

# Accurate Regularized Tucker Decomposition for Image Restoration

Wenwu Gong<sup>a</sup>, Zhejun Huang<sup>a</sup> and Lili Yang<sup>a,\*</sup>

<sup>a</sup>Department of Statistics and Data Science, Southern University of Science and Technology, Guangdong, Shenzhen 518055, PR China

## ARTICLE INFO

### Keywords:

Low Tucker rank decomposition  
Factor priors  
Sparsity  
Alternating proximal gradient  
Image restoration

## ABSTRACT

We propose a new accurate regularized Tucker decomposition (ARTD) method for image restoration (IR), which considers global low-rankness and local similarity of intrinsic image characteristics. Specifically, global low-rankness is represented by a sparse Tucker core, whereas the local similarity is captured using the nonnegative factor matrix and manifold regularization term. Sparse nonnegative Tucker decomposition (SNTD) and graph nonnegative Tucker decomposition (GNTD) can be considered a special case of ARTD. We propose and implement an effective Alternating Proximal Gradient (APG) based algorithm to solve the ARTD model and deduce the closed-form updating rules. Notably, ARTD does not need to tune the Tucker rank and provides an initialization strategy and a first-order feedback control rule to accelerate its convergence. Experiments on real IR problems show that our method outperforms the state-of-the-art methods.

## 1. Introduction

Image restoration (IR), as a fundamental problem in image processing, aims to reconstruct the original image from its corrupted observation and is still facing many challenges. On the one hand, the image has triple channels and is always globally correlated. Hence, tensor completion (TC) [1, 2] is naturally one of the most critical topics in IR. However, a tensor is more challenging to analyze than a matrix due to the complicated nature of higher-order arrays. For example, matrix nuclear-norm minimization and tensor rank minimization suffers from computation burden [3, 4] and cannot work well for images with high-level corruption [5]. Low-rank tensor factorization is essentially not unique and determines the tensor rank is NP-hard [6]. So, recovering the missing entries from a significantly under-sampled tensor has presented various theoretical and computational challenges.

On the other hand, IR is a typical ill-posed inverse problem. Fortunately, it can be solved using low-rank tensor approximation (LRTA) from the perspective of Bayesian [7]. To our knowledge, state-of-the-art LRTA-based IR methods [8, 9] are model-based optimization schemes, where the objective function is based on the low-rank approximation  $\hat{\mathcal{X}}$ , image noiseless degradation process  $\mathcal{X}^0 = \Omega \odot \mathcal{X}$  and the given image priors  $R(\mathcal{X})$ . Mathematically, the widely used MAP (maximum a posterior) model can generally be expressed as

$$\underset{\mathcal{X}}{\text{minimize}} \quad L(\hat{\mathcal{X}}, \mathcal{X}^0) + \lambda R(\mathcal{X}), \quad (1)$$

where  $L(\cdot, \cdot)$  is the loss function to measure the similarity between output image  $\hat{\mathcal{X}}$  and corrupt input image  $\mathcal{X}^0$ , and  $R(\cdot)$  is the regularization term (or prior knowledge).

In reality, an image collected from the real-world has intrinsic characteristics, i.e., global correlation along the spectrum (GCS) [1, 10, 11], local similarity with the adjacent pixels and across space (LSS) [5, 9, 12]. Consequently, the core of IR is to rationally extract prior structures to model such intrinsic image characteristics and then fully utilize such knowledge to rectify the configuration of the clean image in a low-rank approximation manner. Below, we review some commonly utilized prior structures in the TC problem and then briefly discuss the Tucker-based IR techniques applied to color images. The summation of these priors used in some representative TC methods is shown in Table 1.

\* ARTD for IR

\*Corresponding author

Email addresses: 12031299@mail.sustech.edu.cn (W. Gong); huangzj@mail.sustech.edu.cn (Z. Huang); yangll@sustech.edu.cn (L. Yang)

URL: <https://gongwenwu.netlify.app/> (W. Gong)

ORCID(s): 0000000280190582 (W. Gong)

**Table 1**

A summary of existing TC methods utilizing different priors

TC methods	Types of the priors			Forms
	Low-rankness	Sparsity	Smoothness	
HaLRTC [1]	✓			Rank minimization
LRTC-TV [13]	✓		✓	Rank minimization
t-SVD [4]	✓			Rank minimization
FATC [10]	✓			Tensor Train
TT-TV [9]	✓		✓	Tensor Train
SMF [14]	✓		✓	Decomposition
LRTD-TV [15]	✓		✓	Decomposition
SCP [5]	✓		✓	Decomposition
BGCP [16]	✓			Decomposition
STDC [12]	✓		✓	Decomposition
KBR [17]	✓	✓		Decomposition
LR-SETD [18]	✓	✓	✓	Decomposition
SNTD [19]	✓	✓		Decomposition
GNTD [20]	✓		✓	Decomposition
ARTD	✓	✓	✓	Decomposition

### 1.1. Related work

Given GCS information, the low-rankness and sparsity knowledge are the commonly utilized prior structures in the TC problems. For example, a rank-minimization term was proposed for TC in [1], which directly extended the rank of the matrix to higher order by simply summing up ranks (or its relaxations) along all tensor modes. Furthermore, tensor trains with multi-rank are also applicable, even suffering huge computation [21] and the sensitivity of the obtained solution to its random initialization [10]; the tensor tubal rank and its convex surrogate tensor nuclear norm (TNN) [4] is also researched when the tensor is 3-rd order. Another way to tackle TC is based on the tensor sparsity with tensor decomposition forms. Kronecker-basis-representation (KBR) measure [11], interpreted as a regularization for the number of rank-1 Kronecker bases, was proposed to encode the inherent global spectral correlations in IR. And a sparsity-inducing prior combined with CANDECOMP/ PARAFAC (CP) decomposition model was presented in [16] to solve the rank determination using a hierarchical Bayesian probabilistic framework.

To capture LSS knowledge, the total variation (TV) terms have been integrated into the low-rank term in [9, 13–15] to enhance model performance. And the smoothness constraints combined with low-rank CP decomposition were studied in [5] for image processing. Manifold learning, a specified Laplacian graph is constructed to encode the geometrical information of local similarity on a low-dimensional manifold [22], was also proposed to capture the relationships among data in the with-in and with-cross directions [23] for the tensor completion problem. Furthermore, the deep image priors (DIP) [24] method, a newly smoothness priors, was recently proposed by designing a deep convolutional neural network with a rectified linear unit (ReLU) activation function in [25] to encode the intrinsic image characteristics and achieve better performance in IR. This deep learning method, however, needs training image pairs, and the image priors cannot be interpreted explicitly.

Based on the GCS and LSS prior structures, many Tucker-based IR problem methods exist. Chen et al. proposed a method called simultaneous tensor decomposition and completion (STDC) that exploits rank minimization prior and joint-manifold prior by employing the Tucker decomposition to obtain the low-rank approximation of the corrupt input image [12]. Although the proposal reconstructs the missing entries and simultaneously captures the underlying model structure without predefined ranks, this rank-summation term performs poorly in high-level corruption. Guided by nonnegative matrix factorization (NMF), Kim et al. proposed a nonnegative Tucker decomposition (NTD) for face image reconstruction [26]. Furthermore, Xu proposed a sparse nonnegative Tucker decomposition (SNTD) model for tensor completion [19]. However, these nonnegative Tucker decompositions have yet to utilize the prior knowledge entirely and should predefine the low Tucker rank. To faithfully deliver the GCS underlying image tensor, Xie et

al. proposed a KBR tensor sparsity measure to cope with the multispectral image denoising problem [17]. Pan et al. proposed a unified low-rank and sparse enhanced Tucker decomposition (LR-SETD) model for image completion [18]. However, it does not fully utilize the LSS to reconstruct the corrupt image tensor. Thus, the IR performance still has much room for further improvement.

## 1.2. Motivations and contributions

The challenges of IR remain in capturing the images' prior knowledge further to improve the performance in terms of efficiency and accuracy. This paper proposes a new accurate, regularized Tucker decomposition (ARTD) method for IR. Specifically, a sparse Tucker core tensor is used to characterize the GCS knowledge of the tensor image. Compared with KBR sparsity measure [11], we relax orthogonal constraint and use the sparsity of full-size core tensor to encode the low-rankness. Besides this, inspired by the outstanding achievement of convolution neural network with ReLU activation function in image processing, the nonnegative constraints of factor matrices are added in the optimization problem (1). The benefits of adding the nonnegative constraints are twofold: first, it helps to extract image features [27], and second, it further ensures the sparsity of the Tucker core tensor [28]. Furthermore, we provide a specified manifold regularization to capture the LSS knowledge to encode the image's geometrical similarity [29] on a low-dimensional manifold. Our contributions to this paper are summarised as follows.

- 1) A new prior structure for IR is proposed. We relax the orthogonal to be nonnegative and constrain manifold regularization of factor matrices in KBR sparsity measure to enhance the IR performance. To the best of our knowledge, we adopt these priors, which can capture the image's intrinsic characteristics in the spatial and spectral directions simultaneously.
- 2) We propose and implement an effective Alternating Proximal Gradient (APG) based algorithm for solving the ARTD model and deduce the closed-form updating rules for each involved parameter. Specifically, a first-order feedback control rule is established to speed up the algorithm's efficiency. Furthermore, our method can degenerate into graph nonnegative Tucker decomposition (GNTD) [20] and SNTD [19], which are well-studied.
- 3) The experiments on benchmark data with different missing scenarios demonstrate that the proposed method achieves state-of-the-art performance on IR.

The rest of this paper is structured as follows. Section 2 presents some basic tensor notations. The proposed model and the APG-based algorithm with guaranteed convergence and efficient computation are shown in Section 3. Experimental results are demonstrated in Section 4. Finally, Section 5 concludes this paper.

## 2. Notations

Throughout this paper, we use calligraphy font for tensors, such as  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , whose element is denoted as  $x_{i_1 i_2 \dots i_n}$ . The bold uppercase letters for matrices, such as  $\mathbf{U} \in \mathbb{R}^{I_1 \times I_2}$ , bold lowercase letters for vectors, such as  $\mathbf{a} \in \mathbb{R}^{I_1}$ , and Greek alphabet letters for scalars, such as  $\alpha, \beta$ .

The Frobenius norm of a tensor is defined as

$$\|\mathcal{X}\|_F = \sqrt{\sum_{i_1=1}^{I_1} \dots \sum_{i_n=1}^{I_N} x_{i_1 \dots i_n}^2}.$$

We denote the mode- $n$  unfolding (i.e., matricization) of an  $N$ -order tensor  $\mathcal{X}$  by  $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times \prod_{j \neq n} I_j}$ .

Given a tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , Tucker decomposition can be denoted as a core tensor  $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times \dots \times r_N}$  multiplying a matrix  $\mathbf{U}_n \in \mathbb{R}^{I_n \times r_n}$  along each mode  $n$ , i.e.,  $\mathcal{X} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \dots \times_N \mathbf{U}_N$ . Based on the matrix Kronecker product  $\otimes$ , the Tucker decomposition can also be represented by

$$\mathbf{X}_{(n)} = \mathbf{U}_n \mathbf{G}_{(n)} \mathbf{V}_n^T$$

where  $\mathbf{V}_n = (\mathbf{U}_N \otimes \dots \otimes \mathbf{U}_{n+1} \otimes \mathbf{U}_{n-1} \otimes \dots \otimes \mathbf{U}_1)$ . Furthermore, it is not difficult to verify that  $\text{vec}(\mathcal{X}) = (\mathbf{U}_N \otimes \dots \otimes \mathbf{U}_n \otimes \dots \otimes \mathbf{U}_1) \text{vec}(\mathcal{G}) = (\otimes_{n=N}^1 \mathbf{U}_n) \text{vec}(\mathcal{G})$ .

Finally, for a given tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  and the observed index set  $\Omega$ , we define  $\mathcal{X}_\Omega$  is a projection that keeps the entries of  $\mathcal{X}$  in  $\Omega$  while making others be zeros, i.e.,

$$\mathcal{X}_\Omega := \begin{cases} x_{i_1 i_2 \dots i_n}, & \text{if } (i_1, i_2, \dots, i_n) \in \Omega \\ 0, & \text{otherwise.} \end{cases}$$

### 3. Accurate regularized Tucker decomposition

#### 3.1. Manifold regularization

We first introduce the general manifold regularization term for the IR problem to reconstruct the clean image from its corrupted observations. Given an undirected local  $p$ -connected graph with a similarity matrix  $\mathbf{W} \in \mathbb{R}^{N \times N}$  for data  $\mathbf{X} \in \mathbb{R}^{I \times N}$  representation, where  $0 \leq w_{ij} \leq 1$ ,  $i, j = 1, \dots, N$  represents the probability that data point  $x_i, x_j$  are close in the geometry space [30], we construct a manifold regularization term (2) to capture an optimal low-dimensional representation  $\mathbf{U}$  for given data  $\mathbf{X}$ .

$$\sum_{i=1}^N \sum_{j=1}^N w_{ij} \left\| \mathbf{u}_i - \mathbf{u}_j \right\|_2^2 = \text{tr} (\mathbf{U}^T \mathbf{L} \mathbf{U}), \quad \mathbf{L} = \mathbf{D} - \mathbf{W} \quad (2)$$

where  $\mathbf{u}_i$  is the column vector of  $\mathbf{U}^T$  and  $\mathbf{D} \in \mathbb{R}^{N \times N}$  is a diagonal matrix with diagonal elements  $d_{ii} = \sum_{j=1}^N w_{ij}$ ,  $i = 1, \dots, N$ . Note that  $\mathbf{L}$  is a Laplacian matrix designed by some prior knowledge, which enforces the smoothness of the low-dimensional feature  $\mathbf{U}$  and captures the local similarity in tensors [23].

Since the image resides in low-dimensional submanifolds, the manifold regularization can be used to learn the LSS information [12]. More specifically, the proposal ARTD uses the K-nearest neighbor (KNN) method to select  $p$  nearest neighbors for each unfolding image tensor and then construct the similarity matrix  $\mathbf{W}$  using the Euclidean measure to represent the neighbor connections.

The most important parts of manifold regularization are two folds: the Laplacian matrix construction and similarity matrix calculation. In this paper, we apply different strategies to do that [31]. On the one hand, we suppose that the intrinsic manifold is located in the convex envelope of the previously given manifold Laplacian matrices  $\{\mathbf{L}_1, \dots, \mathbf{L}_K\}$ , such that, the ensemble graph Laplacians is denoted as a linear combination of  $K$  candidate Laplacians:

$$\mathbf{L} = \sum_{k=1}^K \tau_k \mathbf{L}_k, \quad \text{s.t. } \sum_{k=1}^K \tau_k = 1, \quad \tau_k \geq 0, \quad (3)$$

where  $\tau_k$  is the combined weight of  $k$ -th graph Laplacian and the constraint  $\tau_k \geq 0$  is used to avoid negative contribution. On the other hand, we use kernel weighting to define the similarity matrix. For each data point  $x_i$  belongs to unfolding matrix  $\mathbf{X}$ , if nodes  $i$  and  $j$  are connected, the kernel weight is defined as (4)

$$w_{ij} = e^{-\left(\|x_i - x_j\|^2\right)/\sigma^2}, \quad (4)$$

where  $\sigma^2$  denotes the divergence, and we set  $\sigma^2 = 1$  in our experiments.

#### 3.2. ARTD model

This paper proposes a new prior structure for the IR problem. A nonnegative and manifold regularized factor matrix combined with a sparse Tucker core tensor is used to capture the image's intrinsic characteristics in a low-rank approximation manner. So, we aim to solve the following optimization problem:

Let  $\mathcal{X}^0 \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  be a corrupt image tensor ( $N = 3$ , and we consider a more general case here) and  $\Omega$  be the observed index set corresponding to the entries of clean image tensor  $\mathcal{X}$ , the ARTD model is

$$\begin{aligned} \underset{\mathcal{G}; \{\mathbf{U}_n\}; \mathcal{X}}{\text{minimize}} \quad & \mathbb{F}(\mathcal{G}, \{\mathbf{U}_n\}, \mathcal{X}) \triangleq \frac{1}{2} \left\| \mathcal{X} - \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \dots \times_N \mathbf{U}_N \right\|_F^2 + \frac{\alpha}{2} \left( \sum_{n=1}^N \text{tr}(\mathbf{U}_n^T \mathbf{L}_n \mathbf{U}_n) \right) + \beta \|\mathcal{G}\|_1, \\ \text{s.t. } & \mathbf{U}_n \in \mathbb{R}_+, \quad \mathcal{X}_\Omega = \mathcal{X}_\Omega^0, \end{aligned} \quad (5)$$

where  $\alpha$  and  $\beta$  are positive penalty parameters, and  $\mathbf{L}_n$  represents the different ensemble Laplacian regularization for image mode- $n$  unfolding matrix. Note that when  $\beta = 0$  and the Tucker core  $\mathcal{G}$  are nonnegative, the ARTD model is transformed into the GNTD model; if  $\alpha = 0$  is further required, the ARTD becomes the NTD model. When  $\alpha = 0$  and the Tucker core  $\mathcal{G}$  is required to be nonnegative, the ARTD model is degraded into the SNTD model.

**Remark:** Motivated by [17], the low-rankness of the Tucker decomposition implies that many elements of the core tensor equal zero; it is, therefore, rational to impose sparsity on the Tucker core tensor. To better understand tensor decomposition, we can utilize nonnegative components. This approach can yield a unique additive representation and extract image features through various parts [19, 32]. Note that the non-negativity constraints on the factor matrix ensure stronger correlations across image data along some channels and provide better compression than unconstrained variants [33]. Then the elements in the core tensor along this mode tend to zeroes, which ensures the Tucker core tensor has more sparsity.

### 3.3. Algorithm for ARTD model

Since the ARTD Model (5) is generally nonconvex but convex in each block of variables, we use the block coordinate descent scheme and apply the alternating proximal gradient [34], an efficient method for solving multi-convex optimization problems to solve (5).

**Updating  $\mathbf{U}$**  We unfold the objective function (5) in mode- $n$  for given tensor  $\mathcal{X}$ , then the latent factor matrices optimization subproblems are given as (6).

$$\begin{aligned} \underset{\mathbf{U}_n \geq 0}{\text{minimize}} \quad & \ell(\mathbf{U}_n) = \frac{1}{2} \left\| \mathbf{X}_{(n)} - \mathbf{U}_n \mathbf{G}_{(n)} \mathbf{V}_n^T \right\|_F^2 + \frac{\alpha}{2} \text{tr} (\mathbf{U}_n^T \mathbf{L}_n \mathbf{U}_n), \\ & \mathbf{V}_n = (\mathbf{U}_N \otimes \cdots \otimes \mathbf{U}_{n+1} \otimes \mathbf{U}_{n-1} \otimes \cdots \otimes \mathbf{U}_1), \end{aligned} \quad (6)$$

where  $\mathbf{L}_n = \mathbf{D}_n - \mathbf{W}_n$  represents the ensemble Laplacian matrix of the unfolding matrix  $\mathbf{X}_n$ .

**Proposition 1:** The objective function of subproblem (6) is differentiable and convex. Furthermore, the gradients  $\nabla_{\mathbf{U}_n} \ell(\mathbf{U}_n)$  are both Lipschitz continuous with the Lipschitz constant

$$L_{\mathbf{U}_n} = \left\| \mathbf{G}_{(n)} \mathbf{V}_n^T \mathbf{V} \mathbf{G}_{(n)}^T \right\|_2 + \alpha \|\mathbf{L}_n\|_2.$$

*Proposition 1* 1 is obvious, and the Lipschitz constant can be calculated by the Hessian matrix of  $\ell(\mathbf{U}_n)$ , we omit the detail here.

Based on the *Proposition 1*, we use the prox-linear updating scheme to obtain the stationary point of the latent factor matrices.

$$\hat{\mathbf{U}}_n = \underset{\mathbf{U}_n \geq 0}{\text{argmin}} \Psi(\mathbf{U}_n, \tilde{\mathbf{U}}_n) \equiv \left\langle \nabla_{\mathbf{U}_n} \ell(\tilde{\mathbf{U}}_n), \mathbf{U}_n - \tilde{\mathbf{U}}_n \right\rangle + \frac{L_{\mathbf{U}_n}}{2} \|\mathbf{U}_n - \tilde{\mathbf{U}}_n\|_F^2, \quad (7)$$

where  $\tilde{\mathbf{U}}_n$  denotes the extrapolated point, and  $\nabla_{\mathbf{U}_n} \ell(\mathbf{U}_n) = \mathbf{U}_n \mathbf{G}_{\mathbf{V}}^n \mathbf{G}_{\mathbf{V}}^{nT} - \mathbf{X}_{(n)} \mathbf{G}_{\mathbf{V}}^{nT} + \alpha \mathbf{L}_n \mathbf{U}_n$ . To solve (7), we consider the KarushKuhnTucker (KKT) conditions as follows

$$\nabla_{\mathbf{U}_n} \Psi(\mathbf{U}_n, \tilde{\mathbf{U}}_n) \geq 0, \quad \mathbf{U}_n \geq 0, \quad \text{and} \quad \nabla_{\mathbf{U}_n} \Psi(\mathbf{U}_n, \tilde{\mathbf{U}}_n) \odot \mathbf{U}_n = 0, \quad (8)$$

where  $\nabla_{\mathbf{U}_n} \Psi(\mathbf{U}_n, \tilde{\mathbf{U}}_n) = \nabla_{\mathbf{U}_n} \ell(\tilde{\mathbf{U}}_n) + L_{\mathbf{U}_n} (\mathbf{U}_n - \tilde{\mathbf{U}}_n)$  and  $\odot$  is the Hadamard product. So, the latent factor matrix  $\{\mathbf{U}_n^k\}$  is updated by

$$\mathbf{U}_n \leftarrow \mathcal{P}_+ \left( \tilde{\mathbf{U}}_n - \frac{1}{L_{\mathbf{U}_n}} \nabla_{\mathbf{U}_n} \ell(\tilde{\mathbf{U}}_n) \right), \quad (9)$$

where  $\mathcal{P}_+(\mathbf{U})$  is the function that projects the negative entries of  $\mathbf{U}$  into zeros and  $\tilde{\mathbf{U}}_n$  is updated by

$$\tilde{\mathbf{U}}_n^k = \mathbf{U}_n^k + \omega_k (\mathbf{U}_n^k - \mathbf{U}_n^{k-1}), \quad \text{for } k \geq 1 \quad (10)$$

with the update step size  $\omega_k$

$$\omega_k = \frac{t^{k-1} - 1}{t^k}, \quad t^k = \frac{1 + \sqrt{4(t^{k-1})^2 + 1}}{2} \quad \text{for } k \geq 1 \text{ and } t^0 = 1 \quad (11)$$

**Updating  $\mathcal{G}$**  We also use the proximal gradient method to update the core tensor  $\mathcal{G}$  using the vectorization form

$$\underset{\mathcal{G}}{\text{minimize}} \frac{1}{2} \left\| \text{vec}(\mathcal{X}) - (\otimes_{n=1}^N \mathbf{U}_n) \text{vec}(\mathcal{G}) \right\|_F^2 + \beta \|\text{vec}(\mathcal{G})\|_1 := f(\mathcal{G}) + g(\mathcal{G}). \quad (12)$$

**Proposition 2:** The objective function of subproblem (12) is the sum of two convex functions, and the gradient  $\nabla_{\mathcal{G}} f(\mathcal{G})$  is Lipschitz continuous with the Lipschitz constant  $L_{\mathcal{G}} = \left\| \otimes_{n=1}^N \mathbf{U}_n^T \mathbf{U}_n \right\|_2 = \prod_{n=1}^N \|\mathbf{U}_n^T \mathbf{U}_n\|_2$ .

**Proof of Proposition 2.** It is straightforward to verify the convex and Lipschitz continuous properties. For the vectorization form, we have

$$\text{vec}(\nabla_{\mathcal{G}} f(\mathcal{G})) = (\otimes_{n=1}^N \mathbf{U}_n^T \mathbf{U}_n) \text{vec}(\mathcal{G}) - (\otimes_{n=1}^N \mathbf{U}_n^T) \text{vec}(\mathcal{X}). \quad (13)$$

Here, the Hessian matrix  $\text{vec}(\nabla_{\mathcal{G}}^2 f(\mathcal{G})) = \otimes_{n=1}^N \mathbf{U}_n^T \mathbf{U}_n$ , which is positive semidefinite and assures the convexity of  $f(\mathcal{G})$ . For any given  $\mathcal{G}_1$  and  $\mathcal{G}_2$ , we have

$$\begin{aligned} \left\| \text{vec}(\nabla_{\mathcal{G}} f(\mathcal{G}_1)) - \text{vec}(\nabla_{\mathcal{G}} f(\mathcal{G}_2)) \right\|_F &= \left\| \otimes_{n=1}^N \mathbf{U}_n^T \mathbf{U}_n (\text{vec}(\mathcal{G}_1) - \text{vec}(\mathcal{G}_2)) \right\|_F \\ &\leq \left\| \otimes_{n=1}^N \mathbf{U}_n^T \mathbf{U}_n \right\|_2 \left\| \text{vec}(\mathcal{G}_1) - \text{vec}(\mathcal{G}_2) \right\|_F \\ &= \prod_{n=1}^N \left\| \mathbf{U}_n^T \mathbf{U}_n \right\|_2 \left\| \text{vec}(\mathcal{G}_1) - \text{vec}(\mathcal{G}_2) \right\|_F. \end{aligned} \quad (14)$$

So, the Lipschitz constant of  $\nabla_{\mathcal{G}} f(\mathcal{G})$  is  $L_{\mathcal{G}} = \prod_{n=1}^N \|\mathbf{U}_n^T \mathbf{U}_n\|_2$ . This completes the proof.  $\square$

Based on *Proposition 2*, we denote the core tensor optimization function as (15)

$$\hat{\mathcal{G}} = \underset{\mathcal{G}}{\text{argmin}} \left\langle \nabla_{\mathcal{G}} f(\tilde{\mathcal{G}}), \mathcal{G} - \tilde{\mathcal{G}} \right\rangle + \frac{L_{\mathcal{G}}}{2} \|\mathcal{G} - \tilde{\mathcal{G}}\|_F^2 + \beta \|\mathcal{G}\|_1, \quad (15)$$

where  $\tilde{\mathcal{G}}$  denotes the extrapolated point and  $\nabla_{\mathcal{G}} f(\mathcal{G})$  can be calculated by (16) using the properties of Kronecker product.

$$\nabla_{\mathcal{G}} f(\mathcal{G}) = \mathcal{G} \times_1 \mathbf{U}_1^T \mathbf{U}_1 \times_2 \cdots \times_N \mathbf{U}_N^T \mathbf{U}_N - \mathcal{X} \times_1 \mathbf{U}_1^T \times_2 \cdots \times_N \mathbf{U}_N^T. \quad (16)$$

So, we have the Tucker core tensor updating rule (17) by using the soft-thresholding operator  $S_{\zeta}(\cdot)$  [19, 35]

$$\hat{\mathcal{G}} = T_{L_{\mathcal{G}}}^{f,g}(\mathcal{G}) = S_{\frac{\beta}{L_{\mathcal{G}}}} \left( \tilde{\mathcal{G}} - \frac{1}{L_{\mathcal{G}}} \nabla_{\mathcal{G}} f(\tilde{\mathcal{G}}) \right), \quad (17)$$

where  $\tilde{\mathcal{G}}$  is updated by (18) with the update step size  $\omega_k$  (11)

$$\tilde{\mathcal{G}}^k = \mathcal{G}^k + \omega_k (\mathcal{G}^k - \mathcal{G}^{k-1}), \text{ for } k \geq 1. \quad (18)$$

**Updating scheme** Suppose the current iterate is  $k$ -step, we update the core tensor  $\mathcal{G}^k$  by

$$\mathcal{G}^{k+1} = S_{\frac{\beta}{L_{\mathcal{G}}^k}} \left( \tilde{\mathcal{G}}^k - \frac{1}{L_{\mathcal{G}}^k} \nabla_{\mathcal{G}} f(\tilde{\mathcal{G}}^k) \right), \quad (19)$$

and the latent factor matrices  $\{\mathbf{U}_n^k\}$  using

$$\mathbf{U}_n^{k+1} = \mathcal{P}_+ \left( \tilde{\mathbf{U}}_n^k - \frac{1}{L_{\mathbf{U}_n^k}} \nabla_{\mathbf{U}_n} \ell(\tilde{\mathbf{U}}_n^k) \right). \quad (20)$$

Then, we check whether the objective function (5) is decreasing during the APG projection. If so, we re-update  $\mathcal{G}^k$  and  $\mathbf{U}_n^k$  along with the extrapolated point updating.

Technically, we propose an initial strategy where the  $\{\mathbf{U}_n\}$  is generated randomly and then processed by normalization. It is observed that the algorithms perform better with this starting point than the Higher-order Single Value Decomposition (HOSVD) [36] in convergence speed. Furthermore, at the end of iteration  $k$ , we apply the dynamic feedback correction mechanism (21), a first-order feedback control rule [37], to re-update tensor  $\mathcal{X}^k$  when having  $\{\mathbf{U}_n^k\}$  and  $\mathcal{G}^k$

$$\mathcal{X}^{k+1}|_{\Omega} = \mathcal{X}^0|_{\Omega} + \gamma(\mathcal{X}^k|_{\Omega} - \mathcal{Z}^k|_{\Omega}), \quad \mathcal{X}^{k+1}|_{\bar{\Omega}} = \mathcal{Z}^k|_{\bar{\Omega}}, \quad (21)$$

where  $\mathcal{Z}^k = \mathcal{G}^k \times_1 \mathbf{U}_1^k \times_2 \dots \times_N \mathbf{U}_N^k$ ,  $\bar{\Omega}$  is the complement set of  $\Omega$ , and  $0 \leq \gamma \leq 1$  is a user defined hyper-parameter to control the correction. By doing this, we show that it can reduce the low-rank approximation error and obtain more stable results.

For the IR problem, we calculate the reconstructed tensor  $\hat{\mathcal{X}} = \mathcal{X}^0|_{\Omega} + \mathcal{Z}^{k+1}|_{\bar{\Omega}}$  as the final result, if one of the following conditions is satisfied.

$$\begin{aligned} & \|(\mathcal{X}^{k+1} - \mathcal{X}^k)\|_F \|\mathcal{X}\|_F^{-1} < tol, \quad \text{for some } k, \\ & \text{or } \frac{|\mathbb{F}_{\Omega}^i - \mathbb{F}_{\Omega}^{i+1}|}{1 + \mathbb{F}_{\Omega}^i} \leq tol, \quad i = k, k+1, k+2, (\text{three consecutive iterations}) \end{aligned} \quad (22)$$

where  $\mathbb{F}_{\Omega}^k$  denotes the objective function under observed index  $\Omega$  at iteration  $k$ , and  $tol$  is a small specified positive value.

The proposed algorithm for ARTD-based image restoration problems can be summarized in Algorithm 1, and we denote the proposed algorithm as APG-RTD for convenience.

---

**Algorithm 1** APG-RTD for Image Restoration

---

- 1: **Input:** To-be-reconstructed image tensor  $\mathcal{X}^0$ , indices of observed entries  $\Omega$ , iteration number  $K$ , and the parameters  $\alpha \geq 0$ ,  $\beta \geq 0$ ,  $0 \leq \gamma \leq 1$  and  $tol = 1e^{-4}$ .
  - 2: **Output:** Reconstructed image tensor  $\hat{\mathcal{X}}$ .
  - 3: Construct positive semi-definite similarity matrix  $\mathbf{W}$ ;
  - 4: Initialize  $\mathcal{G}^0, \{\mathbf{U}_n^0\}$  ( $1 \leq n \leq N$ ) randomly and define  $\mathcal{Z}^0$  as null tensor;
  - 5:  $\mathcal{X}_{\Omega} = \mathcal{X}^0|_{\Omega}, \mathcal{X}_{\bar{\Omega}} = \mathcal{Z}^0|_{\bar{\Omega}}$ ;
  - 6: **while**  $k < K$  **do**
  - 7:   Optimize  $\mathcal{G}$  by (19);
  - 8:   **for**  $n = 1$  to  $N$  **do**
  - 9:     Optimize  $\mathbf{U}_n$  using (20);
  - 10:   **end for**
  - 11:   Update Tucker decomposition  $\mathcal{Z}^k = \mathcal{G}^k \times_1 \mathbf{U}_1^k \times_2 \dots \times_N \mathbf{U}_N^k$ ;
  - 12:   Implement dynamic feedback correction mechanism (21) for input tensor  $\mathcal{X}^k$ ;
  - 13:   **if**  $\mathbb{F}(\mathcal{G}^k, \mathbf{U}_{j \leq n}, \mathbf{U}_{j > n}, \mathcal{X}^k)$  is decreasing **then**
  - 14:     Re-update  $\mathcal{G}^k$  and  $\mathbf{U}_n^k$  respectively;
  - 15:   **else**
  - 16:     Re-update  $\mathcal{G}^k$  and  $\mathbf{U}_n^k$  respectively with  $\tilde{\mathcal{G}}^k = \mathcal{G}^{k-1}$  and  $\tilde{\mathbf{U}}_n^k = \mathbf{U}_n^{k-1}$ ;
  - 17:   **end if**
  - 18:   **until** stopping conditions (22) are satisfied.
  - 19: **end while**
  - 20: **return**  $\hat{\mathcal{X}}_{\Omega} = \mathcal{X}^0|_{\Omega}, \hat{\mathcal{X}}_{\bar{\Omega}} = \mathcal{Z}^{k+1}|_{\bar{\Omega}}$ .
- 

### 3.4. Convergence analysis

Since the ARTD model is nonconvex, obtaining the optimal global solution is impossible. However, it is shown in [34] that the optimization problem (5) with cyclic block coordinate descent updating rule has global convergence to a stationary point. We provide the APG-RTD algorithm's convergence property in **Theorem 1**.

**Theorem 1.** Let  $\Theta^k = \{\{\mathbf{U}_n^k\}, \mathcal{G}^k\}$  be the sequence generated by Algorithm 1, then we have the following conclusions.

- $\{\{\mathbf{U}_n^k\}, \mathcal{G}^k\}$  are all Cauchy sequences.
- If hyper-parameters  $\alpha, \beta$  are nonnegative, then  $\Theta^k$  converges to a stationary point  $\hat{\Theta} = \{\{\hat{\mathbf{U}}_n\}, \hat{\mathcal{G}}\}$  of (5).

**Theorem 1** guarantees the feasibility of solutions obtained using Algorithm 1. The proof of **Theorem 1** is based on the results of *Proposition 1* and *Proposition 2*. Firstly, the Cauchy sequence can be proven by showing the cost function of (5) is sufficient descent at each step, i.e.,  $\sum_{k=2}^{\infty} \|\Theta^k - \Theta^{k+1}\|_F < \infty$ . Then, we can prove that  $\hat{\Theta}$  is a stationary point by verifying the first-order optimality conditions. We omit the detail here, and please refer [19] to see the proof.

### 3.5. Computational complexity analysis

In this section, we analyze the computational complexity of the proposed ARTD method. Following[38], we combine the low-rank approximation with population Tucker decomposition strategies to reduce the computational complexity. Throughout this section, we suppose the input tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$  and the core tensor  $\mathcal{G} \in \mathbb{R}^{r_1 \times \dots \times r_N}$ . We have the basic computational complexity: the computational cost of  $\mathbf{U}_n^T \mathbf{U}_n$  is  $\mathcal{O}(r_n^2 I_n)$  and the mode-n product with the matrix  $\mathbf{U}_n$  of tensor  $\mathcal{G}$  is  $\mathcal{O}(\sum_{n=1}^N \prod_{i=1}^n I_i \prod_{i=1}^N r_i)$ . Furthermore, we reformulate the Kronecker product in  $\mathbf{G}_{\mathbf{V}}^n = \mathbf{G}_{(n)} \mathbf{V}_n^T$  and let

$$\mathcal{Y} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \dots \times_{n-1} \mathbf{U}_{n-1} \times_{n+1} \mathbf{U}_{n+1} \times_{n+2} \dots \times_N \mathbf{U}_N, \quad (23)$$

such that we have  $\mathbf{G}_{\mathbf{V}}^n = \mathcal{Y}_{(n)}$  and its computational cost is

$$\begin{aligned} \mathcal{O}(\mathbf{G}_{\mathbf{V}}^n) &= \mathcal{O}\left(\sum_{j=1}^{n-1} \left(\prod_{i=1}^j I_i\right) \left(\prod_{i=j}^N r_i\right)\right) + \mathcal{O}\left(r_n \left(\prod_{i=1}^{n-1} I_i\right) \sum_{j=n+1}^N \left(\prod_{i=j}^j I_i\right) \left(\prod_{i=j}^N r_i\right)\right) \\ &\leq \mathcal{O}\left(\sum_{n=1}^N \left(\prod_{i=1}^n I_i\right) \left(\prod_{j=n}^N r_j\right)\right) \end{aligned} \quad (24)$$

As can be seen from Algorithm 1, the majority of time cost is spent on tensor-matrix multiplications in which the computation of the gradients and Lipschitz  $\mathbf{U}$  constants makes a significant contribution. Also, we conclude that the computational cost of tensor unfolding, soft-thresholding operator, and projection to nonnegative is negligible compared to gradient computing.

Considering the proposed APG-RTD algorithm for core tensor ‘shrinkage’, the computation of  $\nabla_{\mathcal{G}} f(\mathcal{G})$  requires

$$\mathcal{O}\left(\sum_{n=1}^N r_n^2 I_n + \sum_{n=1}^N r_n \prod_{i=1}^N r_i + \sum_{n=1}^N \left(\prod_{i=1}^n r_i\right) \left(\prod_{j=n}^N I_j\right)\right), \quad (25)$$

where the first part comes from the computation of  $\mathbf{U}_n^T \mathbf{U}_n$ , and the second and third parts are from the computations of the first and second terms in (16), respectively.

Similarly, we use (24) to calculate the computational complexity of factor matrices gradient and require

$$\mathcal{O}\left(r_n^2 \left(\prod_{i \neq n} I_i\right) + r_n^2 I_n\right) + \mathcal{O}\left(\prod_{i=1}^n I_i\right) + \mathcal{O}(r_n^3) + \mathcal{O}(\mathbf{G}_{\mathbf{V}}^n), \quad (26)$$

where the first three parts are respectively from the computations of  $\nabla_{\mathbf{U}_n} \ell(\mathbf{U}_n)$ .

The last part dominates the value of the value of (25) and (26). So, the computational cost of  $\nabla_{\mathcal{G}} f(\mathcal{G})$  is

$$\mathcal{O}\left(\sum_{n=1}^N \left(\prod_{i=1}^n r_i\right) \left(\prod_{j=n}^N I_j\right)\right), \quad (27)$$

and  $\nabla_{\mathbf{U}_n} \ell(\mathbf{U}_n)$  is

$$\mathcal{O}\left(\sum_{n=1}^N \left(\prod_{i=1}^n I_i\right) \left(\prod_{j=n}^N r_j\right)\right). \quad (28)$$

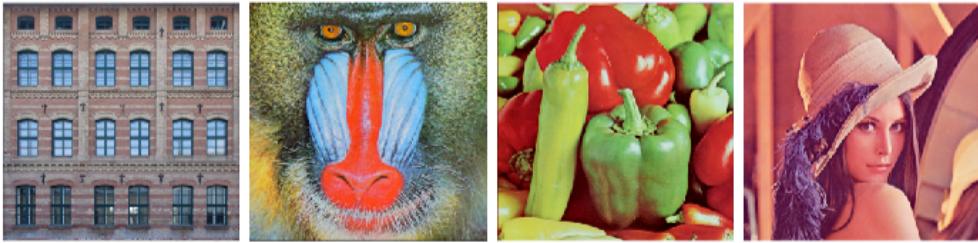
Then, the time complexity of the APG-RTD algorithm in each iteration is approximately estimated as

$$\mathcal{O}\left(\sum_{n=1}^N \left(\prod_{i=1}^n r_i\right) \left(\prod_{j=n}^N I_j\right) + N \cdot \sum_{n=1}^N \left(\prod_{i=1}^n I_i\right) \left(\prod_{j=n}^N r_j\right)\right), \quad (29)$$

where the per-iteration cost is relevant to the tensor sizes  $\prod_{i=1}^n I_i$ , and Algorithm 1 is theoretically efficient [19].

## 4. Experimental results

To validate the effectiveness of the proposed method for the IR problem, we perform both random missing (RM) and structural missing (SM) experiments on popular RGB-color images (shown in Figure 1) that are widely used in IR. For comparison, we select the baseline methods, including HaLRTC [1] and SMF [14], representing the matricization-based method; t-SVD [4] represents the state-of-the-art tensor tubal rank minimization based method; SPC [5] represents state-of-the-art CP decomposition based method; STDC [12], KBR [17], and LR-SETD [18], representing state-of-the-arts Tucker decomposition based method. All experiments are performed using MATLAB on Windows 10 64-bit operating system on a workstation equipped with an Intel(R) Xeon(R) W-2123 CPU with 3.60 GHz, 64 GB RAM.



**Figure 1:** Four color images for experiments. From left to right: House, Baboon, Peppers, Lena

**Performance evaluation.** The peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [39], relative squared error (RSE), and execution time (in seconds) are used to evaluate the performance of various methods quantitatively. The evaluation metrics are defined as

$$\text{PSNR} = 10 \cdot \log_{10} \frac{(\mathcal{X}_{\max})^2}{\|\hat{\mathcal{X}} - \mathcal{X}_{\text{true}}\|_F^2 / |\bar{\Omega}|}, \quad (30)$$

$$\text{RSE} = \frac{\|\hat{\mathcal{X}} - \mathcal{X}_{\text{true}}\|_F}{\|\mathcal{X}_{\text{true}}\|_F}, \quad (31)$$

where  $\hat{\mathcal{X}}$ ,  $\mathcal{X}_{\text{true}}$  and  $\mathcal{X}_{\max}$  represent the reconstructed tensor, ground-truth tensor, and the maximum value in the ground-truth tensor, respectively. Additionally,  $|\bar{\Omega}|$  represents the number of elements of the complement set  $\bar{\Omega}$  of  $\Omega$ . Smaller RSE and execution time, larger PSNR and SSIM indicate better reconstruction performance.

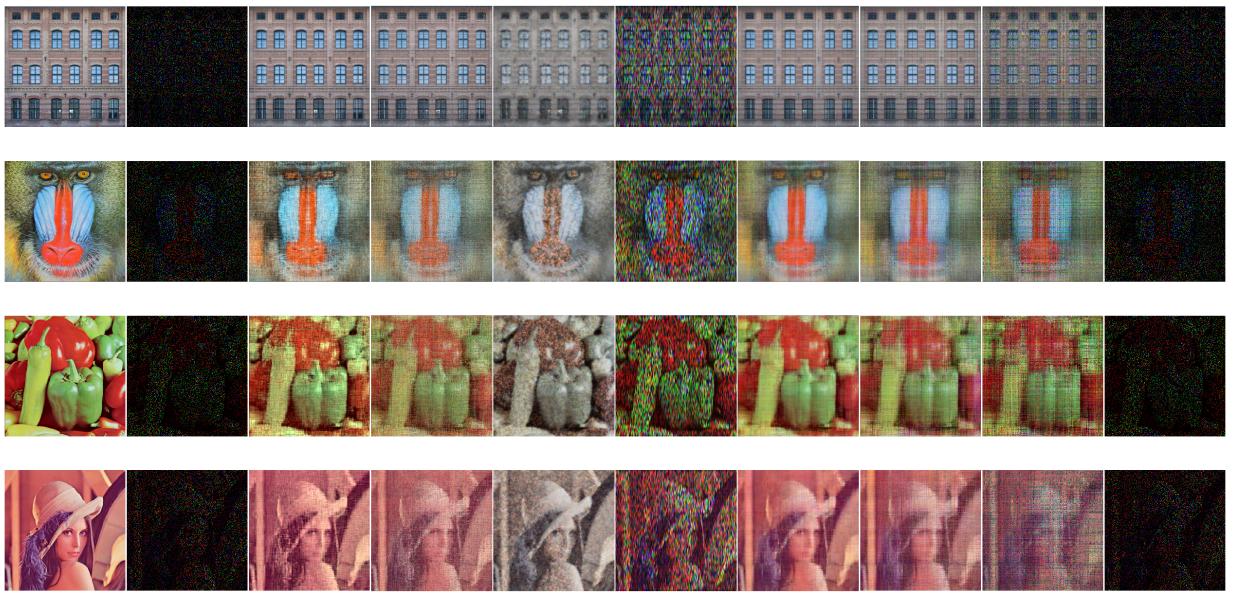
**Parameter Settings.** Throughout this section, we adopt the relative change of the two successive reconstructed image tensor (22) as the stopping criterion for all methods and set the maximum number of iterations as  $K = 300$  for all algorithms. There are three parameters  $\alpha$ ,  $\beta$  and  $\gamma$  in ARTD model. In all our experiments, we determine the parameter  $\alpha$  using different mode- $n$  unfolding matrices given the image tensor,

$$\alpha_n = \frac{\|\mathbf{X}_{(n)}\|_F^2}{2 * \|\mathbf{L}_n\|_F^2}, \quad n = 1, 2, 3. \quad (32)$$

which plays a key role in model performance. Compared with the hand-tuning strategy, our proposal performs more robustly. Next, tuning the low-rankness parameter  $\beta$  can enhance the model performance, and we easily set  $\beta = 1$ . The latter  $\gamma$  is used to correct the low-rank approximation of a clean image, and we just set  $\gamma = 0.2$ .

#### 4.1. Random missing

We randomly sample corrupted images with sampling rates (SRs)  $\{0.01, 0.05, 0.07, 0.1, 0.2, 0.3, 0.4\}$ . Figure 2 shows the visual quality of color image restoration for random sampling entries using seven utilized LRTA methods with SR = 0.1. It can be seen that HaLRTC, STDC, and SMF perform poorly in high-level corruption, which suggests that the global low-rankness is not enough for the IR problem. SPC, KBR, and tSVD can overcome the low precision problem; however, they lose details due to over-smoothness. LRSETD introduced the core tensor sparsity into global low-rankness in Tucker-based models. Still, the smoothness constraint on the clean image cannot capture the local similarity, which makes the restoration results still contain incomplete pixels. Table 2 summarizes the PSNR, SSIM, and execution time of the House image with SRs = 0.01, 0.05, 0.07, 0.1. The best results show that our proposal ARTD has tolerable time consumption and improves performance significantly, so ARTD outperforms the other state-of-the-art methods under high-level corruption.



**Figure 2:** IR results for random missing entries with SR = 0.1. From left to right: the clean image, the corrupted image, and the IR results by our proposal ARTD, KBR, LRSETD, STDC, SPC, t-SVD, SMF, and HaLRTC, respectively. From top to bottom: House, Baboon, Peppers, and Lena.

To further compare the model performance of our proposal ARTD, we plot the PSNR, SSIM, RSE, and execution time for each method in Figure 3. ARTD performs better in all SRs for the House image and obtains higher PSNR and SSIM values when SR>20% for other images. Although the SPC obtains the best IR performance in a tiny sampling (SR<10%, see Figure 3 (b), (c), and (d)), in contrast, our proposed method is more efficient and can reconstruct images with more details.

#### 4.2. Structural missing

Below, we consider the structural missing, Fig. 4 shows the IR performance for missing slices and image masking scenarios. It can be seen that the global low-rankness method, including t-SVD, STDC, SMF, and HaLRTC, almost fails for House and Peppers images, and the results obtained by LRSETD and SPC lost image textures for Baboon and Lena images. Compared with KBR, the proposed method yields a much better visual effect in keeping details and structures for all images, which supports that ARTD works more stable on different types of missing scenarios. To see the details of these methods on IR, we report the PSNR and RSE values in Fig. 4, where the results also show that ARTD performs better in achieving higher PSNR values.

**Table 2**

Numerical performance of the seven methods with random missing entries on the House image. The **best** results are highlighted in bold.

SR Methods	0.01		0.05		0.07		0.1		Time
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
<b>HaLRTC</b>	5.79	0.0020	5.97	0.0082	6.07	0.0110	6.72	0.0337	<b>0.6</b>
<b>SMF</b>	6.89	0.0195	11.22	0.1098	13.24	0.1826	16.14	0.3230	40
<b>STDC</b>	5.85	0.0036	7.15	0.0362	8.39	0.0702	10.92	0.1578	1.6
<b>SPC</b>	16.46	0.2870	22.85	0.6559	24.16	0.7225	<b>27.92</b>	0.8605	62
<b>tSVD</b>	16.69	0.3097	21.71	0.6261	23.14	0.6928	24.73	0.7583	43
<b>KBR</b>	17.71	0.3320	20.79	0.5806	22.27	0.6448	24.97	0.7620	31
<b>LRSETD</b>	<b>18.20</b>	0.2682	20.01	0.4423	20.65	0.5005	21.41	0.5670	<b>14</b>
<b>ARTD</b>	17.51	<b>0.3445</b>	<b>24.22</b>	<b>0.7288</b>	<b>25.49</b>	<b>0.7776</b>	26.68	<b>0.8192</b>	36

**Table 3**

PSNR, SSIM, and Time (in seconds) of different Tucker-based methods for Lena IR in the RM case. The **best** values are highlighted in bold.

SR Methods	0.01			0.05			0.1			0.2		
	PSNR	SSIM	Time									
<b>NTD</b>	13.44	0.0865	85.54	16.20	0.2361	85.06	17.09	0.3170	85.81	19.40	0.4151	84.36
<b>SNTD</b>	13.68	0.1014	85.42	16.37	0.2404	85.39	17.73	0.3170	85.82	20.07	0.4390	85.64
<b>GNTD</b>	13.96	0.0977	86.35	16.55	0.2554	86.35	17.89	0.3436	89.89	20.04	0.4623	89.02
<b>ARTD*</b>	13.27	0.0846	<b>32.48</b>	16.87	0.1982	<b>32.57</b>	18.91	0.2936	<b>33.13</b>	23.61	0.5441	<b>33.28</b>
<b>ARTD</b>	<b>14.75</b>	<b>0.1150</b>	35.05	<b>18.86</b>	<b>0.2983</b>	35.11	<b>20.83</b>	<b>0.4186</b>	35.96	<b>24.97</b>	<b>0.6334</b>	36.30

### 4.3. Discussions

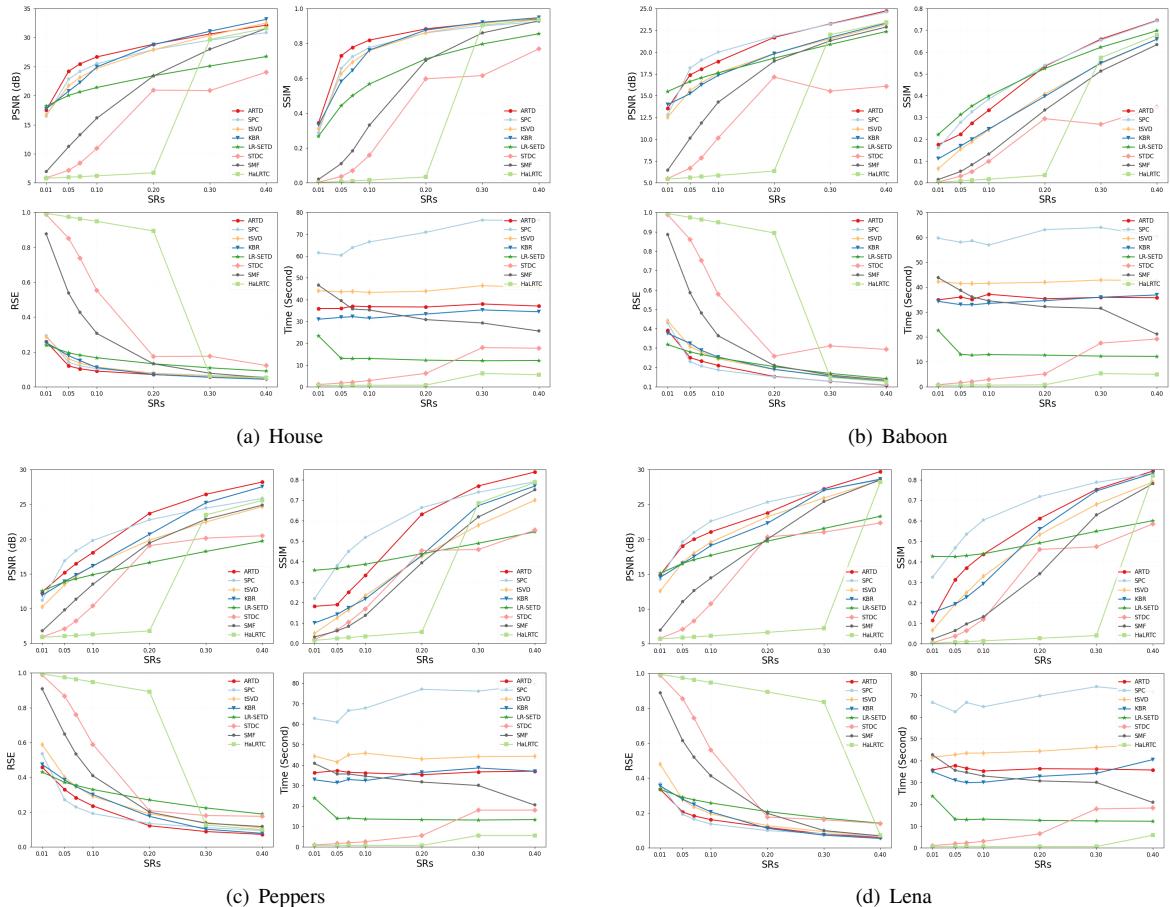
*Contributions of priors:* We discuss the contributions of the new prior structure to the IR performance for Lena under SR = 0.2 in Figure 5. On the one hand, the sparsity core tensor with nonnegative factor matrix constraint encodes the global knowledge under the corrupted image, the same as the SNTD and NTD methods with a given Tucker rank. On the other hand, the local textures of the corrupted image can be captured using factor matrices nonnegative and manifold learning strategy (see ARTD\* and GNTD results). Table 2 summarizes the PSNR, SSIM, and running time of different Tucker-based prior methods for IR. It can be shown that the IR performance has been improved, and the prior knowledge has different results and complements each other, so the proposed method can simultaneously preserve the GCS and LSS knowledge of the underlying image.

*Convergence behaviors:* We have theoretically proven that the sequences generated by **Algorithm 1** converge to a stationary point in **Theorem 1**. Here, we show the numerical convergence of the proposed algorithm. Figure 6 shows the curves of the RSE values versus the iteration number of the proposed ARTD on Lena. Remark that the RSE keeps decreasing as the iteration number increases, and the values stabilize after only about 250 iterations, which implies the proposed algorithm's numerical stability and convergence. Furthermore, we test the Lena under SRs 20% to show that the proposed initialization strategy and the first-order control rule can speed up the convergence of the proposed algorithm.

## 5. Conclusion

This paper proposes a new prior structure for the IR problem. A nonnegative and manifold regularized factor matrix combined with a sparsity Tucker core tensor is used to capture the intrinsic image characteristics, i.e., GCS and LSS, respectively, in a low-rank approximation manner. We have also designed and implemented an effective APG-based algorithm for solving the ARTD model and established the convergence guarantee of the proposed algorithm in

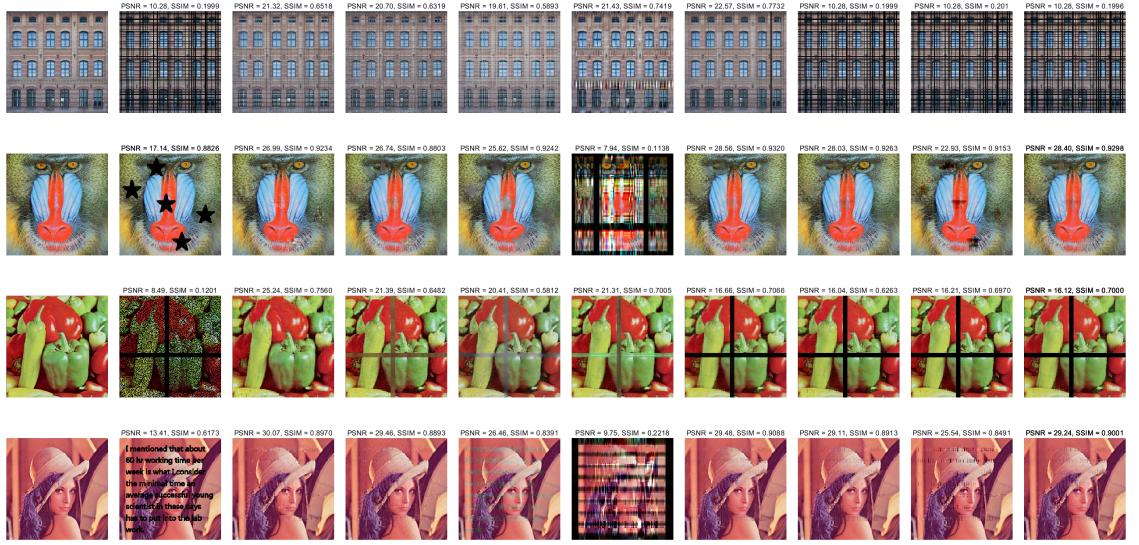
## ARTD for IR



**Figure 3:** Comparison results of evaluation metrics with respect to SRs. (a) ARTD performs better in all SRs. (b)-(d) ARTD is less time-consuming and performs robustly in different image data.

**Theorem 1.** A series of numerical experiments with low percentages of SRs demonstrate that the proposed method outperforms the state-of-the-art methods. ARTD also performs more robustly, as shown in Figure 2 and Figure 4. With the priors of image data considered in this paper, ARTD may be extended to address the spatiotemporal traffic data imputation in future work.

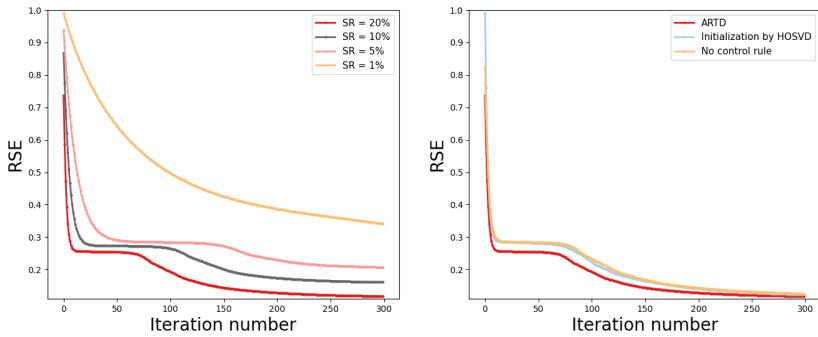
## ARTD for IR



**Figure 4:** IR results for structural missing. From left to right: the clean image, the corrupted image, and the IR results by our proposal ARTD, KBR, LRSETD, STDC, SPC, t-SVD, SMF, and HaLRTC, respectively. From top to bottom: House, Baboon, Peppers, and Lena.



**Figure 5:** IR results given by different Tucker-based prior methods. The SRs is 20%, and the results show that different priors complement each other. From the second column to the bottom: ARTD, ARTD with no manifold regularization (denoted as ARTD\*), GNTD, SNTD, and NTD.



**Figure 6:** RSE comparison under different SRs versus iterations of image Lena.

## References

- [1] J. Liu, P. Musalski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (1) (2013) 208–220.
- [2] X.-L. Zhao, J.-H. Yang, T.-H. Ma, T.-X. Jiang, M. K. Ng, T.-Z. Huang, Tensor completion via complementary global, local, and nonlocal priors, *IEEE Transactions on Image Processing* 31 (2022) 984–999.

- [3] S. Gandy, B. Recht, I. Yamada, Tensor completion and low-n-rank tensor recovery via convex optimization, *Inverse Problems* 27 (2) (2011) 025010.
- [4] Z. Zhang, S. Aeron, Exact tensor completion using t-svd, *IEEE Transactions on Signal Processing* 65 (6) (2017) 1511–1526.
- [5] T. Yokota, Q. Zhao, A. Cichocki, Smooth parafac decomposition for tensor completion, *IEEE Transactions on Signal Processing* 64 (20) (2016) 5423–5436.
- [6] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, C. Faloutsos, Tensor decomposition for signal processing and machine learning, *IEEE Transactions on Signal Processing* 65 (13) (2017) 3551–3582.
- [7] L. Zhang, W. Zuo, Image restoration: From sparse and low-rank priors to deep priors [lecture notes], *IEEE Signal Processing Magazine* 34 (5) (2017) 172–179.
- [8] R. Lin, C. Chen, N. Wong, Coarse to fine: Image restoration boosted by multi-scale low-rank tensor completion (2022).
- [9] J.-H. Yang, X.-L. Zhao, T.-H. Ma, M. Ding, T.-Z. Huang, Tensor train rank minimization with hybrid smoothness regularization for visual data recovery, *Applied Mathematical Modelling* 81 (2020) 711–726.
- [10] C.-Y. Ko, K. Batselier, L. Daniel, W. Yu, N. Wong, Fast and accurate tensor completion with Total Variation regularized tensor trains, *IEEE Transactions on Image Processing* 29 (2020) 6918–6931.
- [11] Q. Xie, Q. Zhao, D. Meng, Z. Xu, S. Gu, W. Zuo, L. Zhang, Multispectral images denoising by intrinsic tensor sparsity regularization, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1692–1700.
- [12] Y.-L. Chen, C.-T. Hsu, H.-Y. M. Liao, Simultaneous tensor decomposition and completion using factor priors, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (3) (2014) 577–591.
- [13] X. Li, Y. Ye, X. Xu, Low-rank tensor completion with Total Variation for visual data inpainting, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2017, pp. 2210–2216.
- [14] Y.-B. Zheng, T.-Z. Huang, T.-Y. Ji, X.-L. Zhao, T.-X. Jiang, T.-H. Ma, Low-rank tensor completion via smooth matrix factorization, *Applied Mathematical Modelling* 70 (2019) 677–695.
- [15] Y. Wang, J. Peng, Q. Zhao, Y. Leung, X.-L. Zhao, D. Meng, Hyperspectral image restoration via total variation regularized low-rank tensor decomposition, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11 (4) (2018) 1227–1243.
- [16] Q. Zhao, L. Zhang, A. Cichocki, Bayesian CP factorization of incomplete tensors with automatic rank determination, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37 (9) (2015) 17511763.
- [17] Q. Xie, Q. Zhao, D. Meng, Z. Xu, Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40 (8) (2018) 1888–1902.
- [18] C. Pan, C. Ling, H. He, L. Qi, Y. Xu, Low-rank and sparse enhanced Tucker decomposition for tensor completion (2020).
- [19] Y. Xu, Alternating proximal gradient method for sparse nonnegative Tucker decomposition, *Mathematical Programming Computation* 5 (3) (2015) 455500.
- [20] Y. Qiu, G. Zhou, Y. Wang, Y. Zhang, S. Xie, A generalized graph regularized non-negative Tucker decomposition framework for tensor data representation, *IEEE Transactions on Cybernetics* 52 (1) (2022) 594–607.
- [21] J. A. Bengua, H. N. Phien, H. D. Tuan, M. N. Do, Efficient tensor completion for color image and video recovery: Low-rank tensor train, *IEEE Transactions on Image Processing* 26 (5) (2017) 2466–2479.
- [22] X. Li, M. K. Ng, G. Cong, Y. Ye, Q. Wu, MR-NTD: Manifold regularization nonnegative Tucker decomposition for tensor data dimension reduction and representation, *IEEE Transactions on Neural Networks and Learning Systems* 28 (8) (2017) 1787–1800.
- [23] A. Narita, K. Hayashi, R. Tomioka, H. Kashima, Tensor factorization using auxiliary information, *Data Mining and Knowledge Discovery* 25 (2012) 298324.
- [24] U. Dmitry, V. Andrea, S. L. Victor, Deep image prior, in: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, p. 94469454.
- [25] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, *Commun. ACM* 60 (6) (2017) 8490.
- [26] Y.-D. Kim, S. Choi, Nonnegative Tucker decomposition, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
- [27] T. K. Sinha, J. Naram, P. Kumar, Nonnegative low-rank tensor completion via dual formulation with applications to image and video completion, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2022, pp. 3732–3740.
- [28] Y.-X. Wang, Y.-J. Zhang, Nonnegative matrix factorization: A comprehensive review, *IEEE Transactions on Knowledge and Data Engineering* 25 (6) (2013) 1336–1353.
- [29] W. Yin, Z. Ma, LE & LLE regularized nonnegative Tucker decomposition for clustering of high dimensional datasets, *Neurocomputing* 364 77–94.
- [30] D. Cai, X. He, J. Han, T. S. Huang, Graph regularized nonnegative matrix factorization for data representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33 (8) (2011) 1548–1560.
- [31] J. J.-Y. Wang, H. Bensmail, X. Gao, Multiple graph regularized nonnegative matrix factorization, *Pattern Recognition* 46 (10) (2013) 2840–2847.
- [32] X. Zhang, M. K. Ng, Sparse nonnegative tensor factorization and completion with noisy observations, *IEEE Transactions on Information Theory* 68 (4) (2022) 2551–2572.
- [33] N. Gillis, Introduction to nonnegative matrix factorization (2017).
- [34] Y. Xu, W. Yin, A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion, *SIAM Journal on Imaging Sciences* 6 (3) (2013) 1758–1789.
- [35] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *Now Foundations and Trends*, 2011.
- [36] L. D. Lathauwer, B. D. Moor, J. Vandewalle, On the best rank-1 and rank-n approximation of higher-order tensors, *SIAM Journal on Matrix Analysis and Applications* 21 (4) (2000) 13241342.

- [37] H. Rauhut, R. Schneider, eljka Stojanac, Low rank tensor recovery via iterative hard thresholding, *Linear Algebra and its Applications* 523 (2017) 220–262.
- [38] G. Zhou, A. Cichocki, Q. Zhao, S. Xie, Efficient nonnegative Tucker decompositions: Algorithms and uniqueness, *IEEE Transactions on Image Processing* 24 (12) (2015) 4990–5003.
- [39] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Transactions on Image Processing* 13 (4) (2004) 600–612.