Robust Tensor Factorization for Color Image and Grayscale Video Recovery

SHIQIANG DU1, (Member, IEEE), YUQING SHI2, WENJIN HU1, (Member, IEEE), WEILAN WANG1, AND JING LIAN3, (Member, IEEE)

1Key Laboratory of China’s Ethnic Languages and Information Technology of Ministry of Education, College of Mathematics and Computer Science, Northwest Minzu University, Lanzhou 730030, China
2College of Electrical Engineering, Northwest Minzu University, Lanzhou 730030, China
3School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

ABSTRACT Low-rank tensor completion (LRTC) plays an important role in many fields, such as machine learning, computer vision, image processing, and mathematical theory. Since rank minimization is an NP-hard problem, one strategy is that it is converted into a convex relaxation tensor nuclear norm (TNN) that requires the repeated calculation of time-consuming SVD, and the other is to convert it into some product of two smaller tensors that are easy to fall into the local minimum. In order to overcome the above shortcomings, we propose a robust tensor factorization (RTF) model for solving LRTC. In RTF, the noisy tensor data with missing entries is decomposed into low-rank tensor and noisy tensor, and then the low-rank tensor is equivalently decomposed into t-products (essentially vectors convolution) of two smaller tensors: orthogonal dictionary tensor and low-rank representation tensor. Meanwhile, the TNN of low-rank representation tensor is adopted to characterize the low-rank structure of the tensor data for preserving global information. Then, an effective iterative update algorithm based on the alternating direction method of multipliers (ADMM) is proposed to solve RTF. Finally, numerical experiments on image recovering and video completion tasks show the effectiveness of the proposed RTF model compared with several state-of-the-art tensor completion models.

INDEX TERMS Tensor completion, tensor factorization, low-rank tensor, tensor nuclear norm.

I. INTRODUCTION Real-world multidimensional data such as images, videos, and social networks usually exhibit low-rank structures due to local similarities, spatial correlations, etc. In addition, some values in the data may be lost in the process of acquiring and storing due to technical failure or human-induced factors. Low-rank tensor completion (LRTC) has become a research hotspot in machine learning and computer vision, because it can recover an ideal tensor by using only a part of the known entries under low-rank constraints. LRTC has achieved state-of-the-art performance in recommendation systems [1], computer vision [2], [3], machine learning [4], [5], multi-energy computed tomography [6], image processing [7], [8], etc.

Many LRTC methods have been proposed and applied to image restoration and video completion [7], [9]–[15]. Among them, the tensor singular value decomposition (t-SVD)-based method has become the most important branch because it can effectively describe tensor spatial information [16]–[19]. LRTC methods based on t-SVD can be divided into two categories. One is to use the tensor nuclear norm (TNN) to maintain the low-rank structures of the tensor while completing the missing entries, the other is to utilize the t-product of two smaller tensor factors to describe the low-rank structures while recovering the missing elements.
Kilmer and Martin [16], Kilmer et al. [20] first propose t-SVD and its related mathematical theoretical framework and apply it to image deblurring. By defining the tensor tubal rank, Zhang et al. [18] introduce a TNN minimization model for tensor completion achieving the state-of-the-art video inpainting results. The exact tensor completion conditions are then analyzed in [21]. On the basis of further perfecting the TNN theory, Lu et al. [14], [22] extend the robust principal component analysis (RPCA) to the tensor RPCA (TRPCA) and apply it to image inpainting. These methods complement the unknown entries in the tensor data while utilizing TNN to preserve the low-rank structures. Since t-SVD is calculated during each iteration, they have a shortcoming that is very time-consuming, especially when the tensor size is large.

When LRTC based on tensor factors fills in missing entries, a low-rank structure of tensor data is characterized by some product of multiple factors with smaller sizes. For methods related to t-SVD decomposition, tensor factorization-based LRTC has two representative methods. One is that Tubal-Alt-Min [23], in which entry missing tensor is parameterized as the t-product of two small tensors, and these two small tensor factors are updated alternately using tensor least squares minimization. The other is tensor completion by tensor factorization (TCTF) [12], which completes tensor completion by decomposing each frontal slice of the unknown tensor into two small matrix factors. These two methods, especially TCTF, not only have a fast calculation speed but also have state-of-the-art completion performance. However, these two methods cannot maintain the global structure information and are prone to fall into local minima. Especially when the proportion of missing entries is large, the complete performance is very poor.

In this paper, we propose a robust tensor factorization (RTF) model that can effectively overcome the above shortcomings for LRTC. In the RTF model, we first decompose the noisy tensor data with missing entries into low-rank tensor and sparse noise tensor, and low-rank tensor is equivalently decomposed into the t-product of two smaller tensors called dictionary tensors and low-rank representation tensors respectively. This equivalent decomposition not only preserves the global low-rank structure in the tensor data but also inherits the property of fast computation of tensor factorization method. In addition, as a special case, if tensor data $\mathbf{X}$ reduces to a matrix, all the tensor factors reduce to the matrix cases. Our model reduces to the model of paper [24]. Therefore, our model is the tensor-based generalization of matrix-based factorization in [24]. The main contributions of this paper are summarized as follows:

1) On the basis of t-product, the low-tubal-rank tensor is equivalently decomposed into the t-product of dictionary tensor and low-rank representation tensor.
2) The TNN constraint on low-rank representation tensor can effectively describe the global structures of tensor data. Meanwhile, tensors with small sizes can improve the computational speed of the algorithm.
3) An efficient iterative update algorithm based on the alternating direction method of multipliers (ADMM) is used for our RTF optimization, and the convergence of the algorithm can also be guaranteed.
4) Compared with other state-of-the-art tensor completion methods, numerical experiments on image recovering and video inpainting tasks demonstrate the effectiveness of our method.

The rest of the paper is structured as follows. We give some notations and related work in Section II. Discrete fourier transformation (DFT) and preliminaries are shown in Section III. Section IV provides the RTF method, the optimization details, convergence and complexity analysis. Extensive numerical experiments on real data are described and analyzed in Section V. Finally, we conclude this work in Section VI.

II. RELATED WORK

Some notations used in this paper are summarized in Table 1 and then review previous work. There are various techniques for tackling the LRTC problem. According to the different decomposition methods of tensor and the definition of tensor rank, the current popular LRTC method can be roughly divided into three categories.

A. CP DECOMPOSITION

CANDECOMP/PARAFAC (CP) decomposition methods [9]–[11], [25], [26] based on CP rank [27]. For a $k$-way tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times \ldots \times n_k}$, the CP decomposition is written as follows

$$\mathbf{X} = \sum_{i=1}^{r} u_i^{(1)} \circ u_i^{(2)} \circ \ldots \circ u_i^{(k)},$$

where $r$ is a positive integer, $u_i^{(j)} \in \mathbb{R}^{n_j}$, for $i = 1, \ldots, r$. In the decomposition of formula (1), the smallest number of $r$ is called the CP rank of tensor $\mathbf{X}$, denoted as $\text{rank}_{cp}(\mathbf{X}) = r$.

LRTC based on CP decomposition is to complete missing entries under the constraint of low CP rank. Due to CP rank is generally NP-hard to compute [28], [29], the LRTC problems of rank minimization are often converted into factor problems to predict missing entries. By expanding the given tensor into matrices along all modes, Xu et al. [10] propose the TMac model to recover the missing entries. Subsequently, Bengua et al. [30] suggest the TMacTT model complete missing elements. The difference from TMac is that the tensor rank is first estimated by using tensor train technology. In order to take into account uncertainty information of latent factors, Zhao et al. [9] present fully Bayesian CP factorization (FBCP) and its extension using mixture prior (FBCPM). However, the CP factorization scheme is prone to overfitting which resulting in severe deterioration of predictive performance. Besides, the CP rank is generally NP-hard to compute.
TABLE 1. Summary of notation.

| Notation | Description | Notation | Description |
|----------|-------------|----------|-------------|
| Scalar   | $x$         | $|x|$     | absolute value |
|          | $x \in \mathbb{C}$ | $x_i$   | $i$-th entry |
|          | $\|x\|_2 = (\sum_i |x_i|^2)^{1/2}$  | $\|x\|_2$  | $\ell_2$-norm |
| Vector   | $\|x\|_1 = \sum_i |x_i|$  | $\ell_1$-norm | |
|          | $x \in \mathbb{C}^n$ | $X_{ij}$ or $x_{ij}$  | $(i,j)$-th entry |
|          | $\ell_1$-norm | $X^*$    | conjugate transpose |
|          | $|X|_\infty = \max_{ij} |x_{ij}|$  | infinity norm | |
|          | $|X|_F = (\sum_{ij} |x_{ij}|^2)^{1/2}$  | Frobenius norm | |
|          | $\sigma_i(X)$ | $\ell_i$-singular value | |
|          | $G \times \ell \times \ldots \times \ell \times \ell$ | $Y_{ijk}$ or $x_{ijk}$  | $(i,j,k)$-th entry |
|          | $X \times \ell \times \ldots \times \ell \times \ell$ | $Y_{(i,j,k)}$ or $x_{(i,j,k)}$  | $j$-th lateral slice |
| Matrix   | $X \in \mathbb{C}^{m \times n}$  | $G_{\ell} \times \ell \times \ldots \times \ell \times \ell$ | mode-3 tube |
|          | $\ell_1$-norm | $X^*$    | conjugate transpose |
|          | $\max_{ij} |x_{ij}|$  | Frobenius norm | |
|          | $|X|_F = (\sum_{ij} |x_{ij}|^2)^{1/2}$  | Frobenius norm | |
| Tensor   | $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$  | $\mathcal{X} \times \ell \times \ldots \times \ell \times \ell$ | tubal rank |
|          | $\mathcal{X}(i,\ldots,\ell)$ | $\mathcal{X} \times \ell \times \ldots \times \ell \times \ell$ | nuclear norm |
|          | $\mathcal{X}(\ell,\ldots,\ell)$ | $\mathcal{X} \times \ell \times \ldots \times \ell \times \ell$ | nuclear norm |
|          | $\mathcal{X}(\ell,\ldots,\ell)$ | $\mathcal{X} \times \ell \times \ldots \times \ell \times \ell$ | nuclear norm |
|          | $\mathcal{X}(\ell,\ldots,\ell)$ | $\mathcal{X} \times \ell \times \ldots \times \ell \times \ell$ | nuclear norm |
|          | $\mathcal{X}$ | $\mathcal{X}(\ell,\ldots,\ell)$ | nuclear norm |
|          | $\mathcal{X}(\ell,\ldots,\ell)$ | $\mathcal{X} \times \ell \times \ldots \times \ell \times \ell$ | nuclear norm |
|          | $\mathcal{X}(\ell,\ldots,\ell)$ | $\mathcal{X} \times \ell \times \ldots \times \ell \times \ell$ | nuclear norm |

B. TUCKER DECOMPOSITION

Tucker decompositoin methods [7] based on Tucker rank [31]. The Tucker decomposition of tensor $\mathcal{X}$ as following

$$\mathcal{X} = \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \times \ldots \times_k U^{(k)},$$

where $U^{(i)} \in \mathbb{R}^{n_i \times m_i}$ and $\mathcal{G} \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_k}$ is the core tensor, the symbol $\times_i$ is the $i$-way (matrix) product. For example, if $i = j$, for above tensor $\mathcal{G}$ and a matrix $U^{(i)} \in \mathbb{R}^{n \times m}$

$$\mathcal{Y} = \mathcal{G} \times_j U^{(i)} \Leftrightarrow Y_{ijkl} = U^{(i)} G_{ijl},$$

where $\mathcal{Y} \in \mathbb{R}^{m_1 \times m_2 \times \ldots \times m_j \times n \times m} \times m_k \times n \times m}$. The Tucker rank of the tensor $\mathcal{X}$ is a $k$-dimensional vector determined by the rank of tensors mode-$i$ matricization $X_{(i)}$, denoted as

$$\text{rank}_k = (\text{rank}(X_{(1)}), \ldots, \text{rank}(X_{(k)})).$$

Using multilinear operations, a $k$-way tensor can be represented as $k$ factor matrices and a core tensor by Tucker decomposition. Due to the Tucker rank defined on the matrix rank, most LRTC methods based on Tucker rank are closely related to the low-rank matrix completion method. Liu et al. [8] define the TNN as the weighted sum of traces norms of flattening matrices and propose three LRTC models to complete visual data. By performing Riemannian optimization techniques on the manifold of tensors, Kressner et al. [7] propose a geometric nonlinear conjugate gradient (GeomCG) for fitting the factors of Tucker decomposition. In order to obtain a compact representation with the smallest core tensor, Yang et al. [32] utilize a group-based log-sum penalty function to discover the structural sparsity of core tensor. In order to reduce the computational cost, Tan et al. [33] suggest a multi-linear low-n-rank factorization model that only requires solving a linear least squares problem per iteration. However, a conceptual drawback of Tucker rank is that its entries are ranks of flattening matrices of the tensor. Due to the upper bound of each rank is usually independent and small, Tucker rank may not be suitable for capturing global (spatial and temporal) information of the tensor.

C. T-SVD DECOMPOSITION

T-SVD decomposition methods [12], [14] based on tubal rank [20]. For the definition of t-SVD, refer to Section III. Compared with the first two categories of LRTC methods, t-SVD-based methods have been shown to be superior in capturing the spatial correlation in real-world data [16]–[20], [34]. Similar to the matrix SVD, t-SVD decomposes a tensor into three tensors on the basis of tensor-tensor product (t-product). Based on the t-product [16] and associated algebraic constructs, any tensor has the t-SVD [20] motivates a new tensor rank referred to as the tubal rank [16], [17], provide a new framework to measure the low-rank structure of tensor. Semerci et al. [35] introduce two TNNs as regularization for the energy selective multi-energy computed tomography problem. Zhang et al. [18] propose tensor completion based on TNN achieves state-of-the-art video inpainting, and the exact tensor completion conditions are demonstrated in [21]. Apart from these, Lu et al. [22] present a solution to tensor decomposition using t-SVD and demonstrated the multi-linear data recovery from sparse noise. Subsequently, a more complete theoretical framework and experiments are presented in [14]. However, these methods all require iterative calculations to obtain the final result, and each iteration has to perform more than one time-consuming SVD.

III. PRELIMINARIES

In this part, we first introduce the discrete fourier transform that is closely related to the t-product, and then present t-SVD and tensor nuclear norm.
A. DISCRETE FOURIER TRANSFORMATION

The Discrete Fourier Transformation (DFT) plays a core role in t-product between tensors. For a vector $v \in \mathbb{R}^n$, its DFT is denoted as $\hat{v}$ and is given by

$$\hat{v} = F_n v \in \mathbb{C}^n,$$

where $F_n$ is the DFT matrix defined as

$$F_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \cdots & \omega^{(n-1)(n-1)} \end{bmatrix} = \mathbb{C}^{n \times n},$$

where $\omega = \exp(-2\pi i/n)$ and $i = \sqrt{-1}$. Then

$$F_n^* F_n = F_n F_n^* = nI_n.$$

Thus $F_n^{-1} = F_n^* / n$. Moreover, $F_n / \sqrt{n}$ is an orthogonal matrix. Indeed, we can compute $\hat{v}$ directly by the Matlab command $\text{fft}$, i.e., $\hat{v} = \text{fft}(v)$, and use the inverse DFT to obtain $v = \text{ifft}(\hat{v})$. Denote the circulant matrix of $v$ as

$$\text{circ}(v) = \begin{bmatrix} v_1 & v_n & \cdots & v_2 \\ v_2 & v_1 & \cdots & v_3 \\ \vdots & \vdots & \ddots & \vdots \\ v_n & v_{n-1} & \cdots & v_1 \end{bmatrix} \in \mathbb{R}^{n \times n}.$$

It is known that it can be diagonalized by the DFT matrix, i.e.,

$$F_n \cdot \text{circ}(v) \cdot F_n^{-1} = \text{Diag}(\hat{v}),$$

where $\text{Diag}(\hat{v})$ denotes a diagonal matrix with its $i$-th diagonal entry as $\hat{v}_i$.

B. T-SVD AND TENSOR NUCLEAR NORM

For a tensor $\mathbf{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, let $\tilde{\mathbf{X}} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ represent the result of DFT of $\mathbf{X}$ along the 3-rd dimension, i.e., each tube $\tilde{\mathbf{X}}(i,j,:) \times n_3$ is obtained by DFT of the corresponding tube $\mathbf{X}(i,j,:) \times n_3$ is used to denote a block diagonal matrix with its $i$-th block on the diagonal as the $i$-th frontal slice $\tilde{\mathbf{X}}(i) \times n_3$ of $\mathbf{X}$, i.e.,

$$\tilde{\mathbf{X}} = \text{bdiag}(\tilde{\mathbf{X}}) = \begin{bmatrix} \tilde{\mathbf{X}}^{(1)} \\ \tilde{\mathbf{X}}^{(2)} \\ \vdots \\ \tilde{\mathbf{X}}^{(n_3)} \end{bmatrix},$$

where $\text{bdiag}(\cdot)$ is an operator which maps the tensor $\tilde{\mathbf{X}}$ to the block diagonal matrix $\tilde{\mathbf{X}}$. We further define the block circulant matrix $b\text{circ}(\mathbf{X}) \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ of $\mathbf{X}$ as

$$b\text{circ}(\mathbf{X}) = \begin{bmatrix} \mathbf{X}^{(1)} & \mathbf{X}^{(n_3)} & \cdots & \mathbf{X}^{(2)} \\ \mathbf{X}^{(2)} & \mathbf{X}^{(1)} & \cdots & \mathbf{X}^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}^{(n_3)} & \mathbf{X}^{(n_3-1)} & \cdots & \mathbf{X}^{(1)} \end{bmatrix}.$$
The tensor nuclear norm $\|X\|_n$ is defined as

$$\|X\|_n = \sum_{i=1}^{r} S(i, i, 1),$$

where $r = \text{rank}_c(X)$.

**Lemma 2** [14]: For tensor $X \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the following properties hold.

$$\|X\|_n = \frac{1}{n_3} \| \text{bcirc}(X) \|_n = \frac{1}{n_3} \| \tilde{X} \|_n.$$  

**IV. RTF MODEL**

In this section, we first give the proposed robust tensor factorization model for LRTC, then introduce an iterative alternating method for RTF optimization, and finally discuss the relationship between RTF and related low-rank tensor methods and analyze its complexity.

**A. ROBUST TENSOR FACTORIZATION MODEL**

Given a tensor $X \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ that only knows some entries, if we use $\mathbf{O}$ to denote the set of the index of the observed entries, the general model of low-rank tensor completion problem is to find a low-rank tensor to fill in the missing values of tensor $X$ under the $\mathbf{O}$ of its entries $\{X_{ijk} | (i, j, k) \in \mathbf{O}\}$. If we use the tubal tank to measure the low-rankness of tensor, the tensor completion problem is to solve an optimization problem as follows

$$\min_{\mathcal{L}} \text{rank}_c(\mathcal{L}), \ s.t. \ \mathcal{P}_G(X) = \mathcal{P}_G(\mathcal{L}),$$

where $\mathcal{P}_G$ is the linear operator that extracts entries in $\mathbf{O}$ and fills the entries not in $\mathbf{O}$ with zeros, $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is the estimated tensor.

It has been proved that tubal rank minimization is a good measure of three-dimensional sparsity. However, due to the discrete nature of the tubal rank function, the optimization problem of (11) is NP-hard. Like solving the rank optimization problem for matrices, an efficient and tractable approach is to relax the tensor tubal rank $\text{rank}_c(\mathcal{L})$ into tensor nuclear norm $\|\mathcal{L}\|_n$. In addition, considering the noise in the model, the problem (11) is changed over to the following robust optimization problem

$$\min_{\mathcal{L}, \mathcal{E}} \|\mathcal{L}\|_n + \lambda \|\mathcal{P}_G(\mathcal{E})\|_1, \ s.t., \ \mathcal{P}_G(X) = \mathcal{P}_G(\mathcal{L} + \mathcal{E}),$$

where tensor $\ell_1$-norm $\| \cdot \|_1$ used to measure the noise, $\mathcal{E}$ is the noise tensor, and $\lambda \geq 0$ is a regularization parameter to balance the tubal rank of the estimated tensor and the noise. If all entries in $\mathcal{X}$ are observable, Eq. (12) is the objective function of TRPCA [14].

Inspired by the equivalent transformation of matrix factorization in the matrix nuclear norm, we expect that the low-tubal rank tensor $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is equivalently converted into the t-products with two small low-rank tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times d \times n_3}$ and $\mathcal{Z} \in \mathbb{R}^{d \times n_2 \times n_3}$, i.e., $\mathcal{L} = \mathcal{A} \ast \mathcal{Z}$, where $d$ is an upper bound on the tubal rank of $\mathcal{L}$, i.e., $d \geq r = \text{rank}_c(\mathcal{L})$.

In this decomposition, $\mathcal{A}$ is always regarded as a dictionary, $\mathcal{Z}$ is the new representation tensor of the tensor $\mathcal{X}$ under the dictionary $\mathcal{A}$, if dictionary $\mathcal{A}$ is an orthogonal tensor, i.e., $\mathcal{A}^\ast \ast \mathcal{A} = \mathcal{I}$. Then, by using the Lemma 3, the problem (12) is converted to the following equivalent problem

$$\min_{\mathcal{Z}, \mathcal{E}} \|\mathcal{Z}\|_n + \lambda \|\mathcal{P}_G(\mathcal{E})\|_1, \ s.t. \ \mathcal{P}_G(X) = \mathcal{P}_G(\mathcal{A} \ast \mathcal{Z} + \mathcal{E}).$$

In the above model, the results of two small tensor t-product are used to estimate the low-rank structure of tensor data, and the $\ell_1$-norm that is robust to noise is used to measure the model error. Therefore, we call the tensor completion model (13) as the robust tensor factorization (RTF) model.

In fact, the solution of our proposed RTF in model (13) is equivalent to the solution of model (12), which is proved in Theorem 2.

**Lemma 3**: Let $\mathcal{X} \in \mathbb{R}^{n_1 \times d \times n_3}$ and $\mathcal{Y} \in \mathbb{R}^{d \times n_2 \times n_3}$ are two tensors, where $\mathcal{X}$ is an orthogonal tensor, i.e., $\mathcal{X}^\ast \ast \mathcal{X} = \mathcal{I}$, then we have $\|\mathcal{X} \ast \mathcal{Y}\|_n = \|\mathcal{Y}\|_n$.

**Proof**: Let the t-SVD of $\mathcal{Y}$ be $\mathcal{Y} = \mathcal{U} \ast \mathcal{S} \ast \mathcal{V}^\ast$, according the definition of the tensor nuclear norm, we can easily obtain $\|\mathcal{Y}\|_n = \|\mathcal{S}\|_n$ and $\mathcal{X} \ast \mathcal{Y} = \mathcal{X} \ast \mathcal{U} \ast \mathcal{S} \ast \mathcal{V}^\ast$. Since $(\mathcal{X} \ast \mathcal{U})^\ast (\mathcal{X} \ast \mathcal{U}) = \mathcal{I}$, $(\mathcal{X} \ast \mathcal{U}) \ast \mathcal{S} \ast \mathcal{V}^\ast$ is actually a t-SVD of $\mathcal{X} \ast \mathcal{Y}$, then, we have $\|\mathcal{X} \ast \mathcal{Y}\|_n = \|\mathcal{S}\|_n = \|\mathcal{X} \ast \mathcal{Y}\|_n$. □

**Theorem 2**: Suppose $(\mathcal{L}', \mathcal{E}')$ is a solution of the model (12) with rank$_c(\mathcal{L}') = r$, then there exists the solution $\mathcal{Z}' = \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{E}'' = \mathbb{R}^{n_1 \times n_2 \times n_3}$ to the model (13) with $d \geq r$, such that $(\mathcal{A}'' \ast \mathcal{Z}'', \mathcal{E}'')$ is also a solution to the model (12), where $\mathcal{A}'' \in \mathbb{R}^{n_1 \times d \times n_3}$ is an orthogonal tensor, i.e., $\mathcal{A}''^\ast \ast \mathcal{A}'' = \mathcal{I}$.

**Proof**: If $(\mathcal{L}', \mathcal{E}')$ is a solution to the optimization problem (12), and rank$_c(\mathcal{L}') = r$, by implementing t-SVD on tensor $\mathcal{L}'$, we can easily find an orthogonal tensor $\mathcal{A}'' \in \mathbb{R}^{n_1 \times d \times n_3}$ and a tensor $\mathcal{Z}'' \in \mathbb{R}^{n_1 \times d \times n_3}$ satisfying $\mathcal{A}'' \ast \mathcal{Z}'' = \mathcal{L}'$ and $\mathcal{P}_G(\mathcal{A}) = \mathcal{P}_G(\mathcal{A}'' \ast \mathcal{Z}'' + \mathcal{E}'')$, where $d \geq r$, $\mathcal{E}'' = \mathcal{E}'$. In addition, according to Lemma 3, we have

$$\min_{\mathbf{L}'', \mathbf{E}''} \|\mathbf{Z}''\|_n + \lambda \|\mathcal{P}_G(\mathcal{E}'')\|_1, \ s.t., \mathcal{A}''^\ast \ast \mathcal{A}'' = \mathcal{I}, \mathcal{P}_G(\mathcal{A}'' \ast \mathcal{Z}'' + \mathcal{E}''),$$

$$= \min_{\mathbf{Z}'', \mathbf{E}''} \|\mathbf{A}'' \ast \mathbf{Z}''\|_n + \lambda \|\mathcal{P}_G(\mathcal{E}'')\|_1, \ s.t., \ \mathcal{P}_G(\mathcal{A}) = \mathcal{P}_G(\mathcal{A}'' \ast \mathcal{Z}'' + \mathcal{E}''),$$

$$= \min_{\mathbf{L}', \mathbf{E}'} \|\mathbf{L}'\|_n + \lambda \|\mathcal{P}_G(\mathcal{E}')\|_1, \ s.t., \ \mathcal{P}_G(\mathcal{X}) = \mathcal{P}_G(\mathcal{L}' + \mathcal{E}').$$

The proof is completed. □

**B. OPTIMIZATION OF RTF**

In order to solve the problem conveniently, we set the unknown entries in the given tensor data $\mathcal{X}$ to zeros, i.e., $\mathcal{X}_{\mathbf{O}} = 0$, where $\mathbf{O}$ is the complement of $\mathbf{O}$. This means that any value can be taken in $\mathcal{X}_{\mathbf{O}}$ such that $\mathcal{P}_G(\mathcal{X}) = \mathcal{P}_G(\mathcal{A} \ast \mathcal{Z} + \mathcal{E})$. Therefore, the constraint with
the operator \( P_{\mathfrak{H}} \) in (13) is simplified into \( \mathfrak{X} = \mathcal{A} * Z + \mathcal{E} \). Hence, the problem (13) is transformed into the following equivalent problem

\[
\min_{\mathfrak{E}, Z, \mathfrak{J}} \| \mathfrak{J} \|_* + \lambda \| P_{\mathfrak{H}}(\mathfrak{E}) \|_1, \quad \\
\text{s.t. } \mathfrak{X} = \mathcal{A} * Z + \mathfrak{E}, \ Z = \mathfrak{J}.
\] (15)

Then, we resort to augmented Lagrangian multiplier method and the corresponding Lagrangian function as following

\[
\Phi(\mathfrak{E}, Z, \mathfrak{J}, \mathfrak{Y}^1, \mathfrak{Y}^2) = \| \mathfrak{J} \|_* + \lambda \| P_{\mathfrak{H}}(\mathfrak{E}) \|_1 \\
+ \langle \mathfrak{Y}^1, \mathfrak{X} - \mathcal{A} * Z - \mathfrak{E} \rangle \\
+ \langle \mathfrak{Y}^2, Z - \mathfrak{J} \rangle \\
+ \frac{\mu}{2} (\| \mathfrak{X} - \mathcal{A} * Z - \mathfrak{E} \|_F^2 + \| Z - \mathfrak{J} \|_F^2),
\] (16)

where \( \mu \) is a penalty parameter improve convergence, \( \mathfrak{Y}^1 \) and \( \mathfrak{Y}^2 \) are Lagrange multipliers. Then, we carry out the alternating direction method of multipliers (ADMM) [37] to update \( \mathfrak{E}, Z, \) and \( \mathfrak{J} \) sequentially, i.e. updating one of the three variables while fixing others.

**Update \( \mathfrak{J} \):** The Lagrangian function (16) is unconstrained, so the minimization of Eq. (16) with respect to \( \mathfrak{J} \) involves the following

\[
\mathfrak{J} = \arg \min_{\mathfrak{J}} \| \mathfrak{J} \|_* + \langle \mathfrak{Y}^1, \mathfrak{X} - \mathcal{A} * Z - \mathfrak{E} \rangle \\
+ \langle \mathfrak{Y}^2, Z - \mathfrak{J} \rangle \\
+ \frac{\mu}{2} (\| \mathfrak{X} - \mathcal{A} * Z - \mathfrak{E} \|_F^2 + \| Z - \mathfrak{J} \|_F^2),
\] (17)

where \( \mathfrak{P} = (Z + \mathfrak{Y}^2/\mu). \) The above problem (17) can be transformed into the complex number domain and solved. By the properties of Lemma 1, the equivalent problem of problem (17) as follows

\[
\min \frac{1}{n_3} (\| \mathfrak{J} \|_* + \frac{\mu}{2} \| \mathfrak{J} - \mathfrak{P} \|_F^2).
\] (18)

Since \( \mathfrak{J} \) is a block-diagonal matrix, the problem (18) can be solved by the following equivalent problem

\[
\min_{\mathfrak{J}} \| \mathfrak{J}^{(i)} \|_* + \frac{\mu}{2} \| \mathfrak{J}^{(i)} - \mathfrak{P}^{(i)} \|_F^2, \quad i = 1, 2, \ldots, n_3,
\] (19)

which can be solved by the singular thresholding (SVT) operator. The SVT operator \( S_\eta \) is defined as

\[
S_\eta(X) = U \Sigma_\eta V^*,
\] (20)

where \( X = U \Sigma V^* \) is the singular value decomposition, and \( \Sigma_\eta(x) = \text{sgn}(x) \max(abs(x) - \eta, 0) \) is the shrinkage operator. Then, the closed-form solution of problem Eq. (19) are

\[
\mathfrak{J}^{(i)} = S_{\mu/2}(\mathfrak{P}^{(i)}), \quad i = 1, 2, \ldots, n_3,
\] (21)

then, we finally get \( \mathfrak{J} = \text{ifft}((\mathfrak{J}, [], 3) \).

**Update \( \mathfrak{Z} \):** Optimization Eq. (15) with respect to \( \mathfrak{Z} \) as follows

\[
\mathfrak{Z} = \arg \min_{\mathfrak{Z}} \langle \mathfrak{Y}^1, \mathfrak{X} - \mathcal{A} * \mathfrak{Z} - \mathfrak{E} \rangle \\
+ \frac{\mu}{2} (\| \mathfrak{X} - \mathcal{A} * \mathfrak{Z} - \mathfrak{E} \|_F^2 + \| \mathfrak{Z} - \mathfrak{J} \|_F^2),
\] (22)

where \( \mathfrak{R} = \mathfrak{X} - \mathfrak{E} + \mathfrak{Y}^1/\mu, \) and \( \mathfrak{R}^2 = -\mathfrak{J} + \mathfrak{Y}^2/\mu. \) Problem (22) is a convex function for the variable \( \mathfrak{Z}. \) Therefore, let Eq. (22) directly differentiate the variable \( \mathfrak{Z} \) and make its result equal to zero. We can get the following equation

\[
-\mathcal{A}^* (\mathfrak{R}^1 - \mathcal{A} * Z) + \mathfrak{Z} + \mathfrak{R}^2 = 0.
\]

From the above equation, it is easy to get the solution about the variable \( \mathfrak{Z} \) as following

\[
\mathfrak{Z} = (\mathcal{A}^* * \mathcal{A} + \mathfrak{I})^{-1} * (\mathcal{A}^* * \mathfrak{R}^1 - \mathfrak{R}^2).
\] (23)

**Update \( \mathfrak{E} \):** Optimizing Eq. (15) with respect to \( \mathfrak{E} \) is converted to the following problem

\[
\mathfrak{E} = \arg \min_{\mathfrak{E}} \langle \mathfrak{Y}^1, \mathfrak{X} - \mathcal{A} * \mathfrak{Z} - \mathfrak{E} \rangle \\
+ \frac{\mu}{2} (\| \mathfrak{X} - \mathcal{A} * \mathfrak{Z} - \mathfrak{E} \|_F^2),
\] (24)

where \( \mathfrak{Q} = \mathfrak{X} - \mathcal{A} * \mathfrak{Z} + \mu^{-1} \mathfrak{Y}^1. \) The problem (24) can be solved by the following two subproblems with respect to \( \mathfrak{E}_{\mathfrak{Q}} \) and \( \mathfrak{E}_{\mathfrak{Z}}, \) respectively.

For \( \mathfrak{E}_{\mathfrak{Q}}, \) the optimization problem (24) is formulated as following

\[
\mathfrak{E}_{\mathfrak{Q}} = \arg \min_{\mathfrak{Q}} \| P_{\mathfrak{H}}(\mathfrak{E}) \|_1 + \frac{\mu}{2\lambda} \| P_{\mathfrak{H}}(\mathfrak{E} - \mathfrak{Q}) \|_F^2.
\] (25)

By the element-wise shrinkage operator \( \Sigma_\eta(x), \) the closed form solution of problem (25) as follows

\[
\mathfrak{E}_{\mathfrak{Q}} = \Sigma_{\lambda/\mu}(\mathfrak{Q}).
\] (26)

For \( \mathfrak{E}_{\mathfrak{Z}}, \) the subproblem with respect to \( \mathfrak{E}_{\mathfrak{Z}} \) as following

\[
\mathfrak{E}_{\mathfrak{Z}} = \arg \min_{\mathfrak{E}_{\mathfrak{Z}}} \frac{\mu}{2\lambda} \| P_{\mathfrak{H}}(\mathfrak{E} - \mathfrak{Q}) \|_F^2,
\] (27)

which is a convex function with respect to \( \mathfrak{E}_{\mathfrak{Z}}, \) we can obtain the closed-form solution by zeroing the gradient of the objective function (27), i.e.,

\[
\mathfrak{E}_{\mathfrak{Z}} = \mathfrak{Q}_{\mathfrak{Z}}.
\] (28)

**Update \( \mathfrak{Y}^1, \mathfrak{Y}^1, \) and \( \mu; \)**

\[
\mathfrak{Y}^1 = \mathfrak{Y}^1 + \mu (\mathfrak{X} - \mathcal{A} * \mathfrak{Z} - \mathfrak{E}),
\]

\[
\mathfrak{Y}^2 = \mathfrak{Y}^2 + \mu (\mathfrak{Z} - \mathfrak{J}),
\]

\[
\mu = \min(\rho \mu, \mu_{\text{max}}).
\] (29)
Algorithm 1 Robust Tensor Factorization for Low-Rank Tensor Completion (RTF)

1: **Input:** data tensor \( X \in \mathbb{R}^{n_1 \times n_2 \times n_3} \), dictionary tensor \( A \in \mathbb{R}^{d \times n_1 \times n_2} \), observed set \( \Omega \), parameter \( \lambda, \epsilon = 1e - 4. 
2: **Output:** \( Z \in \mathbb{R}^{d \times n_2 \times n_3} \).
3: Initialize: \( Z_0, E_0, J_0, Y_0^1, \) and \( Y_0^2; \)
4: While not converged do;
5: Update each frontal slice \( \hat{J}_{t+1}^{(i)} \) for \( i = 1, 2, \ldots, n_3 \) of \( J_{t+1} \) by Eq. (21), where \( P_{t+1} = (Z_t + \frac{Y_t^2}{\mu_t}) \); 
6: Update \( Z_{t+1} \) by Eq. (23), where \( R_{t+1} = X - E_t + Y_t^1/\mu_t, \) and \( R_{t+1}^2 = -J_{t+1} + Y_t^2/\mu_t; \)
7: Update the entries of \( E_{t+1} \) by Eq. (28), where \( Q_{t+1} = X - A \ast Z_{t+1} + \mu_t^{-1}Y_t^1; \)
8: Update multipliers \( Y_{t+1}^1, Y_{t+1}^2, \) and parameter \( \mu_{t+1} \) by Eq. (29);
9: check the convergence conditions: 
\[ \| J_{t+1} - J_t \|_\infty \leq \epsilon, \| Z_{t+1} - Z_t \|_\infty \leq \epsilon, \] and 
\[ \| E_{t+1} - E_t \|_\infty \leq \epsilon. \]

After updating, we check the stopping criterion that the maximum updating difference is less than a predefined threshold \( \epsilon \).

C. CONVERGENCE AND COMPLEXITY

In this section, we analyze the convergence property and the complexity of the proposed method. For convenience, we rewrite the objective function (13) of RTF by assuming the fully-observed data, i.e., \( w_{ijk} = 1 \) for all \( i, j, k \), the Eq. (13) is changed to follows
\[
\min_{Z, E} \| Z \|_* + \lambda \| E \|_1, \quad s.t. \quad X = A \ast Z + E, \]  
(30)

which is the model of TLRR and its convergence has been proved [38]. According to Lemma 3, and let \( E = X - L \), then Eq. (30) is converted to the following
\[
\min_{L, E} \| L \|_* + \lambda \| E \|_1, \quad s.t. \quad X = L + E. \]  
(31)

The Eq. (31) is the objective function of TRPCA [14]. Therefore, if all the entries in the tensor data are known, the objective functions (30) of RTF or TLRR and TRPCA are essentially equivalent. In fact, the proof of TRPCA and TLRR convergence are also valid for tensor completion problems (12) and (13) where only a part of the entries are known.

The main cost of ADMM for solving (15) is to compute \( J, Z, \) and \( E \). Assuming without loss of generality that \( X \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) and \( n_1 > n_2 \), the time complexity for \( J \) involves the \( n_3 \) SVD on matrices with size \( d \times n_2 \), the total cost is \( O(d^2 n_2 n_3) \). The update of \( Z \) in Eq. (23) involves t-product and inversion of tensor, the total computational cost is \( O(d(n_1 + n_2)^3 + d n_1 n_2 n_3) \). The computation cost for \( E \) involves the calculation of \( A \ast Z \) and the update which is \( O(n_1 n_2 n_3(d + \log n_3)) \). The total time complexity for solving the RTF problem is \( O(n_1 n_2 n_3(d + \log n_3)) \), where \( t \) is the number of iterations.

V. EXPERIMENTS

In this section, we conduct numerical experiments to evaluate the effectiveness of our method RTF. We compare it with other state-of-the-art tensor completion methods on color image recovering and grayscale video recovering tasks. The recovering task is to fill in the missing pixel values of a partially observed image or video. All the simulations are conducted on the platform Matlab 2019a under Windows 10 with an Intel Xeon E3-1505M 3.00 GHz CPU and 64 GB memory.

A. COMPARISON METHODS

In each dataset, we evaluate the effectiveness of the proposed RTF and the following state-of-the-art low-rank tensor completion methods on color image recovering and grayscale video recovering tasks.

1) **GeomCG**: [7] performs tensor completion by developing a Riemannian optimization on the low-rank tensor manifold based on Tucker decomposition.
2) **TenALS**: [11] presents a novel alternating minimization method to efficiently and exactly reconstruct a low-rank tensor based on CP decomposition.
3) **TMac**: [10] recovers a low-rank tensor by simultaneously performing low-rank matrix factorizations to the all-mode matricizations of the underlying tensor.
4) **TMacTT**: [30] firstly estimates the tensor rank by the tensor train technology, and then factorizes each modular matrix to complete the tensor completion.
5) **LRTC**: [8] generalizes the matrix trace norm to tensor trace norm and presents a convex optimization problem for tensor completion.
6) **FBCP and FBCMP**: [9] proposes two versions: FBCP and FBCMP, the former is fully Bayesian CP factorization using a hierarchical probabilistic model while the latter is its extension using mixture prior.
7) **TRPCA**: [14] firstly defines the tensor tubal rank to measure the low-rankness of tensor, and then generalizes the matrix RPCA to the tensor RPCA for tensor completion.
8) **TCTF**: [12] proposes a novel low-rank tensor completion method by factorizing the tensor into the t-product of two tensors with smaller sizes.

The code of all these compared methods are provided by their corresponding authors. Assume that \( \hat{X} \) is the recovered tensor of \( X \in \mathbb{R}^{n_1 \times n_2 \times n_3} \), the peak signal-to-noise ratio

1) https://anchp.epfl.ch/index-html/software/geomcg/
2) https://bamdevmisra.in/codes/tensor_completion/
3) https://xu-yangyang.github.io/TMac/
4) https://nbviewer.jupyter.org/github/xinychen/geotensor/blob/master/TMac-Test.ipynb
5) http://www.cs.rochester.edu/~jliu/
6) https://github.com/qbzhao/BCPF
7) https://canyilu.github.io/publications/
8) https://panzhous.github.io/
is evaluated by comparing the PSNR values between the repaired image obtained by various algorithms and the ground truth.

The average PSNR values of all methods with different sample rates are shown in Figure 3. The corresponding average PSNR values are displayed at the top of each bar. Due to FBCP and FBCPMP are from the same paper [9], and the PSNR of FBCP for images is higher, we only show the recovering results of FBCP. In Figure 3, we can find that the average PSNR value of the proposed RTF is consistently higher than other comparison algorithms at five different sample rates. When the sample rate \( \theta \) changes from 10% to 50%, the average PSNR corresponding to RTF is higher than the second TRPCA by 0.40 dB, 0.54 dB, 0.64 dB, 0.68 dB, and 0.60 dB, respectively. According to the corresponding average PSNR value, from high to low, under these five sample rates, the second place is TRPCA, and the third and fourth places are FBCP and TMacTT, respectively. The two worst algorithms are GeomCG and TMac.

For noisy situations, due to space constraints, we only give the PSNR values obtained by all comparison algorithms for recovering 4 images when the sample rate \( \theta = 20\% \), 30\%, and \( \sigma = 0, 5, 10 \) in Table 2. The running time with sampling rate 20% for these 4 noiseless images are demonstrated in Table 3. Figure 4 shows the recovering results of the first two testing images. It can be seen from Table 2 that the repair performance of RTF has always been higher than other algorithms on all images. The algorithm in the second place has changed for different types of images, some are TRPCA, some are FBCP, and some are TMacTT. However, the worst algorithm has always been GeomCG and TMac. From the recovering results in Figure 4, the proposed RTF corresponds to the highest PSNR value, and the clearest repair images are also obtained, which is also very good for small targets, such as eyes and trees. In general, the four algorithms RTF, TRPCA, FBCP, and TMacTT can restore clearer images. The images restored by the three algorithms TCTF, TenALS, and LRTC have obvious streaks, and TMac cannot estimate a large number of missing pixels.

D. VIDEO RESTORATION

A grayscale video is essentially a 3-order tensor. We conduct experiments on the YUV Video Sequences\(^9\) database which is widely used in video recovering. All 18 videos with provided QCIF format in the dataset are used to evaluate the recovering ability of the proposed RTF and comparison algorithms to repair the video data. Each video sequence contains at least 150 frames, we only utilize 30 frames of sequence for the experiment due to the computational limitation. In experiments, we convert color each frame into grayscale, and thus the sequences can be formated as 3-way tensors with size 144 \( \times \) 176 \( \times \) 30. See the first column on the left Figure 6 for four video sequences.

\(^9\)http://r0k.us/graphics/kodak/ 

\(^{10}\)http://trace.eas.asu.edu/yuv/
FIGURE 3. Comparison of the noiseless image recovering performance. We apply the compared methods to recover the images by 10% ~ 50% sample ratio in Kodak and report the average PSNR values on the 24 images. Best viewed in ×2 sized color pdf file.

TABLE 2. The PSNR of the recovering results of all algorithms with different sampling rate $\theta$ and Gaussian noise standard deviation $\sigma$ for the first 4 images of Fig. 2 in Kodak dataset.

| Image | $\theta$ | $\sigma$ | GeomCG | TenALS | TMac | TMacTT | LRTC | FBCP | TRPCA | TCTF | RTF |
|-------|---------|---------|--------|--------|------|--------|------|------|-------|------|-----|
| Women | 20%     | 0       | 9.76   | 21.23  | 10.66| 27.41  | 20.94| 27.35| 27.34  | 21.68| 28.16|
|       | 5       | 9.24   | 21.83  | 9.28   | 26.59| 20.90  | 26.94| 26.61| 21.24  | 27.19|
|       | 10      | 9.13   | 21.28  | 9.27   | 26.11| 20.80  | 26.34| 25.16| 20.60  | 26.47|
|       | 0       | 10.86  | 21.30  | 12.20  | 28.26| 23.09  | 28.90| 29.88| 26.69  | 30.73|
|       | 30%     | 5       | 10.62  | 21.48  | 12.19| 27.48  | 23.01| 28.37| 28.53  | 26.14| 29.15|
|       |         | 10      | 10.15  | 21.37  | 12.15| 26.13  | 22.78| 27.77| 26.51  | 25.12| 27.67|
| Lake  | 20%     | 0       | 9.02   | 24.49  | 10.05| 26.45  | 22.86| 28.01| 28.74  | 20.33| 29.41|
|       | 5       | 8.26   | 24.50  | 10.04  | 25.78| 22.82  | 27.70| 27.85| 20.31  | 28.35|
|       | 10      | 8.02   | 24.22  | 10.02  | 25.45| 22.67  | 26.16| 26.30| 19.59  | 26.52|
|       | 0       | 10.13  | 24.54  | 11.76  | 27.04| 24.70  | 29.05| 31.43| 25.56  | 32.06|
|       | 30%     | 5       | 10.05  | 24.33  | 11.74| 26.64  | 24.59| 28.81| 29.63  | 25.67| 30.24|
|       |         | 10      | 10.00  | 24.51  | 11.71| 26.02  | 24.27| 27.21| 27.33  | 24.85| 27.57|
| Windows | 20%    | 0       | 8.67   | 20.95  | 9.70 | 22.21  | 20.27| 23.37| 24.25  | 19.27| 24.59|
|        | 5       | 8.65   | 20.95  | 8.24  | 21.83| 20.24  | 23.30| 23.94| 19.31  | 24.11|
|        | 10      | 8.66   | 20.93  | 9.66  | 21.18| 20.18  | 23.22| 23.21| 18.73  | 23.23|
|        | 0       | 9.74   | 21.03  | 11.35 | 22.77| 21.87  | 24.39| 26.66| 22.28  | 27.13|
|        | 30%     | 5       | 9.75   | 21.01  | 8.92 | 21.73| 21.81 | 24.17| 25.95  | 22.17| 26.22|
|        |         | 10      | 9.16   | 21.02  | 11.34| 20.30| 21.66 | 23.99| 24.68  | 21.89| 24.74|
| Door  | 20%     | 0       | 10.44  | 24.13  | 11.35| 26.72 | 22.69| 27.37| 27.56  | 19.83| 28.17|
|        | 5       | 10.33  | 24.01  | 11.32| 25.35| 22.62  | 27.05| 26.70| 19.48  | 27.08|
|        | 10      | 10.35  | 23.96  | 11.23| 25.09| 22.40  | 25.30| 25.03| 18.72  | 25.55|
|        | 0       | 11.54  | 24.19  | 12.96| 27.50| 24.43  | 28.60| 29.77| 24.91  | 30.38|
|        | 30%     | 5       | 11.46  | 24.12  | 12.90| 26.13| 24.26 | 28.24| 28.15  | 24.28| 28.56|
|        |         | 10      | 11.21  | 24.13  | 12.74| 25.92| 23.82 | 27.46| 26.85  | 22.83| 27.85|

TABLE 3. The running time (sec.) with sampling rate 20% for 4 noiseless images in Kodak dataset.

| Image   | GeomCG | TenALS | TMac | TMacTT | LRTC | FBCP | TRPCA | TCTF | RTF |
|---------|--------|--------|------|--------|------|------|-------|------|-----|
| Women   | 24.98  | 5.30   | 26.94| 1.50   | 7.14 | 213.09| 41.78 | 6.89 | 12.05|
| Lake    | 25.21  | 5.41   | 19.99| 1.40   | 7.36 | 170.81| 41.13 | 5.86 | 12.42|
| Windows | 25.82  | 5.96   | 24.03| 2.09   | 7.89 | 177.67| 43.15 | 7.27 | 13.77|
| Door    | 25.60  | 5.44   | 62.95| 1.45   | 6.90 | 187.56| 41.46 | 6.10 | 12.43|

The generation method of the video testing tensor is consistent with the image restoration method. For a video with $n_3$ sequences and each frame has size $n_1 \times n_2$, we construct a testing tensor $X \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and randomly set $\theta n_1 n_2 n_3$ entries to be observed. In the case of 5 sampling rates, the average PSNR values corresponding to the restoration results of all 18 noiseless videos by various algorithms are shown in Figure 5. Table 4 demonstrates the PSNR values
corresponding to the recovering results of all algorithms on 4 videos with different sampling rate and Gaussian noise. The noiseless restoration results of one frame in 4 videos are shown in Figure 6, and the corresponding running times are exhibited in Table 5. It should be pointed out that the TMacTT algorithm needs to train a core tensor of unknown size and order when processing video, and different videos require different core tensors. Therefore, in the comparison experiment of video inpainting, the TMacTT was replaced with the FBCPMP. Regardless of the change in sample rate
TABLE 4. The PSNR of four video recovering results of all algorithms with different sampling rate $\theta$ and standard deviation $\sigma$ of Gaussian noise in YUK dataset.

| Video | $\theta$ | $\sigma$ | GeomCG | TenALS | TMac | FBCPMP | LRTC | FBCP | TRPCA | TCTF | RTF |
|-------|----------|----------|--------|--------|------|--------|------|------|-------|------|-----|
| News  | 20%      | 0        | 10.64  | 21.71  | 28.84 | 22.82  | 22.79 | 31.74 | 33.97 | 10.24 | 34.64 |
|       |          | 5        | 10.65  | 21.38  | 28.27 | 22.80  | 22.76 | 31.45 | 31.23 | 10.27 | 32.13 |
|       |          | 10       | 9.13   | 21.62  | 27.60 | 22.74  | 22.39 | 28.14 | 28.29 | 10.15 | 28.89 |
|       | 30%      | 0        | 11.78  | 22.81  | 30.05 | 23.48  | 25.44 | 33.88 | 36.22 | 14.53 | 36.88 |
|       |          | 5        | 11.73  | 22.73  | 29.64 | 23.46  | 25.03 | 33.15 | 32.42 | 14.24 | 33.68 |
|       |          | 10       | 10.27  | 21.93  | 28.67 | 23.40  | 24.55 | 30.52 | 28.71 | 14.07 | 29.79 |
| Coastguard | 20%      | 0        | 7.79   | 24.24  | 24.71 | 24.49  | 24.15 | 27.61 | 28.96 | 9.17  | 29.32 |
|       |          | 5        | 7.39   | 24.06  | 24.60 | 24.43  | 24.04 | 27.42 | 28.09 | 9.16  | 28.28 |
|       |          | 10       | 7.27   | 23.81  | 24.26 | 24.32  | 23.75 | 27.00 | 26.57 | 9.15  | 27.60 |
|       | 30%      | 0        | 8.93   | 25.90  | 25.75 | 25.01  | 25.93 | 28.67 | 30.82 | 12.95 | 31.32 |
|       |          | 5        | 8.39   | 24.61  | 25.62 | 24.96  | 25.72 | 28.30 | 29.38 | 12.99 | 30.79 |
|       |          | 10       | 8.09   | 24.26  | 25.25 | 24.84  | 25.16 | 27.88 | 27.26 | 12.84 | 28.00 |
| Hall  | 20%      | 0        | 6.46   | 21.52  | 28.69 | 23.44  | 23.23 | 33.73 | 35.55 | 8.34  | 36.33 |
|       |          | 5        | 6.19   | 21.33  | 28.39 | 23.41  | 23.13 | 33.13 | 32.34 | 8.30  | 33.88 |
|       |          | 10       | 6.07   | 21.02  | 27.62 | 23.91  | 22.86 | 31.90 | 29.11 | 8.28  | 32.01 |
|       | 30%      | 0        | 7.60   | 22.56  | 31.07 | 24.52  | 25.84 | 36.24 | 37.51 | 12.99 | 38.10 |
|       |          | 5        | 7.33   | 21.88  | 30.65 | 24.65  | 25.59 | 35.11 | 33.42 | 12.89 | 36.31 |
|       |          | 10       | 7.23   | 21.14  | 29.59 | 24.36  | 24.98 | 33.58 | 29.53 | 12.69 | 32.19 |
| Mobile| 20%      | 0        | 6.47   | 16.73  | 17.61 | 19.87  | 18.30 | 20.16 | 21.51 | 8.09  | 21.63 |
|       |          | 5        | 6.22   | 16.49  | 17.57 | 19.88  | 18.23 | 20.04 | 21.35 | 8.08  | 21.30 |
|       |          | 10       | 6.17   | 16.20  | 17.46 | 19.76  | 18.03 | 19.90 | 20.27 | 8.06  | 20.46 |
|       | 30%      | 0        | 7.55   | 17.91  | 18.51 | 20.23  | 20.02 | 20.98 | 23.75 | 11.51 | 23.87 |
|       |          | 5        | 7.36   | 17.87  | 18.47 | 20.21  | 19.89 | 21.02 | 23.15 | 11.55 | 23.16 |
|       |          | 10       | 7.19   | 16.70  | 18.35 | 20.13  | 19.52 | 20.83 | 21.94 | 11.35 | 22.64 |

FIGURE 6. Examples of video recovering results with sample ratio $\theta = 20\%$. (a) Input frame. (b) Noise frame. (c)-(k) are the recovered results by the GeomCG, TenALS, TMac, FBCPMP, LRTC, FBCP, TRPCA, TCTF, and RTF, respectively. Best viewed in $\times 2$ sized color pdf file.

θ or standard deviation σ of Gaussian noise, it can be seen from Figure 5 and Table 4 that consistent with the image recovering results, in most cases, the proposed RTF can obtain the highest PSNR value in video restoration in these comparative algorithms. When the sample rate is 20%, it can be found from Figure 5 that RTF can get the best results in these 18 video recovering experiments. The worst performing algorithms in video repair are GeomCG and TCTF. From the recovering results in Figure 6, we can find that RTF and TRPCA can get the best recover results, and FBCP can also get better recovering frames, but there are more mosaics in the recovery of the fourth complex video Mobile. The repair results of FBCPMP are all blurred, the recover frames of TMac are greatly blurred, and the repair frames of LRTC are more streaked. GeocCG and TCTF cannot get the correct repair results.
TABLE 5. The running time (sec.) with sampling rate 20% for 4 noiseless videos in YUV dataset.

| Video   | GeomCG | TenALS | TMac | FBCPMP | LRTC | FBCP | TRPCA | TCTF | RTF |
|---------|--------|--------|------|--------|------|------|-------|------|-----|
| News    | 42.10  | 3.52   | 10.79| 126.19 | 2.67 | 159.07 | 16.27 | 9.91 | 16.69 |
| Coastguard | 41.52  | 3.65   | 10.37| 124.71 | 2.49 | 227.66 | 14.58 | 9.67 | 11.11 |
| Hall    | 42.18  | 3.79   | 14.01| 124.42 | 2.95 | 153.60 | 15.80 | 10.05 | 16.65 |
| Mobile  | 48.41  | 3.44   | 7.50 | 166.76 | 3.85 | 216.84 | 13.75 | 11.06 | 12.65 |

E. RESULTS AND ANALYSIS

The following conclusions can be drawn from the above tables and figures.

1) Whether in the image recovering or in the video restoration, the proposed RTF can always get the highest PSNR value among all the comparison algorithms. This shows that the proposed RTF can well describe the essential low-rank structure of tensor data, and simultaneously fill the unknown entries.

2) These methods can be divided into four categories: i.e., a) Tucker decomposition, GeomCG; b) CP factorization, TenALS, TMac, TMacTT, FBCP, and FBCPMP; c) Matrix nuclear norm, LRTC; d) t-SVD decomposition, TRPCA, TCTF, and proposed RTF. Essentially, both TRPCA and RTF use TNN that can effectively maintain global information to characterize the low-rank structure of tensor data, while TCTF only uses tensor factorization.

3) Since t-SVD can effectively describe the spatial information between different dimensions of tensor data, the performance of RTF and TRPCA based on t-SVD are better than other algorithms. The performance of TCTF is poor although it is also based on t-SVD decomposition, because factorization destroys the global structure of tensor data. Especially if the sample rate is low. In terms of running time, RTF is nearly 4 times faster than TRPCA in image restoration, while in video restoration they are almost the same.

4) Regardless of the matrix nuclear norm or TNN to describe the low-rank structures of data, these algorithms can effectively maintain the global information of the data and have good recovering ability. The algorithm that characterizes low-rank structure only by factor decomposition tends to have poor data recovery ability because it is easy to fall into local minima.

F. CONVERGENCE ANALYSIS

In section IV, we analyze that the optimal solution of our proposed RTF and TRPCA are equivalent. In addition, when all the entries of data are known, the objective function of RTF is essentially consistent with TLRR. Since TRPCA and TLRR have been proven to be able to converge to the optimal solution, therefore, the proposed RTF must converge. In Figure 7, we give the convergence error of RTF in each iteration of the first 6 images and the first 6 videos. The error is defined as the maximum value of changes in convergence conditions in algorithm 1. According to Figure 7, we can discover that the error decreases rapidly with the number of iterations. When tolerable error $\epsilon = 1e - 4$, whether it is an image or a video, our algorithm can converge in 70 iterations. The error keeps decreasing with the increase of iterations, especially when the number of iterations is greater than 30, the error decreases quickly and is stable. In fact, we found that when the number of iterations is greater than 30, the PSNR value corresponding to the tensor data recovered by RTF is almost in a stable state. In other words, if the number of iterations is set to 30 or the convergence error is set to $1e - 2$, RTF can achieve the results given in the experiment. It is set to $1e - 4$ in the paper, just to be consistent with the comparison algorithms.

VI. CONCLUSION

In this paper, we propose a robust tensor factorization (RTF) framework for the tensor completion problem. Unlike existing low-rank tensor factorization methods, the proposed RTF can not only address large-scale tensor completion problem by using tensor factorization but can also preserve the global information of tensor data. Firstly, on the premise of considering the existence of sparse noise in tensor data, we characterize the low-rank structures of tensor data by the t-product of two tensors with small sizes for tensor completion. Then, we propose an iterative algorithm based on ADMM to solve RTF. While giving the relationship between RTF and existing algorithms, we also analyze the convergence of RTF. Finally, the effectiveness of the RTF is verified in comparison with the state-of-the-art methods in both real-world image restoration and video recovering.

REFERENCES

[1] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, “Tag recommendations based on tensor dimensionality reduction,” in Proc. Conf. Rec. Syst., 2008, pp. 43–50.

[2] D. Tao, X. Li, X. Wu, and S. J. Maybank, “General tensor discriminant analysis and Gabor features for gait recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 10, pp. 1700–1715, Oct. 2007.
[3] D. Goldfarb and Z. Qin, “Robust low-rank tensor recovery: Models and algorithms,” *SIAM J. Matrix. Anal. Appl.*, vol. 35, no. 1, pp. 225–253, Jan. 2014.

[4] M. Signoretto, Q. T. Dinh, L. De Lathauwer, and J. A. K. Suykens, “Learning with tensors: A framework based on convex optimization and spectral regularization,” *Mach. Learn.*, vol. 94, no. 3, pp. 303–351, Mar. 2014.

[5] X. Li, M. K. Ng, G. Cong, Y. Ye, and Q. Wu, “MR-NTD: Manifold regularization nonnegative Tucker decomposition for tensor data dimension reduction and representation,” *IEEE Trans. Neural. Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1787–1800, Aug. 2017.

[6] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller, “Tensor-based formulation and nuclear norm regularization for multienergy computed tomography,” *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1678–1693, Apr. 2014.

[7] D. Kressner, M. Steinlechner, and B. Vander VKecken, “Low-rank tensor completion by Riemannian optimization,” *BIT Numer. Math.*, vol. 54, no. 2, pp. 447–468, Jun. 2014.

[8] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, “Robust recovery of subspace structures by low-rank representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.

[9] Q. Zhao, L. Zhang, and A. Cichocki, “Bayesian CP factorization of incomplete tensors with automatic rank determination,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1751–1763, Sep. 2015.

[10] Y. Xu, R. Hao, W. Yin, and Z. Su, “Parallel matrix factorization for low-rank tensor completion,” *Inverse Problems Imag.*, vol. 9, no. 2, pp. 601–624, 2015.

[11] P. Jain and S. Oht, “Provably tensor factorization with missing data,” in *Proc. Adv. NIPS*, 2014, pp. 1431–1439.

[12] P. Zhou, C. Lu, Z. Lin, and C. Zhang, “Tensor factorization for low-rank tensor completion,” *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1152–1163, Mar. 2018.

[13] X. Zhang, “A nonconvex relaxation approach to low-rank tensor completion,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1659–1671, Jun. 2019.

[14] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, “Tensor robust principal component analysis with a new tensor nuclear norm,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 925–938, Apr. 2020.

[15] J. Lou and Y.-M. Cheung, “Robust low-rank tensor minimization via a new tensor spectral k-support norm,” *IEEE Trans. Image Process.*, vol. 29, pp. 2314–2327, 2019.

[16] M. E. Kilmer and C. D. Martin, “Factorization strategies for third-order tensors,” *Linear Algebra its Appl.*, vol. 435, no. 3, pp. 641–658, Aug. 2011.

[17] D. Martin, R. Shafer, and B. LaRue, “An order-p tensor factorization with applications in imaging,” *SIAM J. Sci. Comp.*, vol. 35, no. 1, pp. A474–A490, 2013.

[18] Z. Zhang, G. Elly, S. Aeron, N. Hao, and M. Kilmer, “Novel methods for multilinear data completion and de-noising based on tensor-svd,” in *Proc. IEEE CVPR*, Jun. 2014, pp. 3842–3849.

[19] Z. Zhang, D. Liu, S. Aeron, and A. Vetro, “An online tensor robust PCA algorithm for sequential 2D data,” in *Proc. IEEE ICASSP*, Mar. 2016, pp. 2434–2438.

[20] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, “Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging,” *SIAM J. Matrix. Anal. Appl.*, vol. 34, no. 1, pp. 148–172, Jan. 2013.

[21] Z. Zhang and S. Aeron, “Exact tensor completion using t-SVD,” *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2017.

[22] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, “Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization,” in *Proc. IEEE CVPR*, Jun. 2016, pp. 5249–5257.

[23] X.-Y. Liu, S. Aeron, V. Aggarwal, and X. Wang, “Low-tubal-rank tensor completion using alternating minimization,” *Proc. SPIE*, vol. 9848, May 2016, Art. no. 984809.

[24] F. Shang, Y. Liu, H. Tong, J. Cheng, and H. Cheng, “Robust bilinear factorization with missing and grossly corrupted observations,” *Inf. Sci.*, vol. 307, pp. 53–72, Jun. 2015.

[25] Y. Wu, H. Tan, Y. Li, J. Zhang, and X. Chen, “A fused CP factorization method for incomplete tensors,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 751–764, Mar. 2019.

[26] X. Chen, Z. Han, Y. Wang, Q. Zhao, D. Meng, L. Lin, and Y. Tang, “A generalized model for robust tensor factorization with noise modeling by mixture of gaussians,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5380–5393, Nov. 2018.

[27] R. A. Harshman, “Foundations of the PARAFAC procedure: Model and conditions for an ‘explanatory’ multimodal factor analysis,” in *Proc. UCLA Work. Papers Phonetics*, vol. 16, 1970, pp. 1–84.

[28] C. J. Hillar and L.-H. Lim, “Most tensor problems are np-hard,” *J. ACM*, vol. 60, no. 6, p. 45, 2013.

[29] C. Mu, B. Huang, J. Wright, and D. Goldfarb, “Square deal: Lower bounds and improved relaxations for tensor recovery,” in *Proc. ICML*, 2014, pp. 73–81.

[30] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do, “Efficient tensor completion for color image and video recovery: Low-rank tensor train,” *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2466–2479, May 2017.

[31] L. R. Tucker, “Some mathematical notes on three–mode factor analysis,” *Psychometrika*, vol. 31, no. 3, pp. 279–311, Sep. 1966.

[32] L. Yang, J. Fang, H. Li, and B. Zeng, “An iterative reweighted method for Tucker decomposition of incomplete tensors,” *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4817–4829, Sep. 2016.

[33] H. Tan, B. Cheng, W. Wang, Y. J. Zhang, and B. Ran, “Tensor completion via a multi-linear low-rank factorization model,” *Neurocomputing*, vol. 133, pp. 161–169, Jun. 2014.

[34] B. Madathil and S. N. George, “Twist tensor total variation regularized-reweighted nuclear norm based tensor completion for video missing area recovery,” *Inf. Sci.*, vol. 423, pp. 376–397, Jan. 2018.

[35] O. Semerci, N. Hao, M. E. Kilmer, and E. L. Miller, “Tensor-based formulation and nuclear norm regularization for multienergy computed tomography,” *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1678–1693, Apr. 2014.

[36] E. Kernfeld, M. Kilmer, and S. Aeron, “Tensor-tensor products with invertible linear transforms,” *Linear Algebra Appl.*, vol. 485, pp. 545–570, 2015.

[37] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, “An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems,” *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 681–695, Mar. 2011.

[38] P. Zhou, C. Lu, J. Feng, Z. Lin, and S. Yan, “Tensor low-rank representation for data recovery and clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Nov. 20, 2019, doi: 10.1109/TPAMI.2019.2954874.

[39] S. Du, Y. Ma, and Y. Ma, “Graph regularized compact low rank representation for subspace clustering,” *Knowl.-Based Syst.*, vol. 118, pp. 56–69, Feb. 2017.

[40] C. Lu, X. Peng, and Y. Wei, “Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms,” in *Proc. IEEE CVPR*, Jun. 2019, pp. 5996–6004.
WENJIN HU (Member, IEEE) received the M.S. and Ph.D. degrees from the Lanzhou University of Technology, in 2008 and 2014, respectively. She is currently an Associate Professor with Northwest Minzu University. Her main research interests include pattern recognition and image processing.

WEILAN WANG received the B.S. degree in mathematics from Northwest Normal University, Lanzhou, China, in 1983. She was a Visiting Scholar with Sun Yat-sen University, Guangzhou, China, in 1987. From 2001 to 2002, she was a Visiting Scholar with Tsinghua University, Beijing, China. From 2006 to 2007, she was a Visiting Scholar with Indiana University, Bloomington, IN, USA. She is currently a Professor with the College of Mathematics and Computer Science, Northwest Minzu University, Lanzhou. Her research interests include image processing, pattern recognition, Tibetan information processing, and so on.

JING LIAN (Member, IEEE) received the B.S. degree in communication engineering and the M.S. degree in signal and information processing from Lanzhou Jiaotong University, Lanzhou, China, in 2005 and 2010, respectively, and the Ph.D. degree from the School of Science and Information, Lanzhou University, Gansu, China, in 2017. He is currently an Associate Professor with the School of Electronic and Information Engineering, Lanzhou Jiaotong University. His current research interests include medical image processing, artificial neural networks, pattern recognition, and computer vision.

***