 \min(\tilde{n}_1, \tilde{n}_2))$ at each iteration. The computation complexity of updating $\mathcal{D}$ and $\mathcal{X}$ are $O(\tilde{n}_1\tilde{n}_2 n_3)$ and $O(4\tilde{n}_1\tilde{n}_2 n_3)$, respectively. So the total cost at each iteration is $O(\tilde{n}_1\tilde{n}_2 n_3 \min(\tilde{n}_1, \tilde{n}_2))$.

3.4 Convergence analysis

In this subsection, we establish global convergence of the WTTC for (6). Before proving the convergence of the proposed Algorithm 1, we first present some lemmas.

Lemma 2. *(Sufficient decrease condition). The sequences* $\{\mathcal{W}^k\}_{k\in\mathbb{N}}$ *generated by WTTC own the following properties*:

(i) $L(\mathcal{W}^{k+1}) - L(\mathcal{W}^k) \le -\frac{\gamma}{2}\|\mathcal{W}^{k+1} - \mathcal{W}^k\|^2$;
(ii) $\lim_{k\to+\infty} \|\mathcal{W}^{k+1} - \mathcal{W}^k\| = 0$.

Proof. (i) From the definition of $L(\mathcal{W})$, it can be seen that

$$L(\mathcal{W}) - L(\mathcal{W};\mathcal{W}^k) = \lambda_A f_{\theta_1}(\mathcal{A};\mathcal{A}^k) + \lambda_H f_{\theta_1}(\mathcal{H};\mathcal{H}^k) + \lambda_V f_{\theta_1}(\mathcal{V};\mathcal{V}^k) + \lambda_D h_{\theta_2}(\mathcal{D};\mathcal{D}^k) - \frac{\gamma}{2}\left\|\mathcal{W} - \mathcal{W}^k\right\|^2,$$

where

$$f_{\theta_1}(\mathcal{C};\mathcal{C}^k) = Q_{\theta_1}(\mathcal{C}^k) - Q_{\theta_1}(\mathcal{C}) + \left\langle \nabla Q_{\theta_1}(\mathcal{C}^k), \mathcal{C} - \mathcal{C}^k \right\rangle,$$

and

$$h_{\theta_2}(\mathcal{D};\mathcal{D}^k) = \Psi_{\theta_2}(\mathcal{D}^k) - \Psi_{\theta_2}(\mathcal{D}) + \left\langle \nabla\Psi_{\theta_2}(\mathcal{D}^k), \mathcal{D} - \mathcal{D}^k \right\rangle.$$

On the other hand, from the convexity of $Q_{\theta_1}(\mathcal{C})$ and $\Psi_{\theta_2}(\mathcal{D})$, we can get $f_{\theta_1}(\mathcal{C};\mathcal{C}^k) \le 0$ and $h_{\theta_2}(\mathcal{D};\mathcal{D}^k) \le 0$. Thus,

$$L(\mathcal{W}) - L(\mathcal{W};\mathcal{W}^k) \le -\frac{\gamma}{2}\left\|\mathcal{W} - \mathcal{W}^k\right\|^2. \tag{15}$$

From (8), we have

$$L(\mathcal{W}^{k+1};\mathcal{W}^k) \le L(\mathcal{W}^k;\mathcal{W}^k) = L(\mathcal{W}^k). \tag{16}$$

Combining (15) and (16), we have

$$L(\mathcal{W}^{k+1}) - L(\mathcal{W}^k) \le -\frac{\gamma}{2}\left\|\mathcal{W}^{k+1} - \mathcal{W}^k\right\|^2, \tag{17}$$

which completes the proof of this statement.

(ii) Summing up (17) for $k = 1, 2, \dots, N-1$, we have

$$\sum_{k=1}^{N-1}\left\|\mathcal{W}^{k+1} - \mathcal{W}^k\right\|^2 \le \frac{2}{\gamma}\left(L(\mathcal{W}^1) - L(\mathcal{W}^N)\right). \tag{18}$$

We then obtain that $\{L(\mathcal{W}^k)\}_{k\in\mathbb{N}}$ is convergent from (i) and $L(\mathcal{W}^k) > 0$. Letting $N \to +\infty$ in (18), we obtain that

$$\sum_{k=1}^{+\infty}\left\|\mathcal{W}^{k+1} - \mathcal{W}^k\right\|^2 < +\infty.$$

Thus, $\lim_{k\to+\infty} \|\mathcal{W}^{k+1} - \mathcal{W}^k\| = 0$. ■

Lemma 3. *(Relative error condition). Let the sequences* $\{\mathcal{W}^k\}_{k\in\mathbb{N}}$ *generated by WTTC. For each positive integer k, define*

$$\mathcal{B}_{\mathcal{A}}^{k+1} = \lambda_A\left(\nabla Q_{\theta_1}(\mathcal{A}^k) - \nabla Q_{\theta_1}(\mathcal{A}^{k+1})\right) + \beta\left(\hat{\mathcal{A}}^k - \hat{\mathcal{A}}^{k+1}\right) + \gamma\left(\mathcal{A}^k - \mathcal{A}^{k+1}\right),$$
$$\mathcal{B}_{\mathcal{H}}^{k+1} = \lambda_H\left(\nabla Q_{\theta_1}(\mathcal{H}^k) - \nabla Q_{\theta_1}(\mathcal{H}^{k+1})\right) + \beta\left(\hat{\mathcal{H}}^k - \hat{\mathcal{H}}^{k+1}\right) + \gamma\left(\mathcal{H}^k - \mathcal{H}^{k+1}\right),$$
$$\mathcal{B}_{\mathcal{V}}^{k+1} = \lambda_V\left(\nabla Q_{\theta_1}(\mathcal{V}^k) - \nabla Q_{\theta_1}(\mathcal{V}^{k+1})\right) + \beta\left(\hat{\mathcal{V}}^k - \hat{\mathcal{V}}^{k+1}\right) + \gamma\left(\mathcal{V}^k - \mathcal{V}^{k+1}\right),$$

$$\mathcal{B}_{D}^{k+1} = \lambda_D\left(\nabla\Psi_{\theta_2}\left(\mathcal{D}^k\right) - \nabla\Psi_{\theta_2}\left(\mathcal{D}^{k+1}\right)\right) + \beta\left(\hat{\mathcal{D}}^k - \hat{\mathcal{D}}^{k+1}\right) + \gamma\left(\mathcal{D}^k - \mathcal{D}^{k+1}\right),$$
$$\mathcal{B}_{\mathcal{X}}^{k+1} = \gamma\left(\mathcal{X}^k - \mathcal{X}^{k+1}\right). \tag{19}$$

Then $\left(\mathcal{B}_{\mathcal{A}}^{k+1}, \mathcal{B}_{\mathcal{H}}^{k+1}, \mathcal{B}_{\mathcal{V}}^{k+1}, \mathcal{B}_{D}^{k+1}, \mathcal{B}_{\mathcal{X}}^{k+1}\right) \in \partial L\left(\mathcal{W}^{k+1}\right)$ *and there exists* $m > 0$ *such that*

$$\left\|\left(\mathcal{B}_{\mathcal{A}}^{k+1}, \mathcal{B}_{\mathcal{H}}^{k+1}, \mathcal{B}_{\mathcal{V}}^{k+1}, \mathcal{B}_{D}^{k+1}, \mathcal{B}_{\mathcal{X}}^{k+1}\right)\right\| \leq m\left\|\mathcal{W}^{k+1} - \mathcal{W}^k\right\|.$$

Proof. From (9), we have

$$\lambda_A\left(\mathcal{U}^{k+1} - \nabla Q_{\theta_1}\left(\mathcal{A}^k\right)\right) + \beta\left(\mathcal{A}^{k+1} - \hat{\mathcal{A}}^k\right) + \gamma\left(\mathcal{A}^{k+1} - \mathcal{A}^k\right) = 0 \tag{20}$$

for some $\mathcal{U}^{k+1} \in \partial\left\|\mathcal{A}^{k+1}\right\|_*$. Combining (19) and (20) and recalling the definition of $L(\mathcal{W})$, one has

$$\begin{aligned}\mathcal{B}_{\mathcal{A}}^{k+1} &= \lambda_A\left(\nabla Q_{\theta_1}\left(\mathcal{A}^k\right) - \nabla Q_{\theta_1}\left(\mathcal{A}^{k+1}\right)\right) + \beta\left(\hat{\mathcal{A}}^k - \hat{\mathcal{A}}^{k+1}\right) + \gamma\left(\mathcal{A}^k - \mathcal{A}^{k+1}\right)\\ &\quad + \lambda_A\left(\mathcal{U}^{k+1} - \nabla Q_{\theta_1}\left(\mathcal{A}^k\right)\right) + \beta\left(\mathcal{A}^{k+1} - \hat{\mathcal{A}}^k\right) + \gamma\left(\mathcal{A}^{k+1} - \mathcal{A}^k\right)\\ &= \lambda_A\left(\mathcal{U}^{k+1} - \nabla Q_{\theta_1}\left(\mathcal{A}^{k+1}\right)\right) + \beta\left(\mathcal{A}^{k+1} - \hat{\mathcal{A}}^{k+1}\right)\\ &\in \partial_{\mathcal{A}} L\left(\mathcal{W}^{k+1}\right).\end{aligned}$$

Similarly, we have $\mathcal{B}_{\mathcal{H}}^{k+1} \in \partial_{\mathcal{H}} L\left(\mathcal{W}^{k+1}\right)$, $\mathcal{B}_{\mathcal{V}}^{k+1} \in \partial_{\mathcal{V}} L\left(\mathcal{W}^{k+1}\right)$, $\mathcal{B}_{D}^{k+1} \in \partial_{D} L\left(\mathcal{W}^{k+1}\right)$ and $\mathcal{B}_{\mathcal{X}}^{k+1} \in \partial_{\mathcal{X}} L\left(\mathcal{W}^{k+1}\right)$. Thus, one has $\left(\mathcal{B}_{\mathcal{A}}^{k+1}, \mathcal{B}_{\mathcal{H}}^{k+1}, \mathcal{B}_{\mathcal{V}}^{k+1}, \mathcal{B}_{D}^{k+1}, \mathcal{B}_{\mathcal{X}}^{k+1}\right) \in \partial L\left(\mathcal{W}^{k+1}\right)$.

From Theorem 3.10 in Reference 38, there exists $\tilde{m} > 0$ such that

$$\left\|\nabla Q_{\theta_1}\left(\mathcal{C}^k\right) - \nabla Q_{\theta_1}\left(\mathcal{C}^{k+1}\right)\right\| \leq \tilde{m}\left\|\mathcal{C}^k + \mathcal{C}^{k+1}\right\|. \tag{21}$$

Since $\nabla\Psi_{\theta_2}$ is Lipschitz continuous and the Lipschitz constant is $1/\theta_2$, we get

$$\left\|\nabla\Psi_{\theta_2}\left(\mathcal{D}^k\right) - \nabla\Psi_{\theta_2}\left(\mathcal{D}^{k+1}\right)\right\| \leq \frac{1}{\theta_2}\left\|\mathcal{D}^k + \mathcal{D}^{k+1}\right\|. \tag{22}$$

Combining (21) with (22) and (19), we obtain

$$\left\|\left(\mathcal{B}_{\mathcal{A}}^{k+1}, \mathcal{B}_{\mathcal{H}}^{k+1}, \mathcal{B}_{\mathcal{V}}^{k+1}, \mathcal{B}_{D}^{k+1}, \mathcal{B}_{\mathcal{X}}^{k+1}\right)\right\| \leq (\lambda + \beta\|W\| + \gamma)\left\|\mathcal{W}^{k+1} - \mathcal{W}^k\right\|,$$

where $\lambda = \max\{\lambda_A\tilde{m}, \lambda_H\tilde{m}, \lambda_V\tilde{m}, \lambda_D/\theta_2\}$. By letting $m = \lambda + \beta\|W\| + \gamma$, we complete the proof of this statement. ■

Lemma 4. *The function $L(\mathcal{W})$ is a Kurdyka–Łojasiewicz (KL) function.*

Proof. By Reference 12, $\|\mathcal{C}\|_*$ is semi-algebraic. Furthermore, References 39 and 40 prove that the Frobenius norm $\|\cdot\|$ and minimax concave penalty function are semi-algebraic functions. Thus, $L(\mathcal{W})$ is semi-algebraic since it is the finite sum of semi-algebraic functions. As $L(\mathcal{W})$ is also a proper continuous function, we know from Theorem 3 in Reference 41 that $L(\mathcal{W})$ is a KL function, which completes the proof of this statement. ■

Finally, we present the convergence results of Algorithm 1.

Theorem 1. *Assume that $L(\mathcal{W})$ is the objective function and the sequence $\left\{\mathcal{W}^k\right\}_{k\in\mathbb{N}}$ generated by Algorithm 1 is bounded. Then the generated sequence $\left\{\mathcal{W}^k\right\}_{k\in\mathbb{N}}$ converges to some critical point of $L(\mathcal{W})$.*

Proof. The boundedness of sequence $\left\{\mathcal{W}^k\right\}_{k\in\mathbb{N}}$ admits a converging subsequence, combining which with continuity of $L(\mathcal{W})$, Lemma 2, Lemma 3, and Lemma 4, we obtain the conclusion according to Theorem 2.9 in Reference 42. The desired result is obtained. ■

4 | NUMERICAL EXPERIMENTS

In this section, we conduct some experiments on real-world dataset to compare the performance of WTTC to show their validity. We employ the peak signal-to-noise rate (PSNR),[43] the structural similarity (SSIM),[43] the feature similarity (FSIM)[44] and the recovery computation time to measure the quality of the recovered results. We compare WTTC for the tensor completion problem with four existing methods, including PSTNN,[34] TNN,[25] HaLRTC,[22] and NTD.[24] WTTC, PSTNN and TNN are specialized to third-order tensors. As (10), (11), (12) and (13) show, we can parallely update $\mathcal{A}$, $\mathcal{H}$, $\mathcal{V}$, and $\mathcal{D}$, but for fair comparison of algorithm running time, we still employ the serial updating scheme in our code. All methods are implemented on the platform of Windows 11 and Matlab (R2020b) with an Intel(R) Core(TM) i5-12500H CPU at 2.50 GHz and 16 GB RAM.

4.1 | Stopping criterion

To measure the precision of the optimal solution obtained by Algorithm 1, we used the relative KKT residual $\xi = \max\{\xi_{\mathcal{A}}, \xi_{\mathcal{H}}, \xi_{\mathcal{V}}, \xi_{\mathcal{D}}\}$ with

$$\xi_{\mathcal{A}} = \frac{\left\|\mathcal{A} - t - SVT_{\lambda_A/\beta}\left(\hat{\mathcal{A}} + \lambda_A \nabla Q_{\theta_1}(\mathcal{A})/\beta\right)\right\|}{1 + \left\|\hat{\mathcal{A}}\right\| + \left\|\lambda_A \nabla Q_{\theta_1}(\mathcal{A})/\beta\right\|}, \quad \xi_{\mathcal{H}} = \frac{\left\|\mathcal{H} - t - SVT_{\lambda_H/\beta}\left(\hat{\mathcal{H}} + \lambda_H \nabla Q_{\theta_1}(\mathcal{H})/\beta\right)\right\|}{1 + \left\|\hat{\mathcal{H}}\right\| + \left\|\lambda_H \nabla Q_{\theta_1}(\mathcal{H})/\beta\right\|},$$

$$\xi_{\mathcal{V}} = \frac{\left\|\mathcal{V} - t - SVT_{\lambda_V/\beta}\left(\hat{\mathcal{V}} + \lambda_V \nabla Q_{\theta_1}(\mathcal{V})/\beta\right)\right\|}{1 + \left\|\hat{\mathcal{V}}\right\| + \left\|\lambda_V \nabla Q_{\theta_1}(\mathcal{V})/\beta\right\|}, \quad \xi_{\mathcal{D}} = \frac{\left\|\mathcal{D} - \operatorname{sgn}(\mathcal{G}) \circ \max\{|\mathcal{G}| - \lambda_D/\beta, 0\}\right\|}{1 + \left\|\hat{\mathcal{D}}\right\| + \left\|\lambda_D \nabla \Psi_{\theta_2}(\mathcal{D})/\beta\right\|},$$

where $W\mathcal{X} := \left\{\hat{\mathcal{A}}, \hat{\mathcal{H}}, \hat{\mathcal{V}}, \hat{\mathcal{D}}\right\}$ and $\mathcal{G} = \left(\hat{\mathcal{D}} + \lambda_D \nabla \Psi_{\theta_2}(\mathcal{D})/\beta\right)$.

In all experiments, the termination precision is set to be $1e-4$ and the maximum iteration steps is set to be 150.

4.2 | Parameters setting

In this subsection, taking the completion of "Wall" image[1] as an example, we evaluate the performance of the proposed method with different parameters $\lambda_A,\ \lambda_H,\ \lambda_V,\ \lambda_D$, and $\theta_1,\ \theta_2$ setting. The sampling rate (SR) is set as 30%.

4.2.1 | Regularization parameters $\lambda_A,\ \lambda_H,\ \lambda_V,\ \lambda_D$ setting

In this part, we fix $\theta_1 = 30,\ \theta_2 = 1$. The quantitative metrics of the results obtained by the proposed method with different regularization parameters setting are reported in Figures 3 and 4. Firstly, λ_D is selected from $\{0.1, 1, 9\}$. Meanwhile, λ_A and $\lambda_H,\ \lambda_V$ are chosen from 1 to 9. From Figure 3, we can find that the recovery effect of $\lambda_D = 1,\ 9$ are better than that of $\lambda_D = 0.1$, and when $\lambda_D = 1,\ 9$, the recovery effect of $\lambda_A = \lambda_H = \lambda_V$ is better than that of λ_A is not equal to $\lambda_H = \lambda_V$. Therefore, in the later analysis, we set $\lambda_A = \lambda_H = \lambda_V$. From Figure 4, we can find that setting $\lambda_A = \lambda_H = \lambda_V = \{2,\ 3\}$ and $\lambda_D = \{3,\ 5,\ 9\}$ is a good choice.

4.2.2 | Parameters $\theta_1,\ \theta_2$ Setting

In this part, we fix $\lambda_A = \lambda_H = \lambda_V = 2$ and $\lambda_D = 5$. We set θ_1 from 10 to 100 and θ_2 from 0.1 to 1. From Figure 5, we can find that $\theta_1 = 20,\ \theta_2 = 0.6$ is the best choice.

According to Sections 4.2.1 and 4.2.2, we set $\lambda_A = \lambda_H = \lambda_V = 2,\ \lambda_D = 5,\ \theta_1 = 20,\ \theta_2 = 0.6$ in the follows experiments.

4.3 | Color image inpainting

In this subsection, we use the USC-SIPI image database[2] to evaluate our proposed method WTTC for color image inpainting. In our test, four images are randomly selected from this database, including "House" and "Peppers" with

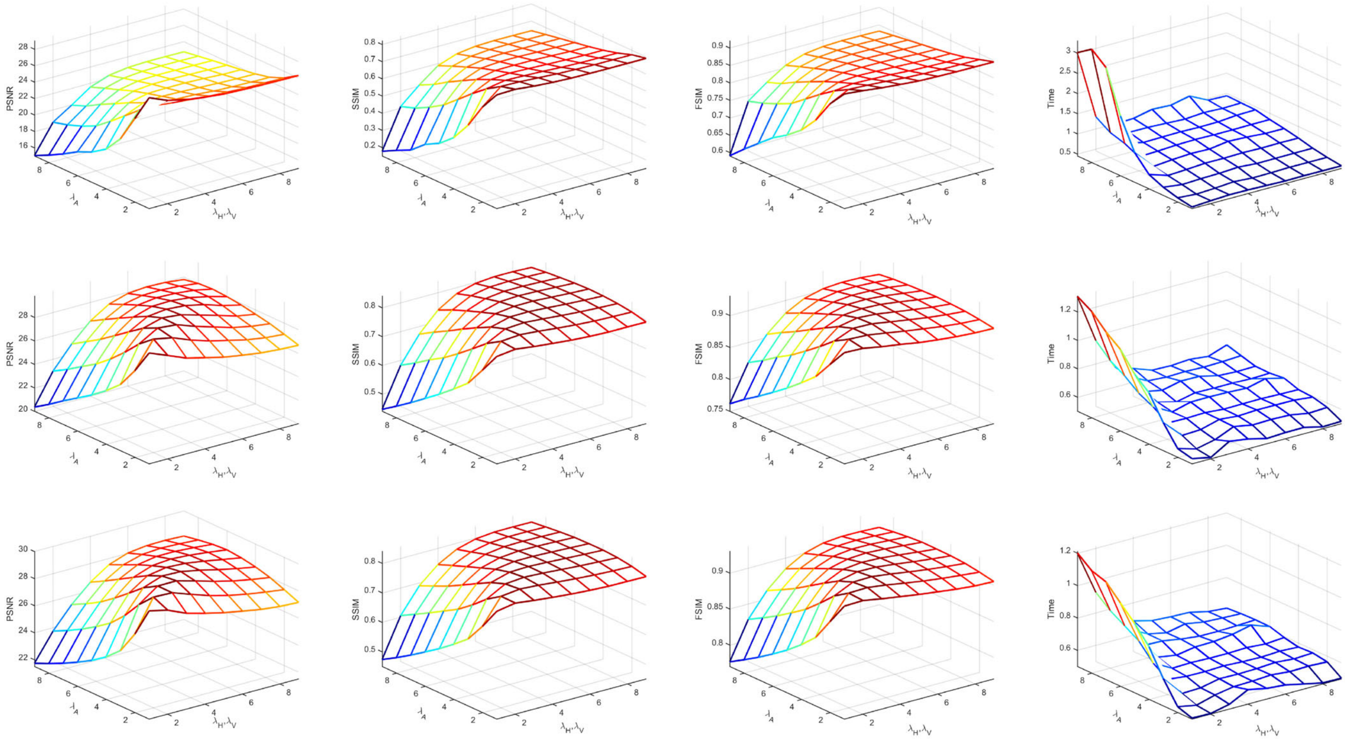

FIGURE 3 The peak signal-to-noise rate, structural similarity, feature similarity, and time of the recovery results by the proposed method with different regularization parameter λ_A, λ_H, λ_V, and λ_D settings. From top to bottom are, respectively, corresponding to $\lambda_D = 0.1, 1, 9$

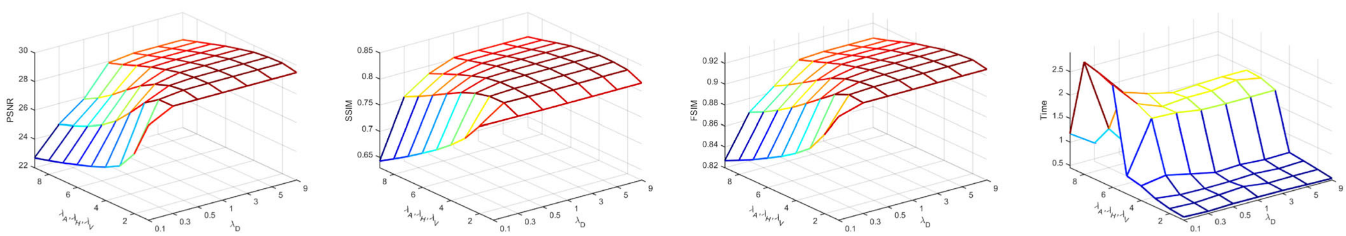

FIGURE 4 The peak signal-to-noise rate, structural similarity, feature similarity, and time of the recovery results by the proposed method with different regularization parameter λ_A, λ_H, λ_V, and λ_D settings

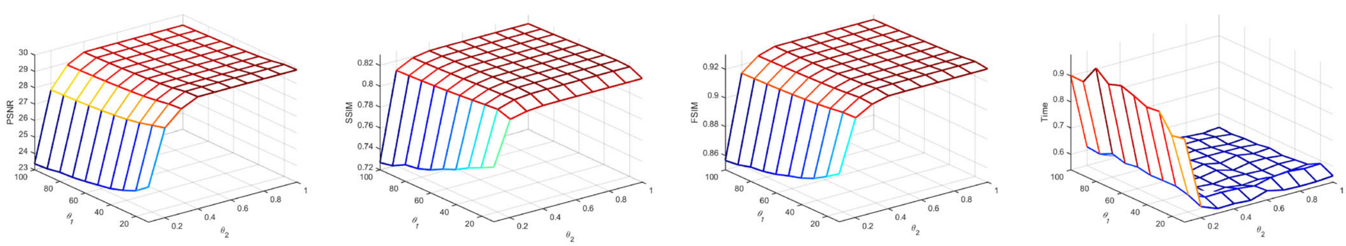

FIGURE 5 The peak signal-to-noise rate, structural similarity, feature similarity, and time of the recovery results by the proposed method with different parameter θ_1, θ_2 settings

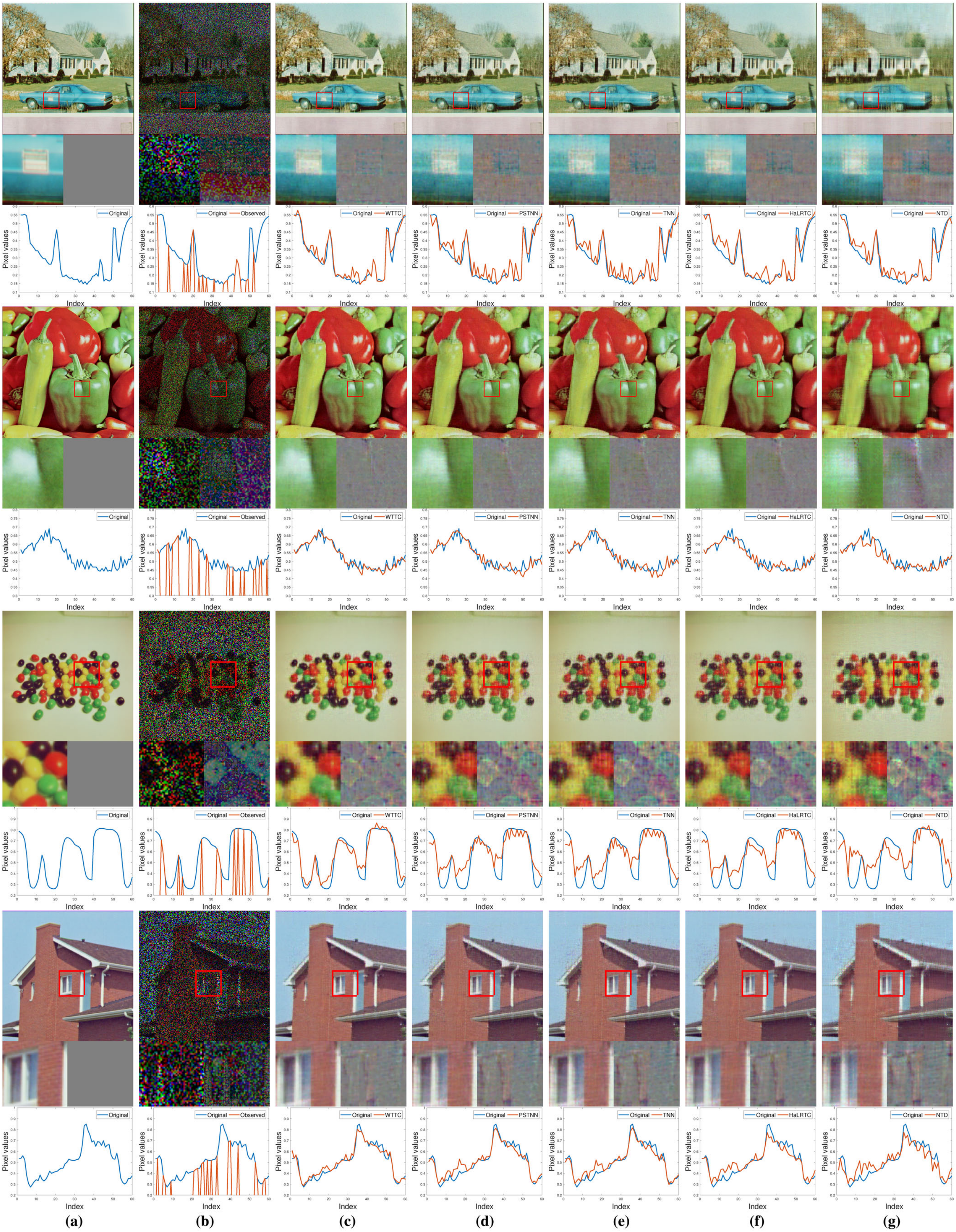

FIGURE 6 Examples of color image inpainting with SR = 30%. From top to bottom are respectively corresponding to "House," "Peppers," "Beans," and "Wall." (a) Original; (b) Observed; (c) WTTC; (d) PSTNN; (e) TNN; (f) HaLRTC; (g) NTD

TABLE 1 Color image inpainting performance comparison: peak signal-to-noise rate (PSNR), structural similarity (SSIM), feature similarity (FSIM), and time

		SR = 10%				SR = 20%				SR = 30%			
Image	**Methods**	**PSNR**	**SSIM**	**FSIM**	**Time**	**PSNR**	**SSIM**	**FSIM**	**Time**	**PSNR**	**SSIM**	**FSIM**	**Time**
House	WTTC	**22.717**	**0.589**	**0.851**	**5.088**	**25.484**	**0.732**	**0.919**	**4.826**	**27.621**	**0.815**	**0.951**	**4.818**
	PSTNN	20.894	0.476	0.807	26.026	23.860	0.639	0.887	31.922	26.201	0.749	0.931	15.239
	TNN	20.535	0.457	0.801	80.707	23.647	0.629	0.883	97.749	26.121	0.747	0.930	99.148
	HaLRTC	20.512	0.494	0.788	15.390	23.525	0.655	0.873	68.541	25.950	0.766	0.923	45.241
	NTD	18.923	0.384	0.733	13.573	21.284	0.519	0.803	13.088	23.013	0.618	0.853	13.641
Sailboat	WTTC	**21.420**	**0.473**	**0.829**	**4.832**	**24.005**	**0.614**	**0.900**	**5.008**	**25.697**	**0.699**	**0.933**	**4.745**
	PSTNN	19.457	0.364	0.800	33.102	22.316	0.527	0.878	31.826	24.295	0.635	0.918	22.515
	TNN	19.100	0.345	0.794	82.774	22.000	0.510	0.873	95.187	24.147	0.628	0.916	52.036
	HaLRTC	19.321	0.413	0.794	84.783	22.137	0.564	0.872	12.112	24.356	0.681	0.917	10.973
	NTD	18.029	0.336	0.739	13.677	20.655	0.475	0.818	10.351	22.140	0.572	0.861	7.758
Peppers	WTTC	**22.902**	**0.487**	**0.814**	**5.158**	**26.333**	**0.652**	**0.904**	**5.018**	**28.317**	**0.731**	**0.942**	**4.922**
	PSTNN	19.864	0.320	0.773	35.389	23.799	0.522	0.872	14.819	26.257	0.641	0.920	14.520
	TNN	19.144	0.295	0.768	59.508	23.199	0.491	0.865	41.981	25.927	0.625	0.916	52.313
	HaLRTC	19.832	0.401	0.797	50.324	23.735	0.579	0.884	10.571	26.452	0.700	0.929	13.847
	NTD	18.227	0.303	0.737	7.655	21.226	0.467	0.815	7.873	22.710	0.560	0.856	10.904
Beans	WTTC	**22.754**	**0.704**	**0.749**	**1.230**	**26.582**	**0.831**	**0.843**	**1.174**	**29.128**	**0.888**	**0.896**	**1.270**
	PSTNN	19.970	0.561	0.678	3.935	23.664	0.732	0.783	3.685	26.549	0.820	0.848	3.467
	TNN	19.709	0.534	0.667	10.732	23.247	0.706	0.767	24.773	26.223	0.805	0.837	21.240
	HaLRTC	19.835	0.576	0.697	2.712	23.496	0.748	0.794	3.312	26.658	0.842	0.863	2.567
	NTD	18.653	0.383	0.614	8.587	21.897	0.559	0.713	9.035	25.081	0.708	0.799	6.877
Wall	WTTC	**23.431**	**0.582**	**0.777**	**1.216**	**27.026**	**0.749**	**0.865**	**1.221**	**29.535**	**0.829**	**0.909**	**1.218**
	PSTNN	20.818	0.394	0.710	3.687	25.054	0.626	0.827	3.538	27.922	0.747	0.884	3.470
	TNN	20.639	0.388	0.707	20.996	24.740	0.606	0.819	9.261	27.777	0.740	0.880	9.100
	HaLRTC	20.710	0.481	0.736	2.686	24.352	0.664	0.828	2.301	27.082	0.774	0.879	2.242
	NTD	20.116	0.358	0.680	8.592	23.504	0.548	0.780	4.930	26.003	0.672	0.838	4.862

Note: The boldface number is the best.

$512 \times 512 \times 3$ pixels, “Beans” and “Wall” with $256 \times 256 \times 3$ pixels. The data of images are normalized in the range [0, 1]. The SRs are set as 10%, 20%, and 30%.

Figure 6 shows the results of the four inpainting tests under SR = 30%. Under each image, we show enlargements of a demarcated patch and the corresponding error map (difference from the Original). Error maps with less color information indicate better restoration performance. At the same time, we also give a graph of some pixels in the enlarged area. The higher the fitting degree of the two curves, the better the recovery effect. As one can see, images recovered by NTD was the least effective, and it yielded a recovery that only recovered the coarse structure, producing significant blurring and artifacts. By comparison with NTD method, HaLRTC, TNN, and PSTNN recover some details, but do not alleviate the blurriness. Compared to other methods, WTTC produces the most visually appealing results with clear and sharp spatial details, because it solves the main part and the detail part of tensors separately.

In Table 1, we present the PSNR/SSIM/FSIM values and running time for different methods of recovering color images under different SRs. Compared to other methods, the proposed WTTC consistently outperforms them in PSNR, SSIM, and FSIM, as well as running time. More precisely, WTTC performs the best with at least 1.5 dB improvement upon the PSNR metric and two times faster than the second fastest method.

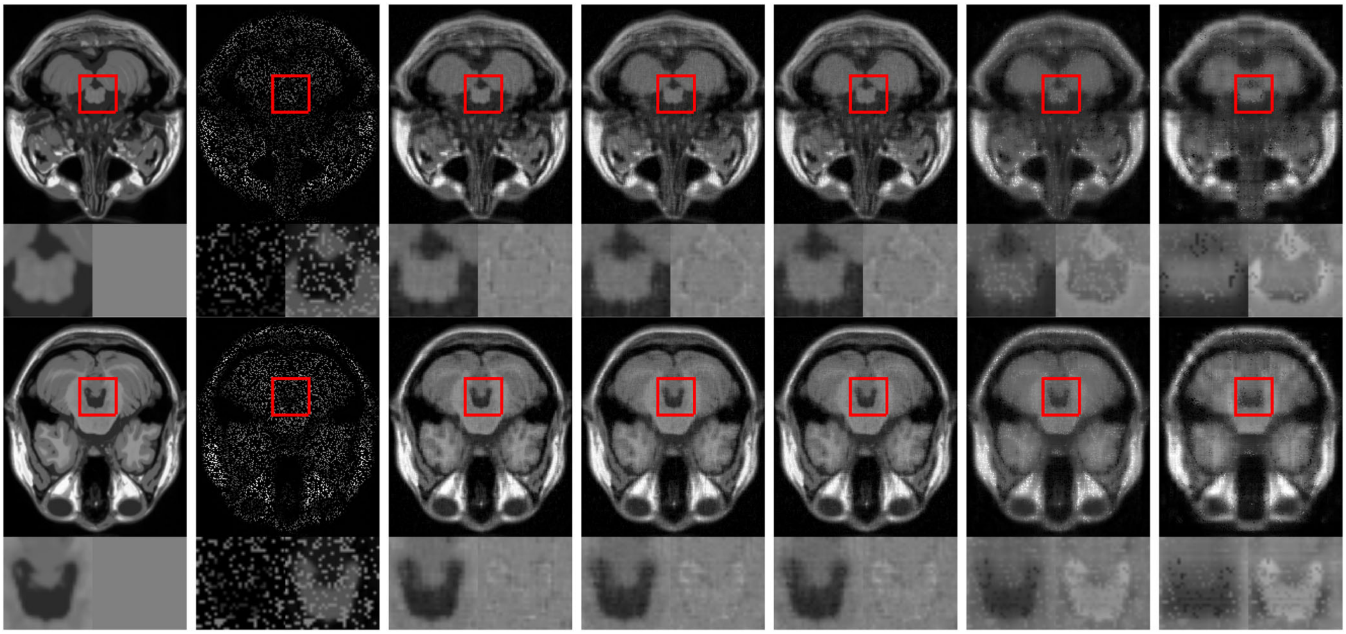

FIGURE 7 Examples of magnetic resonance imaging inpainting with SR = 20%. From top to bottom: the images located at the 20th frontal slice and 40th frontal slice, respectively. (a) Original; (b) Observed; (c) WTTC; (d) PSTNN; (e) TNN; (f) HaLRTC; (g) NTD

TABLE 2 MRI inpainting performance comparison: peak signal-to-noise rate (PSNR), structural similarity (SSIM), feature similarity (FSIM) and Time

	SR = 5%				SR = 10%				SR = 20%			
Methods	PSNR	SSIM	FSIM	Time	PSNR	SSIM	FSIM	Time	PSNR	SSIM	FSIM	Time
WTTC	**23.639**	**0.607**	**0.805**	17.823	**26.234**	**0.715**	**0.856**	13.594	**29.634**	**0.837**	**0.914**	12.370
PSTNN	21.515	0.470	0.759	31.402	24.677	0.645	0.830	30.303	28.198	0.797	0.897	29.117
TNN	21.551	0.488	0.764	55.905	24.256	0.634	0.824	55.473	27.889	0.790	0.893	56.805
HaLRTC	15.844	0.299	0.637	16.947	19.156	0.457	0.725	9.128	23.595	0.687	0.833	4.164
NTD	19.132	0.390	0.711	6.218	21.015	0.504	0.752	6.504	22.878	0.603	0.799	6.651

[1] Note: The boldface number is the best.

4.4 | MRI inpainting

We evaluate the performance of the proposed method and the compared methods on the MRI[3] data, which is of size $217 \times 181 \times 181$, and the first 50 of which are used to construct the third-order tensor due to the computational limitation. The SRs are set as 5%, 10% and 20%.

From Figure 7 and Table 2, we can see that WTTC outperforms other methods on all of them. The Tucker rank-based method HaLRTC and NTD have poor effect on restoring images. TNN, PSTNN, and WTTC are based on recent research on the decomposition of a tensor and avoid the loss of structure information, resulting in better inpainting results. However, the method based on TNN and PSTNN needs to perform SVD on $\lceil (n_3+1)/2 \rceil$ matrices of size $n_1 \times n_2$ in each iteration, so as n_3 increases, the running time required by the method based on TNN and PSTNN increases a lot. The WTTC method changes the size of the matrix from $n_1 \times n_2$ to $\lceil n_1/2 \rceil \times \lceil n_2/2 \rceil$ through wavelet transformation, thus greatly reducing the time required for SVD. At the same time, the wavelet transform separates the main part and the detail part of the tensor, so a better recovery effect is obtained.

FIGURE 8 Examples of video inpainting with SR = 20%. From top to bottom are respectively corresponding to "Bus," "Tempete," "Suzie," and "Foreman." (a) Original; (b) Observed; (c) WTTC; (d) PSTNN; (e) TNN; (f) HaLRTC; (g) NTD

4.5 | Video inpainting

We evaluate our method on the widely used YUV Video Sequences[4]. Each sequence contains at least 150 frames and we use the first 50 frames of the sequences. In the experiments, we test our method and other methods on six videos. The frame sizes of the first three videos are 288×352 pixels and that of the last three are 144×176 pixels. The SRs are set as 10%, 20%, and 30%.

As shown in Figure 8, each test video is shown at the eighth frame. Based on the results of the six tests, WTTC performs better at filling in the missing values. It is better able to deal with the details of the frames. The PSNR, SSIM, and FSIM metrics also shows the best results with WTTC, consistent with Table 3. From time consumption, WTTC uses similar running time as HaLRTC and NTD. In addition, it is about four times faster than PSTNN and 10 times faster than TNN. Those reasons have been discussed above. These results indicate that WTTC performs tensor completion better and runs more efficiently and are consistent with the results for MRI inpainting.

In addition, Figure 9 displays the PSNR, SSIM, FSIM values of each frontal slice of Video "Bus" and "Suzie." As observed, in all frontal slices, the PSNR, SSIM, and FSIM metrics of the proposed WTTC are much higher than those of the other compared methods.

5 | CONCLUSIONS

In this paper, we develop a nonconvex optimization model for third-order tensor completion under wavelet transform. The traditional tensor completion method recoveries the entire large tensor as a whole. When the dimension

TABLE 3 Video inpainting performance comparison: peak signal-to-noise rate (PSNR), structural similarity (SSIM), feature similarity (FSIM), and time

		SR = 10%				SR = 20%				SR = 30%			
Video	**Methods**	**PSNR**	**SSIM**	**FSIM**	**Time**	**PSNR**	**SSIM**	**FSIM**	**Time**	**PSNR**	**SSIM**	**FSIM**	**Time**
Bus	WTTC	**20.471**	**0.471**	**0.738**	18.478	**22.423**	**0.609**	**0.803**	12.084	**24.122**	**0.708**	**0.851**	12.261
	PSTNN	19.693	0.410	0.713	73.539	21.577	0.550	0.782	111.153	23.219	0.656	0.831	67.566
	TNN	19.471	0.395	0.705	174.847	21.424	0.541	0.777	169.464	23.147	0.652	0.829	175.109
	HaLRTC	17.241	0.388	0.592	14.504	19.468	0.525	0.690	11.437	21.115	0.634	0.761	8.567
	NTD	18.274	0.384	0.631	12.621	19.331	0.483	0.688	13.179	20.219	0.559	0.733	13.899
Tempete	WTTC	**22.787**	**0.613**	**0.803**	22.147	**25.029**	**0.748**	**0.866**	19.085	**26.872**	**0.824**	**0.904**	11.851
	PSTNN	22.136	0.566	0.783	73.113	24.406	0.711	0.852	71.209	26.414	0.801	0.895	68.728
	TNN	21.896	0.549	0.776	169.549	24.236	0.702	0.848	175.979	26.322	0.797	0.894	176.696
	HaLRTC	19.062	0.438	0.663	15.164	21.282	0.594	0.762	10.421	23.099	0.708	0.827	7.976
	NTD	19.699	0.426	0.686	12.595	20.818	0.526	0.739	13.371	21.768	0.601	0.782	13.736
Stefan	WTTC	**20.223**	**0.522**	**0.760**	18.607	**22.008**	**0.650**	**0.825**	11.927	**23.518**	**0.737**	**0.869**	11.978
	PSTNN	19.703	0.464	0.728	120.144	21.341	0.593	0.799	88.572	22.848	0.690	0.848	83.054
	TNN	19.571	0.451	0.721	173.245	21.248	0.585	0.795	209.868	22.805	0.687	0.847	195.395
	HaLRTC	18.211	0.485	0.591	12.499	20.002	0.617	0.710	11.172	21.669	0.725	0.794	8.656
	NTD	18.228	0.406	0.617	12.523	19.298	0.506	0.690	21.849	20.269	0.592	0.747	23.822
Carphone	WTTC	**28.194**	**0.841**	**0.907**	7.180	**30.506**	**0.896**	**0.938**	7.511	**32.079**	**0.925**	**0.955**	7.837
	PSTNN	27.595	0.812	0.898	42.355	29.806	0.873	0.930	19.930	31.449	0.907	0.949	18.591
	TNN	27.213	0.799	0.893	40.885	29.628	0.868	0.928	41.294	31.389	0.906	0.949	41.069
	HaLRTC	22.481	0.695	0.828	3.547	25.913	0.816	0.890	2.346	28.607	0.886	0.928	1.985
	NTD	23.081	0.634	0.803	5.403	25.029	0.720	0.847	5.670	26.175	0.773	0.874	5.826
Suzie	WTTC	**28.792**	**0.799**	**0.891**	8.691	**31.549**	**0.871**	**0.929**	7.338	**33.489**	**0.910**	**0.951**	7.770
	PSTNN	27.404	0.751	0.877	74.044	30.238	0.840	0.918	19.263	32.347	0.888	0.942	18.489
	TNN	27.003	0.737	0.871	119.241	30.020	0.834	0.916	41.105	32.264	0.886	0.941	41.785
	HaLRTC	23.726	0.682	0.818	5.546	27.424	0.799	0.883	2.493	30.065	0.867	0.921	2.173
	NTD	24.099	0.612	0.811	5.282	26.373	0.712	0.855	5.628	28.120	0.775	0.885	6.038
Foreman	WTTC	**25.327**	**0.696**	**0.824**	7.513	**27.780**	**0.798**	**0.879**	6.772	**29.617**	**0.856**	**0.912**	7.272
	PSTNN	24.134	0.626	0.803	73.548	26.743	0.751	0.865	21.720	28.906	0.827	0.905	35.034
	TNN	23.776	0.606	0.796	121.072	26.547	0.742	0.861	44.257	28.817	0.824	0.903	82.478
	HaLRTC	19.710	0.485	0.714	5.491	23.043	0.660	0.804	3.739	25.799	0.782	0.867	1.959
	NTD	19.802	0.425	0.709	11.083	21.312	0.518	0.748	5.807	22.945	0.603	0.784	11.655

Note: The boldface number is the best.

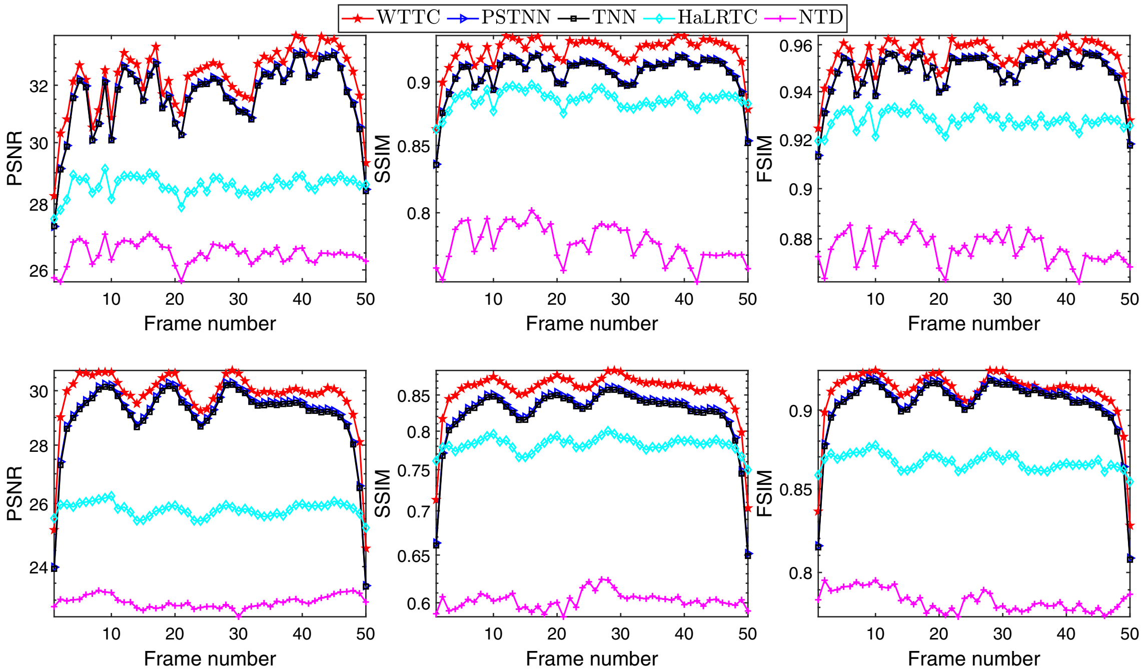

FIGURE 9 All frontal slices obtained by different methods on the video "Carphone" and "Foreman" with SR = 30%

of the tensor is large, the recovery efficiency of the traditional method is often low. To overcome this defect, we divide a large tensor into four small tensors using wavelet transform. These four small tensors are a main part and three detail parts of the original tensor, and the size of each small tensor is about a quarter of the original tensor. In this way, we transform a large tensor completion problem into four small tensor completion problems, which greatly improves the efficiency of the algorithm. Since we recover the main part and the detail part of the tensor separately, compared with the traditional method, the recovery effect of the algorithm is also improved. At the same time, we introduce a nonconvex function to better relax the TNN and l_1-norm of the tensor. The experimental results demonstrated that our proposed models and methods led to impressive improvements over state-of-the-art methods.

CONFLICT OF INTEREST

The authors declare no potential conflict of interest.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ENDNOTES

[1]http://sipi.usc.edu/database/.
[2]http://sipi.usc.edu/database/.
[3]https://brainweb.bic.mni.mcgill.ca/brainweb/.
[4]http://trace.eas.asu.edu/yuv/.

ORCID

Quan Yu https://orcid.org/0000-0002-8051-7477
Minru Bai https://orcid.org/0000-0002-9960-6138

REFERENCES

1. Hu W, Tao D, Zhang W, Xie Y, Yang Y. The twist tensor nuclear norm for video completion. IEEE Trans Neural Netw Learn Syst. 2017;28(12):2961–73.
2. Liu XY, Aeron S, Aggarwal V, Wang X. Low-tubal-rank tensor completion using alternating minimization. IEEE Trans Inf Theory. 2020;66(3):1714–37.

3. Lu C, Peng X, Wei Y. Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms. 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR). Long Beach: IEEE; 2019. p. 5989–97.
4. Qiu D, Bai M, Ng MK, Zhang X. Nonlocal robust tensor recovery with nonconvex regularization. Inverse Probl. 2021;37(3):035001.
5. Yu Q, Zhang X, Chen Y, Qi L. Low Tucker rank tensor completion using a symmetric block coordinate descent method. Numer Linear Algebra Appl. 2022;e2464.
6. Wang PP, Li L, Cheng GH. Low-rank tensor completion with sparse regularization in a transformed domain. Numer Linear Algebra Appl. 2021;28(6):e2387.
7. Song G, Ng MK, Zhang X. Robust tensor completion using transformed tensor singular value decomposition. Numer Linear Algebra Appl. 2020;27(3):e2299.
8. Jiang TX, Ng MK, Zhao XL, Huang TZ. Framelet representation of tensor nuclear norm for third-order tensor completion. IEEE Trans Image Process. 2020;29:7233–44.
9. Zheng YB, Huang TZ, Zhao XL, Jiang TX, Ji TY, Ma TH. Tensor N-tubal rank and its convex relaxation for low-rank tensor recovery. Inform Sci. 2020;532:170–89.
10. Yu Q, Zhang X. T-product factorization based method for matrix and tensor completion problems. Comput Optim Appl. 2022. https://doi.org/10.1007/s10589-022-00439-y
11. Qiu D, Bai M, Ng MK, Zhang X. Robust low-rank tensor completion via transformed tensor nuclear norm with total variation regularization. Neurocomputing. 2021;435:197–215.
12. Zhao X, Bai M, Ng MK. Nonconvex optimization for robust tensor completion from grossly sparse observations. J Sci Comput. 2020;85(2):46.
13. Zhou H, Zhang D, Xie K, Chen Y. Spatio-temporal tensor completion for imputing missing internet traffic data. 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC). Nanjing: IEEE; 2015. p. 1–7.
14. Xie K, Wang L, Wang X, Xie G, Wen J, Zhang G, et al. Accurate recovery of internet traffic data: a sequential tensor completion approach. IEEE/ACM Trans Netw. 2018;26(2):793–806.
15. Zhou H, Zhang D, Xie K, Chen Y. Robust spatio-temporal tensor recovery for internet traffic data. 2016 IEEE Trustcom/BigDataSE/ISPA. Volume 2016. Tianjin: IEEE; 2016. p. 1404–11.
16. Tucker LR. Some mathematical notes on three-mode factor analysis. Psychometrika. 1966;31(3):279–311.
17. Carroll JD, Chang JJ. Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. Psychometrika. 1970;35(3):283–319.
18. Kiers HAL. Towards a standardized notation and terminology in multiway analysis. J Chemometr. 2000;14(3):105–22.
19. Hillar CJ, Lim LH. Most tensor problems are NP-Hard. J ACM. 2013;60(6):1–39.
20. Landsberg J. Tensors: geometry and applications. Vol 128. Providence: American Mathematical Society; 2012.
21. Mu C, Huang B, Wright J, Goldfarb D. Square Deal: Lower Bounds and Improved Relaxations for Tensor Recovery. Proceedings of the 31st international conference on machine learning. ICML'14. Volume 32. Beijing: JMLR.org; 2014. p. II-73–81.
22. Liu J, Musialski P, Wonka P, Ye J. Tensor completion for estimating missing values in visual data. IEEE Trans Pattern Anal Mach Intell. 2013;35(1):208–20.
23. Martin CD, Shafer R, LaRue B. An order-p tensor factorization with applications in imaging. SIAM J Sci Comput. 2013;35(1):A474–90.
24. Xu Y. Alternating proximal gradient method for sparse nonnegative Tucker decomposition. Math Program Comput. 2015;7(1):39–70.
25. Zhang Z, Ely G, Aeron S, Hao N, Kilmer M. Novel methods for multilinear data completion and de-noising based on tensor-SVD. 2014 IEEE conference on computer vision and pattern recognition. Columbus: IEEE; 2014. p. 3842–9.
26. Zhou P, Lu C, Lin Z, Zhang C. Tensor factorization for low-rank tensor completion. IEEE Trans Image Process. 2018;27(3):1152–63.
27. Kilmer ME, Braman K, Hao N, Hoover RC. Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging. SIAM J Matrix Anal Appl. 2013;34(1):148–72.
28. Kilmer ME, Martin CD. Factorization strategies for third-order tensors. Linear Algebra Appl. 2011;435(3):641–58.
29. Semerci O, Hao N, Kilmer ME, Miller EL. Tensor-based formulation and nuclear norm regularization for multienergy computed tomography. IEEE Trans Image Process. 2014;23(4):1678–93.
30. Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. J Am Stat Assoc. 2001;96(456):1348–60.
31. Yang Y, Feng Y, Suykens JAK. Robust low-rank tensor recovery with regularized redescending M-estimator. IEEE Trans Neural Netw Learn Syst. 2016;27(9):1933–46.
32. Ahn M, Pang JS, Xin J. Difference-of-convex learning: directional stationarity, optimality, and sparsity. SIAM J Optim. 2017;27(3):1637–65.
33. Bai M, Zhang X, Shao Q. Adaptive correction procedure for TVL1 image deblurring under impulse noise. Inverse Probl. 2016;32(8):085004.
34. Jiang TX, Huang TZ, Zhao XL, Deng LJ. Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm. J Comput Appl Math. 2020;372:112680.
35. Tao PD, An LTH. Convex analysis approach to DC programming: theory, algorithms and applications. Acta Math Vietnam. 1997;22(1):289–355.
36. Rojo O, Rojo H. Some results on symmetric circulant matrices and on symmetric centrosymmetric matrices. Linear Algebra Appl. 2004;392(15):211–33.
37. Lu C, Feng J, Chen Y, Liu W, Lin Z, Yan S. Tensor robust principal component analysis with a new tensor nuclear norm. IEEE Trans Pattern Anal Mach Intell. 2020;42(4):925–38.
38. Ding C. An introduction to a class of matrix optimization problems. Singapore: National University of Singapore; 2012.
39. Bochnak J, Coste M, Roy MF. Real algebraic geometry. Berlin: Springer; 1998.

40. Yang L, Pong TK, Chen X. Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction. SIAM J Imag Sci. 2017;10(1):74–110.
41. Bolte J, Sabach S, Teboulle M. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math Program. 2014;146(1-2):459–94.
42. Attouch H, Bolte J, Svaiter BF. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. Math Program. 2013;137(1-2):91–129.
43. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process. 2004;13(4):600–12.
44. Zhang L, Zhang L, Mou X, Zhang D. FSIM: a feature similarity index for image quality assessment. IEEE Trans Image Process. 2011;20(8):2378–86.

How to cite this article: Yu Q, Bai M. Nonconvex optimization for third-order tensor completion under wavelet transform. Numer Linear Algebra Appl. 2023;e2489. https://doi.org/10.1002/nla.2489