

Received July 7, 2019, accepted July 26, 2019, date of publication August 12, 2019, date of current version August 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2934482

# Low-Rank Tensor Completion for Image and Video Recovery via Capped Nuclear Norm

XI CHEN<sup>1,2</sup>, JIE LI<sup>1,2</sup>, YUN SONG<sup>1,2</sup>, FENG LI<sup>1,2</sup>, JIANJUN CHEN<sup>3,4</sup>,  
AND KUN YANG<sup>5</sup>, (Senior Member, IEEE)

<sup>1</sup>Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha 410114, China

<sup>2</sup>School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China

<sup>3</sup>National Engineering Laboratory for Information Security Technologies, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

<sup>4</sup>School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100093, China

<sup>5</sup>School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K.

Corresponding author: Yun Song (sonie@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772087, Grant 61504013, and Grant 61303043, in part by the State Scholarship Fund of the China Scholarship Council under Grant 201808430236, in part by the “Double First-Class” International Cooperation and Development Scientific Research Project of the Changsha University of Science and Technology under Grant 2018IC23 and Grant 2018IC25, in part by the Natural Science Foundation of Hunan Province under Grant 2016JJ2005 and Grant 2019JJ50648, in part by the CERNET Innovation Project under Grant NGII20160203, in part by the Outstanding Youth Project of the Hunan Province Education Department under Grant 18B162, in part by the Natural Science Foundation of Hunan Province under Grant 2019JJ50648, and in part by the Open Project of the Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation.

**ABSTRACT** Inspired by the robustness and efficiency of the capped nuclear norm, in this paper, we apply it to 3D tensor applications and propose a novel low-rank tensor completion method via tensor singular value decomposition (t-SVD) for image and video recovery. The proposed tensor capped nuclear norm model (TCNN) handles corrupted low-rank tensors by sparsity enhancement via truncating its partial singular values dynamically. We also develop a simple and efficient algorithm to solve the proposed nonconvex and nonsmooth optimization problem using the Majorization-Minimization (MM) framework. Since the proposed algorithm admits a closed-form solution by optimizing a well-selected approximate version of the original objective function at each iteration, it is very efficient. Experimental results on both synthetic and real-world datasets, clearly demonstrate the superior performance of the proposed method.

**INDEX TERMS** Low-rank tensor completion, tensor singular value decomposition, capped nuclear norm, visual data completion.

## I. INTRODUCTION

The problem of recovering missing elements from partially observed data has attracted widespread attention in many applications, such as collaborative filtering [1], image processing [2]–[4], and video denoising [5], [6]. In some of these applications, the data exhibit strong local or nonlocal inherent similarity and lie in a low-rank structure [7]. Recently, low-rank minimization has proven to be one of the most powerful global constraints for image inverse problems [8]–[10]. Thus, estimating missing values in a partially observed data can be modeled as a low-rank matrix completion problem [11], [12]. Specifically, for a given matrix data  $X \in \mathbb{R}^{I_1 \times I_2}$  of low rank,

The associate editor coordinating the review of this article and approving it for publication was Weizhi Meng.

the matrix completion can be formulated as a constrained rank-minimization problem as follows:

$$\min_{X \in \mathbb{R}^{I_1 \times I_2}} \text{rank}(X), \quad \text{s.t. } y = \mathbf{A}(X), \quad (1)$$

where  $\mathbf{A} : \mathbb{R}^{I_1 \times I_2} \rightarrow \mathbb{R}^m$  is a linear projection operator with  $m \ll I_1 I_2$ . Although the rank function is nonconvex and this problem is NP-hard, the nuclear norm [13], [14] has proven to be a promising surrogate for the rank function to relax this nonconvex problem to a convex problem. Despite the theoretical soundness, the nuclear norm simultaneously penalizes all of the singular values and may make the final solution deviate from the original solution [15], [16]. To more closely approximate the rank function, some nonconvex approaches, e.g., minimax concave

penalty [17], [18], tractable Schatten norms [15], truncated nuclear norm regularization (TNNR) [16] and iterative reweighting algorithms [19], [20], have been proposed. Recently, Sun *et al.* [21] proposed a capped nuclear norm regularization which minimizes the singular values lower than an adaptively learned threshold and yielded a state-of-the-art performance in the robust principal component analysis problem.

Existing studies have mainly focused on developing dynamic matrix completion methods, however, few of them have concentrated on tensors [22] which can be regarded as a high-dimensional extension of matrices. Compared with a matrix, a high-order tensor is a more natural representation of multidimensional data and it can reveal more latent structure information underlying multidimensional data. The tensor completion problem is a natural generalization of the matrix completion and can be formulated as the following rank minimization problem:

$$\min_{\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \dots \times I_N}} \text{rank}(\mathcal{X}), \quad \text{s.t. } y = \mathbf{A}(\mathcal{X}), \quad (2)$$

where  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \dots \times I_N}$  represents the incomplete low-rank tensor and  $\mathbf{A} : \mathbb{R}^{I_1 \times I_2 \times I_3 \times \dots \times I_N} \rightarrow \mathbb{R}^m$  is a linear operator with  $m \ll I_1 I_2 I_3 \dots I_N$ , the order  $N$  is the number of dimensions of the tensor, also known as modes or ways. There is direct evidence suggesting that matrix completion methods can be extended to the tensor case. However, the task is not easy because the definitions of the tensor rank and its convex relaxation become complicated for higher-order tensors [23], [24]. Many previously reported approaches just processed them by unfolding the tensor into matrix. Liu *et al.* [3] first proposed a tensor completion method which tried to minimize the sum of nuclear norms (SNN) of all matrices for an unfolded tensor along each channel. Some studies have revealed that the SNN can characterize the correlations among different channel effectively [25], [26]. However, the computational cost of this approach was high for multiple SVDs. To address these issues, Xu *et al.* [27] extended the matrix factorization method to the tensor case to recover the low-rank tensor. This approach unfolded the tensor data into matrices and then applied the matrix factorization method to each unfolding matrix of the underlying tensor as an alternative of the tensor nuclear norm. This method based on matrix factorization avoided the SVD computational burden of large matrices. However, as noted in [28], unfolding a tensor directly destroyed the multidimensional structure of the original data and led to critical information loss and degraded performance.

Recently, Kilmer and Martin [29] proposed a new tensor decomposition scheme named tensor singular value decomposition (t-SVD) which can be implemented by solving matrix SVDs easily in the Fourier domain. This approach was based on the new definition of tensor-tensor product which enjoyed several properties similar to those of the matrix-matrix product. Related concepts, e.g., tensor tubal rank and tensor multi rank, were also associated with

t-SVD [29], [30]. Based on t-SVD and the tensor tubal rank, Zhang and Aeron [31] defined their tensor nuclear norm (TNN) to replace the tubal rank and solve the low-rank tensor completion problem in Eq. (2). Semerci *et al.* [32] applied this scheme to multi-energy computed tomography (CT) images and achieved promising effects of reconstruction. Lu *et al.* [24], [33] further elaborated the t-SVD factorization and used it to address the tensor robust principal component analysis problem. Note that, all of the t-SVD-based approaches formulated the rank minimization problem by a convex nuclear norm penalty, which treated each singular value equally, and led to a loss of optimality in the representation.

The capped nuclear norm reflects the rank function more accurately than the standard nuclear norm does. In this paper, we propose a novel nonconvex model called the tensor capped nuclear norm (TCNN) which is supposed to capture the hidden information from tensors via capped nuclear norm regularization. The proposed model is defined as the partial sum of singular values of each frontal slice that is obtained in the Fourier domain adaptively. This means the model can characterize the low rankness for each mode. Afterward, we develop a fast algorithm to solve the optimization of the proposed model efficiently. The proposed algorithm is based on the Majorization Minimization (MM) [34], [35] framework and a proximal strategy is applied to guarantee the strict convergence of each subproblem and improve the robustness and stability of the algorithm. In addition, we adopt a new reduced version of t-SVD to address the bottleneck of full SVD computations. Experimental results on synthetic data and real-world data show that the proposed method outperforms existing competing methods.

## II. NOTATIONS AND PRELIMINARIES

In this section, we introduce some notations and definitions used throughout this paper [29], [33]. For brevity, we summarize the main notations in Table.1. The complex conjugate of  $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$  is denoted as  $\text{conj}(\mathcal{A})$ , which takes the complex conjugate of all entries of  $\mathcal{A}$ . The conjugate transpose of a tensor  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is defined as  $\mathcal{A}^* \in \mathbb{R}^{n_2 \times n_1 \times n_3}$ . The transpose is obtained by conjugate transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through  $n_3$ . The inner product between  $A \in \mathbb{C}^{n_1 \times n_2}$  and  $B \in \mathbb{C}^{n_1 \times n_2}$  is defined as  $\langle A, B \rangle = \text{tr}(A^*B)$ , where  $A^*$  is the conjugate transpose of  $A$  and  $\text{tr}(\cdot)$  denotes the trace function.  $\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{i=1}^{n_3} \langle A^{(i)}, B^{(i)} \rangle$  represents the inner product of tensor  $\mathcal{A}$  and  $\mathcal{B}$  in  $\mathbb{C}^{n_1 \times n_2 \times n_3}$ . The identity tensor  $\mathcal{I} \in \mathbb{R}^{n \times n \times n_3}$  is a tensor whose first frontal slice  $A^{(1)}$  is the  $n \times n$  identity matrix and whose other frontal slices are all zeros.

For  $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ , using the MATLAB command `fft`, let  $\bar{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3)$  represent the result of the discrete Fourier transformation (DFT) of  $\mathcal{A}$  along the third dimension. Likewise, we can compute  $\mathcal{A} = \text{ifft}(\bar{\mathcal{A}}, [], 3)$  through the inverse DFT. We define  $\bar{A} \in \mathbb{R}^{n_1 \times n_3 \times n_2 n_3}$  as a block diagonal matrix, where the  $i$ -th block  $\bar{A}^{(i)}$  on the diagonal corresponds

TABLE 1. Summary of main notations in the paper.

$\mathcal{A}$	A tensor	$\mathbf{a}$	A vector
$A$	A matrix	$a$	A scalar
$\mathcal{A}(i, :, :)$	The $i$ -th horizontal slice of $\mathcal{A}$ .	$\text{conj}(\mathcal{A})$	The complex conjugate of $\mathcal{A}$
$\mathcal{A}(:, i, :)$	The $i$ -th lateral slice of $\mathcal{A}$ .	$\mathcal{A}^*$	The conjugate transpose of $\mathcal{A}$
$\mathcal{A}(:, :, i)$	The $i$ -th frontal slice of $\mathcal{A}$ .	$\bar{\mathcal{A}}$	The DFT of $\mathcal{A}$
$A^{(i)}$	$A^{(i)} = \mathcal{A}(:, :, i)$ .	$\ \mathcal{A}\ _1$	$\ \mathcal{A}\ _1 = \sum_{ijk}  \mathcal{A}_{ijk} $
$\mathcal{A}_{ijk}$	The $(i, j, k)$ -th entry of $\mathcal{A}$ .	$\ \mathcal{A}\ _\infty$	$\ \mathcal{A}\ _\infty = \max_{ijk}  \mathcal{A}_{ijk} $
$\mathcal{I}$	The identity tensor.	$\ \mathcal{A}\ _F$	$\ \mathcal{A}\ _F = \sqrt{\sum_{ijk}  \mathcal{A}_{ijk} ^2}$
$\sigma_i(A)$	The $i$ -th largest singular value of $A$ .	$\mathbb{R}$	The field of real number.
$\ \mathcal{A}\ _*$	$\ \mathcal{A}\ _* = \sum_i \sigma_i(A)$	$\mathbb{C}$	The field of complex number.
$\lceil t \rceil$	The nearest integer greater than or equal to $t$	$\lfloor t \rfloor$	The nearest integer less than or equal to $t$

to the  $i$ -th frontal slice of  $\bar{\mathcal{A}}$ , i.e.,

$$\bar{A} = \text{bdiag}(\bar{\mathcal{A}}) = \begin{bmatrix} \bar{A}^{(1)} & & & \\ & \bar{A}^{(2)} & & \\ & & \ddots & \\ & & & \bar{A}^{(n_3)} \end{bmatrix}. \quad (3)$$

The block circulant matrix  $\text{bcirc}(\mathcal{A}) \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}$  of tensor  $\mathcal{A}$  is defined as

$$\text{bcirc}(\mathcal{A}) = \begin{bmatrix} A^{(1)} & A^{(n_3)} & \dots & A^{(2)} \\ A^{(2)} & A^{(1)} & \dots & A^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ A^{(n_3)} & A^{(n_3-1)} & \dots & A^{(1)} \end{bmatrix}. \quad (4)$$

Here, a pair of mapping operators is defined by

$$\text{unfold}(\mathcal{A}) = \begin{bmatrix} A^{(1)} \\ A^{(2)} \\ \vdots \\ A^{(n_3)} \end{bmatrix}, \quad \text{fold}(\text{unfold}(\mathcal{A})) = \mathcal{A}, \quad (5)$$

In particular, the block circulant matrix can be block diagonalized in the Fourier domain, i.e.,

$$(F_{n_3} \otimes I_{n_1}) \cdot \text{bcirc}(\mathcal{A}) \cdot (F_{n_3}^{-1} \otimes I_{n_2}) = \bar{\mathcal{A}}, \quad (6)$$

where  $F_{n_3} \in \mathbb{C}^{n_3 \times n_3}$  denotes the discrete Fourier transform matrix and  $\otimes$  denotes the Kronecker product. In addition, the frontal slices  $\bar{A}$  has the following properties:

$$\begin{cases} \bar{A}^{(1)} \in \mathbb{R}^{n_1 \times n_2}, \\ \text{conj}(\bar{A}^{(i)}) = \bar{A}^{(n_3+2-i)}, \quad i = 2, \dots, \lfloor \frac{n_3+1}{2} \rfloor. \end{cases} \quad (7)$$

$$\|\mathcal{A}\|_F = \frac{1}{\sqrt{n_3}} \|\bar{\mathcal{A}}\|_F. \quad (8)$$

Next some necessary concepts related to this work are as follows:

**Definition 1 (t-Product)** [29], [33]: Let  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  and  $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ , the t-product  $\mathcal{A} * \mathcal{B}$  is defined as a 3-way tensor of size  $n_1 \times n_4 \times n_3$ ,

$$\mathcal{C} = \mathcal{A} * \mathcal{B} = \text{fold}(\text{bcirc}(\mathcal{A}) \cdot \text{unfold}(\mathcal{B})). \quad (9)$$

The t-product is analogous to matrix multiplication; the only difference between them is that the circular convolution replaces the product operation between elements, which are now tubes [29]. By using (7), Lu et al. [33] recently proposed a more efficient method for computing t-product compared with (9).

**Definition 2 (F-Diagonal Tensor)**: [29] Tensor  $\mathcal{A}$  is called f-diagonal if each of its frontal slices is a diagonal matrix.

**Definition 3 (Orthogonal Tensor)**: [29] Tensor  $\mathcal{Q} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is orthogonal if  $\mathcal{Q}^* * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^* = \mathcal{I}$ .

**Theorem 1 (t-SVD)** [29], [33]: Let  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ . The t-SVD of  $\mathcal{A}$  is given by

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*, \quad (10)$$

where  $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$  and  $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$  are both orthogonal tensors.  $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is a f-diagonal tensor, and  $*$  denotes the t-product. Kilmer and Martin [29] first defined the t-SVD and showed that this decomposition can be obtained by computing matrix SVDs in the Fourier domain. That is, let  $\bar{A}^{(i)} = \bar{U}^{(i)} \bar{S}^{(i)} (\bar{V}^{(i)})^*$  ( $i = 1, \dots, n_3$ ) be the full SVD of each frontal slice of  $\bar{\mathcal{A}}$ . Or briefly,  $\bar{\mathcal{A}} = \bar{U} \bar{S} \bar{V}^*$ . Inverse fft function is then performed along the third dimension, i.e.,  $\mathcal{U} = \text{ifft}(\bar{\mathcal{U}}, [1, 3])$ ,  $\mathcal{S} = \text{ifft}(\bar{\mathcal{S}}, [1, 3])$ ,  $\mathcal{V} = \text{ifft}(\bar{\mathcal{V}}, [1, 3])$ . However this decomposition requires the computation of the full SVD for all frontal slices of the tensor. Using property (7), Lu et al. [33] avoided this issue and further proposed a more efficient way of computing t-SVD. Note that the method reported in [29] must compute  $n_3$  matrix SVDs, while the new skinny t-SVD [33] reduces this number to  $\lfloor \frac{n_3+1}{2} \rfloor$ . This reduction significantly decreases the cost of t-SVD computing when  $n_3$  is large.

**Definition 4 (Tensor Tubal Rank)** [33]: For  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , the corresponding t-SVD is  $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*$ . The inverse DFT gives  $\mathcal{S}(i, i, 1) = \frac{1}{n_3} \sum_{j=1}^{n_3} \bar{S}(i, i, j)$ , and the singular values of  $\bar{\mathcal{A}}(:, :, j)$  are the entries on the diagonal of  $\bar{S}(:, :, j)$ . Based on this property, the tensor tubal rank, which is the number of nonzero singular tubes of  $\mathcal{S}$ , is defined as,

$$\text{rank}_t(\mathcal{A}) = \#\{i, \mathcal{S}(i, i, :) \neq 0\} = \#\{i, \mathcal{S}(i, i, 1) \neq 0\}.$$

Hence, the number of nonzero entries of first frontal slice  $\mathcal{S}(i, i, 1)$  is equivalent to the tensor tubal rank, and the entries on the diagonal  $\mathcal{S}(i, i, 1)$  are called the singular values of the tensor  $\mathcal{A}$ .

**Remark 1**: Suppose  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  with tubal rank  $r$ , it has a further reduced version of t-SVD, i.e.,  $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*$ , where  $\mathcal{U} \in \mathbb{R}^{n_1 \times r \times n_3}$  and  $\mathcal{V} \in \mathbb{R}^{n_2 \times r \times n_3}$  satisfying  $\mathcal{U}^* * \mathcal{U} = \mathcal{I}$  and  $\mathcal{V}^* * \mathcal{V} = \mathcal{I}$ ,  $\mathcal{S} \in \mathbb{R}^{r \times r \times n_3}$  is a f-diagonal tensor. The reduced version of t-SVD is faster and more economical for storage. We thus use this version throughout this paper.

**Definition 5 (Tensor Nuclear Norm)** [33]: The tensor nuclear norm of  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is defined as the sum of

the tensor singular values, i.e.,  $\|\mathcal{A}\|_* = \sum_{i=1}^r \sigma_i(\mathcal{A}) = \sum_{i=1}^r \mathcal{S}(i, i, 1)$ , where  $r = \text{rank}_t(\mathcal{A})$ ,  $\mathcal{S}$  is from the t-SVD of  $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*$  and  $\sigma_i(\mathcal{A})$  is the  $i$ -th singular value of  $\mathcal{A}$ . Note that the tensor nuclear norm is only determined by the first frontal slice  $\mathcal{S}(i, i, 1)$ , which is different from works [24], [31], [36].

*Remark 2:* The proposed tensor nuclear norm is based on t-SVD, Lu et al. [24] also defined the tensor nuclear norm as  $\frac{1}{n_3} \|\bar{\mathcal{A}}\|_*$ . The relationship between the two is indicated as follows:

$$\begin{aligned} \|\mathcal{A}\|_* &= \sum_{i=1}^r \mathcal{S}(i, i, 1) = \langle \mathcal{S}, \mathcal{I} \rangle = \frac{1}{n_3} \langle \bar{\mathcal{S}}, \bar{\mathcal{I}} \rangle \\ &= \frac{1}{n_3} \langle \bar{\mathcal{S}}, \bar{\mathcal{I}} \rangle = \frac{1}{n_3} \sum_{j=1}^{n_3} \|\bar{\mathcal{A}}^{(j)}\|_* = \frac{1}{n_3} \|\bar{\mathcal{A}}\|_*. \end{aligned}$$

where  $\frac{1}{n_3} = \frac{1}{\|F_{n_3}\|_F^2}$  and  $F_{n_3}$  denotes the DFT matrix.

### III. TENSOR CAPPED NUCLEAR NORM

#### A. PROBLEM FORMULATION

In the matrix case, given the matrix  $X \in \mathbb{R}^{m \times n}$ , and parameter  $\theta > 0$ , the capped nuclear norm can be represented as  $T(X) = \|X\|_\theta = \sum_{i=1}^{\min(m,n)} \min(\sigma_i(X), \theta)$ . The capped nuclear norm shrinks the singular values that are below a threshold adaptively learned in the optimization. Accordingly, the noisy reference information (corresponds to smallest singular values) can be filtered out; in this way, the capped nuclear norm regularization is robust and stable in reality applications. Thus, it is expected that the capped nuclear norm can be applied for the tensor completion to achieve better performance. Let  $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*$  be the t-SVD of  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ . Using Definition 5 and Remark 2, the tensor capped nuclear norm can be defined as follows:

$$\begin{aligned} \|\mathcal{X}\|_\theta &= \frac{1}{n_3} \|\bar{\mathcal{X}}\|_\theta = \frac{1}{n_3} \sum_{j=1}^{n_3} \|\bar{\mathcal{X}}^{(j)}\|_\theta \\ &= \frac{1}{n_3} \sum_{j=1}^{n_3} \sum_{i=1}^p \min(\sigma_i(\bar{\mathcal{X}}^{(j)}), \theta), \end{aligned} \quad (11)$$

where,  $p = \min(n_1, n_2)$  and  $\theta$  is the truncated threshold parameter. The above definition holds since  $\mathcal{S}(i, i, 1) = \frac{1}{n_3} \sum_{j=1}^{n_3} \bar{\mathcal{S}}(i, i, j)$  and the  $i$ -th singular value  $\sigma_i(\bar{\mathcal{X}}^{(j)})$  is the  $i$ -th entry on the diagonal of  $\bar{\mathcal{S}}(:, :, j)$  [33]. Thus such a definition is similar with the matrix capped nuclear norm.

In this paper, we employ the tensor capped nuclear norm regularization to address the low-rank tensor completion problem. Given a 3-way tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , with the tensor capped nuclear norm, the minimization model can be formulated as follows:

$$\min_{\mathcal{X}} \|\mathcal{X}\|_\theta, \quad \text{s.t. } P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M}). \quad (12)$$

We then relax the constrained problem (12) as follows:

$$\arg \min_{\mathcal{X}} F(\mathcal{X}) = \lambda L(\mathcal{X}) + Q(\mathcal{X}). \quad (13)$$

where  $\lambda > 0$  is a given penalty parameter,  $Q(\mathcal{X}) = \frac{1}{2} \|P_\Omega(\mathcal{X}) - P_\Omega(\mathcal{M})\|_F^2$  and  $L(\mathcal{X}) = \|\mathcal{X}\|_\theta$ . Because the objective function is nonconvex and  $\|\mathcal{X}\|_\theta$  is a concave function [21], it is not easy to solve (13); therefore, we relax the problem to a surrogate and adopt the MM method to optimize the above mentioned model.

We further introduce several useful tools [37]–[39] that will be used later. The details of the nonconvex approximation method are presented in the next subsection.

*Definition 6:* For a matrix  $X \in \mathbb{R}^{n_1 \times n_2}$ . Note that the capped nuclear norm  $T(X)$  of  $X$  has no derivative at  $\sigma_i(X) = \theta$ ; instead, we define the approximate gradient as follows:

$$\partial T(X) = \begin{cases} 1, & \text{if } \sigma_i(X) \leq \theta \\ 0, & \text{if } \sigma_i(X) > \theta. \end{cases} \quad (14)$$

where  $i = 1, \dots, \min(n_1, n_2)$ . Moreover, due to its concave property, given matrices  $X$  and  $Y$ , we have

$$T(X) \leq T(Y) + \langle \partial T(Y), \sigma(X) - \sigma(Y) \rangle. \quad (15)$$

*Theorem 2:* For any  $\lambda > 0$ , let  $X = USV^T$  be the SVD of  $X \in \mathbb{R}^{m \times n}$ ,  $S = \text{diag}(\{\sigma_i\}_{1 \leq i \leq \min(m,n)})$  and  $0 \leq w_1 \leq w_2 \leq \dots \leq w_p$  ( $p = \min(m, n)$ ), the weighted singular value thresholding (WSVT) operator  $D_{\lambda, w}$ , is defined as follows:

$$D_{\lambda, w}(X) = UD_{\lambda, w}(S)V^T, \quad (16)$$

$$D_{\lambda, w}(S) = \text{diag}(\max\{\sigma_i - \lambda w_i, 0\}). \quad (17)$$

For  $Y \in \mathbb{R}^{m \times n}$ , we have a global optimal solution of the following optimization problem:

$$D_{\lambda, w}(Y) = \arg \min_X \frac{1}{2} \|X - Y\|_F^2 + \lambda \sum_{i=1}^p w_i \sigma_i(X). \quad (18)$$

#### B. OPTIMIZATION WITH MM

In this section, we detail the MM algorithm framework for solving the problem (13). Simply, in an optimization problem, a successful MM algorithm substitutes a new simple surrogate function for the original intractable objective function. The key to MM algorithm is to construct an easy to tackle surrogate function based on the output of each iteration. Here, we first construct the desired surrogate (upper bound) function for the problem and then propose an efficient optimization method to iteratively solve the surrogate function.

In each iteration, we impute the current missing tensor data  $\mathcal{X}_k$  and then apply proximal minimization to iteratively obtain the final result.

Given  $\mathcal{X}_k$ , the surrogate function for the  $Q(\mathcal{X})$  can be defined as follows:

$$G(\mathcal{X} | \mathcal{X}_k) = \frac{1}{2} \|\hat{\mathcal{X}} - \mathcal{X}\|_F^2, \quad (19)$$

where  $\hat{\mathcal{X}} = P_\Omega(\mathcal{M}) + P_{\Omega^c}(\mathcal{X}_k)$  and  $P_{\Omega^c}$  is a linear operator that sets the entries in  $\Omega^c$  (i.e., outside  $\Omega$ ) to zero. Note that

$$\begin{aligned} Q(\mathcal{X}) &= \frac{1}{2} \|P_\Omega(\mathcal{M}) - P_\Omega(\mathcal{X})\|_F^2 \\ &\leq \frac{1}{2} \|\hat{\mathcal{X}} - \mathcal{X}\|_F^2 = G(\mathcal{X} | \mathcal{X}_k). \end{aligned} \quad (20)$$

For  $L(\mathcal{X})$ , one general method is to use the first-order approximation to substitute its nonconvex function, although there is no first-order approximation at  $\mathcal{X}_k$ , we can use its approximate gradient of each frontal slice based on Definition 6. Suppose that the t-SVD of  $\mathcal{X}_k$  is  $\mathcal{X}_k = \mathcal{U}_k * \mathcal{S}_k * \mathcal{V}_k^*$ . For  $j$ -th ( $j = 1, \dots, n_3$ ) frontal slice  $\bar{S}_k^{(j)}$  of  $\bar{\mathcal{S}}_k$  (not  $\mathcal{S}_k$ ), assume that there are  $r$  singular values of  $\bar{S}_k^{(j)}$  greater than  $\theta$ . By using Definition 6, we can conclude that  $\langle \partial T(\bar{X}_k^{(j)}), \sigma(\bar{X}) \rangle = \sum_{i=r+1}^p \sigma_i(\bar{X}^{(j)})$ , where the entries  $\sigma(\bar{X}^{(j)})$  on the diagonal of  $\bar{S}^{(j)}$  are the singular values of the  $j$ -th frontal slice  $\bar{X}^{(j)}$ ; therefore, for each frontal slice of  $\mathcal{X}$ , we have

$$T(\bar{X}^{(j)}) \leq T(\bar{X}_k^{(j)}) + \langle \partial T(\bar{X}_k^{(j)}), \sigma(\bar{X}^{(j)}) - \sigma(\bar{X}_k^{(j)}) \rangle = \sum_{i=r+1}^p \sigma_i(\bar{X}^{(j)}) + D_k. \quad (21)$$

where  $D_k = T(\bar{X}_k^{(j)}) - \langle \partial T(\bar{X}_k^{(j)}), \sigma(\bar{X}_k^{(j)}) \rangle$  is a constant given  $\mathcal{X}_k$ . Thus, the corresponding surrogate function of  $L(\mathcal{X})$  can be approximated as

$$H(\mathcal{X} | \mathcal{X}_k) = \frac{1}{n_3} \sum_{j=1}^{n_3} T(\bar{X}^{(j)}) = \frac{1}{n_3} \sum_{j=1}^{n_3} \sum_{i=r+1}^p \sigma_i(\bar{X}^{(j)}) + C_k = \sum_{i=r+1}^p \sigma_i(\mathcal{X}) + C_k. \quad (22)$$

where  $C_k = \frac{1}{n_3} \sum_{j=1}^{n_3} D_k$  is a constant given  $\mathcal{X}_k$ , and  $\sigma_i(\mathcal{X})$  is the  $i$ -th singular values of  $\mathcal{X}$ . Consequently, the surrogate (upper bound) function of  $F(\mathcal{X})$  takes the form

$$J(\mathcal{X} | \mathcal{X}_k) = \lambda H(\mathcal{X} | \mathcal{X}_k) + G(\mathcal{X} | \mathcal{X}_k). \quad (23)$$

Combining (20), (21) and (22), it is clear that we have  $F(\mathcal{X}) \leq J(\mathcal{X} | \mathcal{X}_k)$  such that  $F(\mathcal{X}) = J(\mathcal{X} | \mathcal{X}_k)$  if and only if  $\mathcal{X} = \mathcal{X}_k$ , which implies that  $J(\mathcal{X} | \mathcal{X}_k)$  is a global upper bound for  $F(\mathcal{X})$  and meets the requirements of MM. This condition ensures that when we optimize the surrogate function, we are constantly optimizing the original objective function. Thus  $J(\mathcal{X} | \mathcal{X}_k)$  is a suitable surrogate of  $F(\mathcal{X})$ . Because the traditional MM method only ensures non-increment of the original objective function, there is no convergence guarantee for  $F(\mathcal{X})$ . To address this issue, we further add the proximal term  $\|\mathcal{X} - \mathcal{X}_k\|_F^2$  to the surrogate function (23) to ensure the convergence of the algorithm. Thus, we can reformulate problem (23) as follows

$$\begin{aligned} \mathcal{X}_{k+1} &= \arg \min_{\mathcal{X}} J(\mathcal{X} | \mathcal{X}_k) + \frac{\mu}{2} \|\mathcal{X} - \mathcal{X}_k\|_F^2 \\ &= \arg \min_{\mathcal{X}} \frac{1}{2} \|\hat{\mathcal{X}} - \mathcal{X}\|_F^2 + \frac{\mu}{2} \|\mathcal{X} - \mathcal{X}_k\|_F^2 \\ &\quad + \lambda \sum_{i=r+1}^p \sigma_i(\mathcal{X}) + \lambda C_k. \end{aligned} \quad (24)$$

where  $\mu$  is a scalar proximal parameter. Discarding the constant terms, minimizing (24) amounts to solving

$$\mathcal{X}_{k+1} = \arg \min_{\mathcal{X}} \frac{1}{2} \|\mathcal{X} - \frac{1}{\mu+1} (\hat{\mathcal{X}} + \mu \mathcal{X}_k)\|_F^2 + \frac{\lambda}{\mu+1} \|\mathcal{X}\|_r. \quad (25)$$

By using property (8) and Remark 2, denote  $\mathcal{Y} = \frac{1}{1+\mu} (\hat{\mathcal{X}} + \mu \mathcal{X}_k)$ , problem (25) is equivalent to

$$\begin{aligned} \mathcal{X}_{k+1} &= \arg \min_{\mathcal{X}} \frac{1}{n_3} \left( \frac{1}{2} \|\bar{\mathcal{X}} - \bar{\mathcal{Y}}\|_F^2 + \frac{\lambda}{1+\mu} \|\bar{\mathcal{X}}\|_r \right) \\ &= \arg \min_{\mathcal{X}} \frac{1}{n_3} \sum_{j=1}^{n_3} \left( \frac{1}{2} \|\bar{X}^{(j)} - \bar{Y}^{(j)}\|_F^2 + \frac{\lambda}{1+\mu} \|\bar{X}^{(j)}\|_r \right). \end{aligned} \quad (26)$$

where  $\|\bar{X}^{(j)}\|_r = \sum_{i=r+1}^p \sigma_i(\bar{X}^{(j)})$ . From Theorem 2, we then let the weights  $w_i = \partial T(\bar{X}^{(j)})$  ( $i = 1, \dots, p$ ), and we can obtain  $w_1 = 0, \dots, w_r = 0, w_{r+1} = 1, \dots, w_p = 1$ . It can be seen that the problem (25) is a special case of (18) and the  $j$ -th subproblem of (26) can be solved by the WSVT. Now we first introduce our defined tensor capped singular value thresholding operator and then show that problem (25) also has a real closed-form solution, similar to the proximal operator for the matrix case.

Recently, Lu et al. [33] proposed tensor singular value thresholding (t-SVT), which is a natural extension of the matrix SVT. Let  $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*$  be the t-SVD of  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ ,  $p = \min(n_1, n_2)$ , for each  $r$  ( $0 < r < p$ ) and a fixed threshold  $\tau > 0$ . Based on t-SVT. We thus define the tensor capped Singular Value Thresholding (t-CSVT) operator as follows

$$\mathcal{D}_{\tau,r}(\mathcal{Y}) = \mathcal{U} * \mathcal{D}_{\tau,r}(\mathcal{S}) * \mathcal{V}^* \quad (27)$$

where  $\mathcal{D}_{\tau,r}(\mathcal{S}) = \text{fft}(\text{diag}(\bar{\mathcal{S}}_{ii}), [], 3)$  and  $\bar{\mathcal{S}}_{ii}$  is

$$\bar{\mathcal{S}}_{ii} = \begin{cases} \bar{\mathcal{S}}_{ii}, & \text{if } i \leq r \\ (\bar{\mathcal{S}}_{ii} - \tau)_+, & \text{if } i > r. \end{cases} \quad (28)$$

That is, in the Fourier domain, this operator simply applies the shrinkage operation to the singular values  $\bar{\mathcal{S}}$  of the frontal slices of  $\mathcal{X}$ . Unlike t-SVT, which shrinks all singular values to zero, t-CSVT only shrinks the singular values with indices greater than  $r$ . The detail procedure of t-CSVT is shown in Algorithm 1. Further, we have the following theorem based on Theorem 2.

**Theorem 3:** For any  $\tau > 0$  and tensor  $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ ,  $p = \min(n_1, n_2)$ ,  $0 < r < p$ , the t-CSVT operator (27) obeys:

$$\mathcal{D}_{\tau,r}(\mathcal{Y}) = \arg \min_{\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}} \frac{1}{2} \|\mathcal{X} - \mathcal{Y}\|_F^2 + \tau \|\mathcal{X}\|_r. \quad (29)$$

Let  $\mathcal{Y} = \mathcal{U} * \mathcal{S} * \mathcal{V}^*$  be the t-SVD of  $\mathcal{Y}$ . Note that the entries of  $\mathcal{S}$  are real. By using property (7) and Lemma 2.1 in [33], we can conclude that  $\mathcal{D}_{\tau,r}(\mathcal{S})$  is real and so do the  $\mathcal{D}_{\tau,r}(\mathcal{Y})$ . Secondly, by using Theorem 2, we know that the  $j$ -th frontal slice of  $\mathcal{D}_{\tau,r}(\mathcal{Y})$  is a global real closed-form solution of the  $j$ -th subproblem of (29). Hence, problem (29) has a global solution  $\mathcal{D}_{\tau,r}(\mathcal{Y})$ .

With t-CSVT, at each iteration, problem (25) applies this shrinkage operation to  $\frac{1}{1+\mu} \hat{\mathcal{X}} + \mu \mathcal{X}_k$  to obtain the update  $\mathcal{X}_{k+1}$ . Because all the processes are carried out in the Fourier domain, by using the inverse Fourier transform operation along the 3rd dimension, we can iteratively obtain the optimized solution of the objective function. The main step in the

**Algorithm 1** Tensor Capped Singular Value Thresholding (t-CSVT)

---

**Require:**  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ ,  $p = \min(n_1, n_2)$ , truncation parameter  $r$  ( $0 < r < p$ ),  $\tau > 0$ .

**Ensure:**  $\mathcal{D}_{\tau,r}(\mathcal{X})$  as defined in (26).

- 1: Compute  $\tilde{\mathcal{X}} = \text{fft}(\mathcal{X}, [], 3)$ .
- 2: Perform matrix SVT on each frontal slice of  $\tilde{\mathcal{X}}$  by
- 3: **for**  $j = 1, \dots, \lceil \frac{n_3+1}{2} \rceil$  **do**  
 $[U, S, V] = \text{SVD}(\tilde{\mathcal{X}}^{(j)})$ ;
- 4: **for**  $k = 1, \dots, p$  **do**
- 5:   If  $k > r$   
 $S_{kk} = (S_{kk} - \tau)_+$ ;
- 6: **end for**
- 7:  $S = \text{diag}(S_{kk})$ ;
- 8: Compute  $\tilde{W}^{(j)} = U \cdot S \cdot V^*$ ;
- 9: **end for**
- 10: **for**  $j = \lceil \frac{n_3+1}{2} \rceil, \dots, n_3$  **do**
- 11:    $\tilde{W}^{(j)} = \text{conj}(\tilde{W}^{(n_3-j+2)})$ ;
- 12: **end for**
- 13: **return**  $\mathcal{D}_{\tau,r}(\mathcal{X}) = \text{ifft}(\tilde{\mathcal{W}}, [], 3)$ .

---

**Algorithm 2** Tensor Capped Nuclear Norm for Low-Rank Tensor Completion

---

**Input:** the original incomplete data  $\mathcal{M}$ , indices of known elements  $\Omega$  and unknown elements  $\Omega^c$ , regularization parameter  $\lambda$ , proximal parameter  $\mu$ , threshold parameter  $\theta$ ;

**Initialize:**  $\hat{\mathcal{X}}_1 = P_{\Omega}(\mathcal{M}) + P_{\Omega^c}(0)$ , the zero vector  $\mathbf{r} = \mathbf{0}$ ,  $\mu = 1e-2$ ,  $\varepsilon = 1e-5$ ,  $k = 0$ .

**while not converged do**

1. Update  $\mathcal{X}_{k+1}$  with t-CSVT.  
 $\mathcal{X}_{k+1} = \arg \min_{\mathcal{X}} \frac{1}{2} \|\mathcal{X} - \frac{1}{1+\mu}(\hat{\mathcal{X}} + \mu \mathcal{X}_k)\|_F^2 + \frac{\lambda}{1+\mu} \|\mathcal{X}\|_r$ ;
2. Update  $\mathbf{r}$  along with each frontal slices of  $\mathcal{X}_{k+1}$   
**for**  $j = 1, \dots, \lceil \frac{n_3+1}{2} \rceil$  **do**  
 $\mathbf{r}(j) = \sum_{i=1}^{\min(n_1, n_2)} I(\sigma_i(X_{k+1}^{(j)}) > \theta)$ ;
- end**
3. Update  $\hat{\mathcal{X}}_{k+1}$  by  $\hat{\mathcal{X}}_{k+1} = P_{\Omega}(\mathcal{M}) + P_{\Omega^c}(\mathcal{X}_{k+1})$ ;
4. Check the convergence conditions:  
 $\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F \leq \varepsilon$ ,  $\|\hat{\mathcal{X}}_{k+1} - \hat{\mathcal{X}}_k\|_F \leq \varepsilon$ ;
5.  $k = k + 1$ .

**end**

Output: the recovered tensor.

---

proposed method is depicted in Algorithm 2. The truncation parameter  $r$  of  $j$ -th frontal slice of  $\tilde{\mathcal{X}}$  corresponding to the  $j$ -th entry  $\mathbf{r}(j)$  in vector  $\mathbf{r}$ , which is dynamically updated in each iteration. Note that step 2 of Algorithm 2, i.e., the calculation of the truncation parameter  $\mathbf{r}(j)$ , can be obtained in step 1; all that is required is to apply the inverse DFT to  $S$  and we can count the number of each entry  $\mathbf{r}(j)$  in the original domain. Unlike many other studies [3], [28], [31], [40],

the alternating direction method of multipliers (ADMM) framework is applied to solve the tensor completion problem. Our new method is formulated via the Majorize-Minimization approach, in which each iteration has a closed-form solution and convergence is guaranteed.

**IV. CONVERGENCE ANALYSIS**

In this section, the convergence of the proposed method is analyzed. We first demonstrate that the objective function (25) decreases sufficiently and then prove, accordingly, that any limit point of Algorithm 2 is a stationary point of problem (25). We first recall certain definitions [34] adopted in later proofs.

*Definition 7:* Let  $f : D \rightarrow \mathbb{R}$  be a function where  $D \in \mathbb{R}^m$  is an open set. The directional derivative of  $f$  at point  $x$  in the feasible direction  $\Delta$  is defined by

$$f'(x; \Delta) \triangleq \liminf_{\alpha \downarrow 0} \frac{f(x + \alpha \Delta) - f(x)}{\alpha}. \quad (30)$$

A point  $x$  is a (minimizing) stationary point of  $f$  if  $f'(x; \Delta) \geq 0$  for all  $\Delta$  such that  $x + \Delta \in D$ .

*Definition 8:* A function  $f$  is said to decrease sufficiently on the sequence  $\{x_k\}$  if there exists a constant  $\alpha > 0$  such that

$$f(x_k) - f(x_{k+1}) \geq \alpha \|x_k - x_{k+1}\|^2, \quad \forall k \quad (31)$$

According to the above mentioned definitions, we have the following theorems.

*Theorem 4:* Denote  $\mathcal{D}$  as the set of solutions generated by Algorithm 2, and the sequence  $\{\mathcal{X}_k\} \subseteq \mathcal{D}$  satisfies the following properties:

- 1)  $F(\mathcal{X})$  has a sufficient decrease in the sequence  $\{\mathcal{X}_k\}$ .  
Indeed,

$$F(\mathcal{X}_k) - F(\mathcal{X}_{k+1}) \geq \frac{\mu}{2} \|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F^2. \quad (32)$$

- 2)  $\lim_{k \rightarrow \infty} (\mathcal{X}_{k+1} - \mathcal{X}_k) = 0$ .
- 3) The sequence  $\{\mathcal{X}_k\}$  is bounded.

*Proof:* Since  $\mathcal{X}_{k+1}$  is optimal to (23), we have

$$J(\mathcal{X} | \mathcal{X}_{k+1}) + \frac{\mu}{2} \|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F^2 \leq J(\mathcal{X} | \mathcal{X}_k) + \frac{\mu}{2} \|\mathcal{X}_k - \mathcal{X}_k\|_F^2. \quad (33)$$

This implies that  $J(\mathcal{X}_k | \mathcal{X}_k) - J(\mathcal{X}_{k+1} | \mathcal{X}_k) \geq \frac{\mu}{2} \|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F^2$ . Combining with  $F(\mathcal{X}_k) = J(\mathcal{X}_k | \mathcal{X}_k)$  and  $F(\mathcal{X}) \leq J(\mathcal{X} | \mathcal{X}_k)$ , we have

$$F(\mathcal{X}_k) - F(\mathcal{X}_{k+1}) \geq J(\mathcal{X}_k | \mathcal{X}_k) - J(\mathcal{X}_{k+1} | \mathcal{X}_k) \geq \frac{\mu}{2} \|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F^2. \quad (34)$$

Thus  $F(\mathcal{X})$  shows a sufficient decrease.

Summing the above mentioned inequalities in (34) for  $k \geq 1$ , we have

$$F(\mathcal{X}_1) \geq \sum_{k=1}^{\infty} \frac{\mu}{2} \|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F^2. \quad (35)$$

Based on the positive definiteness of  $\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F^2$ , we can infer that  $\lim_{k \rightarrow \infty} (\mathcal{X}_{k+1} - \mathcal{X}_k) = 0$ .

Furthermore, as  $F(\mathcal{X}) \rightarrow \infty$  if and only if  $\mathcal{X} \rightarrow \infty$ , we can conclude that the sequence  $\{\mathcal{X}_k\}$  is bounded.

*Theorem 5:* Denote  $\mathcal{D}$  as the set of solutions generated by Algorithm 2, the sequence  $\{\mathcal{X}_k\} \subseteq \mathcal{D}$ . Assume that  $Q(\mathcal{X}|\mathcal{X}_k) = J(\mathcal{X}|\mathcal{X}_k) + \frac{\mu}{2}\|\mathcal{X} - \mathcal{X}_k\|_F^2$ ; the approximation function  $Q(\mathcal{X}|\mathcal{X}_k)$  satisfies the following:

$$Q'(\mathcal{X}|\mathcal{X}_k; \Lambda)|_{\mathcal{X}=\mathcal{X}_k} = F'(\mathcal{X}_k; \Lambda) \quad \forall \Lambda \quad \text{with } \mathcal{X}_k + \Lambda \in \mathcal{D}. \quad (36)$$

*Proof:* Together with  $F(\mathcal{X}_k) = J(\mathcal{X}_k|\mathcal{X}_k)$  and  $F(\mathcal{X}) \leq J(\mathcal{X}|\mathcal{X}_k)$ , for any point  $\mathcal{X} \in \mathcal{D}$ , we can easily obtain

$$\begin{aligned} F(\mathcal{X}_k) &= Q(\mathcal{X}_k|\mathcal{X}_k). \\ F(\mathcal{X}) &\leq Q(\mathcal{X}|\mathcal{X}_k). \end{aligned} \quad (37)$$

Not that for any fixed  $\mathcal{X}_k \in \mathcal{D}$ , the function  $G(\mathcal{X}) = Q(\mathcal{X}|\mathcal{X}_k) - F(\mathcal{X}) \geq 0$  achieves its global minimum at the point  $\mathcal{X} = \mathcal{X}_k$ . Hence, it is easy to determine that the first order optimality condition implies

$$Q'(\mathcal{X}|\mathcal{X}_k; \Lambda)|_{\mathcal{X}=\mathcal{X}_k} - F'(\mathcal{X}_k; \Lambda) = 0 \quad \forall \Lambda, \quad (38)$$

which completes the proof.

*Theorem 6:* Denote  $\mathcal{D}$  as the set of solutions generated by Algorithm 2, and the sequence  $\{\mathcal{X}_k\} \subseteq \mathcal{D}$ . Then any limit point of  $\{\mathcal{X}_k\}$  is a stationary point to (13).

*Proof:* From Theorem 4, we know that  $F(\mathcal{X})$  is bounded; therefore, the sequence  $\{\mathcal{X}_k\}$  generated by Algorithm 2 has limit point  $\mathcal{X}_t$  and there exists a subsequence  $\{\mathcal{X}_{k_j}\}$  converging to limit point  $\mathcal{X}_t$ , i.e.,  $\mathcal{X}_{k_j} \rightarrow \mathcal{X}_t$ . Let

$$Q(\mathcal{X}|\mathcal{X}_{k_j}) = J(\mathcal{X}|\mathcal{X}_{k_j}) + \frac{\mu}{2}\|\mathcal{X} - \mathcal{X}_{k_j}\|_F^2,$$

Note that  $\mathcal{X}_{k_{j+1}} = \arg \min_{\mathcal{X}} Q(\mathcal{X}|\mathcal{X}_{k_j})$ . Thus, together with Theorem 4 and (37), we have

$$\begin{aligned} Q(\mathcal{X}_{k_{j+1}}|\mathcal{X}_{k_{j+1}}) &= F(\mathcal{X}_{k_{j+1}}) \leq Q(\mathcal{X}_{k_{j+1}}|\mathcal{X}_{k_j}) \\ &\leq Q(\mathcal{X}|\mathcal{X}_{k_j}) \quad \forall \mathcal{X} \in \mathcal{D}. \end{aligned} \quad (39)$$

A more straightforward consequence of (39) is that the sequence of the surrogate function values is non-increasing, and we also have

$$Q(\mathcal{X}_t|\mathcal{X}_t) \leq Q(\mathcal{X}|\mathcal{X}_t) \quad \forall \mathcal{X} \in \mathcal{D}, \quad (40)$$

which implies

$$Q'(\mathcal{X}|\mathcal{X}_k; \Lambda)|_{\mathcal{X}=\mathcal{X}_t} \geq 0 \quad \forall \Lambda \quad \text{with } \mathcal{X}_t + \Lambda \in \mathcal{D}. \quad (41)$$

Combining this result with Theorem 5, we obtain

$$F'(\mathcal{X}_t; \Lambda) \geq 0 \quad \forall \Lambda \quad \text{with } \mathcal{X}_t + \Lambda \in \mathcal{D}. \quad (42)$$

implying that  $\mathcal{X}_t$  is a stationary point of  $F(\mathcal{X})$ .

## V. EXPERIMENTAL RESULTS

In this section, we conduct experiments to evaluate our proposed algorithm. We first verify the effectiveness of the algorithm under certain parameters on several synthetic data sets and then compare it with other methods on real-world data including natural images and video data sequences. The competing approaches include matrix and tensor completion algorithms: truncated nuclear norm regularization (TNNR) [16], low rank tensor completion (LRTC) [3], tensor nuclear norm method (TNN) [31], low tubal rank tensor completion (Tubal-NN) [40] and tensor completion by tensor factorization (TCTF) [28]. The codes of TNNR,<sup>1</sup> LRTC,<sup>2</sup> TNN,<sup>3</sup> Tubal-NN<sup>4</sup> and TCTF<sup>5</sup> are provided by the corresponding authors. The parameters of the compared methods are adjusted to be optimal and the optimal results are reported. Each algorithm stops when the maximum number of iterations is reached or  $\|\mathcal{X}_{k+1} - \mathcal{X}_k\|_F$  is sufficiently small. Supposing we denote  $\mathcal{M}$  as the original tensor data and  $\mathcal{X}_{rec}$  as the final recovered output. In general, the relative square error (RSE) and peak signal-to-noise ratio (PSNR) are commonly used criteria to evaluate the recovery performance of different algorithms and are defined as follows:

$$\text{RSE} = \|\mathcal{X}_{rec} - \mathcal{M}\|_F / \|\mathcal{M}\|_F \quad (43)$$

$$\text{MSE} = \|P_{\Omega^c}(\mathcal{X}_{rec} - \mathcal{M})\|_F^2 / T \quad (44)$$

$$\text{PSNR} = 10 \times \log_{10}\left(\frac{255^2}{\text{MSE}}\right) \quad (45)$$

where  $T$  is the total number of missing entries in a tensor. We suppose that the maximum pixel value on real-world data is 255. All experiments are executed on a PC with an Intel Core i7 CPU @2.70 GHz and 8 GB Memory.

### A. SYNTHETIC EXPERIMENTS

First, we validate the exact tensor completion effectiveness of our proposed method; we simply consider the tensors  $\mathcal{M} \in \mathbb{R}^{n \times n \times n}$  with varying dimensions  $n = 50, 100$  and  $300$ . The corresponding tubal rank  $r$  is determined by  $\mathcal{M} = \mathcal{P} * \mathcal{Q}$ , where the entries of  $\mathcal{Q} \in \mathbb{R}^{r \times n \times n}$  and  $\mathcal{P} \in \mathbb{R}^{n \times r \times n}$  are independently sampled from an  $\mathcal{N}(0, 1/n)$  distribution. We uniformly select the  $pn^3$  positions of  $\mathcal{M}$  to construct the support set  $\Omega$ , where the sampling ratio  $p$  is defined as the percentage of the observed entries. For the choice of the parameters, we can set a series of lambdas  $\lambda \in [1000]$  and thresholds  $\theta \in [10, 3000]$  to find the best parameters. Our experimental results show that fixing the regularization parameter  $\lambda$  and only adjusting the truncated threshold  $\theta$  are sufficient to achieve the optimal results. In the experiments, we empirically set the parameters  $\lambda = 80, 100$  and  $500$  and test  $\theta$  from  $[1000, 3000]$  to choose an optimal value for each case manually. As shown in Table.2, our method yields very

<sup>1</sup><https://sites.google.com/site/zjuyaohu/>

<sup>2</sup><http://www.cs.rochester.edu/~jliu/publications.html>

<sup>3</sup><http://www.ece.tufts.edu/~shuchin/software.html>

<sup>4</sup><https://canyilu.github.io/publications/>

<sup>5</sup><https://panzhous.github.io/>

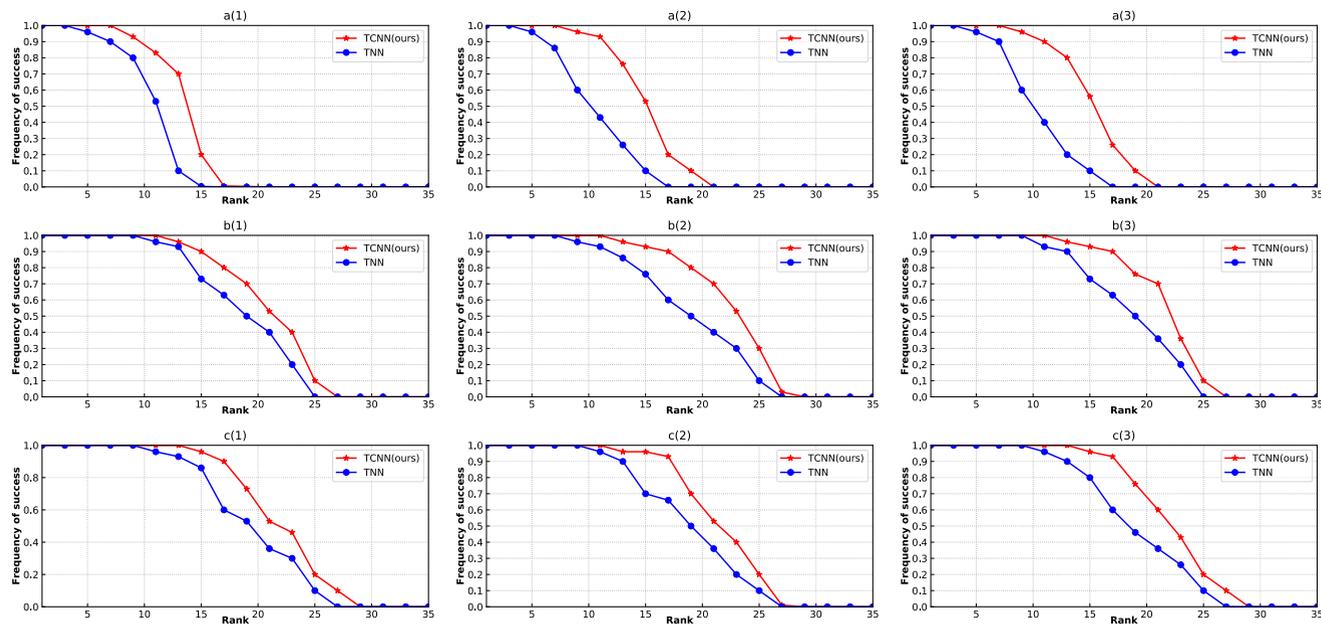


FIGURE 1. Frequency of success comparison of TNN and TCNN (ours) on synthetic data. The tubal rank  $r$  is equivalent to 1, 3, 5, . . . , 35, respectively.

TABLE 2. Exact low-rank tensor completion on random data for varying tensor dimensions with different parameter values.

#	$n$	$r$	$p$	$\lambda$	$rank_t(\mathcal{X}_{rec})$	$\frac{\ \mathcal{X}_{rec} - \mathcal{M}\ _F}{\ \mathcal{M}\ _F}$
a(1)	50	3	0.5	80	3	2.4e-6
a(2)	50	5	0.6	100	5	3.8e-6
a(3)	50	10	0.8	500	10	9.6e-6
b(1)	100	3	0.5	80	3	1.6e-6
b(2)	100	5	0.6	100	5	5.4e-7
b(3)	100	10	0.8	500	10	5.5e-7
c(1)	300	10	0.5	80	10	7.8e-6
c(2)	300	15	0.6	100	15	3.2e-6
c(3)	300	30	0.8	500	30	2.4e-6

small relative errors, less than  $10^{-5}$ , and the correct tubal rank estimation of  $\mathcal{M}$  in all cases. These results verifies the effectiveness of our parameter selection strategy.

Second, both TNN and our TCNN are based on the tensor tubal rank definitions, which reflects the intrinsic low-rank property in the corresponding Fourier space. Because the two approaches use the similar tensor rank definitions, we conduct experiments to compare the methods in detail based on synthetic data. We also consider the associated parameters, i.e., the tensor size  $n$ , sampling rate  $p$  and parameter  $\lambda$ , as indicated in Table.2. Then, we vary the tubal rank  $r$  from 1 to 35 in increment of 2. For fair comparison, we run each method 30 times and regard the recovered  $\mathcal{X}_{rec}$  to be successful if  $\|\mathcal{X}_{rec} - \mathcal{M}\|_F / \|\mathcal{M}\|_F \leq 10^{-3}$ . Fig.1 plots the experimental results of these two methods. Although the size of the tensors,

the sampling ratio and the parameter  $\lambda$  are different, we can observe a visible improvement of our method over TNN. All these results are consistent with those in Table.2. Thus, it is further verified that the newly developed method is robust to the parameters.

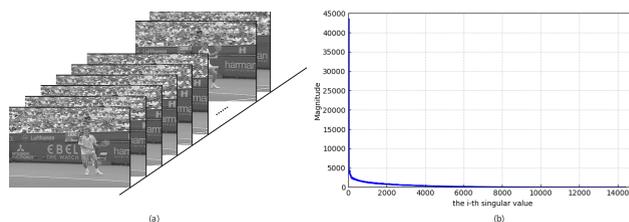
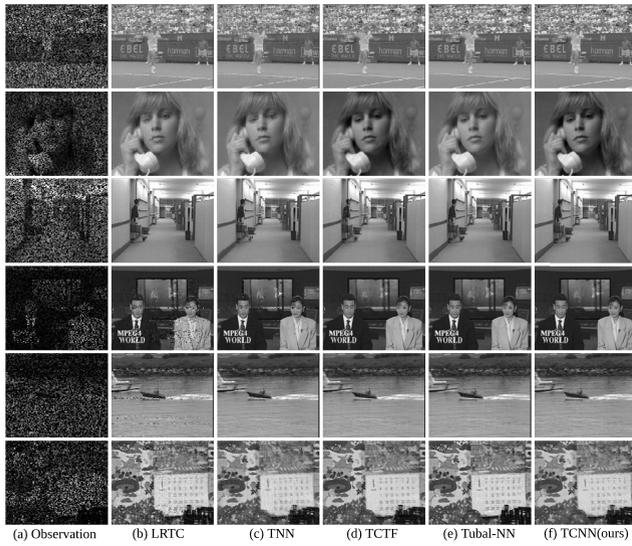


FIGURE 2. Illustration of the low tubal rank property in the video sequence. (a) First 50 Frames of the sequence Stefan. (b) Singular values of the first 50 frames in the sequence Stefan.

### B. TENSOR COMPLETION FOR VIDEO RECOVERY

In the task of video recovery, we compare our method to the four tensor based methods LRTC [3], TNN [31], TCTF [28] and Tubal-NN [40]. Essentially, a grayscale video is a 3-way tensor. In this section, we evaluate the performance of the proposed TCNN on grayscale video sequences from <http://trace.eas.asu.edu/yuv/>. Because the given video sequences are color videos, we convert them to grayscale to form the 3-way tensor. We plot the singular values of a test video of size  $288 \times 352$  and 50 frames in Fig.2. It can be observed that there are many singular values whose magnitudes are very close to zero, indicating that the video has much redundant information and its low tubal rank structure is notable. Thus, these videos can be approximated by the tensors of the low tubal rank very well.



**FIGURE 3.** The 25th frame of the five sequences reconstructed by five methods. The first three rows are the results with the sampling ratio  $p = 0.5$  and the last three rows are the results with sampling ratio  $p = 0.3$ .

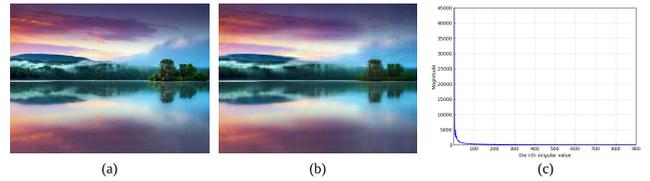
**TABLE 3.** The PSNR values and run times (seconds) for various methods on test videos at different sampling ratios  $p$ .

Videos	LRTC		TNN		TCTF		Tubal-NN		TCNN(ours)	
	PSNR	time								
Sampling ratio $p=0.8$										
Stefan	38.15	790.0	39.43	586.3	31.62	33.24	40.66	70.50	<b>41.95</b>	155.0
Suzie	43.17	360.0	46.11	120.5	36.40	15.00	50.36	30.02	<b>51.86</b>	62.34
Hall	40.93	360.2	41.20	120.5	33.15	17.76	<b>46.58</b>	31.20	46.29	62.05
<b>Average</b>	<b>40.75</b>	<b>503.4</b>	<b>42.25</b>	<b>275.7</b>	<b>33.72</b>	<b>22.00</b>	<b>45.87</b>	<b>43.91</b>	<b>46.70</b>	<b>93.13</b>
Sampling ratio $p=0.5$										
Stefan	17.39	728.1	20.11	580.6	24.65	33.50	28.01	72.50	<b>28.70</b>	157.4
Suzie	22.60	362.0	24.05	121.3	26.52	17.42	<b>31.01</b>	31.42	30.71	62.43
Hall	24.29	360.3	25.63	122.5	26.40	17.66	33.21	31.66	<b>34.54</b>	64.13
<b>Average</b>	<b>21.42</b>	<b>483.4</b>	<b>23.26</b>	<b>274.8</b>	<b>25.86</b>	<b>22.86</b>	<b>30.74</b>	<b>45.19</b>	<b>31.32</b>	<b>94.65</b>
Sampling ratio $p=0.3$										
News	14.36	365.7	19.44	122.0	24.67	18.01	28.33	32.01	<b>29.20</b>	63.39
Coastguard	15.10	364.0	19.80	121.7	23.14	17.30	27.30	31.40	<b>27.46</b>	62.22
Mobile	17.20	360.5	19.55	120.5	21.50	17.76	<b>25.09</b>	31.47	24.79	62.01
<b>Average</b>	<b>15.55</b>	<b>363.4</b>	<b>19.59</b>	<b>121.4</b>	<b>23.10</b>	<b>17.69</b>	<b>26.90</b>	<b>31.62</b>	<b>27.15</b>	<b>62.54</b>

For each sequence, we use the first 50 frames for the comparison due to the computational limitations. Fig.3 displays the 25th frame of the five test videos. The frame sizes of the videos in the first row (Stefan) are  $288 \times 352$  pixels, and the others are all  $144 \times 176$  pixels. In the first three rows, the sampling ratio is set as  $p = 0.5$ . In the last three rows, the sampling ratio is  $p = 0.3$ . Fig.3(a) shows examples of 25th incomplete frame of the videos. The 25 frame examples of the recovered video are shown in Fig.3(b)-3(f), respectively. Table.3 shows the corresponding PSNR values and run times of all the methods for the six test videos at the sampling rates: 0.8, 0.5 and 0.3. The experimental results show that our method performs excellent reconstruction compared with the competing methods, and the run time is tolerable. As can be



**FIGURE 4.** The images (1-10) used in this section.



**FIGURE 5.** Illustration of the low tubal rank property of the images that we used in our experiment. (a) The original image (b) approximation by its tubal rank  $r = 30$ . (c) The singular values of the image (a).



**FIGURE 6.** Examples for image recovery performance comparison. The first three rows are the results with sampling ratio  $p = 0.5$  and the last two rows are the results with sampling ratio  $p = 0.3$ .

seen, the LRTC (Fig. 3(b)) is the worst, and when the sampling ratio drops from 0.5 to 0.3, the PSNR value is nearly halved. Thus, it is verified that the LRTC method is not robust to video recovery. The TNN (Fig. 3(c)) performs better PSNR than the LRTC method. However, our method is superior in PSNR and runs much faster than the TNN in nearly all cases. TCTF (Fig. 3(d)) runs faster than our benefits from the tensor factorization, but our method yields a higher PSNR (greater than 4 dB on average). The only method that comes within 0.5 dB to ours in some cases, is the Tubal-NN (Fig.3(e)). This also confirms that the internal low tubal rank property (exploited by both our algorithm and Tubal-NN) can provide a strong cue for video reconstruction.

### C. TENSOR COMPLETION FOR IMAGE RECOVERY

In the task of image reconstruction, we compare the proposed TCNN method with the other four methods, including the matrix and tensor completion algorithms: TNNR [16], LRTC [3], TNN [31] and TCTF [28]. Matrix completion algorithms, such as TNNR, process each color channel

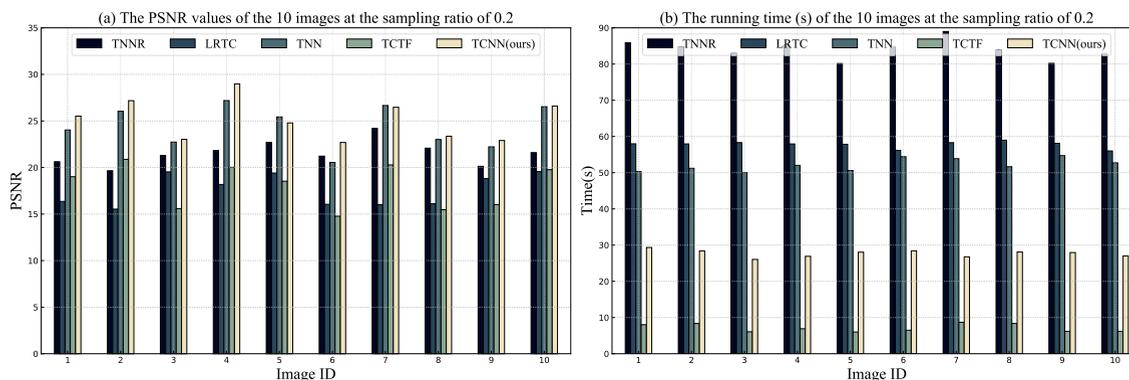


FIGURE 7. Comparison of the PSNR values (a) and the running time (b) on all 10 images (Fig.4). The sampling ratio is set as  $p = 0.2$ .

TABLE 4. The PSNR values and run times (seconds) for various methods on test images at different sampling ratios  $p$ .

Images	TNNR		LRTC		TNN		TCTF		TCNN(ours)	
	PSNR	time	PSNR	time	PSNR	time	PSNR	time	PSNR	time
Sampling ratio $p=0.8$										
Starfish	26.95	65.20	33.88	55.20	39.44	53.31	33.87	6.22	<b>41.90</b>	24.00
Flower	31.89	60.24	44.89	55.03	40.11	52.87	37.03	6.33	<b>48.28</b>	23.67
Leaf	31.29	63.77	39.01	56.20	39.22	52.66	36.88	6.63	<b>40.01</b>	24.01
Sun	29.92	66.33	38.62	59.77	33.87	53.26	30.61	7.20	<b>41.89</b>	25.60
Mountain	25.40	66.36	35.57	58.20	39.00	56.99	32.19	6.34	<b>41.09</b>	25.75
<b>Average</b>	<b>29.09</b>	<b>64.38</b>	<b>30.99</b>	<b>56.88</b>	<b>38.33</b>	<b>53.82</b>	<b>34.12</b>	<b>6.54</b>	<b>42.63</b>	<b>24.61</b>
Sampling ratio $p=0.5$										
Starfish	25.62	70.90	20.36	55.98	29.03	53.30	29.05	7.03	<b>30.52</b>	26.35
Flower	29.65	64.80	22.53	55.95	32.04	53.22	32.88	6.36	<b>34.18</b>	24.38
Leaf	28.70	60.14	24.41	56.84	30.43	53.60	29.53	7.01	<b>32.26</b>	24.05
Sun	26.04	69.00	23.85	60.16	29.12	52.37	28.91	7.15	<b>32.07</b>	27.01
Mountain	24.17	67.35	22.68	57.30	28.22	54.05	27.65	7.76	<b>30.94</b>	27.12
<b>Average</b>	<b>26.84</b>	<b>66.44</b>	<b>22.77</b>	<b>57.25</b>	<b>29.76</b>	<b>53.30</b>	<b>29.60</b>	<b>7.06</b>	<b>31.99</b>	<b>25.78</b>
Sampling ratio $p=0.3$										
Starfish	23.45	72.36	19.89	57.20	26.06	53.11	20.33	7.30	<b>26.80</b>	27.08
Flower	26.46	66.62	20.01	56.04	28.97	55.99	20.34	5.38	<b>30.23</b>	24.01
Leaf	25.00	65.86	18.69	58.70	25.75	53.25	19.18	6.60	<b>27.34</b>	26.12
Sun	22.56	65.01	21.90	57.92	25.30	54.04	20.14	6.42	<b>26.84</b>	26.09
Mountain	22.17	66.05	20.01	57.04	<b>25.66</b>	54.27	20.28	7.40	22.55	27.60
<b>Average</b>	<b>23.93</b>	<b>67.18</b>	<b>20.10</b>	<b>57.38</b>	<b>26.35</b>	<b>54.13</b>	<b>20.05</b>	<b>6.62</b>	<b>26.75</b>	<b>26.18</b>

separately and then combine the results to form the final result. However, by directly applying the matrix completion method to each channel separately, the recovery performance may be reduced. We now illustrate the performance of matrix and tensor completion based on ten images, as shown in Fig. 4. The size of each image is  $400 \times 300$ . Fig. 5 also illustrates that a color image can be well approximated by a low tubal rank tensor because the information is dominated by the top  $r$  singular values. Thus the incomplete image can be recovered by the low-rank approximation method. Examples of the recovered visual results are shown in Fig. 6 and the corresponding PSNR values and run times are summarized in Table. 4.

Fig. 6(a) presents five example images of Fig. 4 with the sampling ratios: 0.5 and 0.3. Examples of the results recovered by five methods are illustrated in Figs. 6(b)-6(f). In most cases, the results of TNNR (Fig. 6(b)) still contain some quite

blurry parts and are the worst of the five methods. In tensor cases, the LRTC converges much faster than the TNNR does, but the corresponding PSNR is apparently inferior to that of the other tensor methods in all cases. Moreover, the result of TNNR is slightly superior to that of LRTC with respect to PSNR in some cases. The results of TNN (Fig. 6(d)) and TCTF (Fig. 6(e)) are much clearer than those of TNNR (Fig. 6(b)) and LRTC (Fig. 6(c)). However, the edge details of these images are not well restored, and a certain amount of information is still missing. As can be seen, our TCNN (Fig. 6(f)) produce the best visual results. We further compare the PSNR value and running time of each approach on all ten images at the sampling ratio of  $p = 0.2$ , and the results are reported in Fig. 7. Our method achieves an improvement of at least 1dB in PSNR for most test images. These results demonstrate that our TCNN yields the best recovery performance, and the operation speed also meets the requirement of tensor completion.

## VI. CONCLUSIONS

In this work, we studied the nonconvex optimization regularized by tensor capped nuclear norm for low-rank tensor completion. Capped nuclear norms regularization is a better rank minimization approximation than the standard nuclear norm. The proposed tensor nuclear capped norm is based on t-SVD and utilizes the low-rankness of all-mode unfoldings of the tensor. Hence, the performance of our approach is considerably improved. We further developed an MM framework and a proximal technique to efficiently solve the non-convex formulation. In addition, the convergence of the objective function was analyzed. Experimental results show that the proposed approach performs better in recovering videos and images than other competing tensor completion methods. For the run time, the results also demonstrated that the proposed method is accelerated with the advantage of the capped nuclear norm.

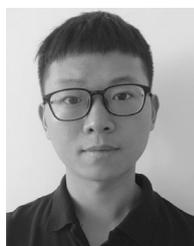
## REFERENCES

[1] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proc. 4th ACM Conf. Recommender Syst.*, Sep. 2010, pp. 79–86.

- [2] Y. Li, Y. Zhou, J. Yan, J. Yang, and X. He, "Tensor error correction for corrupted values in visual data," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 2321–2324.
- [3] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.
- [4] D. Zhang, T. Yin, G. Yang, M. Xia, L. Li, and X. Sun, "Detecting image seam carving with low scaling ratio using multi-scale spatial and spectral entropies," *J. Vis. Commun. Image Represent.*, vol. 48, pp. 281–291, Oct. 2017.
- [5] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust video denoising using low rank matrix completion," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 1791–1798.
- [6] Y. Song, G. Yang, H. Xie, D. Zhang, and S. Xingming, "Residual domain dictionary learning for compressed sensing video recovery," *Multimedia Tools Appl.*, vol. 76, no. 7, pp. 10083–10096, 2017.
- [7] W. Dong, G. Shi, X. Li, K. Peng, J. Wu, and Z. Guo, "Color-guided depth recovery via joint local structural and nonlocal low-rank regularization," *IEEE Trans. Multimedia*, vol. 19, no. 2, pp. 293–301, Feb. 2017.
- [8] X. Fang, Y. Xu, X. Li, Z. Lai, and W. K. Wong, "Robust semi-supervised subspace clustering via non-negative low-rank representation," *IEEE Trans. Cybern.*, vol. 46, no. 8, pp. 1828–1838, Aug. 2016.
- [9] J. Chen and J. Yang, "Robust subspace segmentation via low-rank representation," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1432–1445, Aug. 2014.
- [10] Y. Yang, W. Hu, Y. Xie, W. Zhang, and T. Zhang, "Temporal restricted visual tracking via reverse-low-rank sparse learning," *IEEE Trans. Cybern.*, vol. 47, no. 2, pp. 485–498, Feb. 2017.
- [11] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, p. 717, Dec. 2009.
- [12] Y. Song, Y. Zeng, X. Li, B. Cai, and G. Yang, "Fast CU size decision and mode decision algorithm for intra prediction in HEVC," *Multimedia Tools Appl.*, vol. 76, no. 2, pp. 2001–2017, 2017.
- [13] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.
- [14] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2009, pp. 2080–2088.
- [15] F. Shang, Y. Liu, and J. Cheng, "Tractable and scalable Schatten quasi-norm approximations for rank minimization," *Artif. Intell. Statist.*, pp. 620–629, May 2016.
- [16] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2117–2130, Sep. 2013.
- [17] S. Zhang, H. Qian, W. Chen, and Z. Zhang, "A concave conjugate approach for nonconvex penalized regression with the MCP penalty," in *Proc. 27th AAAI Conf. Artif. Intell.*, Jun. 2013, pp. 1027–1033.
- [18] Q. Tang, K. Yang, D. Zhou, Y. Luo, and F. Yu, "A real-time dynamic pricing algorithm for smart grid with unstable energy providers and malicious users," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 554–562, Aug. 2016.
- [19] Y. Peng, J. Suo, Q. Dai, and W. Xu, "Reweighted low-rank matrix recovery and its application in image restoration," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2418–2430, Dec. 2014.
- [20] L. Liu, L. Chen, C. L. P. Chen, Y. Y. Tang, and C. M. Pun, "Weighted joint sparse representation for removing mixed noise in image," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 600–611, Mar. 2017.
- [21] Q. Sun, S. Xiang, and J. Ye, "Robust principal component analysis via capped norms," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2013, pp. 311–319.
- [22] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
- [23] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *J. ACM*, vol. 60, no. 6, Nov. 2013, Art. no. 45.
- [24] 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 Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5249–5257.
- [25] M. Morup, D. M. Dunlavy, E. Acar, and T. G. Kolda, "Scalable tensor factorizations with missing data," Sandia Nat. Lab., Albuquerque, NM, USA, Tech. Rep., 2010.
- [26] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, 2011, Art. no. 025010.
- [27] Y. Xu, R. Hao, W. Yin, and Z. Su, "Parallel matrix factorization for low-rank tensor completion," Dec. 2013, *arXiv:1312.1254*. [Online]. Available: <https://arxiv.org/abs/1312.1254>
- [28] 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.
- [29] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, 2011.
- [30] C. D. Martin, R. Shafer, and B. LaRue, "An order- $p$  tensor factorization with applications in imaging," *SIAM J. Sci. Comput.*, vol. 35, no. 1, pp. A474–A490, 2013.
- [31] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2015.
- [32] 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.
- [33] C. Lu, J. Feng, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
- [34] Z. Lin, C. Xu, and H. Zha, "Robust matrix factorization by majorization minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 208–220, Jan. 2018.
- [35] S. Kuang, H. Chao, and Q. Li, "Matrix completion with capped nuclear norm via majorized proximal minimization," *Neurocomputing*, vol. 316, pp. 190–201, Nov. 2018.
- [36] L.-H. Lim and P. Comon, "Multirray signal processing: Tensor decomposition meets compressed sensing," *Comp. Rendus Mécanique*, vol. 338, no. 6, pp. 311–320, Jun. 2010.
- [37] K. Chen, H. Dong, and K.-S. Chan, "Reduced rank regression via adaptive nuclear norm penalization," *Biometrika*, vol. 100, no. 4, pp. 901–920, 2013.
- [38] C. Lu, J. Tang, S. Yan, and Z. Lin, "Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 829–839, Feb. 2016.
- [39] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang, "Compressive sensing via nonlocal low-rank regularization," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3618–3632, Aug. 2014.
- [40] C. Lu, J. Feng, Z. Lin, and S. Yan, "Exact low tubal rank tensor recovery from Gaussian measurements," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, Jul. 2018, pp. 2504–2510.



**XI CHEN** received the M.S. degree from the National University of Defense Technology, in 1991. He is currently a Professor with the School of Computer and Communication Engineering, Changsha University of Science and Technology, China. His major research interests include video coding, machine learning, and pattern recognition.



**JIE LI** is currently pursuing the M.S. degree in computer science with the Changsha University of Science and Technology. His research interests include video processing, matrix completion, and sparse representation.



**YUN SONG** received the M.S. degree in computer science from the Changsha University of Science and Technology, in 2008, and the Ph.D. degree in computer science from Hunan University, China, in 2016. He is currently an Associate Professor with the School of Computer and Communication Engineering, Changsha University of Science and Technology. His major research interests include video coding, multimedia communication and security, compressive sensing, video analysis, and image processing.



**FENG LI** received the Ph.D. degree from Sun Yat-sen University, in 2006. He is currently a Professor with the School of Computer and Communication Engineering, Changsha University of Science and Technology, China. His major research interests include video coding, image processing, and pattern recognition.



**JIANJUN CHEN** received the M.S. degree from the Changsha University of Science and Technology, in 2015. He is currently pursuing the Ph.D. degree in computer science with the University of Chinese Academy of Sciences. His research interests include image processing and pattern recognition.



**KUN YANG** (M'02–SM'08) received the B.Sc. and M.Sc. degrees from the Computer Science Department, Jilin University, Changchun, China, and the Ph.D. degree from the Department of Electronic and Electrical Engineering, University College London, London, U.K. He is currently a Chair Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., where he is also leading the Network Convergence Laboratory. He has authored or coauthored over 80 journal papers. His current research interests include wireless networks, network convergence, the future Internet technology and network virtualization, and mobile cloud computing and networking. He is a Fellow of the IET.

...