Exploiting Global Low-rank Structure and Local Sparsity Nature for Tensor Completion

Yong Du, Guoqiang Han*, Yuhui Quan, Zhiwen Yu, *Senior Member, IEEE*, Hau-San Wong, C. L. Philip Chen, *Fellow, IEEE*, Jun Zhang, *Fellow, IEEE*

Abstract—In the era of data science, a huge amount of data has emerged in the form of tensors. In many applications, the collected tensor data is incomplete with missing entries, which affects the analysis process. In this paper, we investigate a new method for tensor completion, in which a low-rank tensor approximation is used to exploit the global structure of data, and sparse coding is used for elucidating the local patterns of data. Regarding the characterization of low-rank structures, a weighted nuclear norm for tensor is introduced. Meanwhile, an orthogonal dictionary learning process is incorporated into sparse coding for more effective discovery of the local details of data. By simultaneously using the global patterns and local cues, the proposed method can effectively and efficiently recover the lost information of incomplete tensor data. The capability of the proposed method is demonstrated with several experiments on recovering MRI data and video data, and the experimental results have shown the excellent performance of the proposed method in comparison with recent related methods.

Index Terms—tensor completion, sparse coding, weighted nuclear norm, orthogonal dictionary learning

I. Introduction

IN many data mining and machine learning scenarios, the collected data are in the form of multi-dimensional arrays, *e.g.* multi-spectral images, image patch stacks, videos, and magnetic resonance imaging (MRI) data, which can be represented as tensors. The tensor data (*i.e.* data in the form of tensor) encodes very rich structural information, and tensor analysis is widely used in gait recognition [1, 2], object recognition [3], dynamic texture recognition [4], image recovery [5–7], medical image processing [8, 9], and many machine learning fields [10–17].

The tensor data collected in real world are often incomplete, which may be caused by occlusions, noise, partial damages, difficulties in collection, or data loss during transition. For example, people may wear sunglasses in surveillance videos. In recommendation systems, the preferences of users are often only available partially. Such incompleteness of tensor data may significantly decrease the quality of the data, making the analysis process very difficult.

This incompleteness problem can be remedied through tensor completion, which is to recover the complete tensor

Yong Du, Guoqiang Han, Yuhui Quan, Zhiwen Yu and Jun Zhang are with the School of Computer Science and Engineering at South China University of Technology, China. Email: csyongdu@mail.scut.edu.cn, {csgqhan, csyhquan, zhwyu}@scut.edu.cn. *Asterisk indicates the corresponding author.*

C. L. P. Chen is with the Faculty of Science and Technology, University of Macau and Dalian Maritime University, Dalian 116026, China (e-mail: philip.chen@ieee.org).

Hau-San Wong is with the Department of Computer Science in City University of Hong Kong.

from the incomplete one. There have been some approaches proposed for this purpose, which can be roughly classified into two categories: the matching-based approaches (*e.g.* [18, 19]) that find optimal patch correspondences to fill the holes in data, and the low-rank approximation approaches (*e.g.* [20, 21]) which exploit the low-rank structures of the original tensor data for the completion. The matching-based approaches are often designed for inpainting large regions, while the low-rank approximation approaches are applicable to both damaged regions and random missing entries.

A. Related Work

This paper focuses on the development of an effective low-rank approximation approach for tensor completion. In the following, we first review the existing low-rank tensor completion methods.

The basic idea of the low-rank approximation approach is that high-dimensional data are likely to lie in some low-dimensional spaces, since they often exhibit large similarities. Such low-dimensionality of data can be well characterized by low-rank constraints on the data, and hence the global structure of tensor data encoded in their similarities can be recovered via low-rank minimization. Nevertheless, unlike the rank of a matrix, the rank of a tensor has several non-equivalent definitions which are built upon different types of tensor decompositions. By using different definitions of rank of a tensor, different kinds of low-rank tensor completion approaches have been proposed. Currently, there are mainly three main types of tensor decompositions: CP decomposition, Tucker decomposition, and t-SVD decomposition.

CP decomposition [22–26] decomposes a tensor into the sum of a group of rank-1 factors, and its rank (CP-rank) is defined as the minimum number of the rank-1 factors. Based on the CP decomposition, Yokota et al. [25] proposed a low-rank tensor approximation model with the smoothness constraints defined by total variation and quadratic variation. For acceleration, Liu et al. [23] proposed a tensor completion model based on factor matrix trace norm minimization, in which the cost of performing singular value decompositions can be noticeably reduced. Liu et al. [26] used the nuclear norm with CP decomposition to approximately measure the CP-rank, which leads to a convex optimization problem. However, estimating the CP-rank of high-dimensional data is non-trivial, and for practical problems it is often computationally difficult to determine the CP-rank or the best low-rank CP approximation of a tensor data set beforehand [27]. To address

this issue, Zhao et al. [24] proposed a hierarchical probabilistic model with a fully Bayesian treatment to determinate the CP-rank automatically.

In t-SVD decomposition [27–31], a tensor is expressed by a defined t-product operation of three factors. The structure of t-SVD is similar to that of SVD applied to a matrix. Based on t-SVD, Zhang et al. [28] defined the tensor multi-rank and tensor tubal-rank, which are used to construct the low-rank tensor completion models. Based on the tubal-rank, Liu et al. [29] proposed a model similar to [28]. In the same spirit, Lu et al. generalized the robust principal component analysis technique from the matrix case to the tensor case [30]. In [31], the twist tensor nuclear norm is developed based on t-svd and used to design a convex low-rank tensor approximation model.

Tucker decomposition [32–36] decomposes a tensor into a core tensor and a set of matrices, and the rank is defined based on the core tensor. Building upon the Tucker decomposition, Liu et al. [32] proposed a tensor nuclear norm which is defined by the rank of the matrix generated by unfolding the tensor along each mode. Liu et al. [34] proposed a similar method with much lower computational complexity. Han et al. [35] modified the tensor nuclear norm to form a truncated version, which ignores large singular values and thus yields better approximation to the rank. They also introduced a sparse regularization framework based on a multi-dimensional DCT dictionary to construct a minimization model for tensor completion. Hosono et al. [36] proposed to use the weighted version of the tensor nuclear norm in a denoising model, as it is a better surrogate to the rank function than either the nuclear norm or truncated nuclear norm.

Besides, Imaizumi et al. [37] developed a convex tensor completion model based on tensor train decomposition. It is worth mentioning that tensor completion is related to matrix completion, as tensor is the generalization of matrix. Interested readers may refer to [38–45] for the low-rank matrix completion methods which are related to ours.

B. Contributions

The key in tensor completion is how to relate the unknown values with the known ones by utilizing the global structures as well as the local patterns in the tensor. In this paper, we proposed an effective method for tensor completion, which simultaneously utilizes a low-rank approximation as the global guidance, and a sparse representation as the local clue for tensor completion. For low-rank approximation, we introduce the weighted nuclear norm with an effective weighting scheme to measure the rank of a tensor. For sparse representation, we integrate an orthogonal dictionary learning process into the sparse coding stage for more effective discovery of the local patterns of the tensor data. In the experimental results, the proposed method has shown superior performance when compared to a number of state-of-the-art methods, while its computational efficiency is acceptable.

Compared with [32] which only utilizes the global low-rank structure of tensors, we consider additional local information of the data set for completion via sparse coding. Compared with [35] which uses the truncated tensor nuclear

norm for low-rank approximation, we employ the weighted nuclear norm as the surrogate of rank. The performance of the truncated tensor nuclear norm minimization is influenced by the number of truncated singular values, of which the optimal value is not known in advance. Moreover, this ignores a subset of singular values which might be useful for the recovery process. In contrast, weighted nuclear norm minimization with an adaptive weighting scheme in our method allows different singular values to be shrunk with different thresholds, which yields a better approximation to the rank. In addition, compared to [35], we learn a dictionary instead of using a fixed dictionary for sparse coding. By using dictionary learning, our method can learn more meaningful dictionary atoms (*e.g.*, the atoms in the DCT dictionary is orientation-less and periodic, while our learned dictionary has more flexibility and adapts to the data), and recover the local details of data more accurately.

The remainder of this paper is organized as follows. Section II introduces the preliminaries of our work, and Section III presents the proposed model and the corresponding optimization approach. Section IV provides details of the experimental evaluation on the proposed method. The final section concludes the paper.

II. PRELIMINARIES

A. Notations and Definitions

Throughout the paper, we use the following notations. The operation $\circ$ denotes outer product. Tensors are denoted by uppercase calligraphic letters, matrices and sets are denoted by uppercase boldfaced letters, vectors are denoted by lowercase boldfaced letters, and scalars are denoted by normal letters (lowercase or uppercase). For example, $\mathcal{X} \in \mathbb{R}^{Q_1 \times Q_2 \cdots \times Q_N}$ denotes an Nth-order tensor, $\boldsymbol{X} \in \mathbb{R}^{Q_1 \times Q_2}$ denotes a $Q_1 \times Q_2$ matrix, and $\boldsymbol{x} \in \mathbb{R}^Q$ denotes a column vector with Q elements.

For a matrix $\boldsymbol{X}$, its rank is denoted by $rank(\boldsymbol{X})$, and its nuclear norm $\|\boldsymbol{X}\|_*$ is defined by the sum of all singular values of $\boldsymbol{X}$, *i.e.*

$$\|\boldsymbol{X}\|_* = \sum_{i=1}^{L} \sigma_i, \tag{1}$$

where σ_i is the ith largest singular value of $\boldsymbol{X}$, and L is the number of singular values of $\boldsymbol{X}$. $\boldsymbol{X}_{\boldsymbol{\Omega}}$ denotes the elements of $\boldsymbol{X}$ in the index set $\boldsymbol{\Omega}$. The identity matrix with appropriate size is denoted by $\mathbf{I}$.

For an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{Q_1 \times Q_2 \times \cdots \times Q_N}$, the mode-$n$ unfolding operation, or the mode-n matricization, is denoted by

$$\mathcal{X}_{(n)} \in \mathbb{R}^{Q_n \times (Q_1 \times \cdots \times Q_{(n-1)} \times Q_{(n+1)} \times \cdots \times Q_N)},$$

which is a mapping from $\mathcal{X}(q_1, q_2, \cdots, q_N)$ to $\mathcal{X}_{(n)}(q_n, j)$ indexed by

$$j = 1 + \sum_{k=1, k \neq n}^{N} (q_k - 1) J_k, \tag{2}$$

where $J_k = \prod_{m=1, m \neq n}^{k-1} Q_m$. The folding operation $fold[\cdot]$ is the reverse mapping of $\mathcal{X}_{(n)}$, that is, $fold[\mathcal{X}_{(n)}] = \mathcal{X}$.

Furthermore, let the operator $\mathcal{P}$ denote the following operations: sample p distinct patches in $\mathcal{X}$, with a size of $S_1 \times S_2 \times \cdots \times S_N$, where $p = \frac{Q_1}{S_1} \times \frac{Q_2}{S_2} \times \cdots \times \frac{Q_N}{S_N}$, then vectorize these patches, and stack them in a matrix. Let $\mathcal{B} = \mathcal{P}^{-1}$ be the inverse operation of $\mathcal{P}$, *i.e.*, stacking the patches back into a tensor.[1] It is easy to verify that both $\mathcal{P}$ and $\mathcal{B}$ preserve the Frobenius norm.

Given two equal-size tensors $\mathcal{X}$ and $\mathcal{Y}$, their inner product $\langle \mathcal{X}, \mathcal{Y} \rangle$ is defined as the sum of the products of the corresponding entries of $\mathcal{X}$ and $\mathcal{Y}$. The Frobenius norm of $\mathcal{X}$ is then defined by $\|\mathcal{X}\|_F = \langle \mathcal{X}, \mathcal{X} \rangle^{\frac{1}{2}}$. The ℓ_0-norm[2] of a tensor $\mathcal{X}$, denoted by $\|\mathcal{X}\|_0$, is defined as the number of non-zero elements in $\mathcal{X}$.

B. Low-rank Tensor Completion

Tensor is a generalization of matrix, which can be viewed as a multi-dimensional array. The dimensionality of the array is called the order of the tensor. By this definition, a scalar is a 0th-order tensor, a vector is a 1st-order tensor, and a matrix is a 2nd-order tensor. The rank of a matrix can also be generalized to the rank of a tensor, which can be used to measure the variation of the data encoded by a tensor. The rank of a tensor is often defined via the minimum number of its components under certain types of decomposition. Based on the different types of decomposition, there are a number of different definitions for its rank.

Two representative tensor decompositions include the CP decomposition and Tucker decomposition, both of which are generally expressed as a sum of outer products of vectors. Given an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{Q_1 \times Q_2 \times \cdots \times Q_N}$, the CP decomposition on $\mathcal{X}$ can be regarded as a generalization of the matrix singular value decomposition (SVD) to tensor, which yields

$$\mathcal{X} = \sum_{i=1}^{M} \lambda_i \boldsymbol{a}_i^{(1)} \circ \boldsymbol{a}_i^{(2)} \circ \cdots \circ \boldsymbol{a}_i^{(N)}, \tag{3}$$

where $\circ$ denotes the outer product which implies that $\mathcal{X}(q_1, q_2, \cdots, q_N) = \sum_{i=1}^{M} \lambda_i \boldsymbol{a}_i^{(1)}(q_1) \cdot \boldsymbol{a}_i^{(2)}(q_2) \cdots \boldsymbol{a}_i^{(N)}(q_N)$, M is expected to be the minimum number of rank-1 tensors needed to express $\mathcal{X}$, and it is called the CP-rank of the tensor $\mathcal{X}$, and $\boldsymbol{a}_i^{(k)}$ is a vector in $\mathbb{R}^{Q_k}$.

In comparison, the Tucker decomposition expresses $\mathcal{X}$ as

$$\mathcal{X} = \sum_{i_1=1, i_2=1, \cdots, i_N=1}^{M_1, M_2, \cdots, M_N} \lambda_{i_1, i_2, \cdots, i_N} \boldsymbol{a}_{i_1}^{(1)} \circ \boldsymbol{a}_{i_2}^{(2)} \circ \cdots \circ \boldsymbol{a}_{i_N}^{(N)}, \tag{4}$$

where $(M_1, M_2, \cdots, M_N)$ is viewed as the multi-linear rank of the tensor $\mathcal{X}$. By simple manipulation, the decomposition (4) can be rewritten using a core tensor $\mathcal{S} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_N}$ as follows:

$$\mathcal{X} = \mathcal{S} \times \boldsymbol{A}^{(1)} \times \boldsymbol{A}^{(2)} \times \cdots \times \boldsymbol{A}^{(N)}, \tag{5}$$

---

[1]Without loss of generality, we assume $\frac{Q_i}{S_i}$ is an integer for all i. Otherwise, we can extend the tensor to meet this requirement.

[2]According to [46], ℓ_0-norm is neither a norm nor a pseudo-norm. We call it a norm here for convenience.

where $\{\boldsymbol{A}^{(n)} \in \mathbb{R}^{Q_n \times M_n}, n = 1, 2, \cdots, N\}$ is the set of factor matrices with $\boldsymbol{A}^{(n)} = [\boldsymbol{a}_1^{(n)}, \cdots, \boldsymbol{a}_{M_n}^{(n)}]$, and $\times$ denotes the tensor product. The mode-n tensor product of the tensor $\mathcal{S}$ by a matrix $\boldsymbol{A}^{(n)}$ is expressed as

$$(\mathcal{S} \times \boldsymbol{A}^{(n)})(m_1, m_2, \cdots, m_{n-1}, q_n, m_{n+1}, \cdots, m_N)$$
$$= \sum_{m_n} \mathcal{S}(m_1, m_2, \cdots, m_N) \cdot \boldsymbol{A}^{(n)}(q_n, m_n). \tag{6}$$

The low-rank tensor completion problem can be formulated as follows:

$$\min_{\mathcal{X}} rank(\mathcal{X}), \quad \text{s.t. } \mathcal{X}_{\boldsymbol{\Omega}} = \mathcal{M}_{\boldsymbol{\Omega}}, \tag{7}$$

where $\boldsymbol{\Omega}$ is an index set, in which the elements of $\mathcal{M}$ in the set $\boldsymbol{\Omega}$ are given and the remaining elements are missing. The rank function is non-convex, which makes the minimization problem difficult to solve. A popular alternative to the rank is the nuclear norm which relaxes the former to its convex surrogate. Consider the Tucker decomposition, one possible relaxation scheme is as follows [32]:

$$\min_{\mathcal{X}} \|\mathcal{X}\|_*, \quad \text{s.t. } \mathcal{X}_{\boldsymbol{\Omega}} = \mathcal{M}_{\boldsymbol{\Omega}}, \tag{8}$$

where the tensor nuclear norm $\|\mathcal{X}\|_*$ is defined by

$$\|\mathcal{X}\|_* = \sum_{i=1}^{N} \alpha_i \|\mathcal{X}_{(i)}\|_*, \tag{9}$$

with $\alpha_i \geq 0$ and $\sum_{i=1}^{N} \alpha_i = 1$. To improve the approximation to the rank, the truncated nuclear norm is generalized from the matrix to the tensor case in [35].

The low-rank tensor completion models focus on exploiting the global features of the data for completion, as the rank of a tensor is determined by all its entries. However, these models do not consider the local structure of the data which is very important in many applications (*e.g.* local smoothness of a video sequences in space and time).

C. Sparse Coding and Dictionary Learning

In the last two decades, sparse coding and dictionary learning [47–53] have become important techniques for discovering the low-dimensional structures of high-dimensional data. Sparse coding aims at expressing the given data with a dictionary (*i.e.* a set of atoms), such that the linear combination of a few dictionary atoms can approximate the data well. In other words, a signal $\boldsymbol{y}$ is expressed with a dictionary $\boldsymbol{D}$ as

$$\boldsymbol{y} \approx \boldsymbol{D}\boldsymbol{c}, \tag{10}$$

where the coefficient vector $\boldsymbol{c}$ is sparse (*i.e.*, most entries are zeros). The sparsity pattern of $\boldsymbol{c}$ combined with the dictionary can reveal the essential structure of the data and leads to a compact representation.

The dictionary for sparse coding can be analytic (like wavelets) or learned from data. In general, dictionary learning can noticeably improve the representational power of a dictionary and leads to better sparsity in sparse coding, as the learned dictionary is adapted to the data. The plain sparse

coding model with dictionary learning is often formulated as follows:

$$\underset{\boldsymbol{D},\{\boldsymbol{c}_i\}}{\arg\min} \sum_{i=1}^{K} \|\boldsymbol{y}_i - \boldsymbol{D}\boldsymbol{c}_i\|_2^2 + \lambda_i\|\boldsymbol{c}_i\|_0, \qquad (11)$$

where $\{\boldsymbol{y}_i\} \subset \mathbb{R}^Z$ is a set of signals to be approximated, $\boldsymbol{c}_i$ is the sparse code of $\boldsymbol{y}_i$, $\boldsymbol{D} = [\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_E] \in \mathbb{R}^{Z \times E}$ is a dictionary which often has the constraints $\|\boldsymbol{d}_j\|_2 = 1$, $1 \leq j \leq E$, and $\{\lambda_i\}$ is the set of balance factors.

In many applications, over-complete dictionaries are used, as the redundancy of the dictionary facilitates the finding of sparse solutions. However, the more redundant the dictionary is, the higher the computational cost becomes. A balance between performance and computational cost can be achieved by using orthogonal dictionaries. In fact, recent studies [54, 55] have shown that using an orthogonal dictionary can lead to a very efficient algorithm, while achieving comparable performance to that of a redundant dictionary in image restoration.

It is worth mentioning that traditional sparse coding approaches are developed with respect to the matrix form, and they are not applicable to general tensor data. Thus, there are some recent works that investigate sparse coding techniques for tensor data, which are referred to as tensor sparse coding; see *e.g.* [4, 56–60].

III. OUR METHOD

In this section, we present our method, which aims at simultaneously utilizing the global structures and local patterns of data for tensor completion. We first present the minimization model of our method in Section III-A, and then a numerical algorithm is proposed in Section III-B for solving the minimization problem.

A. Model

The basic idea of the proposed method is to introduce the weighted nuclear norm for a tensor to characterize the global low-rank structure of the data, and use a sparse representation with dictionary learning to capture the local patterns of the data. Based on this idea, our model is formulated as follows:

$$\min_{\mathcal{X},\boldsymbol{D},\boldsymbol{C}} \quad \|\mathcal{X}\|_{*,\boldsymbol{W},\boldsymbol{\alpha}} + \frac{\beta}{2}\|\mathcal{X} - \mathcal{B}(\boldsymbol{D}\boldsymbol{C})\|_F^2 + \lambda\|\boldsymbol{C}\|_0$$
$$\text{s.t.} \quad \mathcal{X}_{\boldsymbol{\Omega}} = \mathcal{M}_{\boldsymbol{\Omega}}, \quad \boldsymbol{D}^\top\boldsymbol{D} = \mathbf{I}, \qquad (12)$$

where $\mathcal{X} \in \mathbb{R}^{Q_1 \times Q_2 \times \cdots \times Q_N}$ is the incomplete tensor to be recovered, $\mathcal{B}$ is the tensor patch re-constructor defined in Section II-A, $\boldsymbol{D}$ is an orthogonal dictionary to be learned, $\boldsymbol{C}$ is the set of sparse codes from $\mathcal{X}$ under $\boldsymbol{D}$, $\beta, \lambda > 0$ are the balance factors, $\mathcal{M}$ is a latent tensor, and $\mathcal{X}_{\boldsymbol{\Omega}} = \mathcal{M}_{\boldsymbol{\Omega}}$ means that the elements of $\mathcal{X}$ and $\mathcal{M}$ on the support $\boldsymbol{\Omega}$ are equal.

The weighted tensor nuclear norm $\|\mathcal{X}\|_{*,\boldsymbol{W},\boldsymbol{\alpha}}$ is defined by

$$\|\mathcal{X}\|_{*,\boldsymbol{W},\boldsymbol{\alpha}} = \sum_{i=1}^{N} \alpha_i\|\mathcal{X}_{(i)}\|_{*,\boldsymbol{w}_i}, \qquad (13)$$

where $\boldsymbol{W} = [\boldsymbol{w}_1^\top, \boldsymbol{w}_2^\top, \cdots, \boldsymbol{w}_N^\top]^\top$, $\boldsymbol{w}_i \in \mathbb{R}^{J_i}$ is the weighting vector for the weighted nuclear norm of $\mathcal{X}_{(i)}, i = 1, \cdots, N$,

J_i is the total number of the singular values of $\mathcal{X}_{(i)}$, $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_N]$ is subject to $\alpha_i \geq 0$ for all i and $\sum_{i=1}^{N} \boldsymbol{\alpha}_i = 1$, and $\|\boldsymbol{X}\|_{*,\boldsymbol{w}}$ is the weighted nuclear norm for a matrix, which is defined as

$$\|\boldsymbol{X}\|_{*,\boldsymbol{w}} = \sum_{j=1}^{J} \sigma_j\boldsymbol{w}(j), \qquad (14)$$

where σ_j is the jth largest singular value of $\boldsymbol{X}$, and J is the total number of singular values. Motivated by [36, 43], we define the weight vector $\boldsymbol{w}_i$ as

$$\boldsymbol{w}_i = \delta/\Big(\sigma(\mathcal{X}_{(i)}) + \epsilon\Big), \quad i = 1, 2, \cdots, N, \qquad (15)$$

where $\delta \in \mathbb{R}^+$ is a constant, $\sigma(\mathcal{X}_{(i)}) \in \mathbb{R}^{J_i}$ is the vector of singular values of $\mathcal{X}_{(i)}$, and ϵ is a small value to avoid division by zero. As $\mathcal{X}_{(i)}$ is unknown, we iteratively update $\boldsymbol{w}_i$. In other words, in the kth iteration, we calculate $\boldsymbol{w}_i^{(k)}$ by

$$\boldsymbol{w}_i^{(k)} = \delta/\Big(\sigma(\mathcal{X}_{(i)}^{(k)}) + \epsilon\Big), \quad i = 1, 2, \cdots, N. \qquad (16)$$

It is effective to determine the weights using our proposed strategy, since larger singular values represent more important components of the data and should be shrunk less. Thus, we assume the weights are inversely proportional to the singular values.

There are two components in the model (12). The first component $\|\mathcal{X}\|_{*,\boldsymbol{W},\boldsymbol{\alpha}}$ is to enforce a low-rank structure on the recovered result $\mathcal{X}$, which is to use global similarity as well as the low-dimensionality of data for the recovery. The second component $\frac{\beta}{2}\|\mathcal{X} - \mathcal{B}(\boldsymbol{D}\boldsymbol{C})\|_F^2 + \lambda\|\boldsymbol{C}\|_0$ is to enforce sparsity on the recovered result $\mathcal{X}$, *i.e.*, the local patches of $\mathcal{X}$ can be sparsely represented by some dictionary $\boldsymbol{D}$, which is used to elucidate the local patterns in the tensor for better results. In order to improve the effectiveness of sparse representation, we propose to learn the dictionary $\boldsymbol{D}$. As shown in [54, 55], using an orthogonal dictionary can lead to a speedup without a noticeable performance loss in data recovery. Thus, we also introduce a sparsity constraint to the dictionary. The learned dictionary in the proposed model is adaptive to the local structure of the data. The sparse representation framework of using a third-order tensor for illustration is shown in Figure 1.
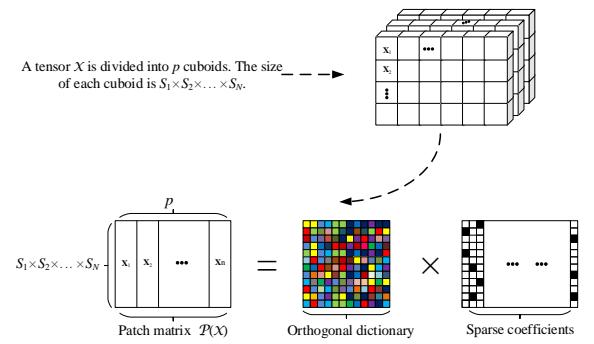

Fig. 1. The procedure of determining the sparse codes

The benefits of our model can be summarized into two aspects. Regarding the low-rank approximation with the weighted tensor nuclear norm, it can enforce a low-rank structure for the data. The weighted tensor nuclear norm is better than the truncated or original tensor nuclear norm in terms of its capability to approximate the rank, and it can capture the global information of the data well by considering the singular values. Regarding the sparsity regularization with the learned dictionary, it can effectively discover the local structural information, and utilize the information for tensor completion. The sparsity constraint also helps to eliminate the artifacts of the reconstructed data, as can be observed in the experiments.

B. Framework of Algorithm

We utilize the Alternating Direction Method of Multipliers (ADMM) to solve the proposed minimization model in (12). Since the ith-mode unfolding $\mathcal{X}_{(i)}$ for $i = 1, \cdots, N$ are dependent on each other, the optimization problem (12) is inseparable with respect to each $\mathcal{X}_{(i)}$. Thus, we first introduce a series of auxiliary tensors $\mathcal{G}_1, \cdots, \mathcal{G}_N$, and reformulate the problem (12) as follows:

$$\min_{\mathcal{X}, \boldsymbol{D}, \boldsymbol{C}} \quad \sum_{i=1}^{N} \alpha_i \|\mathcal{G}_{i,(i)}\|_{*,\boldsymbol{w}_i} + \frac{\beta}{2}\|\mathcal{X} - \mathcal{B}(\boldsymbol{D}\boldsymbol{C})\|_F^2 + \lambda\|\boldsymbol{C}\|_0$$
$$\text{s.t.} \quad \mathcal{X} = \mathcal{G}_i, i = 1, \cdots, N,$$
$$\mathcal{X}_{\boldsymbol{\Omega}} = \mathcal{M}_{\boldsymbol{\Omega}}, \ \boldsymbol{D}^\top \boldsymbol{D} = \mathbf{I}. \quad (17)$$

Then, the augmented Lagrangian function of (17) is given by

$$L(\mathcal{X}, \mathcal{G}_i, \boldsymbol{C}, \boldsymbol{D}, \mathcal{Y}_i) = \sum_{i=1}^{N} \alpha_i\|\mathcal{G}_{i,(i)}\|_{*,\boldsymbol{w}_i} + \langle \mathcal{X} - \mathcal{G}_i, \mathcal{Y}_i \rangle$$
$$+ \frac{\beta}{2}\|\mathcal{X} - \mathcal{G}_i\|_F^2 + \frac{\beta}{2}\|\mathcal{X} - \mathcal{B}(\boldsymbol{D}\boldsymbol{C})\|_F^2$$
$$+ \lambda\|\boldsymbol{C}\|_0, \quad (18)$$

where $\beta > 0$ is a penalty parameter, and $\mathcal{Y}_i$ is a Lagrangian multiplier. ADMM solves the problem (17) with the following scheme:

$$\begin{cases} \mathcal{X}^{(k+1)} = \underset{\mathcal{X}_{\boldsymbol{\Omega}}=\mathcal{M}_{\boldsymbol{\Omega}}}{\arg\min} \ L(\mathcal{X}, \mathcal{G}_i^{(k)}, \boldsymbol{C}^{(k)}, \boldsymbol{D}^{(k)}, \mathcal{Y}_i^{(k)}), \\ \mathcal{G}_i^{(k+1)} = \underset{\mathcal{G}_i}{\arg\min} \ L(\mathcal{X}^{(k+1)}, \mathcal{G}_i, \boldsymbol{C}^{(k)}, \boldsymbol{D}^{(k)}, \mathcal{Y}_i^{(k)}), \\ \boldsymbol{C}^{(k+1)} = \underset{\boldsymbol{C}}{\arg\min} \ L(\mathcal{X}^{(k+1)}, \mathcal{G}_i^{(k+1)}, \boldsymbol{C}, \boldsymbol{D}^{(k)}, \mathcal{Y}_i^{(k)}), \\ \boldsymbol{D}^{(k+1)} = \underset{\boldsymbol{D}^\top \boldsymbol{D}=\mathbf{I}}{\arg\min} \ L(\mathcal{X}^{(k+1)}, \mathcal{G}_i^{(k+1)}, \boldsymbol{C}^{(k+1)}, \boldsymbol{D}, \mathcal{Y}_i^{(k)}), \\ \mathcal{Y}_i^{(k+1)} = \mathcal{Y}_i^{(k)} + \beta(\mathcal{X}^{(k+1)} - \mathcal{G}_i^{(k+1)}). \end{cases}$$

We develop an iterative alternating algorithm based on the above scheme, which is summarized in Algorithm 1. In the following we will provide details on each step of the algorithm.

Algorithm 1 The proposed method for tensor completion

Input:
The observed tensor $\mathcal{M}$, maximum number of iterations k_{max}, the index set of the known elements $\boldsymbol{\Omega}$.
Output:
The reconstructed data $\mathcal{X}$.
1: Initialization: $\alpha_i s, \lambda, \mathcal{X}_{\boldsymbol{\Omega}}^{(0)} = \mathcal{M}_{\boldsymbol{\Omega}}, \mathcal{X}_{\overline{\boldsymbol{\Omega}}}^{(0)}, \mathcal{G}_i^{(0)} = 0, \mathcal{Y}_i^{(0)} = 0, \boldsymbol{D}^{(0)}, \boldsymbol{C}^{(0)}, \epsilon, \beta$.
2: While not meeting the stop criterion do
3:   Update $\mathcal{X}^{(k+1)}$ by Eq. (20);
4:   Update the weights $\boldsymbol{w}_i^{(k+1)}$ by Eq. (21);
5:   Update $\mathcal{G}_i^{(k+1)}$ by Eq. (23);
6:   Update $\boldsymbol{C}^{(k+1)}$ by Eq. (26);
7:   Update the dictionary $\boldsymbol{D}^{(k+1)}$ by Eq. (29);
8:   Update $\mathcal{Y}_i^{(k+1)}$ by Eq. (30).
9: end while
10: **return** $\mathcal{X}^{(k+1)}$.

C. Detailed Steps of Algorithm

1) Calculation of $\mathcal{X}^{(k+1)}$: When $\mathcal{G}_i^{(k)}, \boldsymbol{C}^{(k)}, \boldsymbol{D}^{(k)}, \mathcal{Y}_i^{(k)}$ are available, the calculation of $\mathcal{X}^{(k+1)}$ is as follows:

$$\mathcal{X}^{(k+1)} = \underset{\mathcal{X}_{\boldsymbol{\Omega}}=\mathcal{M}_{\boldsymbol{\Omega}}}{\arg\min} \ L(\mathcal{X}, \mathcal{G}_i^{(k)}, \boldsymbol{C}^{(k)}, \boldsymbol{D}^{(k)}, \mathcal{Y}_i^{(k)})$$
$$= \underset{\mathcal{X}_{\boldsymbol{\Omega}}=\mathcal{M}_{\boldsymbol{\Omega}}}{\arg\min} \ \sum_{i=1}^{N} \left\langle \mathcal{X} - \mathcal{G}_i^{(k)}, \mathcal{Y}_i^{(k)} \right\rangle$$
$$+ \frac{\beta}{2}\left\|\mathcal{X} - \mathcal{G}_i^{(k)}\right\|_F^2 + \frac{\beta}{2}\|\mathcal{X} - \mathcal{B}(\boldsymbol{D}^{(k)}\boldsymbol{C}^{(k)})\|_F^2$$
$$= \underset{\mathcal{X}_{\boldsymbol{\Omega}}=\mathcal{M}_{\boldsymbol{\Omega}}}{\arg\min} \ \sum_{i=1}^{N} \left\|\mathcal{X} - \mathcal{G}_i^{(k)} + \frac{\mathcal{Y}_i^{(k)}}{\beta}\right\|_F^2$$
$$+ \|\mathcal{X} - \mathcal{B}(\boldsymbol{D}^{(k)}\boldsymbol{C}^{(k)})\|_F^2. \quad (19)$$

This is a least-square problem which has the explicit solution given by

$$\begin{cases} \mathcal{X}_{\overline{\boldsymbol{\Omega}}}^{(k+1)} = \frac{1}{N+1} \cdot \{\sum_{i=1}^{N}(\mathcal{G}_i^{(k)} - \frac{\mathcal{Y}_i^{(k)}}{\beta}) \\ \qquad\qquad + \mathcal{B}(\boldsymbol{D}^{(k)}\boldsymbol{C}^{(k)})\}, \\ \mathcal{X}_{\boldsymbol{\Omega}}^{(k+1)} = \mathcal{M}_{\boldsymbol{\Omega}}, \end{cases} \quad (20)$$

where $\overline{\boldsymbol{\Omega}}$ denotes the complement of $\boldsymbol{\Omega}$, *i.e.* the index set of the missing values.

2) Calculation of $\mathcal{G}_i^{(k+1)}$: After $\mathcal{X}$ is updated, we update the weights $\boldsymbol{w}_i$ in the weighted tensor nuclear norm as follows:

$$\boldsymbol{w}_i^{(k+1)} = \delta/[\sigma(\mathcal{X}_{(i)}^{(k+1)}) + \epsilon], i = 1, 2, \cdots, N. \quad (21)$$

Then, when $\mathcal{X}^{(k+1)}$, $\boldsymbol{C}^{(k)}$, $\boldsymbol{D}^{(k)}$, $\mathcal{Y}_i^{(k)}$ have been calculated, we compute $\mathcal{G}_i^{(k+1)}$ as follows:

$$
\begin{aligned}
\mathcal{G}_i^{(k+1)} =& \underset{\mathcal{G}_i}{\arg\min} \quad L(\mathcal{X}^{(k+1)}, \mathcal{G}_i, \boldsymbol{C}^{(k)}, \boldsymbol{D}^{(k)}, \mathcal{Y}_i^{(k)}) \\
=& \underset{\mathcal{G}_i}{\arg\min} \quad \alpha_i \|\mathcal{G}_{i,(i)}\|_{*, \boldsymbol{w}_i^{(k+1)}} + \left\langle \mathcal{X}^{(k+1)} - \mathcal{G}_i, \mathcal{Y}_i^{(k)} \right\rangle \\
& + \frac{\beta}{2} \|\mathcal{X}^{(k+1)} - \mathcal{G}_i\|_F^2 \\
=& \underset{\mathcal{G}_i}{\arg\min} \quad \alpha_i \|\mathcal{G}_{i,(i)}\|_{*, \boldsymbol{w}_i^{(k+1)}} \\
& + \frac{\beta}{2} \|\mathcal{X}^{(k+1)} - \mathcal{G}_i + \frac{\mathcal{Y}_i^{(k)}}{\beta}\|_F^2 \\
=& \underset{\mathcal{G}_i}{\arg\min} \quad \alpha_i \|\mathcal{G}_{i,(i)}\|_{*, \boldsymbol{w}_i^{(k+1)}} \\
& + \frac{\beta}{2} \|\mathcal{G}_{i,(i)} - (\mathcal{X}^{(k+1)} + \frac{\mathcal{Y}_i^{(k)}}{\beta})_{(i)}\|_F^2.
\end{aligned}
\tag{22}
$$

The above problem has the following closed-form solution:

$$
\mathcal{G}_i^{(k+1)} = fold[\boldsymbol{U}_i S_{\frac{\alpha_i}{\beta} \boldsymbol{w}_i^{(k+1)}}(\boldsymbol{\Sigma}) \boldsymbol{V}_i^\top],
\tag{23}
$$

where $(\mathcal{X}^{(k+1)} + \frac{\mathcal{Y}_i^{(k)}}{\beta})_{(i)} = \boldsymbol{U}_i \boldsymbol{\Sigma} \boldsymbol{V}_i^\top$ is the SVD of $(\mathcal{X}^{(k+1)} + \frac{\mathcal{Y}_i^{(k)}}{\beta})_{(i)}$, and $S_{\boldsymbol{w}}(\boldsymbol{\Sigma})$ is the generalized soft-thresholding operator defined with the weight vector $\boldsymbol{w}$ as follows:

$$
S_{\boldsymbol{w}}(\boldsymbol{\Sigma})(i,j) = \begin{cases} \max\left(\boldsymbol{\Sigma}(i,j) - \boldsymbol{w}(i), 0\right), & i = j; \\ 0, & i \neq j. \end{cases}
\tag{24}
$$

3) Calculation of $\boldsymbol{C}^{(k+1)}$: After $\mathcal{X}^{(k+1)}$, $\mathcal{G}_i^{(k+1)}$, $\boldsymbol{D}^{(k)}$, $\mathcal{Y}_i^{(k)}$ are obtained, $\boldsymbol{C}^{(k+1)}$ is computed by

$$
\begin{aligned}
\boldsymbol{C}^{(k+1)} =& \underset{\boldsymbol{C}}{\arg\min} \quad L(\mathcal{X}^{(k+1)}, \mathcal{G}_i^{(k+1)}, \boldsymbol{C}, \boldsymbol{D}^{(k)}, \mathcal{Y}_i^{(k)}) \\
=& \underset{\boldsymbol{C}}{\arg\min} \frac{\beta}{2} \|\mathcal{X}^{(k+1)} - \mathcal{B}(\boldsymbol{D}^{(k)} \boldsymbol{C})\|_F^2 + \lambda \|\boldsymbol{C}\|_0 \\
=& \underset{\boldsymbol{C}}{\arg\min} \frac{\beta}{2} \|\mathcal{P}[\mathcal{X}^{(k+1)} - \mathcal{B}(\boldsymbol{D}^{(k)} \boldsymbol{C})]\|_F^2 + \lambda \|\boldsymbol{C}\|_0 \\
=& \underset{\boldsymbol{C}}{\arg\min} \frac{\beta}{2} \|\mathcal{P}(\mathcal{X}^{(k+1)}) - \boldsymbol{D}^{(k)} \boldsymbol{C}\|_F^2 + \lambda \|\boldsymbol{C}\|_0 \\
=& \underset{\boldsymbol{C}}{\arg\min} \frac{\beta}{2} \|\boldsymbol{D}^{(k)\top} \mathcal{P}(\mathcal{X}^{(k+1)}) - \boldsymbol{C}\|_F^2 + \lambda \|\boldsymbol{C}\|_0,
\end{aligned}
\tag{25}
$$

This problem can be solved by

$$
\boldsymbol{C}^{(k+1)} = H_{\sqrt{\frac{2\lambda}{\beta}}}\left(\boldsymbol{D}^{(k)\top} \mathcal{P}(\mathcal{X}^{(k+1)})\right).
\tag{26}
$$

where $H_\theta(\cdot)$ denotes the element-wise hard thresholding operator defined by

$$
H_\theta(x) = \begin{cases} x, & |x| > \theta, \\ 0, & otherwise. \end{cases}
\tag{27}
$$

4) Calculation of $\boldsymbol{D}^{(k+1)}$: After obtaining $\mathcal{X}^{(k+1)}$, $\mathcal{G}_i^{(k+1)}$, $\boldsymbol{C}^{(k+1)}$ and $\mathcal{Y}_i^{(k)}$, we compute the dictionary $\boldsymbol{D}^{(k+1)}$ as follows:

$$
\begin{aligned}
\boldsymbol{D}^{(k+1)} =& \underset{\boldsymbol{D}^\top \boldsymbol{D} = \boldsymbol{I}}{\arg\min} L(\mathcal{X}^{(k+1)}, \mathcal{G}_i^{(k+1)}, \boldsymbol{C}^{(k+1)}, \boldsymbol{D}, \mathcal{Y}_i^{(k)}) \\
=& \underset{\boldsymbol{D}^\top \boldsymbol{D} = \boldsymbol{I}}{\arg\min} \|\mathcal{X}^{(k+1)} - \mathcal{B}(\boldsymbol{D} \boldsymbol{C}^{(k+1)})\|_F^2 \\
=& \underset{\boldsymbol{D}^\top \boldsymbol{D} = \boldsymbol{I}}{\arg\min} \|\mathcal{P}[\mathcal{X}^{(k+1)} - \mathcal{B}(\boldsymbol{D} \boldsymbol{C}^{(k+1)})]\|_F^2 \\
=& \underset{\boldsymbol{D}^\top \boldsymbol{D} = \boldsymbol{I}}{\arg\min} \|\mathcal{P}(\mathcal{X}^{(k+1)}) - \boldsymbol{D} \boldsymbol{C}^{(k+1)}\|_F^2,
\end{aligned}
\tag{28}
$$

This problem has an explicit solution given by [34, 54]:

$$
\boldsymbol{D}^{(k+1)} = \boldsymbol{U} \boldsymbol{V}^\top,
\tag{29}
$$

where $\mathcal{P}(\mathcal{X}^{(k+1)}) \boldsymbol{C}^{(k+1)\top} = \boldsymbol{U} \boldsymbol{\Sigma} \boldsymbol{V}^\top$ is the SVD of the matrix $\mathcal{P}(\mathcal{X}^{(k+1)}) \boldsymbol{C}^{(k+1)\top}$.

5) Calculation of $\mathcal{Y}_i^{(k+1)}$: After $\mathcal{X}^{(k+1)}$, $\mathcal{G}_i^{(k+1)}$, $\boldsymbol{C}^{(k+1)}$, $\boldsymbol{D}^{(k+1)}$ are calculated, we compute $\mathcal{Y}_i^{(k+1)}$ by:

$$
\mathcal{Y}_i^{(k+1)} = \mathcal{Y}_i^{(k)} + \beta(\mathcal{X}^{(k+1)} - \mathcal{G}_i^{(k+1)}).
\tag{30}
$$

D. Initialization

To begin, the above algorithm requires the initial estimates of $\boldsymbol{C}$ and $\boldsymbol{D}$, which are obtained as follows. The initial dictionary $\boldsymbol{D}^{(0)} \in \mathbb{R}^{(S_1 \times S_2 \times \cdots \times S_N) \times (S_1 \times S_2 \times \cdots \times S_N)}$ is set to a multi-dimensional DCT dictionary. The initial estimate $\mathcal{X}^{(0)}$ is generated by setting all elements of $\mathcal{X}_{\overline{\Omega}}$ to the mean value of the elements of $\mathcal{M}_\Omega$, and $\mathcal{X}_\Omega = \mathcal{M}_\Omega$. The initial sparse code $\boldsymbol{C}^{(0)}$ is computed by solving

$$
\min_{\boldsymbol{C}} \quad \frac{\beta}{2} \|\mathcal{P}(\mathcal{X}^{(0)}) - \boldsymbol{D}^{(0)} \boldsymbol{C}\|_F^2 + \lambda \|\boldsymbol{C}\|_0.
\tag{31}
$$

Same as (26), this problem has the explicit solution given by

$$
\boldsymbol{C}^{(0)} = H_{\sqrt{\frac{2\lambda}{\beta}}}\left(\boldsymbol{D}^{(0)\top} \mathcal{P}(\mathcal{X}^{(0)})\right).
\tag{32}
$$

IV. EXPERIMENTS

In this section, we evaluated the proposed method by applying it to tensor completion. We consider third-order tensor data of three different types: color images, MRI volume data and video data. The evaluation was conducted using Matlab R2012b (64bit) on a PC with an Intel Core i5-4590 CPU (3.30GHz) and 8GB memory. In the remainder of the paper, we denote our method as WTNNDL.

A. Test Data

We select six test images which are shown in Figure 2 to evaluate the performance of the proposed method on the recovery of color images. The size of each image is $255 \times 255 \times 3$. Regarding the evaluation on MRI, we use the MRI volume dataset called CThead, which is a subset of the datasets in the "University of North Carolina Volume Rendering Test Data Set" archive.[3] The size of the dataset is $252 \times 252 \times 99$. See Figure 3 for some samples from this dataset. Regarding the evaluation on videos, we used

[3]downloaded from http://graphics.stanford.edu/data/voldata/.

four videos which are respectively selected from the classes of Biking, BabyCrawling, JugglingBall and Lunges from the UCF-101 action dataset [61]. In the simulation, we use a third-order tensor representation, and the color videos are converted to grayscale videos. The size of each video is $240 \times 320 \times 80$. See Figure 4 for the test data.
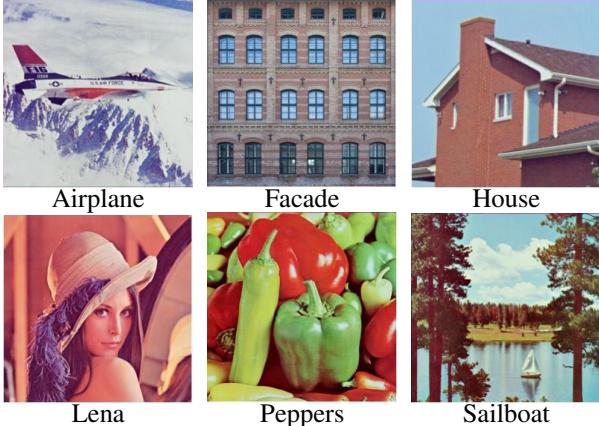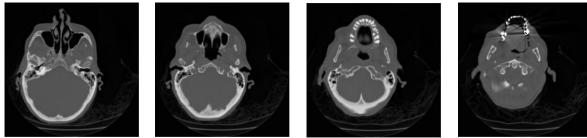

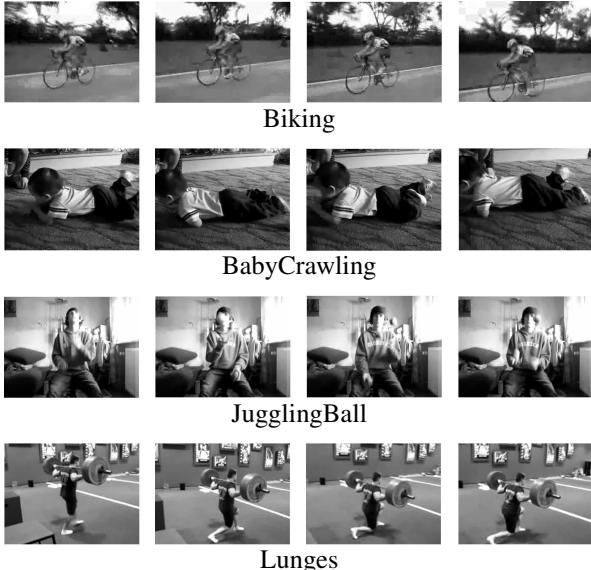Fig. 2. Test images

Fig. 3. Sample slices from MRI dataset

Biking

BabyCrawling

JugglingBall

Lunges

Fig. 4. Test videos

used for measuring the quality of the recovered results. One is the relative square error (RSE) defined as

$$RSE(\mathcal{X}, \mathcal{M}) = 20 * \log_{10}(\|\mathcal{X} - \mathcal{M}\|_F)/\|\mathcal{M}\|_F. \quad (33)$$

The other is the peak signal-to-noise ratio (PSNR) defined as

$$PSNR(\mathcal{X}, \mathcal{M}) = 10 * \log_{10}(TV_{max}^2/\|\mathcal{X} - \mathcal{M}\|_F^2), \quad (34)$$

where V_{max} corresponds to the upper bound of the voxel value in $\mathcal{M}$ which is set to 255 in the experiment, and T denotes the total number of voxels in $\mathcal{M}$.

For comparison, we selected four state-of-the-art approaches which have published results or available codes, including

- SPC-QV [25]: integrating CP decomposition for incomplete tensors and the efficient selection of models for minimizing the tensor rank, with the quadratic variation constraint to promote smoothness.
- HaLRTC [32]: utilizing a general tensor nuclear norm defined on Tucker decomposition for low-rank tensor completion.
- FaLRTC [32]: using a smoothed version of the tensor nuclear norm with an accelerated numerical algorithm for the low-rank approximation of tensor.
- TTNNL1 (TTNN+3DDCT) [35]: using the truncated tensor nuclear norm for low-rank approximation, and a ℓ_1-sparse regularization term combined with the 3-dimensional DCT bases. For clarity, we denote it as TTNN+3DDCT in the following.

To analyze the effectiveness of each component in the proposed approach, we construct three baseline methods for comparison, including

- WTNN: constructed from the proposed method by discarding the sparse coding module. This method is used to test the effectiveness of sparse coding.
- TTNNDL: constructed by replacing the weighted nuclear norm in the proposed method with the truncated tensor nuclear norm.
- WTNN+3DDCT: constructed by discarding the dictionary learning process and fix the dictionary to be the initial dictionary (*i.e.* 3-dimensional DCT).

The parameters of the different algorithms are finely tuned for ensuring fairness of the comparison. For clarity, the components in all the compared methods are listed in Table I.

TABLE I. Components in the compared methods

Methods	Surrogate of rank	Type of dict.
SPC-QV	CP decomposition	n/a
HaLRTC	traditional nuclear norm	n/a
FaLRTC	smoothed version of nuclear norm	n/a
TTNN+3DDCT	truncated nuclear norm	DCT
TTNNDL	truncated nuclear norm	learned dict.
WTNN	weighted nuclear norm	n/a
WTNN+3DDCT	weighted nuclear norm	DCT
WTNNDL	weighted nuclear norm	learned dict.

B. Protocols

Given the original complete data $\mathcal{M}$ (ground truth) and the recovered data $\mathcal{X}$ from the incomplete one, two criteria are

C. Implementation Details

In the implementation of our method, the dictionary was initialized by multidimensional DCT. The procedure will stop

TABLE II. PSNR values of the recovered results on the MRI data (dB)

Sampling Rate	10%	20%	30%	40%	50%	60%	70%	80%	90%
HaLRTC	22.90	26.42	29.11	31.61	34.17	36.74	39.49	42.93	47.36
FaLRTC	22.82	26.42	29.14	31.60	34.14	36.78	39.62	42.99	47.41
TTNN+3DDCT	28.19	30.37	32.50	34.61	36.71	38.92	41.24	43.85	47.53
TTNNDL	29.14	33.44	36.28	38.22	39.97	41.44	43.47	45.15	48.25
WTNN	30.70	34.03	36.50	38.46	40.64	42.58	44.66	47.05	50.73
WTNN+3DDCT	31.46	34.51	37.03	39.12	41.20	43.07	44.96	47.37	50.88
WTNNDL	**31.75**	**34.99**	**37.33**	**39.31**	**41.26**	**43.12**	**45.06**	**47.51**	**51.27**

TABLE III. Negative RSE of the recovered results on the MRI data (-dB)

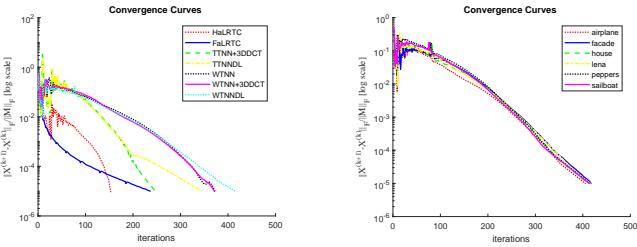| Sampling Rate	10%	20%	30%	40%	50%	60%	70%	80%	90%
HaLRTC	9.59	13.11	15.80	18.29	20.86	23.43	26.18	29.60	34.07
FaLRTC	9.51	13.11	15.83	18.29	20.83	23.47	26.30	29.68	34.11
TTNN+3DDCT	14.88	17.05	19.19	21.30	23.40	25.61	27.92	30.54	34.20
TTNNDL	15.83	20.12	22.96	24.91	26.65	28.13	30.14	31.84	34.94
WTNN	17.39	20.72	23.19	25.14	27.33	29.27	31.34	33.72	37.39
WTNN+3DDCT	18.15	21.20	23.72	25.81	27.89	29.76	31.65	34.07	37.59
WTNNDL	**18.39**	**21.78**	**24.02**	**26.00**	**27.94**	**29.82**	**31.74**	**34.19**	**38.06**

Fig. 5. Left: The convergence curves of different methods on 'Lena'. Right: The convergence curves of WTNNDL on test images.

TABLE IV. PSNR of the recovered 'Airplane' utilizing the proposed methods with l_2-norm, l_1-norm and l_0-norm(dB)

Sampling Rate	l_2-norm	l_1-norm	l_0-norm
40%	31.09	34.14	**34.72**
50%	34.24	36.59	**37.85**
60%	37.52	39.61	**40.41**
70%	40.96	42.77	**43.26**
80%	44.57	46.18	**46.48**
90%	49.22	50.22	**50.74**

when the maximum number of iterations is reached, or there is no further change in the norm of the tensor data. The parameters in the numerical algorithm were set as follows: the maximum number of iterations $k_{max} = 500$, $\epsilon = 10^{-16}$ and $\delta = 1000$, the initial value of the penalty factor $\beta = 10^{-6}$. In the proposed model, the constants $\alpha_i s$ were set as $\alpha_1 = \alpha_2 = \alpha_3 = 1/3$. To obtain the best performance, the parameters S_1, S_2, S_3 and λ of the proposed method are determined by manual tuning. We also applied cross-validation [62, 63] to adjust the parameters. The parameter values are $S_1 = S_2 = S_3 = 3, \lambda = 1.5$ for the color images, $S_1 = S_2 = S_3 = 9, \lambda = 0.02$ for the MRI data, and $\lambda = 0.1, S_1 = S_2 = S_3 = 8$ for the video data.

D. Results and Analysis

For color image completion, the incomplete data is generated by randomly sampling the pixels of the images, with different sampling ratios $\in \{40, 50, 60, 70, 80, 90\%\}$. The results obtained by different methods are plotted in Figure 6. It can be seen that WTNNDL performs better than the competing methods, which demonstrates the capability of the proposed method. In contrast to multi-dimensional DCT, the dictionary learned by the proposed method can discover better local patterns and thus lead to less artifacts. See more visual results in the supplementary materials.

We are interested in the effectiveness of introducing the ℓ_0-norm into our model. Thus, we construct two baseline methods by replacing the ℓ_0-norm with the ℓ_1-norm or the ℓ_2-norm for regularization, and evaluate their performance. Table IV summarizes the PSNR values of the recovered image 'Airplane' using the baseline methods. The results on more test images are included in the supplementary materials. It can be seen from the results that the performance using the ℓ_2-norm decreases significantly, which demonstrates the benefit of using sparse representation in the proposed method. The ℓ_1-norm regularization yields similar performance as the ℓ_0-norm case, with a slight decrease. This is not surprising as the ℓ_1-norm can be viewed as the convex relaxation of the ℓ_0-norm in sparse representation.

Since a theoretical proof regarding the convergence of our algorithm is not available, we study its convergence curves. The plots of the negative logarithm of the stopping criterion $\frac{\|\mathcal{X}^{(k+1)} - \mathcal{X}^{(k)}\|_F}{\|\mathcal{M}\|_F}$ of different methods on the image 'Lena' based on a sampling ratio of 60% are plotted on the left side of Fig. 5.[4] The right side of Fig. 5 shows the convergence curves of the proposed method under different test images. It can be seen from both figures that the value of the stopping criterion becomes stable as the number of iteration increases. This indicates that the algorithm works stably.

In the MRI completion task, we generated the missing entries by randomly sampling the voxels from the original data with the sampling rate varying from 10% to 90%. The results of PSNR and RSE are summarized in Table II and Table III

---

[4]We exclude SPC-QV for comparison as it uses a different stopping criterion.

respectively, and the best results are denoted in bold. It can be seen that our method achieves the best performance among all the compared methods. In particular, we notice that WTNN, WTNN+3DDCT and WTNNDL, which are all based on the weighted tensor nuclear norm, exhibit superior performance when compared to other methods which use the traditional tensor nuclear norm or truncated tensor nuclear norm. This demonstrates that the weighted tensor nuclear norm in the proposed method is more effective than the traditional tensor nuclear norm and truncated tensor nuclear norm.

To demonstrate the effectiveness of the dictionary learning module in the proposed method, we compared the performance of TTNNDL and TTNN+3DDCT. It can be seen that TTNNDL has a PSNR improvement of $0.7-3.8$ dB over TTNN+3DDCT. Moreover, the proposed WTNNDL method achieves a higher improvement over WTNN+3DDCT with a gap of $0.1-0.4$ dB. These improvements come from the capability of the learned dictionary to reveal the detailed structure of the data.

We visualize the recovered MRI slices by different methods in Figure 7, with sampling ratios 10%, 20%, and 40%. It is observed that the reconstructed results by WTNNDL have a higher quality with less artifacts than the compared methods. By comparing the visual results of TTNNDL vs. TTNN+3DDCT, it can also be seen that the results from dictionary learning are better than that using fixed DCT dictionaries.

Regarding the video completion, the PSNR values of the reconstructed video data by different methods are summarized and compared in Table VI under the different sampling rates (SR) from 10% to 90%. On all the four videos, our method consistently performs better than the other compared methods. Similar to the results in MRI data recovery, the results in video completion demonstrates the effectiveness of our method. On one hand, the benefits of using the weighted nuclear norm can be shown by comparing the results of the proposed method with HaLRTC, FaLRTC, TTNN+3DDCT and TTNNDL, where a $1.9-9.7$ dB PSNR improvement on average was observed in WTNNDL. On the other hand, the improvement of using dictionary learning can be demonstrated by observing the $2.2-3.6$ dB PSNR improvement of TTNNDL over TTNN+3DDCT, and the $0.4-1.9$ dB PSNR improvement of WTNNDL over WTNN+3DDCT. To further demonstrate these aspects, we plot the RSE values of TTNN+3DDCT, TTNNDL, WTNN+3DDCT and WTNNDL on the four videos in Figure 8.

In addition, we notice that the results of WTNN+3DDCT are sometimes less satisfactory than those of WTNN from Table VI. One possible reason could be that the multi-dimensional DCT dictionary used in WTNN+3DDCT is constructed by the tensor product of 1D DCT bases. As a result, the atoms in the DCT dictionary have no orientations. Such a property may not be conducive to the representation of many types of data (*e.g.*, image edges are often not horizontal or vertical). In contrast, the proposed method can learn a more meaningful dictionary whose atoms are adaptive to the local structure of the data. Thus, the proposed WTNNDL can achieve better results than both WTNN and WTNN+3DDCT. For visual comparison, we show the recovered results with

sampling rate 30% in Figure 9.

E. Remarks on Computational Complexity

To evaluate the efficiency of the proposed method, we estimate the computational complexity of the test methods as follows.

Given an Nth-order incomplete tensor $\mathcal{X} \in \mathbb{R}^{Q_1 \times Q_2 \times \cdots \times Q_N}$, at each iteration the complexity of HaLRTC is approximated as $O(Q_1 Q_2 \cdots Q_N)$. Regarding TTNN+3DDCT, the computational cost is mainly expended on the SVD operations and multi-dimensional DCT, and results in the complexity $O(Q_1^3 + Q_2^3 + \cdots + Q_N^3 + Q_1 Q_2 \cdots Q_N(2log(Q_1 Q_2 \cdots Q_N) + r_1 + r_2 + \cdots + r_N)$. Here $r_n, n = 1, 2, \cdots, N$, denotes the number of untruncated singular values of the mode-n matricization of the tensor.

In contrast, the complexity of WTNN is $O(2(Q_1^3 + Q_2^3 + \cdots + Q_N^3))$, which is due to the SVD decomposition operations in the updating procedures of $\mathcal{G}_i$ and $\boldsymbol{w}_i$. Thus if $Q_1^3 + Q_2^3 + \cdots + Q_N^3 \ll Q_1 Q_2 \cdots Q_N(2log(Q_1 Q_2 \cdots Q_N) + r_1 + r_2 + \cdots + r_N)$, WTNN is more efficient than TTNN+3DDCT.

Regarding TTNNDL, the complexity of the algorithm is $O(Q_1^3 + Q_2^3 + \cdots + Q_N^3 + Q_1 Q_2 \cdots Q_N S_1 S_2 \cdots S_N)$. The complexity of WTNN3DDCT is $O(2(Q_1^3 + Q_2^3 + \cdots + Q_N^3) + Q_1 Q_2 \cdots Q_N(2log(Q_1 Q_2 \cdots Q_N) + Q_1 + Q_2 + \cdots + Q_N))$. In addition, the complexity of our method WTNNDL is $O(2(Q_1^3 + Q_2^3 + \cdots + Q_N^3) + Q_1 Q_2 \cdots Q_N S_1 S_2 \cdots S_N)$. If $S_1 S_2 \cdots S_N \ll 2log(Q_1 Q_2 \cdots Q_N) + Q_1 + Q_2 + \cdots + Q_N$, WTNNDL is more efficient than WTNN+3DDCT. Otherwise, the complexities of these two methods are in the same order.

In Table V, the average running time of the compared methods is reported with the sampling rates ranging from 10% to 90%. It can be seen that our method requires more time in comparison to HaLRTC and WTNN. This is not surprising as our method has more components which lead to a more complex numerical algorithm. Compared to TTNN+3DDCT and TTNNDL, our method also takes more computational time because additional operations are required mainly due to the updating of weights. Compared to WTNN+3DDCT, the efficiency of the proposed method is acceptable.

TABLE V. The average running time of the compared methods (sec)

Methods	CThead	Biking	Baby.	Juggling	Lunges
HaLRTC	136.6	108.4	114.5	106.4	95.1
TTNN+3DDCT	465.6	415.1	427.7	419.4	416.5
TTNNDL	493.7	433.5	438.1	439.4	434.6
WTNN	320.9	268.5	289.4	264.8	260.3
WTNN+3DDCT	717.3	559.6	568.7	546.3	565.5
WTNNDL	754.4	591.5	597.9	584.1	586.8

V. Conclusion

In this paper, we propose an effective method for tensor completion, which simultaneously finds a globally low-rank approximation and locally sparse representation from the incomplete tensor. The low-rank approximation is performed with weighted nuclear norm regularization, and sparse coding is applied through a dictionary learning process. By exploiting both global information and local patterns, the proposed
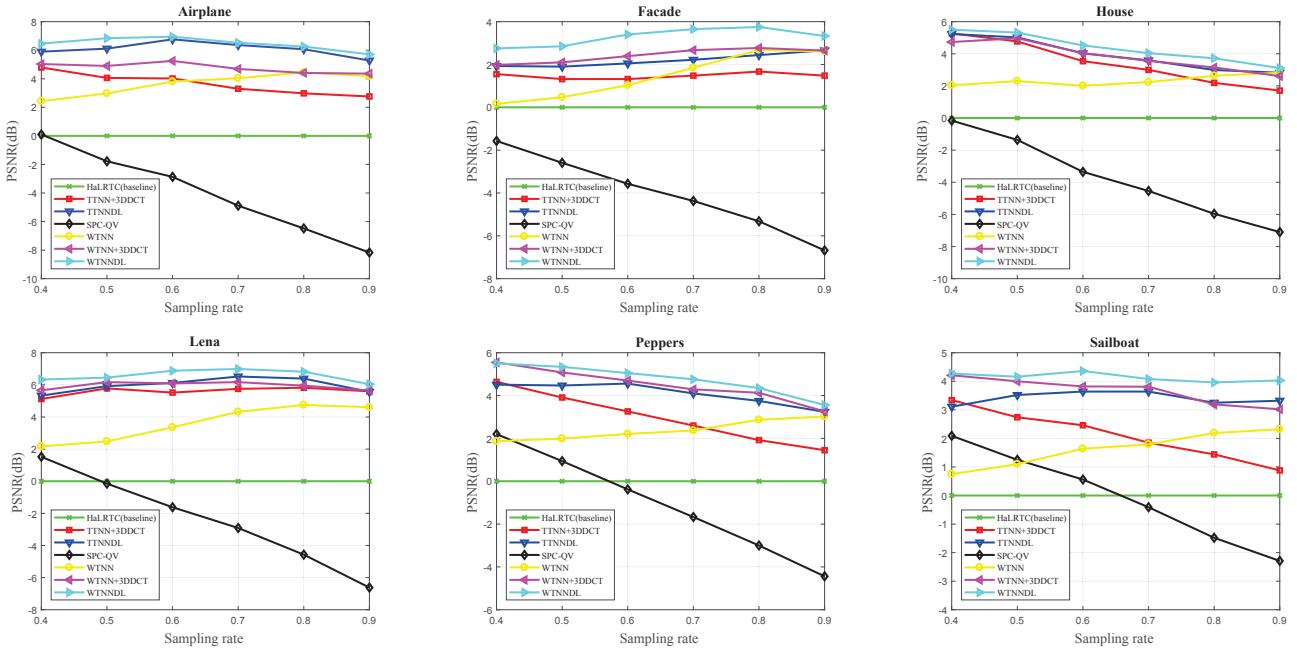
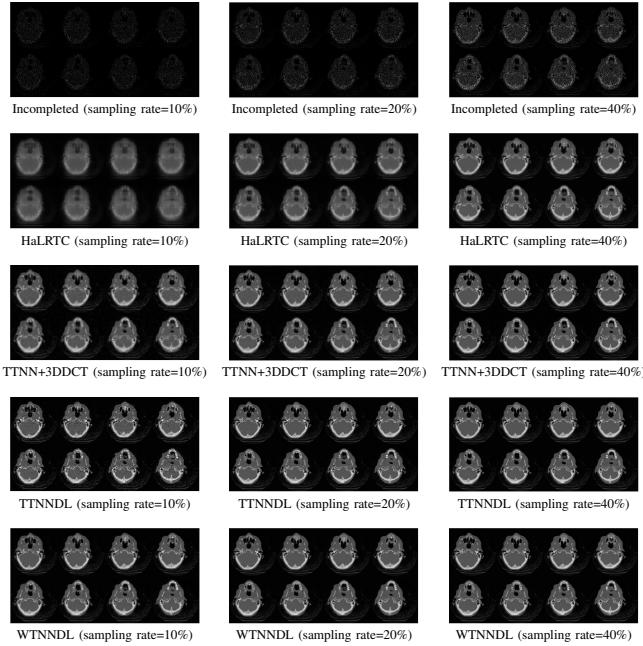Fig. 6. Comparison of PSNR subtracted by baseline(HaLRTC) for the test images obtained using the test methods.

Fig. 7. Experimental results of the different methods on the MRI dataset with different sampling rates.

Fig. 8. RSE of the recovery results by TTNN + 3DDCT, TTNNDL, WTNN + 3DDCT, WTNNDL on the video data.

method can well recover the details from the incomplete data. We evaluated the proposed method on MRI data and real-life videos, and the experimental results show that our method consistently achieves excellent performance.

In the future, we would like to investigate the acceleration of the proposed method, as well as develop numerical algorithms with a faster rate of convergence. In addition, we will further study other surrogates of the tensor rank that are more effective for the tensor completion problem. Finally, we would like to apply the proposed method to other possible applications.

TABLE VI. PSNR values of the recovered results on the video data (dB)

Video	SR (%)	HaLRTC	FaLRTC	TTNN+3DDCT	TTNNDL	WTNN	WTNN+3DDCT	WTNNDL
Biking	10%	18.08	18.05	23.95	24.00	26.94	27.32	**28.34**
	20%	21.86	21.85	26.60	28.94	31.01	30.96	**32.35**
	30%	25.23	25.26	28.95	33.84	33.81	33.57	**35.32**
	40%	28.52	28.54	31.19	36.19	36.27	35.84	**37.84**
	50%	31.69	31.69	33.32	38.10	38.63	38.01	**40.18**
	60%	34.77	34.79	35.57	39.86	40.96	40.19	**42.47**
	70%	37.92	37.90	38.05	41.97	43.72	42.60	**44.91**
	80%	41.36	41.34	41.06	44.56	46.78	45.57	**47.77**
	90%	45.97	45.91	45.47	48.80	50.75	49.74	**51.69**
BabyCrawling	10%	19.10	19.08	24.27	23.75	25.86	26.78	**26.88**
	20%	22.39	22.41	26.37	27.51	29.37	29.83	**30.22**
	30%	24.95	24.97	28.45	31.88	32.25	32.42	**32.87**
	40%	27.38	27.38	30.63	34.56	34.79	34.80	**35.23**
	50%	29.81	29.82	32.94	36.65	37.22	37.11	**37.54**
	60%	32.43	32.39	35.43	38.55	39.68	39.44	**39.95**
	70%	35.33	35.34	38.17	40.54	42.27	41.99	**42.44**
	80%	38.81	38.85	41.43	42.86	45.26	44.83	**45.39**
	90%	43.66	43.62	45.94	46.78	48.98	48.87	**49.44**
JugglingBall	10%	21.47	21.48	27.79	28.31	30.33	31.42	**32.60**
	20%	25.38	25.38	30.79	34.67	34.69	35.38	**36.94**
	30%	28.63	28.63	33.42	38.85	37.97	38.47	**40.10**
	40%	31.77	31.80	35.90	40.72	40.76	41.10	**42.68**
	50%	34.95	34.93	38.35	42.17	43.42	43.60	**45.21**
	60%	38.17	38.18	40.77	43.76	46.20	46.02	**47.58**
	70%	41.65	41.58	43.39	45.90	49.01	48.66	**50.10**
	80%	45.55	45.36	46.54	48.92	52.19	51.74	**52.95**
	90%	50.52	50.21	50.96	53.76	56.31	55.90	**56.78**
Lunges	10%	17.62	17.59	23.70	23.72	26.73	27.08	**28.09**
	20%	21.27	21.29	26.55	28.84	31.47	31.38	**32.62**
	30%	24.74	24.73	29.35	34.34	34.92	34.86	**36.25**
	40%	28.16	28.13	32.19	36.83	37.80	37.78	**39.11**
	50%	31.62	31.60	35.00	38.79	40.54	40.45	**41.74**
	60%	35.18	35.11	37.87	40.54	43.19	43.12	**44.42**
	70%	38.90	38.84	40.85	42.73	46.38	46.07	**47.21**
	80%	43.02	42.94	44.24	45.55	49.99	49.51	**50.42**
	90%	48.27	48.18	48.98	50.40	54.20	53.96	**54.63**

REFERENCES

[1] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Mpca: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Netw.*, vol. 19, no. 1, pp. 18–39, Jan. 2008.

[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] W. K. Wong, Z. Lai, Y. Xu, J. Wen, and C. P. Ho, "Joint tensor feature analysis for visual object recognition," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2425–2436, Nov. 2015.

[4] Y. Quan, Y. Huang, and H. Ji, "Dynamic texture recognition via orthogonal tensor dictionary learning," in *Proc. IEEE Int. Conf. Comput. Vision*, Dec. 2015, pp. 73–81.

[5] Y. Peng, D. Meng, Z. Xu, C. Gao, Y. Yang, and B. Zhang, "Decomposable nonlocal tensor dictionary learning for multi-spectral image denoising," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, Jun. 2014, pp. 2949–2956.

[6] W. Dong, G. Li, G. Shi, X. Li, and Y. Ma, "Low-rank tensor approximation with laplacian scale mixture modeling for mul-tiframe image denoising," in *Proc. IEEE Int. Conf. Comput. Vision*, Dec. 2015, pp. 442–449.

[7] Q. Xie, Q. Zhao, D. Meng, Z. Xu, S. Gu, W. Zuo, and L. Zhang, "Multispectral images denoising by intrinsic tensor sparsity regularization," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, Jun. 2016, pp. 1692–1700.

[8] D. H. J. Poot and S. Klein, "Detecting statistically significant differences in quantitative MRI experiments, applied to diffusion tensor imaging," *IEEE Trans. Med. Imag.*, vol. 34, no. 5, pp. 1164–1176, May 2015.

[9] S. F. Roohi, D. Zonoobi, A. A. Kassim, and J. L. Jaremko, "Dynamic MRI reconstruction using low rank plus sparse tensor decomposition," in *Proc. IEEE Int. Conf. Image Proc.*, Sep. 2016, pp. 1769–1773.

[10] A. Barmpoutis, "Tensor body: Real-time reconstruction of the human body and avatar synthesis from rgb-d," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1347–1356, Oct. 2013.

[11] X. Cao, X. Wei, Y. Han, and D. Lin, "Robust face clustering via tensor decomposition," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2546–2557, Nov. 2015.

[12] N. Qi, Y. Shi, X. Sun, and B. Yin, "Tensr: Multi-dimensional tensor sparse representation," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, Jun. 2016, pp. 5916–5925.

[13] J. Chen, B. Jia, and K. Zhang, "Trifocal tensor-based adaptive visual trajectory tracking control of mobile robots," *IEEE Trans. Cybern.*, vol. PP, no. 99, pp. 1–15, 2017.

[14] B. Ma, L. Huang, J. Shen, and L. Shao, "Discriminative tracking using tensor pooling," *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2411–2422, Nov. 2016.

[15] Y. Sun, J. Gao, X. Hong, B. Mishra, and B. Yin, "Heterogeneous
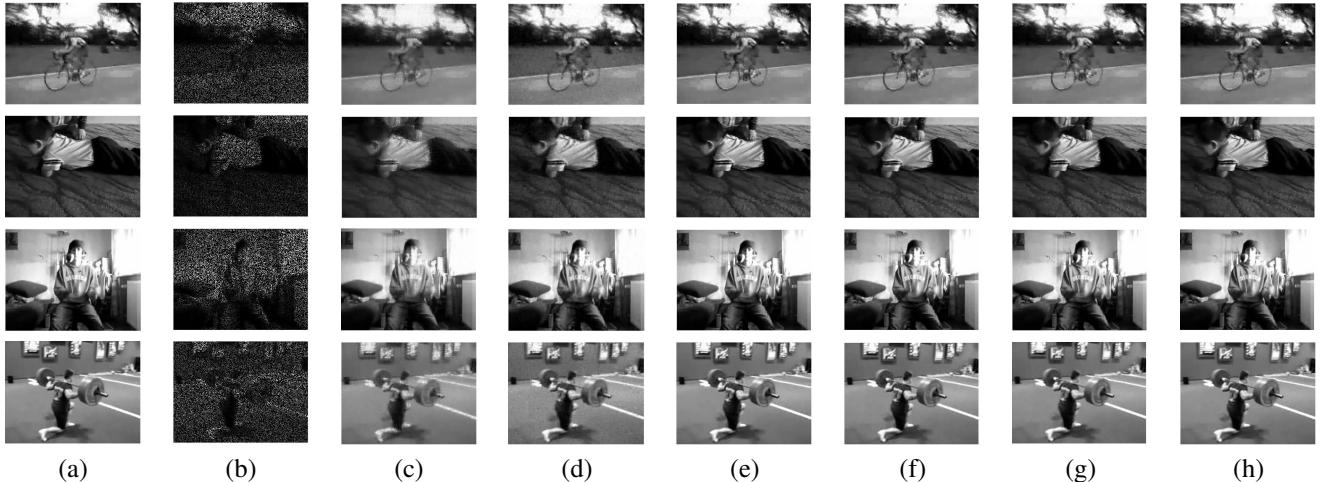
Fig. 9. Completion results under a sampling rate of 30%. (a) Ground Truth. (b) Sampling rate = 0.3. (c) HaLRTC. (d) TTNN + 3DDCT. (e) TTNNDL. (f) WTNN. (g) WTNN + 3DDCT. (h) WTNNDL. From the first row to the last are the results of Biking, BabyCrawling, JugglingBall, and Lunges.

tensor decomposition for clustering via manifold optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 476–489, Mar. 2016.

[16] X. Nie, Y. Yin, J. Sun, J. Liu, and C. Cui, "Comprehensive feature-based robust video fingerprinting using tensor model," *IEEE Trans. Multimedia*, vol. 19, no. 4, pp. 785–796, Apr. 2017.

[17] S. K. Biswas and P. Milanfar, "Linear support tensor machine with lsk channels: Pedestrian detection in thermal infrared images," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4229–4242, Sep. 2017.

[18] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.

[19] K. A. Patwardhan, G. Sapiro, and M. Bertalmio, "Video inpainting under constrained camera motion," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 545–553, Feb. 2007.

[20] Y. Yang, Y. Feng, and J. A. K. Suykens, "Robust low-rank tensor recovery with regularized redescending m-estimator," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1933–1946, Sep. 2016.

[21] 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.

[22] V. De Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1084–1127, 2008.

[23] Y. Liu, F. Shang, H. Cheng, J. Cheng, and H. Tong, "Factor matrix trace norm minimization for low-rank tensor completion," in *Proc. SIAM Int. Conf. Data Mining*. SIAM, 2014, pp. 866–874.

[24] 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.

[25] T. Yokota, Q. Zhao, and A. Cichocki, "Smooth parafac decomposition for tensor completion," *IEEE Trans. Signal Process.*, vol. 64, no. 20, pp. 5423–5436, Oct. 2016.

[26] Y. Liu, F. Shang, L. Jiao, J. Cheng, and H. Cheng, "Trace norm regularized candecomp/parafac decomposition with missing data," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2437–2448, Nov. 2015.

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

[28] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-SVD," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, Jun. 2014, pp. 3842–3849.

[29] X.-Y. Liu, S. Aeron, V. Aggarwal, and X. Wang, "Low-tubal-rank tensor completion using alternating minimization," *arXiv preprint arXiv:1610.01690*, 2016.

[30] 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. Vision and Pattern Recognition*, 2016, pp. 5249–5257.

[31] W. Hu, D. Tao, W. Zhang, Y. Xie, and Y. Yang, "The twist tensor nuclear norm for video completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. PP, no. 99, pp. 1–13, 2017.

[32] 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.

[33] C. Mu, B. Huang, J. Wright, and D. Goldfarb, "Square deal: Lower bounds and improved relaxations for tensor recovery," in *Proc. Int. Conf. Mach. Learning*, 2014, pp. 73–81.

[34] Y. Liu, F. Shang, W. Fan, J. Cheng, and H. Cheng, "Generalized higher order orthogonal iteration for tensor learning and decomposition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2551–2563, Dec. 2016.

[35] Z.-F. Han, C.-S. Leung, L.-T. Huang, and H. C. So, "Sparse and truncated nuclear norm based tensor completion," *Neural Proc. Letters*, vol. 45, no. 3, pp. 729–743, 2017.

[36] K. Hosono, S. Ono, and T. Miyata, "Weighted tensor nuclear norm minimization for color image denoising," in *Proc. IEEE Int. Conf. Image Proc.* IEEE, 2016, pp. 3081–3085.

[37] M. Imaizumi, T. Maehara, and K. Hayashi, "On tensor train rank minimization: Statistical efficiency and scalable algorithm," in *Proc. Advances in Neural Info. Proc. Syst.*, 2017, pp. 3933–3942.

[38] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, p. 717, 2009.

[39] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.

[40] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2980–2998, 2010.

[41] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and bregman iterative methods for matrix rank minimization," *Mathematical Programming*, vol. 128, no. 1, pp. 321–353, 2011.

[42] 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.

[43] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, Jun. 2014, pp. 2862–2869.

[44] C. Lu, J. Tang, S. Yan, and Z. Lin, "Generalized nonconvex nonsmooth low-rank minimization," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, Jun. 2014, pp. 4130–4137.

[45] Q. Liu, Z. Lai, Z. Zhou, F. Kuang, and Z. Jin, "A truncated nuclear norm regularization method based on weighted residual error for matrix completion," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 316–330, Jan. 2016.

[46] F. Shang, J. Cheng, Y. Liu, Z.-Q. Luo, and Z. Lin, "Bilinear factor matrix norm minimization for robust pca: Algorithms and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017.

[47] Y. Xie, W. Zhang, C. Li, S. Lin, Y. Qu, and Y. Zhang, "Discriminative object tracking via sparse representation and online dictionary learning," *IEEE Trans. Cybern.*, vol. 44, no. 4, pp. 539–553, Apr. 2014.

[48] X. Lu, Y. Yuan, and P. Yan, "Alternatively constrained dictionary learning for image superresolution," *IEEE Trans. Cybern.*, vol. 44, no. 3, pp. 366–377, Mar. 2014.

[49] Z. Wen, B. Hou, and L. Jiao, "Discriminative dictionary learning with two-level low rank and group sparse decomposition for image classification," *IEEE Trans. Cybern.*, vol. PP, no. 99, pp. 1–14, 2017.

[50] C. Bao, H. Ji, Y. Quan, and Z. Shen, "Dictionary learning for sparse coding: Algorithms and convergence analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 7, pp. 1356–1369, Jul. 2016.

[51] W. Huang, F. Sun, L. Cao, D. Zhao, H. Liu, and M. Harandi, "Sparse coding and dictionary learning with linear dynamical systems," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, Jun. 2016, pp. 3938–3947.

[52] A. Cherian and S. Sra, "Riemannian dictionary learning and sparse coding for positive definite matrices," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. PP, no. 99, pp. 1–13, 2017.

[53] X. Lu, Y. Yuan, and X. Zheng, "Joint dictionary learning for multispectral change detection," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 884–897, Apr. 2017.

[54] C. Bao, J. F. Cai, and H. Ji, "Fast sparsity-based orthogonal dictionary learning for image restoration," in *Proc. IEEE Int. Conf. Comput. Vision*, Dec. 2013, pp. 3384–3391.

[55] J.-F. Cai, H. Ji, Z. Shen, and G.-B. Ye, "Data-driven tight frame construction and image denoising," *Applied and Computational Harmonic Analysis. Time-Frequency and Time-Scale Analysis, Wavelets, Numerical Algorithms, and Applications*, vol. 37, no. 1, pp. 89–105, 2014.

[56] G. Duan, H. Wang, Z. Liu, J. Deng, and Y. W. Chen, "K-cpd: Learning of overcomplete dictionaries for tensor sparse coding," in *Proc. Int. Conf. Pattern Recognition*, Nov. 2012, pp. 493–496.

[57] Y. Zhang, Z. Jiang, and L. S. Davis, "Discriminative tensor sparse coding for image classification." in *Proc. British Mach. Vision Conf.*, 2013.

[58] Y. Fu, J. Gao, Y. Sun, and X. Hong, "Joint multiple dictionary learning for tensor sparse coding," in *Proc. Int. Joint Conf. Neural Networks (IJCNN)*, Jul. 2014, pp. 2957–2964.

[59] F. Roemer, G. D. Galdo, and M. Haardt, "Tensor-based algorithms for learning multidimensional separable dictionaries," in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Proc.*, May 2014, pp. 3963–3967.

[60] P. Koniusz and A. Cherian, "Sparse coding for third-order super-symmetric tensor descriptors with application to texture recognition," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, Jun. 2016, pp. 5395–5403.

[61] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[62] H. Kim, G. H. Golub, and H. Park, "Missing value estimation for dna microarray gene expression data: local least squares imputation," *Bioinformatics*, vol. 21, no. 2, pp. 187–198, 2004.

[63] C. G. Li and R. Vidal, "A structured sparse plus structured low-rank framework for subspace clustering and completion," *IEEE Trans. Signal Process.*, vol. 64, no. 24, pp. 6557–6570, Dec. 2016.

Yong Du is a Ph.D. candidate in the School of Computer Science and Engineering in South China University of Technology. He received the B.Sc. and Master degree from Jiangnan University in 2011 and 2014 respectively. His research interests include computer vision, machine learning and data mining.
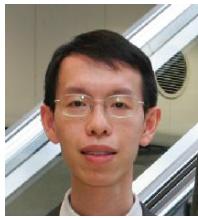
Guoqiang Han is a professor at the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China. He is the head of the School of Computer Science and Engineering in SCUT. He received his B.Sc. degree from the Zhejiang University in 1982, and the Master and Ph.D. degree from the Sun Yat-Sen University in 1985 and 1988 respectively. His research interests include multimedia, computational intelligence, machine learning and computer graphics. He has published over 100 research papers.

Yuhui Quan received his Ph.D. degree in Computer Science from South China University of Technology in 2013. He worked as the postdoc research fellow in Mathematics at National University of Singapore from 2013 to 2016. He is currently the associate professor at School of Computer Science and Engineering in South China University of Technology. His research interests include computer vision, sparse representation, and machine learning.

Zhiwen Yu (S'06-M'08-SM'14) is a Professor in School of Computer Science and Engineering, South China University of Technology, and an adjunct professor in Sun Yat-Sen university. He is a senior member of IEEE, ACM, CCF (China Computer Federation) and CAAI (Chinese Association for Artificial Intelligence). Dr. Yu obtained the PhD degree from City University of Hong Kong in 2008. The research areas of Dr. Yu focus on data mining, machine learning, bioinformatics and pattern recognition. Until now, Dr. Yu has been published more than 100 referred journal papers and international conference papers, including TKDE, TEC, TCYB, TMM, TCVST, TCBB, TNB, PR, INS, Bioinformatics, and so on. Please refer to the homepage for more details: http://www.hgml.cn/yuzhiwen.

Hau-San Wong is currently an associate professor in the Department of Computer Science, City University of Hong Kong. He received the BSc and MPhil degrees in Electronic Engineering from the Chinese University of Hong Kong, and the PhD degree in Electrical and Information Engineering from the University of Sydney. He has also held research positions in the University of Sydney and Hong Kong Baptist University. His research interests include multimedia information processing, multimodal human-computer interaction and machine learning.

C. L. Philip Chen (S'88-M'88-SM'94-F'07) After having worked at U.S. for 23 years as a tenured professor, as a department head and associate dean in two different universities, he is currently the Dean of the Faculty of Science and Technology, University of Macau, Macau, China and a Chair Professor of the Department of Computer and Information Science. His research area is in systems and cybernetics.

Dr. Chen is a Fellow of the IEEE, AAAS, Chinese Association of Automation, and HKIE. He was the President of IEEE Systems, Man, and Cybernetics Society (2012-2013). Currently, he is the Editor-in-Chief of IEEE Transactions on Systems, Man, and Cybernetics: Systems (2014-) and an associate editor of IEEE Transactions on Fuzzy Systems, IEEE Transactions on Cybernetics, and a steering committee member of IEEE Transactions on Big Data. He is also the Chair of TC 9.1 Economic and Business Systems of IFAC (International Federation of Automatic Control). He has been an ABET (Accreditation Board of Engineering and Technology Education) Program Evaluator for Computer Engineering, Electrical Engineering, and Software Engineering programs.

Dr. Chen received 2016 Outstanding Electrical and Computer Engineers (OECE) award from Purdue University, West Lafayette, where he received his Ph.D. degree in 1988 after he received his M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, in 1985.

Jun Zhang (M'02-SM'08-F'16) received the Ph.D. degree in Electrical Engineering from the City University of Hong Kong, Kowloon, Hong Kong, in 2002. Since 2004, he has been with Sun Yat-Sen University, Guangzhou, China, where he is currently an IEEE Fellow, a Cheung Kong Professor. He has authored seven research books and book chapters, and over 100 technical papers in his research areas. He is currently an Associate Editor of the IEEE Transactions on Evolutionary Computation and the IEEE Transactions on Cybernetics. He is the Founding and Current Chair of the IEEE Guangzhou Subsection, the Founding and Current Chair of ACM Guangzhou Chapter.