Factorized Tensor Dictionary Learning for Visual Tensor Data Completion

Ruotao Xu, Yong Xu and Yuhui Quan*

***Abstract*—This paper aims at developing a dictionary-learning-based method for completing the visual tensor data with missing elements. Traditional dictionary learning approaches suffer from very high computational costs when processing high-dimensional tensor data. Some existing approaches for acceleration impose orthogonality constraints or rank-one decompositions on dictionary atoms; however, the expressibility of the resulting dictionary is rather limited. To address such issues, we propose a convolutional analysis model for tensor dictionary learning, where the update of sparse coefficients during dictionary learning is simple and fast. Furthermore, we propose an orthogonality-constrained convolutional factorization scheme for dictionary construction, in which each tensor dictionary atom is factorized by the convolution of two atoms selected from two orthogonal factor dictionaries respectively. This factorization scheme enables us to efficiently learn an expressive dictionary with over-completeness and non-rank-one atoms. Based on our convolutional analysis model and factorization scheme, an effective yet efficient dictionary learning method is proposed for visual tensor completion. Extensive experiments show that, our method not only outperforms existing dictionary-based approaches with relatively-low time cost, but also outperforms recent low-rank approaches.**

***Index Terms*—Tensor dictionary learning, Tensor completion, Convolutional Sparse coding, Factorized dictionary learning**

I. Introduction

TENSORS are a generalization of matrices, which can be used for modeling multi-dimensional data. In multimedia and image processing, there is a large amount of visual data in the form of tensor, such as color images, videos, multi-spectral images, arrays of image patches, and magnetic resonance imaging volume data; see *e.g.* [1–9]. Therefore, the studies on visual tensor data processing are of great values and have plenty of applications, *e.g.* object recognition [10], video description [2–4], image recovery [6–9], medical imaging [5], and compression of deep networks [11].

In many scenarios, there are inevitably a large number of missing elements in the collected visual tensor data. The task of recovering these missing elements is referred to as visual tensor data completion, which is very useful for the subsequent processing on the data. Visual tensor data often contain rich and strong local structures, *e.g.* local space-time patterns of videos. The discovery and exploitation of such structures is one key for the visual tensor data completion [12–14].

One popular tool that has proven its effectiveness in discovering and exploiting the local structures of visual data is the so-called sparse dictionary learning, *i.e.* dictionary learning together with sparse coding, which has the applications ranging from recovery [15–18] to recognition [2, 19]. However, most existing sparse dictionary learning approaches, when used for visual tensor completion, suffer from very high computational costs or low performance. To address these issues, in this paper, we propose a novel dictionary learning scheme for tensor data, based on which a sparse coding model is proposed for visual data completion.

A. Related Work

1) Sparse dictionary learning for tensors: In the last two decades, a lot of sparse dictionary learning methods (*e.g.* [14–17, 20]) have been proposed for the local processing of 2D images, which learn dictionaries from image patches. During the processing, these patch-based methods ignore the consistency among the pixels that lie in different patches but correspond to the same pixel in the image. This may have a negative effect on the result. The convolutional dictionary learning methods [17, 21–23] proposed recently avoid the inconsistency among pixel values across different patches by using the convolutional coding formula instead of the patch-based one. One disadvantage of these methods is that, the convolutional form increases the difficulty of optimization and brings additional time cost in the numerical solver.

By using tensor patches [12] or high-dimensional convolutions [24], both the patch-based and convolutional dictionary learning methods can be straightforwardly extended to processing tensor data. However, due to their limited computational scalability, such approaches are very time-consuming when processing the tensor data of high dimensionality. For acceleration, some approaches (*e.g.* [16]) impose the orthogonality constraint on the dictionary. The orthogonality constraint not only leads to an analytic solution for dictionary update, but also results in an analysis form of sparse coding that has a very fast analytic solver for the sparse approximation subproblem.

Nevertheless, the orthogonality constraint disable the over-completeness of the dictionary, *i.e.* an orthogonal dictionary cannot have atoms whose number is larger than the atom size.

All the authors are with the School of Computer Science and Engineering at South China University of Technology, Guangzhou 510006, China. Yong Xu is also with the Peng Cheng Laboratory, Shenzhen 518055, China, as well as the Communication and Computer Network Laboratory of Guangdong, Guangzhou 510006, China. Yuhui Quan is also with Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, Guangzhou 510006, China. Email: xu.ruotao@mail.scut.edu.cn, yxu@scut.edu.cn, csyhquan@scut.edu.cn.

This work is supported in part by National Natural Science Foundation of China under Grant 61872151, Grant 61672241, and Grant U1611461, in part by Natural Science Foundation of Guangdong Province under Grant 2017A030313376, Grant 2016A030308013, and Grant 2020A1515011128, in part by Science and Technology Program of Guangdong Province under Grant 2019A050510010 and Grant 20140904-160, and in part by Science and Technology Program of Guangzhou under Grant 201802010055.

Asterisk indicates the corresponding author.

In practice, the overcompleteness of the dictionary is important to the success of sparse-coding-based methods in many tasks; see *e.g.* [14, 15, 25]. There are also the so-called analysis dictionary learning methods (*e.g.* [26–28]) which use the analysis operator as the dictionary for learning. These methods have fast solvers for the sparse approximation subproblem, and avoid imposing orthogonality constraints. However, it is challenging to design effective constraints on the dictionary of analysis form, so as to avoid trivial solutions of the learned dictionary while having a fast solver for the dictionary update.

There are some dictionary learning approaches [29–34] that exploit the tensor form of dictionary atoms for acceleration. Most of these approaches impose certain factorizations, such as canonical polyadic decomposition (CPD) and Tucker decomposition (TD), on the dictionary, by which fast solvers for the dictionary update subproblem can be obtained. A seminal work can be traced back to Hazan *et al.*'s method [29]. They applied CPD with non-negativity and sparsity constraints. Duan *et al.* [30] also used CPD but without non-negativity constraints for sparse coding. A similar CPD-based approach was proposed by Dantas *et al.* [32]. Briefly, the CPD factorizes a tensor into a summation of weighted rank-one tensors. Based on CPD, the dictionary is expressed as the outer-product of 1D atoms, and the sparse codes are defined by the coefficients in the summation. From this view, the separable filter learning proposed by Rigamonti *et al.* [31] is a matrix version of the CPD-based convolutional tensor dictionary learning. Zhang *et al.* [33] applied the CPD-based tensor dictionary learning to processing the tensors of non-local similar patch stacks for image restoration. Stevens *et al.* [34] conducted the CPD-based tensor dictionary learning under a probabilistic framework of sparse coding.

In comparison to the CPD-based ones, the TD-based approaches express a tensor into a set of matrices and a small core tensor. With such a decomposition, the dictionary atoms are defined by the factor matrices, and the coding results of the dictionary are encoded in the core tensor; see [35] for a pioneering work. Hawe *et al.* [36] expressed the dictionary as the Kronecker product of two factor dictionaries, which is the matrix form of Zubair *et al.*'s method. Roemer *et al.* [37] and Qi *et al.* [38] developed efficient numerical solvers for the TD-based dictionary learning. Peng *et al.* [1] combined structured sparse coding with TD-based dictionary learning for denoising non-local similar patches of multi-spectral images. Ju *et al.* [39] proposed a non-parametric approach for tensor dictionary learning by combining the TD-based dictionary learning with a stochastic process. Zubair [40] used TD-based dictionary learning for exploiting the block-sparse representations of signals for classification. Fu *et al.* [41] proposed to jointly learn multiple tensor dictionaries based on TD. Quan *et al.* [2] introduced the orthogonal constraint on each factor matrices in TD, resulting in a very fast numerical solver.

Both the CPD-based and TD-based approaches for dictionary learning have shown significant improvement on the computational scalability over the traditional ones. Nevertheless, their resulting dictionary atoms are rank-1 tensors with limited expressive power. One limitation is that a rank-1 atom cannot express local structures with different orientations except the ones with the horizontal/vertical orientation, *e.g.*, an image patch only containing an edge which is neither horizontal nor vertical is not rank-1, whereas structures with orientations are very important for visual data processing, particularly that the orientations become richer as the order of the tensor increases. It is noted that there are some tensor dictionary learning approaches proposed for specific types of data, *e.g.* positive definite matrices [42, 43] and third-order super-symmetric tensor descriptors [44], which are not the focus of this paper.

2) Visual tensor completion: Many tensor completion methods focus on the utilization of the global low rank property of tensor data, and they build up low-rank approximation models for the completion. Mathematically, the rank of a tensor is not uniquely defined. Based on different definitions on tensor rank, there are different kinds of models for low-rank tensor completion; see *e.g.* CPD-based approaches [45–47], TD-based approaches [48–50], and t-SVD-based approaches [51–54].

The low-rank tensor completion methods ignore one important characteristic of visual tensor data, *i.e.*, inclusion of rich and strong local patterns. To exploit such a characteristic for improvement, some approaches were proposed by combining low-rank approximation models with sparse coding. Liu *et al.* [13] induced piece-wise smoothness on the processed tensor using total variation (TV) penalty. Han *et al.* [55] induced local smoothness on the recovered tensor by promoting the sparsity under a discrete cosine transform (DCT) dictionary. With a similar purpose, Xiong *et al.*[8] exploited the local sparsity prior of visual data by transferring the field-of-experts filters learned from natural images to tensor data. Instead of using fixed dictionaries, Du *et al.* [12] employed orthogonal dictionary learning for exploiting data-adaptive features for completion. Yang *et al.* [56] applied the sparse dictionary learning to the vectors along one dimension of tensor. There are also a few methods that exploit the self-recurrence of local structures of visual tensor data. For instance, Xie *et al.* [57] matched and stacked similar blocks of the given tensor and then applied low-rank approximation to the stack for recovery.

With the development of deep learning, there are some deep approaches proposed recently for tensor completion; see *e.g.* [58–60]. However, most of these approaches need a large amount of data for training and the training is computationally infeasible when the dimensionality of the tensor data is high.

B. Motivations and Basic Ideas

Let $\boldsymbol{x}$ denote a given signal and $\{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_M\}$ denote the patches sampled from $\boldsymbol{x}$. Let $\boldsymbol{B}$ denote the dictionary and $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_M]$. For the local processing on $\boldsymbol{x}$, most conventional sparse dictionary learning approaches consider the following synthesis model: $\boldsymbol{Y} = \boldsymbol{B}\boldsymbol{C}$ where $\boldsymbol{C}$ is sparse. Such approaches have two weaknesses: *(i)* The approaches intrinsically assume the independence of $\boldsymbol{y}_1, \cdots, \boldsymbol{y}_M$. However, since two patches may contain the elements that correspond to the same one in $\boldsymbol{x}$,the independence assumption is invalid and may cause performance drop [17]. *(ii)* The synthesis form often results in a time-consuming numerical solver on $\boldsymbol{C}$, as solving $\boldsymbol{C}$ is an NP-hard problem. These two weaknesses become more noticeable when processing tensor data with high order/dimensionality. First, an element of a higher-order tensor is likely to repeat more times in the sampled patches, implying that more dependencies among different patches are ignored. Second, as the dimensionality of the tensor increases, the solving $\boldsymbol{C}$ becomes much more time-consuming when using the pursuit algorithms (*e.g.* [15]).

To overcome the above weaknesses, in this paper, we propose a convolutional analysis dictionary learning approach. Let $\boldsymbol{d}_1, \cdots, \boldsymbol{d}_K$ denote a series of analysis dictionary atoms and $*$ denote the convolution. Basically, we consider the sparse model of the convolutional analysis form as follows:

$$\boldsymbol{d}_i * \boldsymbol{x} = \boldsymbol{c}_i, \quad \text{where } \boldsymbol{c}_i \text{ is sparse for all } i.$$

In our model, we use the convolutional form of sparse coding to avoid the independence assumption in the patch-based form. Meanwhile, the analysis form is used so that a fast algorithm for the subproblem regarding sparse approximation can be developed, even with the convolutional form.

Owing to the analysis form, our dictionary learning model needs an effective dictionary construction scheme which is nontrivial and challenging. Without any constraint, it can be seen that $\boldsymbol{d}_i$ will degenerate to $\boldsymbol{0}$, which is a trivial solution. Constraining $\|\boldsymbol{d}_i\|_2 = 1$ for all i, as [15] does, can avoid $\boldsymbol{d}_i = \boldsymbol{0}$, whereas it is likely to generate another trivial solution: $\boldsymbol{d}_i = \boldsymbol{d}^*, \forall i$, where $\boldsymbol{d}^*$ is the atom that leads to the sparsest solution. There are some constraints investigated in existing literature [26–28] for analysis dictionary learning. However, such constraints may introduce much additional time cost and thus are not suitable for processing tensor data.

To address the above challenge, we propose a convolutional factorization scheme for constructing the tensor dictionary. A p^{th}-order tensor dictionary atom is factorized as the convolution of two (or more) p^{th}-order tensors selected from two (or more) orthogonal tensor dictionaries (called factor dictionaries) respectively. This scheme can prevent the aforementioned trivial solutions in analysis dictionary learning:

- Each factor dictionary is orthogonal and thus must not contain $\boldsymbol{0}$ as its atom. Furthermore, the convolution of two non-zero atoms with finite support will generate nonzero response. Therefore, the resulting composite dictionary cannot have $\boldsymbol{0}$ as its atoms.
- The atoms of an orthogonal factor dictionary are definitely different from one another. Besides, convolving an atom from a factor dictionary with two different atoms from another factor dictionary respectively will yield two different results. Thus, it is impossible that $\boldsymbol{d}_i = \boldsymbol{d}^*, \forall i$.

In addition, the proposed scheme allows the dictionary to be overcomplete, as the convolutional composition of two orthogonal dictionaries is not an orthogonal dictionary and its number of atoms can be larger than the size of its atoms.

Combining the proposed dictionary factorization scheme with the convolutional analysis sparse coding, we propose an efficient sparse coding model for visual tensor completion. Benefiting from the convolutional factorization on dictionary, the proposed model is very effective, with state-of-the-art results achieved in the experiments. Moreover, we present an efficient solver for the proposed model, where each subproblems has an explicit solution that can be efficiently computed:

- Using the commutativity of convolution, a factor dictionary can be moved to the front of the convolutional composition. Then by exploiting the relation between convolution and patch-based representation, the update of the factor dictionary can be formulated as a problem of finding an optimal orthogonal transform (dictionary) in least squares approximation, which has an analytic SVD-based solution.
- As will be shown, benefiting from the orthogonality of the factor dictionaries and their convolutional composition, the transform associated with the composite dictionary has the tight frame property. Such a property allows the update of the recovered tensor to have an explicit solution that can be computed by simple operations including convolutions and summations.
- Due to the use of analysis dictionary learning, the sparse coding subproblem can be done by a simple element-wise thresholding operation.

The efficiency of the proposed approach will be analyzed and demonstrated by experiments.

C. Contributions

The contributions of this paper are two-fold. Firstly, a new tensor dictionary learning scheme is proposed. Compared with the vectorized dictionary learning methods (*e.g.* [15]), the proposed one avoids vectorization of tensor patches which is arguably not suitable for processing tensors [42]. Compared with convolutional dictionary learning approaches [17, 21–23], the proposed one has higher computational scalability. As a byproduct, it involves much fewer parameters when increasing the size of dictionary, which reduces the possibility of overfitting when learning a large dictionary. Compared to the existing orthogonal dictionary learning approaches [12, 16], the proposed one allows learning an overcomplete rather than orthogonal dictionaries. Compared to existing tensor dictionary learning approaches [29–34], our convolutional factorization scheme for dictionary construction avoids their main weakness, *i.e.* imposing rank-1 structures onto dictionary atoms. In addition, the work in this paper also contributes a practical solution to convolutional analysis dictionary learning.

Secondly, we propose an effective and efficient approach to visual tensor completion, which has several advantages over existing dictionary-based methods. Compared with the ones using analytic dictionaries (*e.g.* [55]) or transferred dictionaries (*e.g.* [8]), the proposed approach employs a data-driven dictionary which has better adaptivity. Compared with [12] which iterates orthogonal dictionary learning and low-rank approximation, ours can learn a dictionary with overcompleteness and better expressive power for the completion. In addition, it performs better than [12] without using the low rank prior. Compared with the ones using CPD-based or TD-based dictionary learning, the proposed approach avoids imposing the rank-1 assumption on dictionary atoms and thus can recover more interesting local structures in the data. Last but not least, there are few model parameters to be tuned in the proposed approach. The advantages of the proposed approach are demonstrated with the experiments on four types of visual tensor data. The experimental results show that, the proposed approach not only outperforms existing sparse coding methods and low-rank approximation methods, but even outperforms the methods that combine sparse coding and low-rank approximation.

D. Notations and Organization

Through the paper, unless specified, we use uppercase hollow letters for sets, uppercase calligraphic letters for tensors, uppercase boldfaced letters for matrices, lowercase boldfaced letters for vectors, and normal letters for scalar. For example, $\mathcal{A} \in \mathbb{R}^{S_1\times S_2\times\cdots\times S_P}$ denotes a P^{th}-order tensor of size $S_1 \times S_2 \times \cdots \times S_P$, $\boldsymbol{A} \in \mathbb{R}^{S_1\times S_2}$ denotes a matrix of size $S_1 \times S_2$, and $\boldsymbol{a} \in \mathbb{R}^{S}$ denotes a column vector with S elements. Particularly, $\boldsymbol{I}, \mathbf{0}$ denote the identity matrix and the zero matrix with appropriate sizes respectively. For matrix concatenation, commas are used for adding elements column-wisely and semicolons are used for adding elements row-wisely. Given two equal-size tensors $\mathcal{A}$ and $\mathcal{B}$, their inner product $\langle \mathcal{A}, \mathcal{B}\rangle$ is defined as the sum of the products of the corresponding entries of $\mathcal{A}$ and $\mathcal{B}$. The Euclidean norm of $\mathcal{A}$ is defined by $\|\mathcal{A}\|_2 = \langle \mathcal{A}, \mathcal{A}\rangle^{\frac{1}{2}}$. The ℓ_0-norm of $\mathcal{A}$, denoted by $\|\mathcal{A}\|_0$, is defined as the number of non-zeros in $\mathcal{A}$.

Followings are some operators used throughout the paper. Let $*, \circ, \otimes$ denote the convolution, tensor product and Kronecker product respectively. Let $\mathcal{P}_M : \mathbb{R}^{S_1\times S_2\times\cdots\times S_P} \to \mathbb{R}^{M^P\times(\prod_{p=1}^{P} S_p)}$ denote the patch extraction operator which extracts the patches of size $\underbrace{M \times M \times \cdots \times M}_{P}$ using a sliding window on the given tensor with periodic boundary extension. Let $\mathcal{V} : \mathbb{R}^{S_1\times S_2\times\cdots\times S_N} \to \mathbb{R}^{\prod_{p=1}^{P} S_p}$ denote the vectorization operator on the given tensor. Let $\mathcal{S}(\mathcal{A})$ denote the convolution matrix corresponding to the convolution kernel $\mathcal{A}$ with circular boundary condition, such that $\mathcal{S}(\mathcal{A})\mathcal{V}(\mathcal{B}) = \mathcal{V}(\mathcal{A} * \mathcal{B})$. In 1D case, given $\mathcal{A} = [a_1, \cdots, a_n]$, $\mathcal{S}(\mathcal{A})$ can be expressed as a circulant matrix:

$$\mathcal{S}(\mathcal{A}) = \begin{bmatrix} a_n & \cdots & a_1 & & & \\ & a_n & \cdots & a_1 & & \\ & & \ddots & \ddots & \ddots & \\ a_{n-1} & \cdots & & & & a_n \end{bmatrix}.$$

Recall that discrete convolution is indeed about the weighted summations of neighbors for each element in a tensor. Therefore, for an arbitrary order of tensor, $\mathcal{S}(\mathcal{A})$ can still be expressed as a matrix whose rows store the convolution kernel in appropriate positions. Note that $\mathcal{S}(\mathcal{A})^\top$ is also a convolution matrix whose corresponding kernel is the flipping (mirror symmetry) of $\mathcal{A}$. Let $\delta_{i,j}$ denote the Kronecker delta.

The rest of this paper is organized as follows. Section II presents the proposed scheme for dictionary construction. Section III presents the proposed dictionary-learning-based approach for visual tensor completion. Section IV is for the experimental evaluation. Section V concludes the paper.

II. Dictionary Construction Scheme

Given a tensor $\mathcal{X} \in \mathbb{R}^{S_1\times S_2\cdots\times S_P}$, our goal is about finding a series of analysis dictionary atoms $\mathcal{D}_1, \cdots, \mathcal{D}_M \in \mathbb{R}^{S_1\times S_2,\cdots\times S_P}$ such that the atoms can sparsify $\mathcal{X}$ effectively. We consider the following convolutional analysis dictionary learning model:

$$\min_{\mathcal{D}_1,\cdots,\mathcal{D}_M} \quad \sum_{m=1}^{M} \|\mathcal{D}_m * \mathcal{X}\|_0, \tag{1}$$

where the ℓ_0 norm is employed as the sparsity measurement. Such a minimization model needs some effective constraints on $\mathcal{D}_1, \cdots, \mathcal{D}_M$ to avoid trivial solutions. In the remainder of this section, we first present our dictionary construction scheme that imposes effective constraints on the dictionary. Then, we show an important property of the dictionary constructed by our scheme, which is helpful for understanding the essence of our method as well as for developing efficient numerical solvers for our model. Lastly, we revisit the existing dictionary learning methods and compare them with our approach.

A. Orthogonality-constrained Convolutional Factorization for Dictionary Construction

To avoid trivial solutions in (1) and learn effective dictionaries, we define a P^{th}-order tensor dictionary atom as the convolution of two P^{th}-order tensors (called atoms of a factor dictionary, or factor dictionary atoms). For convenience, we use a double index on the dictionary atoms and define the tensor dictionary as

$$\mathbb{D} = \{\mathcal{D}_{1,1}, \mathcal{D}_{1,2}, \cdots, \mathcal{D}_{I,J}\}. \tag{2}$$

Then two orthogonal factor dictionaries are defined by

$$\begin{aligned} \mathbb{D}_1 &= \{\mathcal{E}_i \in \mathbb{R}^{\overbrace{M\times\cdots\times M}^{P}} : \langle \mathcal{E}_{i_1}, \mathcal{E}_{i_2}\rangle = \delta_{i_1,i_2}, \forall i_1, i_2\}_{i=1}^{I}, \\ \mathbb{D}_2 &= \{\mathcal{F}_j \in \mathbb{R}^{\overbrace{N\times\cdots\times N}^{P}}, \langle \mathcal{F}_{j_1}, \mathcal{F}_{j_2}\rangle = \delta_{j_1,j_2}, \forall j_1, j_2\}_{j=1}^{J}, \end{aligned} \tag{3}$$

where $I = M^P$,$J = N^P$. Instead of directly learning the dictionary $\mathbb{D}$, we learn $\mathcal{D}_{i,j}$ with the following factorization:

$$\mathcal{D}_{i,j} = \mathcal{E}_i * \mathcal{F}_j \quad \forall i, j. \tag{4}$$

Then the size of $\mathcal{D}_{i,j}$ is $\overbrace{(M + N - 1) \times \cdots \times (M + N - 1)}^{P}$. Note that the orthogonality of factor dictionaries $\mathbb{D}_1, \mathbb{D}_2$ does not imply the orthogonality of the composite dictionary $\mathbb{D}$. In fact, we can see that the composite dictionary is overcomplete by noting its atom size is smaller than the number of atoms. Based on above, the model of (1) can be rewritten as

$$\begin{aligned} \min_{\mathbb{D}_1,\mathbb{D}_2} \quad & \sum_{i,j=1}^{I,J} \|\mathcal{D}_{i,j} * \mathcal{X}\|_0 \\ \text{s.t.} \quad & \mathcal{D}_{i,j} = \mathcal{E}_i * \mathcal{F}_j, \ \forall i, j. \end{aligned} \tag{5}$$

Our dictionary construction scheme can be extended to the cases of three or more factor dictionaries. For convenience, we indicate the dictionary atom with multiple indices $(k_1, k_2, \cdots, k_Q)$ and the atom of factor dictionary with single index. Let $\{\mathcal{D}_{k_1,k_2,\cdots,k_Q} \in \mathbb{R}^{N_1\times\cdots\times N_P} | k_q =$

$1, \cdots, \prod_{p=1}^{P} n_{q,p}; q = 1, \cdots, Q\}$ denote the set of dictionary atoms. The factorization is given by

$$\mathcal{D}_{k_1,k_2,\cdots,k_Q} = \mathcal{A}^{(1)}_{k_1} * \mathcal{A}^{(2)}_{k_2} * \cdots * \mathcal{A}^{(Q)}_{k_Q}, \qquad (6)$$

where the factor dictionary atoms

$$\mathcal{A}^{(q)}_{k_q} \in \mathbb{R}^{n_{q,1}\times n_{q,2}\times\cdots n_{q,P}}, k_q = 1, \cdots, \prod_{p=1}^{P} n_{q,p}$$

satisfy $\sum_{q=1}^{Q} n_{q,p} - Q = N_p$ and

$$< \mathcal{A}^{(q)}_r, \mathcal{A}^{(q)}_s >= \delta_{r,s}, \ \forall r, s. \qquad (7)$$

for $q = 1, \cdots, Q$. In practice, we set $Q = 2$ which suffices our need.

B. Interpretation From Tight Frame

For a dictionary $\mathbb{D}$ constructed by the proposed factorization scheme, its associated transform (system) is defined by

$$\boldsymbol{S}_{\mathcal{D}} = [\mathcal{S}(\mathcal{D}_{1,1}); \mathcal{S}(\mathcal{D}_{2,1}); \cdots; \mathcal{S}(\mathcal{D}_{I,J})]. \qquad (8)$$

Since $\mathcal{S}(\mathcal{D}_{i,j})\mathcal{V}(\mathcal{X}) = \mathcal{V}(\mathcal{D}_{i,j} * \mathcal{X})$, $\boldsymbol{S}_{\mathcal{D}}\mathcal{V}(\mathcal{X})$ is equivalent to calculating $\mathcal{D}_{i,j} * \mathcal{X}$ for all i, j. By definition, we have $\sum_{i,j=1}^{I,J} \|\mathcal{D}_{i,j} * \mathcal{X}\|_0 = \|\boldsymbol{S}_{\mathcal{D}}\mathcal{V}(\mathcal{X})\|_0$. The system $\boldsymbol{S}_{\mathcal{D}}$ has one important property called tight frame property.

A system $\boldsymbol{W} \in \mathbb{R}^{m\times n}$ $(m \geq n)$ is a tight frame if and only if $\|\boldsymbol{W}\boldsymbol{x}\|_2 = c\|\boldsymbol{x}\|_2$ for any $\boldsymbol{x} \in \mathbb{R}^n$, where c is a positive constant. Any tight frame system has the perfect reconstruction property $\boldsymbol{W}^\top\boldsymbol{W} = c\boldsymbol{I}$. Let $\boldsymbol{R}_s$ denote the transform for the patch-based representation on a tensor with circular boundary condition, defined by $\boldsymbol{R}_s\mathcal{V}(\mathcal{A}) = \mathcal{V}(\mathcal{P}_s(\mathcal{A}))$. Take the 1D case for instance. Given $\boldsymbol{a} = [a_1, \cdots, a_K]^\top$, $\boldsymbol{R}_3\boldsymbol{a} = [\underline{a_K, a_1, a_2},\ \underline{a_1, a_2, a_3},\ \cdots,\ \underline{a_{K-2}, a_{K-1}, a_K},\ \underline{a_{K-1}, a_K, a_1}]^\top$. The transform $\boldsymbol{R}_s$ is a very basic operation in image processing. Meanwhile $\boldsymbol{R}_s$ is a tight frame satisfying $\|\boldsymbol{R}_s\mathcal{V}(\mathcal{A})\|_2 = s^p\|\mathcal{V}(\mathcal{A})\|_2$ for any p^{th}-order tensor $\mathcal{A}$, as it reduplicates each element in the input with s^p times.

A system is called a convolutional tight frame if it is a tight frame with the following form:

$$[\mathcal{S}(\mathcal{A}_1); \cdots; \mathcal{S}(\mathcal{A}_K)].$$

The following proposition shows that the system $\boldsymbol{S}_{\mathcal{D}}$ associated with the dictionary constructed by our scheme is actually a convolutional tight frame.

Lemma 2.1: Given a set of orthonormal atoms $\{\mathcal{A}_k \in \mathbb{R}^{\overbrace{m\times m\cdots\times m}^{P}}, k = 1, \cdots, m^P : \forall k_1, k_2, \langle\mathcal{A}_{k_1}, \mathcal{A}_{k_2}\rangle = \delta_{k_1,k_2}\}$. Let $\boldsymbol{S}_{\mathcal{A}}$ denote the system constructed by

$$\boldsymbol{S}_{\mathcal{A}} = [\mathcal{S}(\mathcal{A}_1); \cdots; \mathcal{S}(\mathcal{A}_K)], \qquad (9)$$

where $K = m^P$. Then $\boldsymbol{S}_{\mathcal{A}}$ is a convolutional tight frame.

Proof Let $\boldsymbol{A} = [\mathcal{V}(\bar{\mathcal{A}}_1), \mathcal{V}(\bar{\mathcal{A}}_2), \cdots, \mathcal{V}(\bar{\mathcal{A}}_K)]^\top$ where $\bar{\mathcal{A}}_i$ denotes the flipping of $\mathcal{A}_i$. Due to the orthonormality of $\{\mathcal{A}_k\}_k$, $\boldsymbol{A}$ is an orthogonal matrix. There exists a permutation matrix $\boldsymbol{P}$ such that

$$\boldsymbol{S} = \boldsymbol{P}\begin{pmatrix}\boldsymbol{A} & & \\ & \ddots & \\ & & \boldsymbol{A}\end{pmatrix}\boldsymbol{R}_m. \qquad (10)$$

Since $\boldsymbol{P}^\top\boldsymbol{P} = \boldsymbol{I}, \boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{I}, \boldsymbol{R}_m^\top\boldsymbol{R}_m = m^P\boldsymbol{I}$, we have $\boldsymbol{S}^\top\boldsymbol{S} = m^P\boldsymbol{I}$ which completes the proof.

Proposition 2.2: Given two orthogonal dictionaries

$$\{\mathcal{E}_i \in \mathbb{R}^{\overbrace{M\times\cdots\times M}^{P}} : \langle\mathcal{E}_{i_1}, \mathcal{E}_{i_2}\rangle = \delta_{i_1,i_2}, \forall i_1, i_2\}_{i=1}^{I},$$

$$\{\mathcal{F}_j \in \mathbb{R}^{\overbrace{N\times\cdots\times N}^{P}} : \langle\mathcal{F}_{j_1}, \mathcal{F}_{j_2}\rangle = \delta_{j_1,j_2}, \forall j_1, j_2\}_{j=1}^{J},$$

where $I = M^P, J = N^P$. Let $\mathbb{D}$ denote the dictionary constructed by

$$\mathbb{D} = \{\mathcal{D}_{i,j} : \mathcal{D}_{i,j} = \mathcal{E}_i * \mathcal{F}_j\}_{i=1,j=1}^{I,J}. \qquad (11)$$

Then its associated system

$$\boldsymbol{S}_{\mathcal{D}} = [\mathcal{S}(\mathcal{D}_{1,1}); \mathcal{S}(\mathcal{D}_{2,1}); \cdots; \mathcal{S}(\mathcal{D}_{I,J})] \qquad (12)$$

is a convolutional tight frame satisfying $\boldsymbol{S}_{\mathcal{D}}^\top\boldsymbol{S}_{\mathcal{D}} = c\boldsymbol{I}$ where $c = (MN)^P$.

Proof Let $\boldsymbol{S}_{\mathcal{E}} = [\mathcal{S}(\mathcal{E}_1); \cdots; \mathcal{S}(\mathcal{E}_I)]$ and $\boldsymbol{S}_{\mathcal{F}} = [\mathcal{S}(\mathcal{F}_1); \cdots; \mathcal{S}(\mathcal{F}_J)]$. For any i, j, we have

$$\begin{aligned}\mathcal{S}(\mathcal{D}_{i,j})\mathcal{V}(\mathcal{X}) &= \mathcal{D}_{i,j} * \mathcal{X} = \mathcal{E}_i * \mathcal{F}_j * \mathcal{X} \\ &= \mathcal{S}(\mathcal{E}_i)\mathcal{V}(\mathcal{F}_j * \mathcal{X}) = \mathcal{S}(\mathcal{E}_i)\mathcal{S}(\mathcal{F}_j)\mathcal{V}(\mathcal{X}). \qquad (13)\end{aligned}$$

Therefore, we can rewrite $\boldsymbol{S}_{\mathcal{D}}$ as

$$\boldsymbol{S}_{\mathcal{D}} = \begin{pmatrix}\mathcal{S}(\mathcal{E}_1)\mathcal{S}(\mathcal{F}_1) \\ \mathcal{S}(\mathcal{E}_2)\mathcal{S}(\mathcal{F}_1) \\ \vdots \\ \mathcal{S}(\mathcal{E}_I)\mathcal{S}(\mathcal{F}_J)\end{pmatrix} = \begin{pmatrix}\boldsymbol{S}_{\mathcal{E}}\mathcal{S}(\mathcal{F}_1) \\ \boldsymbol{S}_{\mathcal{E}}\mathcal{S}(\mathcal{F}_2) \\ \vdots \\ \boldsymbol{S}_{\mathcal{E}}\mathcal{S}(\mathcal{F}_J)\end{pmatrix} = (\boldsymbol{I}\otimes\boldsymbol{S}_{\mathcal{E}})\boldsymbol{S}_{\mathcal{F}}. \qquad (14)$$

From Lemma. 2.1, we have $\boldsymbol{S}_{\mathcal{E}}^\top\boldsymbol{S}_{\mathcal{E}} = M^P\boldsymbol{I}$ and $\boldsymbol{S}_{\mathcal{F}}^\top\boldsymbol{S}_{\mathcal{F}} = N^P\boldsymbol{I}$. Then it is straightforward to show that $\boldsymbol{S}_{\mathcal{D}}^\top\boldsymbol{S}_{\mathcal{D}} = (MN)^P\boldsymbol{I}$, which completes the proof.

From Proposition 2.2, learning a dictionary under our factorization scheme can be viewed as learning a multi-level convolutional tight system for sparsifying the input tensor data. Convolutional tight frame systems have many good properties in local image processing; see *e.g.* [61]. One benefit from the tight frame property of $\boldsymbol{S}_{\mathcal{D}}$ is that we can avoid computing $(\boldsymbol{S}_{\mathcal{D}}^\top\boldsymbol{S}_{\mathcal{D}})^{-1}$ as it is proportional to $\boldsymbol{I}$. This is useful for developing related numerical solvers; see Sec. III-B.

C. Revisit of Existing Dictionary Construction Schemes

The proposed dictionary construction scheme has its merits over the existing ones. For comparison, recall the notations used by (1). Let $\boldsymbol{x} = \mathcal{V}(\mathcal{X}), \boldsymbol{d}_m = \mathcal{V}(\mathcal{D}_m) \in \mathbb{R}^{1\times(\prod_{p=1}^{P} S_p)}$ denote the vectorization of input data $\mathcal{X}$ and dictionary atom $\mathcal{D}_m$ respectively. Let $\boldsymbol{D} = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_M]$. In the following, we summarize the dictionary construction schemes of existing approaches, as well as their pros and cons:

- Plain [14, 15]: Represent $\mathcal{X}, \{\mathcal{D}_m\}_m$ by $\boldsymbol{x}, \boldsymbol{D}$ and call existing matrix-oriented dictionary learning algorithms. Their computational costs are huge in processing big tensors.
- Orthogonality [16]: Similar to above but add $\boldsymbol{D}^\top\boldsymbol{D} = \boldsymbol{I}$. On one hand, such an orthogonality constraint can lead to explicit solutions in both sparse approximation and

dictionary update, with computational efficiency noticeably improved. On the other hand, the orthogonality constraint limits the expressive power of model, as the dictionary cannot be overcomplete.

- CPD-Separability [30, 32–34]: $\mathcal{D}_m := \boldsymbol{d}_1^{(m)} \circ \boldsymbol{d}_2^{(m)} \circ \cdots \circ \boldsymbol{d}_P^{(m)}$ for all m. Each dictionary atom is factorized into one-dimensional atoms, allowing the dictionary learning to be conducted separately along each dimension of the tensor. Then, the dictionary learning processing can be significantly accelerated. However, the factorization also imposes the independence among the dimensions of the tensor, limiting the expressive power of the learned dictionary. For instance, the learned dictionary atoms are all rank-1 tensors without orientations. See Fig. 1 for an illustration.
- TD-Separability [1, 35, 37–41]: $\boldsymbol{D} := \boldsymbol{D}_1 \otimes \boldsymbol{D}_2 \otimes \cdots \otimes \boldsymbol{D}_P$. Similar to the CPD case, such a factorization also imposes the rank-1 property on each dictionary atom, which disables the learning of oriented atoms. See also Fig. 1.
- Separability + Orthogonality [2]: $\boldsymbol{D} := \boldsymbol{D}_1 \circ \boldsymbol{D}_2 \cdots \circ \boldsymbol{D}_P$ and $\boldsymbol{D}_p^\top \boldsymbol{D}_p = \boldsymbol{I}$ for all p. This scheme enjoys both advantages of the two structures for further acceleration; however, it also inherits the disadvantages of the both.

In summary, the existing approaches have limitations in either computational efficiency or expressive power.

In comparison, the proposed model uses the convolutional composition of $\mathcal{E}_i$ and $\mathcal{F}_j$ which avoids imposing the rank-1 property onto the atoms, and thus it overcomes the main weakness in most existing tensor dictionary learning approaches. The orthogonal constraints on both $\{\mathcal{E}_i\}_i$ and $\{\mathcal{F}_j\}_j$, as can be seen in the next section, result in the explicit solution with efficient computation to each subproblem in dictionary learning. See Fig. 1 for an example of the learned dictionary by the proposed method and its comparison to other approaches.

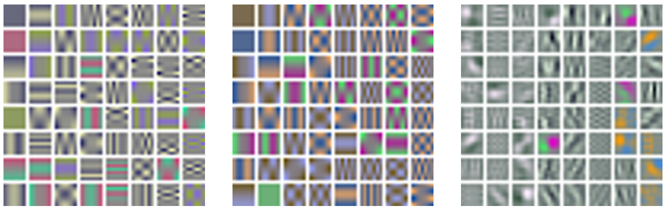
Note that the cardinal of $\mathbb{D}$ is $I \times J = (MN)^P$, which is larger than $(M+N-1)^P$, the size of $\mathcal{D}_{i,j}$, for M, $N \geq 3$. This implies that our approach allows learning overcomplete dictionaries, which distinguishes itself from the existing orthogonal dictionary learning approaches. Also note that, the number of unknowns in our approach is $M^{2P} + N^{2P}$. In comparison, using a dictionary of the same size in conventional dictionary learning involves $(M+N-1)^P(MN)^P$ unknowns, which is much larger than ours and may encounter the curse of dimensionality when M, N, P are large. In other words, our approach introduces much fewer parameters when increasing the size of the dictionary, with lower possibility of overfitting when scaling to big tensors.

III. Visual Tensor Completion Model

This section presents a dictionary-learning-based approach for visual tensor completion, which is built upon the dictionary construction scheme proposed in Section II.

A. Model

Let $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{S_1 \times \cdots \times S_P}$ denote the truth tensor and its observation with missing elements. Let Ω denote the index set of the known elements in $\mathcal{Y}$ and $\bar{\Omega}$ denote the index set of the missing elements. The goal of tensor completion is to recover $\mathcal{Y}_{\bar{\Omega}}$ from $\mathcal{Y}_\Omega$, or equivalently, recover $\mathcal{X}$ from $\mathcal{Y}$. Considering the local characteristic of a visual tensor, we propose a sparse-dictionary-learning-based completion model as follows:



(a) CPD-based dictionary learning [30]

(b) TD-based dictionary learning [2]

(c) Proposed dictionary learning model (1)

Fig. 1. Dictionaries learned on the color image 'Lena' by different methods. The top 81 atoms with highest responses are shown. It can be seen that our learned dictionary atoms have much richer orientations than the other methods.

$$\begin{aligned} &\min_{\mathcal{X},\mathbb{D}} \sum_{i,j=1}^{I,J} \|\mathcal{D}_{i,j} * \mathcal{X}\|_0 \\ \text{s.t.}\quad &\mathcal{X}_\Omega = \mathcal{Y}_\Omega,\ \mathcal{D}_{i,j} = \mathcal{E}_i * \mathcal{F}_j,\ \forall i,j, \\ &\langle \mathcal{E}_{i_1}, \mathcal{E}_{i_2} \rangle = \delta_{i_1,i_2}, \langle \mathcal{F}_{j_1}, \mathcal{F}_{j_2} \rangle = \delta_{j_1,j_2}, \forall i_i, i_2, j_1, j_2. \end{aligned} \tag{15}$$

The above tensor completion model is built upon the convolutional analysis dictionary learning model of (1). Note that when $\mathcal{Y}$ has no missing elements (*i.e.* $\bar{\Omega} = \emptyset$), the above model becomes the dictionary learning model defined in (1).

By using the convolution under a learned dictionary for local processing, the model of (15) can exploit the local structures effectively for the completion. Furthermore, benefiting from the proposed factorization scheme on the dictionary, the model can learn an overcomplete dictionary with oriented atoms from the damaged input $\mathcal{Y}$ for the recovery of missing elements. In addition, the model can be efficiently solved, which is shown in the next. It is worth mentioning that the proposed model has few parameters to be tuned. The parameters needed to be set are only the sizes of factor dictionaries $\{\mathcal{E}_i\}_{i=1}^I, \{\mathcal{F}_j\}_{j=1}^J$.

B. Numerical Algorithm

For solving the problem (15), we rewrite it into

$$\begin{aligned} &\min_{\mathcal{X},\{\mathcal{D}_{i,j}\}} \sum_{i,j=1}^{I,J} \|\mathcal{C}_{i,j}\|_0 \\ \text{s.t.}\quad &\mathcal{C}_{i,j} = \mathcal{D}_{i,j} * \mathcal{X},\ \mathcal{X}_\Omega = \mathcal{Y}_\Omega,\ \mathcal{D}_{i,j} = \mathcal{E}_i * \mathcal{F}_j, \forall i,j, \\ &\langle \mathcal{E}_{i_1}, \mathcal{E}_{i_2} \rangle = \delta_{i_1,i_2}, \langle \mathcal{F}_{j_1}, \mathcal{F}_{j_2} \rangle = \delta_{j_1,j_2},\ \forall i_1, i_2, j_1, j_2. \end{aligned} \tag{16}$$

Define $L(\mathcal{X}, \{\mathcal{C}_{i,j}\}, \{\mathcal{D}_{i,j}\}) = \sum_{i,j=1}^{I,J} \|\mathcal{C}_{i,j}\|_0 + \rho\|\mathcal{C}_{i,j} - \mathcal{D}_{i,j} * \mathcal{X}\|_2^2$. Then the problem (16) is solved by

$$\left\{\begin{array}{rcl} \mathcal{X}^{(t+1)} & = & \arg\min\limits_{\mathcal{X}} L(\mathcal{X}, \{\mathcal{C}_{i,j}^{(t)}\}, \{\mathcal{D}_{i,j}^{(t)}\}) \\ \{\mathcal{C}_{i,j}^{(k+1)}\} & = & \arg\min\limits_{\{\mathcal{C}_{i,j}\}} L(\mathcal{X}^{(t+1)}, \{\mathcal{C}_{i,j}\}, \{\mathcal{D}_{i,j}^{(t)}\}) \\ \{\mathcal{D}_{i,j}^{(t+1)}\} & = & \arg\min\limits_{\{\mathcal{D}_{ij}\}} L(\mathcal{X}^{(t+1)}, \{\mathcal{C}_{i,j}^{(t+1)}\}, \{\mathcal{D}_{i,j}\}) \\ \rho^{(t+1)} & = & \gamma \rho^{(t)} \end{array}\right. , \tag{17}$$

subject to the constraints in (16), for $t = 0, 1, \cdots$ and $\gamma > 1$ (set to 1.05 in our implementation). The solution to each subproblem of (17) is given in the following.

Update of recovered tensor. The subproblem regarding $\mathcal{X}$ in (17) is as follows:

$$\min_{\mathcal{X}} \quad \sum_{i,j=1}^{I,J} \|\mathcal{D}_{i,j}^{(t)} * \mathcal{X} - \mathcal{C}_{i,j}^{(t)}\|_2^2, \tag{18}$$

subject to $\mathcal{X}_\Omega = \mathcal{Y}_\Omega$. Regarding this problem, its solution is revealed by the following proposition.

Proposition 3.1: Let $\mathcal{D}_{i,j}$ denote the dictionary defined in (15), and $\bar{\mathcal{D}}_{i,j}$ denote the flipping of $\mathcal{D}_{i,j}$. The problem

$$\min_{\mathcal{X}} \quad \sum_{i,j=1}^{I,J} \|\mathcal{D}_{i,j} * \mathcal{X} - \mathcal{C}_{i,j}\|_2^2, \tag{19}$$

has the unique solution given by

$$\mathcal{X} = \frac{1}{M^P N^P} \sum_{i,j=1}^{I,J} \bar{\mathcal{D}}_{i,j} * \mathcal{C}_{i,j}, \tag{20}$$

Proof Let $\boldsymbol{S}_\mathcal{D} = [\mathcal{S}(\mathcal{D}_{1,1}); \cdots; \mathcal{S}(\mathcal{D}_{I,J})]$. Then, we have

$$\sum_{i,j=1}^{I,J} \|\mathcal{D}_{i,j} * \mathcal{X} - \mathcal{C}_{i,j}\|_2^2 = \|\boldsymbol{S}_\mathcal{D}\mathcal{V}(\mathcal{X}) - \boldsymbol{C}\|_2^2, \tag{21}$$

where $\boldsymbol{C} = [\mathcal{V}(\mathcal{C}_{1,1}), \mathcal{V}(\mathcal{C}_{2,1}), \cdots, \mathcal{V}(\mathcal{C}_{I,J})]$. Setting the derivative of $\|\boldsymbol{S}_\mathcal{D}\mathcal{V}(\mathcal{X}) - \boldsymbol{C}\|_2$ over $\mathcal{V}(\mathcal{X})$ to zero yields

$$\mathcal{V}(\mathcal{X}) = (\boldsymbol{S}_\mathcal{D}^\top \boldsymbol{S})^{-1} \boldsymbol{S}_\mathcal{D}^\top \boldsymbol{C} = \frac{1}{M^P N^P} \boldsymbol{S}_\mathcal{D}^\top \boldsymbol{C}, \tag{22}$$

where we use $\boldsymbol{S}_\mathcal{D}^\top \boldsymbol{S}_\mathcal{D} = \frac{1}{M^P N^P}\boldsymbol{I}$ from Proposition 2.2. Then, by definition we have

$$\mathcal{X} = \mathcal{V}^{-1}(\boldsymbol{S}_\mathcal{D}^\top \boldsymbol{C}) = \frac{1}{M^P N^P} \sum_{i,j}^{I,J} \bar{\mathcal{D}}_{i,j} * \mathcal{C}_{i,j}. \tag{23}$$

The proof is completed.

Based on Proposition 3.1, we solve the problem of (18) by calculating $\mathcal{X}^{(t+1)} = \sum_{i,j} \bar{\mathcal{D}}_{i,j}^{(t)} * \mathcal{C}_{i,j}^{(t)} / (M^P N^P)$, where $\bar{\mathcal{D}}^{(t)}$ is the flipping of $\mathcal{D}^{(t)}$. Then we project the solution to the constraint $\mathcal{X}_\Omega = \mathcal{Y}_\Omega$ by updating $\mathcal{X}_\Omega^{(t+1)} = \mathcal{Y}_\Omega$.

Update of sparse code tensors. The problem regarding $\mathcal{C}_{i,j}$ in (17) is as follows:

$$\min_{\mathcal{C}_{i,j}} \quad \|\mathcal{C}_{i,j}\|_0 + \rho\|\mathcal{D}_{i,j}^{(t)} * \mathcal{X}^{(t+1)} - \mathcal{C}_{i,j}\|_2^2 \tag{24}$$

for all i, j. This problem has the closed-form solution given by Proposition 3.2.

Proposition 3.2: The solution of (24) is given by

$$\mathcal{C}_{i,j} = \mathcal{T}_{\sqrt{\frac{1}{\rho}}}(\mathcal{D}_{i,j}^{(t)} * \mathcal{X}^{(t+1)}), \tag{25}$$

where $\mathcal{T}_\lambda(\cdot)$ is the element-wise hard thresholding operation defined by

$$(\mathcal{T}_\lambda(\mathcal{A}))(i_1, i_2, \cdots, i_P) = \mathcal{T}_\lambda(\mathcal{A}(i_1, i_2, \cdots, i_P)), \tag{26}$$

$$\mathcal{T}_\lambda(x) = x \text{ if } |x| \geq \lambda \text{ and } 0 \text{ otherwise.} \tag{27}$$

for a P^{th}-order tensor $\mathcal{A}$ and a scalar x.

Proof For simplicity, we use linear indexing on the tensor, *i.e.* $\mathcal{A}(k)$ denotes the k-th element of the tensor $\mathcal{A}$. Let $\mathbb{I} : \mathbb{R} \to \{0, 1\}$ denote the function outputting 1 if the input condition is true and 0 otherwise. Define $\mathcal{Z}_{i,j} = \mathcal{D}_{i,j}^{(t)} * \mathcal{X}^{(t+1)}$. The problem (24) can be rewritten as

$$\min_{\mathcal{C}_{i,j}} \sum_k (\mathbb{I}(\mathcal{C}_{i,j}(k) \neq 0) + \rho(\mathcal{Z}_{i,j}(k) - \mathcal{C}_{i,j}(k))^2). \tag{28}$$

In above, each term in the summation over k is independent of one another. Thus, the problem of (28) is equivalent to solving a series of subproblems simultaneously. That is, for each k,

$$\min_{\mathcal{C}_{i,j}(k)} \quad \mathbb{I}(\mathcal{C}_{i,j}(k) \neq 0) + \rho(\mathcal{Z}_{i,j}(k) - \mathcal{C}_{i,j}(k))^2. \tag{29}$$

For each k, the problem is a single-variable minimization. It can be solved by simple comparison and calculation, and the solution is given by

$$\mathcal{C}_{i,j}(k) = \begin{cases} \mathcal{Z}_{i,j}(k), & |\mathcal{Z}_{i,j}(k)| > 1/\sqrt{\rho} \\ 0, & \text{otherwise} \end{cases}, \tag{30}$$

for all k, which is equivalent to (25). The proof is done.

Update of Dictionaries. The update of the dictionary atoms $\mathcal{D}_{i,j}$ is as follows:

$$\begin{aligned} \min_{\mathcal{D}_{i,j}} \quad & \sum_{i,j=1}^{I,J} \|\mathcal{D}_{i,j} * \mathcal{X}^{(t+1)} - \mathcal{C}_{i,j}^{(t+1)}\|_F^2 \\ \text{s.t.} \quad & \mathcal{D}_{i,j} = \mathcal{E}_i * \mathcal{F}_j, \langle \mathcal{E}_{i_1}, \mathcal{E}_{i_2} \rangle = \delta_{i_1,i_2}, \\ & \langle \mathcal{F}_{j_1}, \mathcal{F}_{j_2} \rangle = \delta_{j_1,j_2}, \forall i, j, i_1, i_2, j_1, j_2. \end{aligned} \tag{31}$$

It can be seen that the update of $\mathcal{D}_{i,j}$ involves the update of the factor atoms $\mathcal{E}_i$ and $\mathcal{F}_j$. We sequentially update $\mathcal{E}_i, \mathcal{F}_j$ and then calculate $\mathcal{D}_{i,j} = \mathcal{E}_i * \mathcal{F}_j$. The update of $\mathcal{E}_i$ is about solving

$$\begin{aligned} \min_{\{\mathcal{E}_i\}_i} \quad & \sum_{i=1}^{I} \|\mathcal{E}_i * \mathcal{F}_j^{(t)} * \mathcal{X}^{(t+1)} - \mathcal{C}_{i,j}^{(t+1)}\|_2^2 \\ \text{s.t.} \quad & \langle \mathcal{E}_{i_1}, \mathcal{E}_{i_2} \rangle = \delta_{i_1,i_2}, \ \forall i_i, i_2. \end{aligned} \tag{32}$$

Then, using the commutativity of convolution: $\mathcal{E}_i^{(t+1)} * \mathcal{F}_j = \mathcal{F}_j * \mathcal{E}_i^{(t+1)}$, the update of $\mathcal{F}_j$ is done by solving

$$\begin{aligned} \min_{\{\mathcal{F}_j\}_j} \quad & \sum_{j=1}^{J} \|\mathcal{F}_j * \mathcal{E}_i^{(t+1)} * \mathcal{X}^{(t+1)} - \mathcal{C}_{i,j}^{(t+1)}\|_2^2 \\ \text{s.t.} \quad & \langle \mathcal{F}_{j_1}, \mathcal{F}_{j_2} \rangle = \delta_{j_1,j_2}, \ \forall j_i, j_2. \end{aligned} \tag{33}$$

Note that the problems of (32) and (33) share the same form:

$$\min_{\{\mathcal{A}_k\}_k} \sum_{k=1}^{K} \|\mathcal{A}_k * \mathcal{Y} - \mathcal{B}_k\|_2^2, \ \text{ s.t. } \langle \mathcal{A}_{k_1}, \mathcal{A}_{k_2} \rangle = \delta_{k_1,k_2}, \forall k_1, k_2, \tag{34}$$

which has the unique solution given by Proposition 3.3.

Proposition 3.3: Let $\boldsymbol{A} = [\mathcal{V}(\bar{\mathcal{A}}_1), \mathcal{V}(\bar{\mathcal{A}}_2), \cdots, \mathcal{V}(\bar{\mathcal{A}}_K)]^\top$ where $\bar{\mathcal{A}}_k \in \mathbb{R}^{S \times \cdots \times S}$ denotes the flipping of $\mathcal{A}_k \in \mathbb{R}^{S \times \cdots \times S}$, $\boldsymbol{Y} = \mathcal{P}_s(\mathcal{Y})$ and $\boldsymbol{B} = [\mathcal{V}(\mathcal{B}_1), \mathcal{V}(\mathcal{B}_2), \cdots, \mathcal{V}(\mathcal{B}_K)]^\top$. The solution of (34) is given by $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{V}^\top$ where $\boldsymbol{U}, \boldsymbol{V}$ denote the orthogonal matrices defined by the following SVD: $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top = \boldsymbol{B}\boldsymbol{Y}^\top$.

Proof Based on the definition as well as calculation process of convolution, the problem of (34) can be rewritten into

$$\min_{\boldsymbol{A}} \|\boldsymbol{A}\boldsymbol{Y} - \boldsymbol{B}\|_\text{F}^2, \quad \text{s.t. } \boldsymbol{A}^\top\boldsymbol{A} = I. \tag{35}$$

The problem of (35) is about finding an optimal orthogonal transform in least squares approximation, which has the unique solution given by [16]:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{V}^{\top}. \tag{36}$$

The proof is completed.

C. Complexity Analysis

Let $S = \prod_{p=1}^{P} S_p$ denote the number of elements in the processed tensor $\mathcal{Y}$ and $Z = (M+N-1)^P$ denote the number of elements in each dictionary atom $\mathcal{D}_{i,j}$. Suppose the number of nonzero entries for each position, *i.e.* nonzero entries in each row of $\boldsymbol{C}$, is a constant μ. Also recall that the number of atoms in the dictionary is $N = IJ$, the computational complexity of the proposed tensor completion method is analyzed as follows. The update of $\mathcal{X}$ mainly involves the convolution operations, which need μZS dominant operations. The update of $\{\mathcal{C}_{i,j}\}_{i,j}$ involves the convolution and element-wise hard thresholding operations, where the total number of dominant operations is $IJZS + IJS$. The update of factor dictionary $\{\mathcal{E}_i\}_i$ involves convolution, matrix product and SVD, requiring $J^2S + \mu IS + I^3$ dominant operations. Similarly, the total number of dominant operations in calculating $\{\mathcal{F}_j\}_j$ is $I^2S + \mu JS + J^3$. Assuming that $\mu \ll IJ \ll S$, the proposed method has an iteration complexity of $\mathcal{O}(IJZS)$. For comparison, we consider the complexity of applying K-SVD (accelerated version) [62] and L0DL [14] for tensor completion, where the dictionary learning and tensor recovery steps are alternated in the outer loop. For good results, several inner iterations of dictionary update are needed in one outer iteration for these two methods. Suppose the number of such inner iterations is k_o. The complexity of K-SVD and L0DL in tensor completion is $\mathcal{O}(k_oIJZS+k_o\mu^2IJS)$ and $\mathcal{O}(k_oIJZS)$ respectively.

IV. Experiments

A. Protocols and Implementation Details

The proposed method is evaluated using four types of data, including color images, videos, multi-spectral images, and magnetic resonance imaging (MRI) data. The peak signal-to-noise ratio (PSNR) is used for measuring the quality of the recovered tensor. Given the recovered tensor $\mathcal{Y}$ and its ground truth $\mathcal{X}$, the PSNR is computed by

$$\text{PSNR}(\mathcal{Y}, \mathcal{X}) = 10 * \log_{10}(NV_{max}^2)/\|\mathcal{Y} - \mathcal{X}\|_F^2,$$

where V_{max} is the maximum possible value in the ground-truth tensor $\mathcal{X}$, and N is to the number of elements in $\mathcal{X}$.

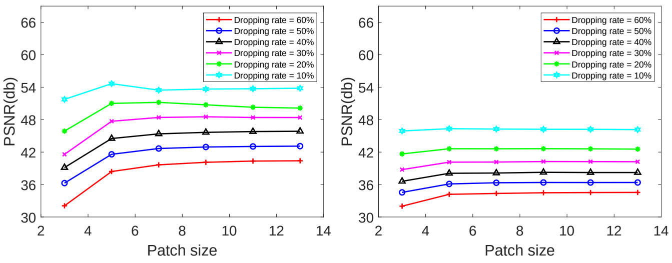
The implementation details of our method are as follows. Each factor dictionary is initialized by the multi-dimensional DCT. The algorithm stops when the number of iterations reaches 500, or $\|\mathcal{X}^{(t+1)} - \mathcal{X}^{(t)}\|_F/\|\mathcal{Y}_\Omega\|_F \le 10^{-4}$. The parameters of our method are set as follows. The parameters of our model only include the atom sizes of $\mathcal{E}_i$ and $\mathcal{F}_j$,[1] which are set according to the size of input data. In particular, we set each atom to have equal spatial dimensions and allow the other dimensions to be different from the spatial dimensions. Our numerical algorithm involves two parameters: γ and initial ρ, which are set to 1.05 and 0.01 respectively.

For comparison, we select seven approaches which have the published results or available codes, including

- Three representative low-rank approaches: HaLRTC [48] (multi-rank), SPCTC [63] (CPD-rank), and t-SVD [51] (tubal rank).
- Two conventional vectorized dictionary learning methods: K-SVD [18] and L0DL [14]. The patch sizes are set the same as the proposed method for fair comparison.
- Two very recent hybrid approaches (low-rank approximation + sparse coding): KBRTC [57] (smoothed tensor nuclear norm + core tensor sparsity), and WTNNDL [12] (weighted tensor nuclear norm + orthogonal dictionary learning).

To better demonstrate the effectiveness of the proposed model, we constructed two baseline methods for comparison:

- Orthogonal dictionary learning. To show the performance gain from the double orthogonal dictionaries over a single one, our model is modified so that it only learns an orthogonal dictionary, and the subproblem of dictionary learning is solved by the orthogonal dictionary learning algorithm of [16]. The patch size is set the same as ours. Such a baseline is denoted by ODLTC.
- Separable orthogonal tensor dictionary learning. To show the benefits from our dictionary factorization over the existing factorization schemes, the dictionary in our model is replaced with the separable orthogonal dictionary [2]. The dictionary learning subproblem is then solved by the algorithm of [2]. The patch size is set the same as ours. The resulting baseline method is denoted by SODLTC.



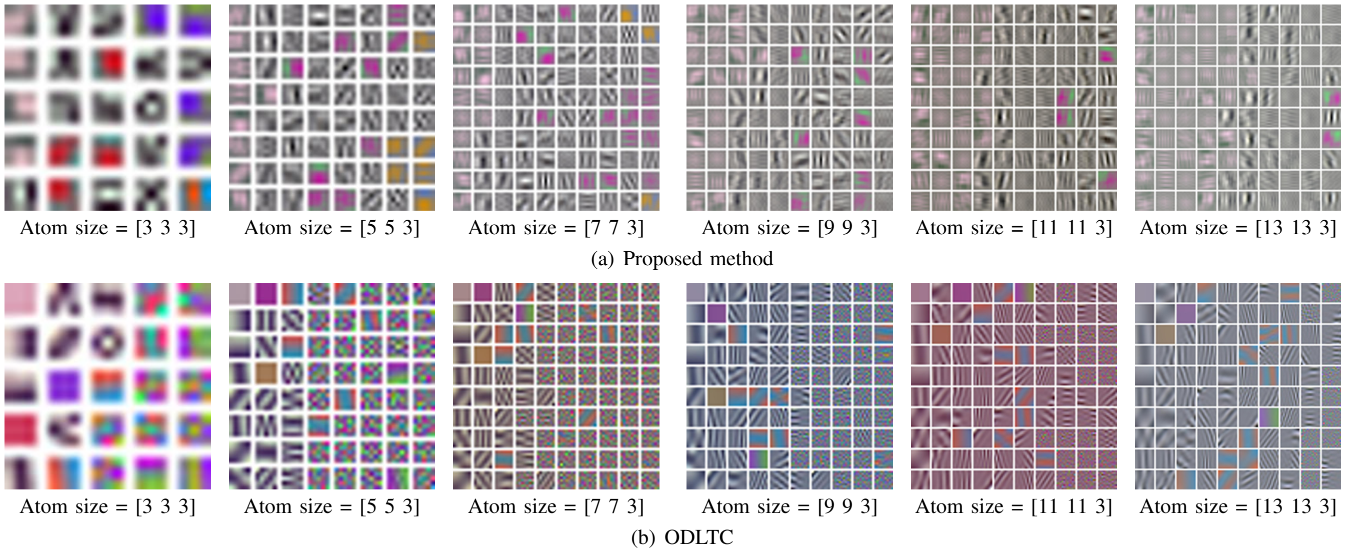
(a) Proposed method (b) ODLTC

Fig. 2. PSNR values (dB) of the proposed method and ODLTC using different atom sizes on image 'Lena' with different dropping rates.

B. Influence of Atom Sizes of Factor Dictionaries

The sizes of the factor dictionaries $\{\mathcal{E}_i\}_i, \{\mathcal{F}_j\}_j$ are the only model parameters to be set. We investigate the influence of such sizes on the completion of color images. Firstly, the sizes of atoms $\mathcal{E}_i, \mathcal{F}_j$ are set to $L \times L \times 1$ and $L \times L \times 3$ respectively for all i, j, with L varying from 2 to 7. Accordingly, the size of the composite dictionary atom $\mathcal{D}_{i,j}$ is $3, 5, 7, 9, 11, 13$ respectively. In Fig. 2 we plot the PSNR values on the completion of image 'Lena' with different pixel missing (dropping) rates, by

[1] Owing to the orthogonality constraints, once the atom sizes are set, the size of each factor dictionary as well as the composite dictionary is fixed.



(a) Proposed method

(b) ODLTC

Fig. 3. Dictionaries learned by the proposed method and ODLTC with different atom sizes on image 'Lena' with pixel dropping rate of 70%. Due to space limitation, we only show top-K atoms that have highest responses on the image.

applying our method using different atom sizes. Overall, the performance of our method increases when the size of atoms grows up, and the performance saturates when the atom size is sufficiently large.

For comparison, we also show the results of ODLTC in Fig. 2 under the same setting. The overall performance of ODLTC also increases with the increase of atom size. However, the improvement is not as big as ours. One reason is probably that as the number of atoms increases, ODLTC may learn some undesired patterns due to the orthogonality constraint on the dictionary. To demonstrate this, in Fig. 3 we show the dictionaries learned by our method and ODLTC, with different dictionary sizes. It can been seen that as the dictionary (atom) size increases, ODLTC may learn some noisy patterns. In comparison, the atoms learned by our method become smoother and contain larger-scale structures when the atom size increases, without noisy patterns generated.

C. Evaluation on Color Images

For color image completion, six classic images shown in Fig. 4 are used for the evaluation. The size of each image is 255 × 255 × 3. The missing elements are generated by randomly dropping the image pixels with the ratio of $10\%, 20\%, 30\%, 40\%, 50\%, 60\%$ respectively. The atom sizes of the two factor dictionaries are set to $5\times5\times1$ and $3\times3\times3$, respectively. The PSNR values of the recovered results are listed in Table I. It can be seen that the proposed method performs the best for all dropping rates. Particularly, compared with the baseline methods including ODLTC and SODLTC, as well as the dictionary-learning-based methods including K-SVD and L0DL, the proposed method shows noticeable improvement. Such improvement has demonstrated the dictionary structure we use is effective for tensor completion. It is also worth mentioning that compared with the hybrid methods including KBRTC and WTNNDL, ours still shows improvements even without the low rank prior.

Fig. 4. Color images used for evaluation.

TABLE I
AVERAGE PSNR VALUES (DB) OF RECOVERED RESULTS ON COLOR IMAGES WITH DIFFERENT DROOPING RATES.

Dropping Rate	10%	20%	30%	40%	50%	60%
HaLRTC	44.48	41.35	36.61	33.87	31.33	28.90
SPCTC	38.81	35.60	33.73	32.26	31.07	29.88
t-SVD	45.34	42.20	37.10	34.25	31.66	29.18
K-SVD	44.37	40.87	38.68	36.89	35.33	33.76
L0DL	44.32	40.89	38.67	36.90	35.32	33.76
KBRTC	48.58	45.30	39.65	36.68	33.54	30.83
WTNNDL	48.83	46.43	41.70	39.06	36.60	34.16
ODLTC	44.51	40.62	38.56	36.69	34.72	33.13
SODLTC	42.82	38.03	36.87	35.09	33.48	31.75
Ours	**50.67**	**48.48**	**44.37**	**41.85**	**39.42**	**36.72**

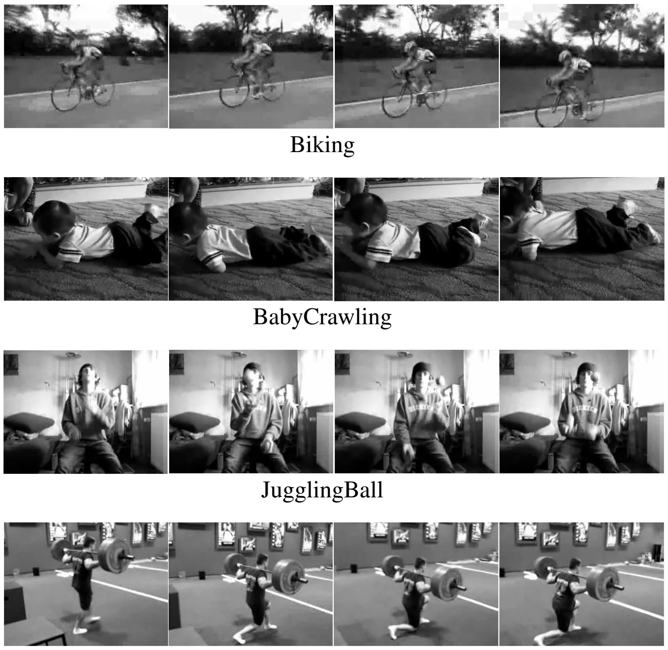
D. Evaluation on Videos

Following [12], we use four videos for the test on video completion, including Biking, BabyCrawling, JugglingBall and Lunges, which are from the UCF-101 action dataset[2]. The size of each video is 240 × 320 × 80. See Fig. 5 for some sampled frames of the videos. The incomplete test data are produced by randomly dropping the pixels in each video with ratios $20\%, 30\%, \cdots, 90\%$ respectively. The atom sizes of the two factor dictionaries are set to $10 \times 10 \times 1$ and $7 \times 7 \times 4$ respectively. We use all the previous methods (including existing approaches and our constructed baselines) mentioned in Section IV-A for comparison. The results in terms of average PSNR are summarized and compared in

[2] http://crcv.ucf.edu/data/UCF101.php

Table II. It can be seen that the proposed method outperforms others under the dropping ratios from 20% to 90%.

In Fig. 6, we show the recovered results of the 54-th frame in 'BabyCrawling' for visual comparison. It can be seen that the low-rank methods including HaLRT, SPCTC and t-SVD produced blurred results with obvious artifacts. The dictionary-learning-based methods including K-SVD and L0DL fail to recover the region of the clothes which has complex patterns, and they also blur the ear of the baby in the close-up. The results of KBRTC and WTNNDL are more acceptable, but with a spot of artifacts. In comparison, our recovery result is visually better.



Biking

BabyCrawling

JugglingBall

Lunges

Fig. 5. Sample key frames of four test videos.

TABLE II
AVERAGE PSNR VALUES (DB) OF RECOVERED RESULTS IN VIDEO COMPLETION WITH DIFFERENT DROPPING RATES.

Method	20%	30%	40%	50%	60%	70%	80%	90%
HaLRTC	42.20	38.46	35.14	32.04	28.97	25.90	22.75	19.08
SPCTC	34.71	32.83	31.43	30.30	29.26	28.25	27.08	25.21
t-SVD	42.38	38.58	35.37	32.51	29.87	27.32	24.77	21.94
KSVD	31.18	28.70	26.88	25.40	24.11	22.94	21.66	20.14
L0DL	31.59	29.00	27.11	25.62	24.24	23.01	21.67	20.13
KBRTC	50.36	47.34	44.66	42.15	39.54	36.71	33.38	27.69
WTNNDL	49.17	46.25	43.65	41.26	38.78	36.14	33.06	29.07
ODLTC	31.20	28.98	27.22	25.70	24.23	22.73	21.03	17.66
SODLTC	30.67	28.46	26.78	25.34	23.94	22.57	21.17	19.17
Ours	**50.62**	**47.75**	**45.17**	**42.68**	**40.14**	**37.32**	**34.02**	**29.41**

E. Evaluation on Multi-Spectral Images

The evaluation on multi-spectral images is done on The Columbia MSI dataset[3]. The dataset contains 32 real-world scenes of a variety of real-world materials and objects, each with the spatial resolution of 512×512 and the spectral resolution of 31. Following [57], each image was resized to 256×256 for all spectral bands in our experiments. We set the atom sizes of the two factor dictionaries to $8 \times 8 \times 6$ and $4 \times 4 \times 5$ respectively. The recovery results are compared to those of the compared methods used in Section IV-D. The dropping rate is varied from $70\%, 75\%, 80\%, 85\%, 90\%$.

The recovery results of different methods in terms of average PSNR are listed in Table III. It can be seen that the proposed method achieved the highest PSNR values for the dropping rates of $70\%, 75\%, 80\%, 85\%$ and perform the second best at the dropping rate of 90%. One reason that our method performed worse than KBRTC at the dropping rate of 90% is probably that there are too many elements missing under such a high dropping rate, making the dictionary learning less effective. In comparison, KBRTC is a hybrid approach which not only uses sparsity prior but also resorts to low rank prior, leading to better result in this setting. For visual comparison, we show results of the band centered at 630nm in Jelly-Beans with the dropping rate of 90%. It can be observed that the proposed method is superior in the recovery of both the fine-grained textures and coarse-grained structures.

TABLE III
AVERAGED PSNR VALUES (DB) OF RECOVERED RESULTS ON MULTI-SPECTRAL IMAGES WITH DIFFERENT DROPPING RATES.

Method	70%	75%	80%	85%	90%
HaLRTC	38.90	37.21	35.30	33.06	30.17
SPCTC	40.67	40.05	39.35	38.42	37.03
t-SVD	43.19	41.57	39.71	37.46	34.58
KSVD	23.02	22.24	21.53	20.92	20.36
L0DL	23.37	22.52	21.68	20.99	20.39
KBRTC	51.24	49.81	48.07	**45.85**	**42.72**
WTNNDL	45.38	46.03	42.39	42.58	38.34
ODLTC	26.18	26.05	24.29	23.34	22.04
SODLTC	26.47	25.69	24.71	23.33	21.86
Ours	**51.60**	**50.08**	**48.30**	**45.85**	42.32

F. Evaluation on MRI Volume Data

Following [12], we use the CThead dataset for the evaluation on MRI data recovery. The CThead dataset is a subset of the dataset of the University of North Carolina Volume Rendering Test DataSet[4]. The size of each volume in the dataset is $252 \times 252 \times 99$. Some samples of this dataset are shown in Fig. 8. We set the atom sizes of the two factor dictionaries as $8 \times 8 \times 3$ and $5 \times 5 \times 5$ respectively. The recovery results are compared with all the methods mentioned in Section IV-D, and the average PSNR results w.r.t. the dropping ratio from 20% to 90% are listed in Table IV. It can be seen that the proposed method has noticeable PSNR gain (around 1dB) over the second best method.

Some visual inspections are done in Fig. 9. The baselines ODLTC and SODLTC failed to represent the data well when the dropping rate is high, while our method still produces good results. Compared with the other state-of-the-art methods, ours produced clearer results with less artifacts.

[3] http://www1.cs.columbia.edu/CAVE/databases/multispectral/

[4] http://graphics.stanford.edu/data/voldata

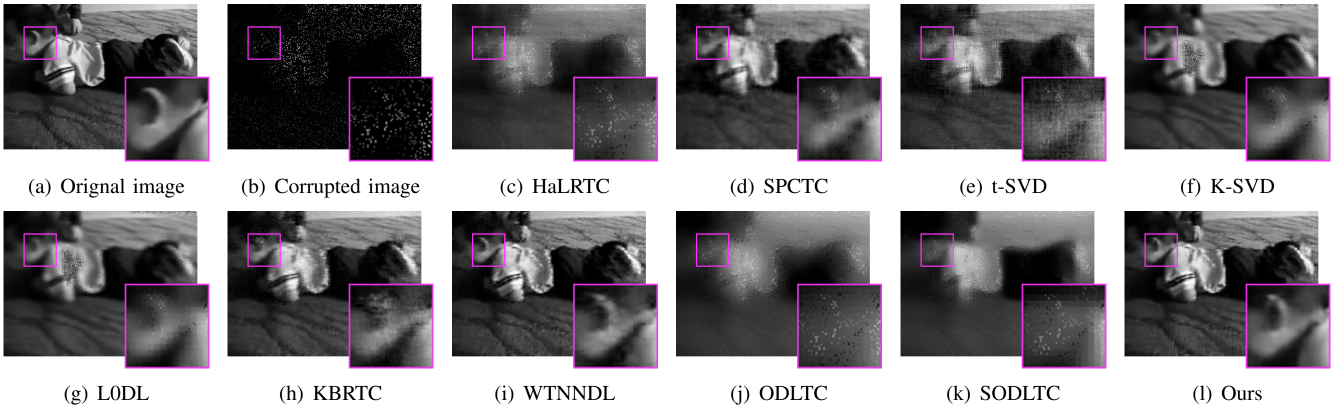


Fig. 6. Visual results in video completion. (a) Original frame in BabyCrawling. (b) Corrupted frame. (c)-(j) The recovered results by different methods.

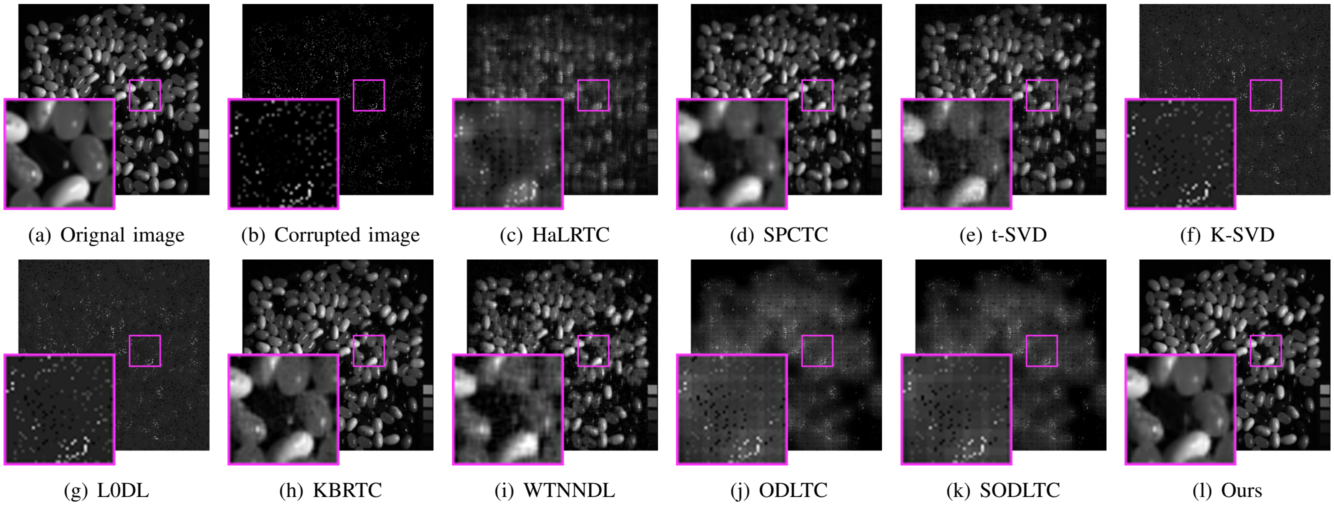


Fig. 7. Visual results in multi-spectral image completion. (a) Original band in Jelly-Beans. (b) Corrupted image. (c)-(j) The recovered results by different methods.

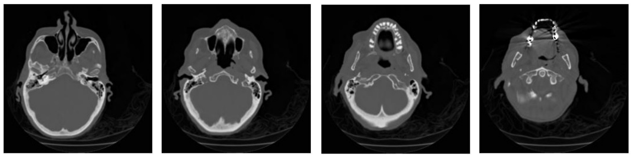


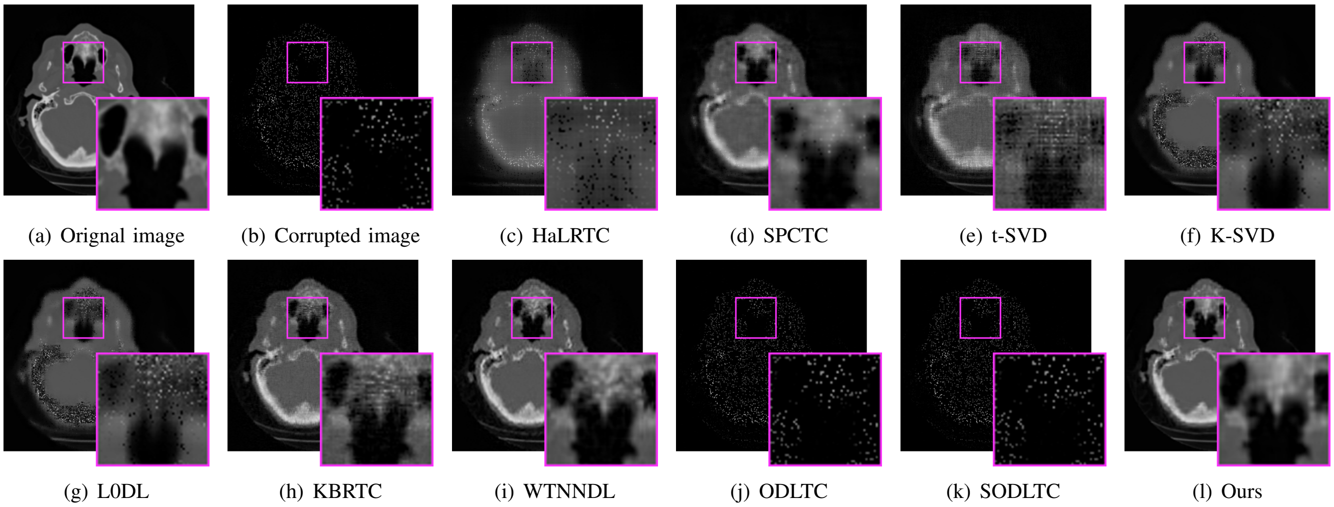
Fig. 8. Sample slices from MRI data.

TABLE IV
PSNR VALUES (DB) OF RECOVERED RESULTS ON MRI VOLUME DATA WITH DIFFERENT DROPPING RATES.

Dropping Rate	20%	30%	40%	50%	60%	70%	80%	90%
HaLRTC	42.92	39.7	36.76	34.19	31.67	29.14	26.39	22.89
SPCTC	39.29	37.36	35.9	34.66	33.51	32.33	31.03	29.04
t-SVD	43.8	40.88	38.24	35.93	33.75	31.54	29.22	26.54
KSVD	35.49	35.05	32.34	28.26	25.65	23.28	21.64	20.46
L0DL	34.93	34.73	33.02	29.19	26.04	23.49	21.66	20.48
KBRTC	48.39	45.72	43.46	41.61	39.67	37.59	35.02	30.40
WTNNDL	47.99	45.43	43.26	41.46	39.61	37.45	35.04	31.78
ODLTC	30.91	28.82	27.08	25.87	24.47	23.28	21.55	13.77
SODLTC	30.81	28.62	27.34	26.12	24.63	23.26	21.52	13.77
Ours	**49.02**	**46.79**	**44.49**	**42.82**	**41.02**	**38.91**	**36.36**	**32.83**

G. Time Cost

To evaluate the computational efficiency of our method, we compare the average running time of ours and other methods on the four datasets. The results are reported in Table V. It can be seen that our approach runs much faster than the conventional dictionary-learning-based methods K-SVD and L0DL. Such an advantage comes from the proposed dictionary learning model. Compared with SODLTC which is the baseline method using separable dictionary learning, the efficiency of the proposed method is comparable. In comparison to SODLTC, our approach can learn an overcomplete instead of orthogonal dictionary. Compared to the low-rank methods, ours is slower. However, the low-rank approaches cannot exploit the local characteristics of visual tensor data. While the hybrid approach, KBRTC, is faster than ours, it uses a fixed dictionary which is not adaptive to data. Another hybrid approach, WTNNDL, is also faster than ours. But note that the dictionary of WTNNDL is smaller than ours. In addition, WTNNDL uses orthogonal dictionary learning which disables the completeness of dictionary. Thus, the performance of WTNNDL is worse than ours, as shown in previous experimental results.



(a) Orignal image (b) Corrupted image (c) HaLRTC (d) SPCTC (e) t-SVD (f) K-SVD

(g) L0DL (h) KBRTC (i) WTNNDL (j) ODLTC (k) SODLTC (l) Ours

Fig. 9. Visual results in MRI volume data completion. (a) The orignal image of the 68-th slice in CTHead. (b) The corrupted image. (c)-(j) The recovered image obtained by the competing methods and the proposed method.

TABLE V
AVERAGE RUNNING TIME (SECONDS) OF COMPARED METHODS ON DIFFERENT DATASETS.

Type	Color Image	Video	MSI	MRI
HaLRTC	0.64	0.88	0.19	5.26
SPCTC	18.01	2144.92	356.19	2530.26
t-SVD	19.46	56.88	11.04	346.34
KSVD	256.83	29293.45	3890.01	24596.90
L0DL	260.97	32525.13	4796.02	26922.00
KBRTC	58.28	1264.51	505.09	1366.49
WTNNDL	24.49	275.86	238.79	339.77
ODLTC	31.66	243.32	33.28	212.48
SODLTC	913.04	11130.92	297.75	2945.71
Ours	25.96	8727.84	1497.16	5660.66

H. Convergence Behavior Analysis

It is difficult to analyze the theoretical convergence of the proposed algorithm. Instead, we study the convergence behavior of the proposed algorithm in color image completion, in terms of

- normalized decrement $\|\mathcal{X}^{(t+1)} - \mathcal{X}^{(t)}\|_F / \|\mathcal{Y}_\Omega\|_F$ (used in the stopping criterion) over iterations;
- decay of objective function value $\sum_{i,j=1}^{I,J} \|\mathcal{D}_{i,j} * \mathcal{X}\|_0$ over iterations.

The results are shown in Fig. 10. It can be seen that, for all test color images, both the normalized decrement between two successive results and the objective function value decay very fast over iterations, without oscillations observed. The objective function value converges to some relatively-small values for different images, as different images have different sparsity degrees under a learned dictionary. The normalized decrement between two successive results converges be zero eventually after sufficient iterations, for all test images. Such a feature is attractive for practical use and provides a good stopping criterion.

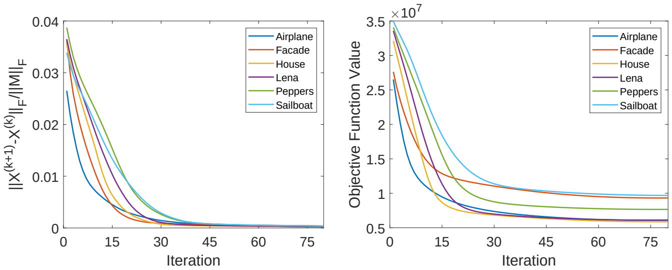


Fig. 10. Convergence behavior analysis of the proposed algorithm.

V. Conclusion

There is an increasing amount of visual data emerging in the form of tensor. Learning dictionaries from such visual tensor data has become a critical module in many tasks. Conventional dictionary learning approaches have very high computational cost when run on a high-order/dimensional tensor. Imposing the orthogonal constraint on the dictionary may accelerate the computation; however, the learned dictionary cannot be overcomplete. Many existing tensor dictionary learning approaches reduce the computational cost by using CPD or TD on the dictionary. Nevertheless, they cannot generate atoms with rich orientations, as the decompositions they use impose the rank-1 constraint on each atom. In short, existing dictionary learning approaches have the efficiency or effectiveness issues when dealing with tensor data, and thus they are not the good choices for the task of visual tensor completion.

In this paper, we proposed a novel tensor dictionary learning scheme for visual tensor data. It employs a convolutional analysis model for dictionary learning, with an orthogonality-constrained convolutional factorization on the dictionary. Compared with conventional dictionary learning approaches, the proposed one has lower computational cost. Compared to the existing tensor dictionary learning approaches, the proposed one has the advantages of learning an overcomplete dictionary with atoms of different orientations.

Built upon the proposed tensor dictionary learning scheme, we proposed a visual tensor completion approach which enjoys both effectiveness and efficiency. The experiments on four types of visual data have demonstrated the advantages of the proposed approach. It not only outperformed existing sparse approaches and low-rank approaches, but also performed better than the ones that utilize both sparsity prior and low rank prior.

References

[1] P. Yi, D. Meng, Z. Xu, C. Gao, Y. Yi, and B. Zhang, "Decomposable nonlocal tensor dictionary learning for multispectral image denoising," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2014.

[2] Y. Quan, H. Yan, and J. Hui, "Dynamic texture recognition via orthogonal tensor dictionary learning," in *Proc. IEEE Int. Conf. Comput. Vision*, 2016.

[3] 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, 2016.

[4] X. Zhou, L. Chen, and X. Zhou, "Structure tensor series-based large scale near-duplicate video retrieval," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1220–1233, 2012.

[5] 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, 2015.

[6] W. Dong, G. Li, G. Shi, L. Xin, and M. Yi, "Low-rank tensor approximation with laplacian scale mixture modeling for multiframe image denoising," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015.

[7] X. Qi, Z. Qian, D. Meng, Z. Xu, S. Gu, W. Zuo, and Z. Lei, "Multispectral images denoising by intrinsic tensor sparsity regularization," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2016.

[8] B. Xiong, Q. Liu, J. Xiong, S. Li, S. Wang, and D. Liang, "Field-of-experts filters guided tensor completion," *IEEE Trans. Multimedia*, vol. 20, no. 9, pp. 2316–2329, 2018.

[9] B. Du, M. Zhang, L. Zhang, R. Hu, and D. Tao, "Pltd: Patch-based low-rank tensor decomposition for hyperspectral images," *IEEE Trans. Multimedia*, vol. 19, no. 1, pp. 67–79, 2016.

[10] X. Tan, F. Wu, X. Li, S. Tang, W. Lu, and Y. Zhuang, "Structured visual feature learning for classification via supervised probabilistic tensor factorization," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 660–673, 2015.

[11] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov, "Tensorizing neural networks," in *Adv. Neural Inf. Process. Syst.*, 2015, pp. 442–450.

[12] Y. Du, G. Han, Y. Quan, Z. Yu, H. Wong, C. L. P. Chen, and J. Zhang, "Exploiting global low-rank structure and local sparsity nature for tensor completion," *IEEE Trans.Syst. Man Cybern.*, pp. 1–13, 2018.

[13] Y. Liu, Z. Long, and C. Zhu, "Image completion using low tensor tree rank and total variation minimization," *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 338–350, 2018.

[14] 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, 2016.

[15] M. Aharon, M. Elad, and A. M. Bruckstein, "K -svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.

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

[17] V. Papyan, Y. Romano, M. Elad, and J. Sulam, "Convolutional dictionary learning via local processing," *Proc. IEEE Int. Conf. Comput. Vision*, pp. 5306–5314, 2017.

[18] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, 2007.

[19] Y. Quan, H. Teng, T. Liu, and Y. Huang, "Weakly-supervised sparse coding with geometric prior for interactive texture segmentation," *IEEE Signal Processing Letters*, vol. 27, pp. 116–120, 2020.

[20] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *Proc. IEEE Int. Conf. Acoustics*, 1999.

[21] F. Heide, W. Heidrich, and G. Wetzstein, "Fast and flexible convolutional sparse coding," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2015.

[22] F. Huang and A. Anandkumar, "Convolutional dictionary learning through tensor factorization," *Computer Science*, pp. 1–30, 2015.

[23] J. Sulam, V. Papyan, Y. Romano, and M. Elad, "Multilayer convolutional sparse modeling: Pursuit and dictionary learning," *IEEE Trans. Signal Process.*, vol. 66, no. 15, pp. 4090–4104, 2018.

[24] A. Bibi and B. Ghanem, "High order tensor formulation for convolutional sparse coding," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017.

[25] Y. Quan, Y. Xu, Y. Sun, and Y. Huang, "Supervised dictionary learning with multiple classifier integration," *Pattern Recognition*, vol. 55, pp. 247–260, 2016.

[26] R. Rubinstein, T. Peleg, and M. Elad, "Analysis k-svd: A dictionary-learning algorithm for the analysis sparse model," *IEEE Trans. Signal Process.*, vol. 61, no. 3, pp. 661–677, 2012.

[27] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Analysis operator learning for overcomplete cosparse representations," in *Proc. Eur. Signal Process. Conf.* IEEE, 2011, pp. 1470–1474.

[28] S. Hawe, M. Kleinsteuber, and K. Diepold, "Analysis operator learning and its application to image reconstruction," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2138–2150, 2013.

[29] T. Hazan, S. Polak, and A. Shashua, "Sparse image coding using a 3d non-negative tensor factorization," in *IEEE Int. Conf. Comput. Vision*, 2005.

[30] G. Duan, H. Wang, Z. Liu, J. Deng, and Y. W. Chen, "K-cpd: Learning of overcomplete dictionaries for tensor sparse coding," in *Proc. IEEE Int. Conf. Pattern Recognition*, 2013.

[31] A. Sironi, B. Tekin, R. Rigamonti, V. Lepetit, and P. Fua, "Learning separable filters," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2013.

[32] C. F. Dantas, J. E. Cohen, and R. Gribonval, "Learning fast dictionaries for sparse representations using low-rank tensor decompositions," in *Int. Conf. Latent Variable Anal. Signal Separation.* Springer, 2018, pp. 456–466.

[33] X. Zhang, X. Yuan, and L. Carin, "Nonlocal low-rank tensor factor analysis for image restoration," *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pp. 8232–8241, 2018.

[34] A. Stevens, Y. Pu, Y. Sun, G. Spell, and L. Carin, "Tensor-dictionary learning with deep kruskal-factor analysis," *Int. Conf. on Artif. Intell. and Statist.*, pp. 121–129, 2017.

[35] S. Zubair and W. Wang, "Tensor dictionary learning with sparse tucker decomposition," in *Int. Conf. Digit. Signal Process.*, 2013.

[36] S. Hawe, M. Seibert, and M. Kleinsteuber, "Separable dictionary learning," *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pp. 438–445, 2013.

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

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

[39] F. Ju, Y. Sun, J. Gao, Y. Hu, and B. Yin, "Nonparametric tensor

dictionary learning with beta process priors," *Neurocomputing*, vol. 218, pp. 120–130, 2016.

[40] S. Zubair and W. Wang, "Signal classification based on block-sparse tensor representation," in *Int. Conf. Digit. Signal Process.*, 2014.

[41] Y. Fu, J. Gao, Y. Sun, and H. Xia, "Joint multiple dictionary learning for tensor sparse coding," in *Int. Joint Conf. Neural Netw.*, 2014.

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

[43] R. Sivalingam, D. Boley, V. Morellas, and N. Papanikolopoulos, "Tensor dictionary learning for positive definite matrices," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4592–4601, 2015.

[44] 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 Pattern Recognition*, 2016.

[45] 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.

[46] 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, 2015.

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

[48] 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, 2013.

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

[50] R. Hao and Z. Su, "A patch-based low-rank tensor approximation model for multiframe image denoising," *J. Comput. Appl. Math.*, vol. 329, pp. 125–133, 2018.

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

[52] 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," *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pp. 5249–5257, 2016.

[53] J. Ren, X. Li, and J. Haupt, "Robust pca via tensor outlier pursuit," in *Proc. Conf. Signals Syst. Comput.*, 2016.

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

[55] Z. Han, C. Leung, L. Huang, and H. C. So, "Sparse and truncated nuclear norm based tensor completion," *Neural Process. Lett.*, vol. 45, no. 3, pp. 729–743, 2017.

[56] J. Yang, Y. Zhu, K. Li, J. Yang, and C. Hou, "Tensor completion from structurally-missing entries by low-tt-rankness and fiber-wise sparsity," *IEEE J. Selected Topics Signal Process.*, vol. PP, pp. 1–1, 10 2018.

[57] Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1888–1902, 2018.

[58] C. Zheng, T.-J. Cham, and J. Cai, "Pluralistic image completion," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2019, pp. 1438–1447.

[59] W.-H. Xu, X.-L. Zhao, T.-X. Jiang, Y. Wang, and M. Ng, "Deep plug-and-play prior for low-rank tensor completion," *arXiv preprint arXiv:1905.04449*, 2019.

[60] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Deep video inpainting," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 2019, pp. 5792–5801.

[61] J. F. Cai, H. Ji, Z. Shen, and G. B. Ye, "Data-driven tight frame construction and image denoising," *Appl. Comput. Harmonic Anal.*, vol. 37, no. 1, pp. 89–105, 2014.

[62] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit," Computer Science Department, Technion, Tech. Rep., 2008.

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

Ruotao Xu received his B.Eng. degree in computer science from South China University of Technology, Guangzhou, China, in 2015, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. His research interests include image processing, sparse coding, and deep learning.

Yong Xu received his B.Sc., M.Sc. and Ph.D. degrees in mathematics from Nanjing University, Nanjing, China, in 1993, 1996, and 1999, respectively. He was a Post-Doctoral Research Fellow in computer science with South China University of Technology, Guangzhou, China, from 1999 to 2001, where he is currently a professor with the School of Computer Science and Engineering. His current research interests include computer vision and pattern recognition.

Yuhui Quan received his doctoral degree in computer science from South China University of Technology, Guangzhou, China, in 2013. He was a Post-Doctoral Research Fellow in mathematics with the National University of Singapore, Singapore, from 2013 to 2016. He is currently an Associate Professor with the School of Computer Science and Engineering, South China University of Technology. His research interests include computer vision, image processing, deep learning, and sparse representation.