TenSR: Multi-Dimensional Tensor Sparse Representation

Na Qi[1], Yunhui Shi[1], Xiaoyan Sun[2], Baocai Yin[13]
[1]Beijing Key Laboratory of Multimedia and Intelligent Software Technology,
College of Metropolitan Transportation, Beijing University of Technology
q1987n@emails.bjut.edu.cn, syhzm@bjut.edu.cn

[2]Microsoft Research
xysun@microsoft.com

[3]Faculty of Electronic Information and Electrical Engineering,
Dalian University of Technology
ybc@bjut.edu.cn

Abstract

The conventional sparse model relies on data representation in the form of vectors. It represents the vector-valued or vectorized one dimensional (1D) version of an signal as a highly sparse linear combination of basis atoms from a large dictionary. The 1D modeling, though simple, ignores the inherent structure and breaks the local correlation inside multidimensional (MD) signals. It also dramatically increases the demand of memory as well as computational resources especially when dealing with high dimensional signals. In this paper, we propose a new sparse model ***TenSR*** *based on tensor for MD data representation along with the corresponding MD sparse coding and MD dictionary learning algorithms. The proposed TenSR model is able to well approximate the structure in each mode inherent in MD signals with a series of adaptive separable structure dictionaries via dictionary learning. The proposed MD sparse coding algorithm by proximal method further reduces the computational cost significantly. Experimental results with real world MD signals, i.e. 3D Multi-spectral images, show the proposed TenSR greatly reduces both the computational and memory costs with competitive performance in comparison with the state-of-the-art sparse representation methods. We believe our proposed TenSR model is a promising way to empower the sparse representation especially for large scale high order signals.*

1. Introduction

In the past decade, sparse representation has been widely used in a variety of tasks in computer vision such as image denoising [10, 7, 22, 8], image super-resolution [37, 33, 36], face recognition [34, 39], and pattern recognition [16, 13]. Generally speaking, a classic sparse model represents a vector-valued signal by a linear combination of certain atoms of an overcomplete dictionary. Higher-order signals (*e.g.* images and videos) need to be dealt with primarily by vectorizing them and applying any of the available vector techniques [29]. Researches on the conventional one dimensional (1D) sparse representation include the proposal of 1D sparse model [4, 9], sparse coding [23, 32, 3], and dictionary learning algorithms [1, 18]. Though simple, the 1D sparse model suffers from high memory as well as high computational costs especially when handling high dimensional data since the vectorized data will be quite long and must be measured using very large sampling matrices.

Recent research has demonstrated the advantages of maintaining the higher-order data in their original form [31, 26, 14, 40, 24, 29, 27, 6]. For image data, the two dimensional (2D) sparse model is proposed to make use of the intrinsic 2D structure and local correlations within images and has been applied to image denoising [26] and super-resolution [25]. The 2D dictionary learning problem is solved by the two-phase block-coordinate-relaxation approach. Given the 2D dictionaries, the 1D sparse coding algorithms are extended to solve the 2D sparse coding problem [12, 11] or converted to 1D problem and solved via the kronecker product [26]. By learning 2D dictionaries for images, the 2D sparse model helps to greatly reduce the time complexity and memory cost for image processing [26, 14, 25]. On the other hand, the 2D sparse model is difficult to be extended for multidimensional (MD) sparse modeling due to the use of 1D sparse coding method.

Tensors are also introduced in the sparse representation of vectors to approximate the structure in each mode of MD signals. Due to the equivalence of the constrained Tucker model and the Kronecker representation of a tensor, the tensor is assumed to be represented by separable given dictionaries, known as Kronecker dictionaries, with a sparsity constraint, such as multi-way sparsity and block sparsity [5]. The corresponding Kroneker-OMP and N-way Block

OMP (N-BOMP) algorithms are also presented for recovery of MD signals with fixed dictionaries. Furthermore, dictionary learning method based on tensor factorization are proposed in [40], and some algorithms are presented to approximate tensor based on tensor decomposition [20] or tensor low-rank approximation [19, 28, 24]. However, to the best of our knowledge, there is no unified framework for tensor-based sparse representation presented in literature.

In this paper, we propose the first **Ten**sor **S**parse model for MD signal **R**epresentation (TenSR in short) along with the corresponding sparse coding and dictionary learning algorithms. Our proposed sparse coding algorithm is a iterative shrinkage thresholding method, which can be easily implemented via the n-mode product of tensor by matrix and element-wise thresholding. We also formulate the dictionary learning problem as an optimization problem solved via a two-phase block-coordinate-relaxation approach including sparse coding and dictionary updating. Both dictionary learning and sparse coding are without Kronecker product so as to greatly reduce the computation burden. In addition, the efficiency of our sparse coding can be further improved through parallel computing. Dictionaries of every dimension (mode) can be updated by a series of quadratically constrained quadratic programming (QCQP) problem via Lagrange dual method. The advantages of our proposed TenSR model as well as the sparse coding and dictionary learning algorithms are demonstrated with the real world 3D signal processing. To summarize, this paper makes the following contributions:

- We propose the first tensor sparse model TenSR for MD signal representation. To the best of knowledge, this is the first paper presenting this theory along with the corresponding sparse coding and dictionary learning algorithms.
- We propose the novel sparse coding and dictionary learning algorithms based on tensor operation rather than Kronecker product, which help in not only revealing structure in each mode inherent in MD signals but also significantly reducing computational as well as memory cost for MD signal processing.
- Our proposed TenSR model is able to empower the sparse representation especially when dealing with high dimensional data (*e.g.* 3D multi-spectral images as demonstrated in experiments) by greatly reducing the processing cost but meanwhile achieving comparable performance with regard to the conventional 1D sparse model.

The rest of this paper is organized as follows. Section 2 reviews the related work on sparse representation for 1D, 2D, and MD signals. Section 3 presents our MD tensor sparse model TenSR, the corresponding MD sparse coding and dictionary learning algorithms, followed by the complexity analysis. In Section 4, we demonstrate the effectiveness of our TenSR model by simulation experiment and 3D multi-spectral image denoising problem. Finally, Section 5 concludes this paper.

2. Related Work

In this section, we briefly review the related work on sparse representation towards 1D, 2D and MD signals.

1D signal (vector) The conventional 1D sparse model represents a vector $\mathbf{x}$ by the linear combination of a few atoms from a large dictionary $\mathbf{D}$, denoted as $\mathbf{x} = \mathbf{D}\mathbf{b}, \|\mathbf{b}\|_0 \leq L$, where L is the sparsity of $\mathbf{b}$. The computational techniques for approximating sparse coefficient $\mathbf{b}$ under a given dictionary $\mathbf{D}$ and $\mathbf{x}$ includes greedy pursuit (*e.g.* OMP [23]) and convex relaxation optimization, such as Lasso [32] and FISTA [3]. Rather than using fixed dictionaries, dictionary learning algorithms [1, 18, 22] are also investigated, which substantially improve the performance of sparse representation [10, 37, 33, 36, 16]. However, the efficiency of 1D sparse coding as well as dictionary learning degrades rapidly as the dimensionality increases.

2D signal (matrix) A matrix $\mathbf{X}$ is sparse modeled by two dictionaries $\mathbf{D}_1$, $\mathbf{D}_2$, and a sparse coefficient matrix $\mathbf{B}$, denoted as $\mathbf{X} = \mathbf{D}_1\mathbf{B}^T\mathbf{D}_2^T, \|\mathbf{B}\|_0 \leq L$, where L is the sparsity of $\mathbf{B}$ [26]. Given dictionaries $\mathbf{D}_1$ and $\mathbf{D}_2$, the 2D sparse model can be easily converted to 1D sparse model $\mathbf{x} = \mathbf{D}\mathbf{b}, \|\mathbf{b}\|_0 \leq L$, where $\mathbf{D} = \mathbf{D}_2 \otimes \mathbf{D}_1$. The dictionaries $\mathbf{D}_1$ and $\mathbf{D}_2$ are learned by the two-phase block-coordinate-relaxation approach via sparse coding and dictionary updating [26] while the sparse coding problem can be solved by 2DSL0 through steepest ascent [12] or the greedy algorithm 2D-OMP [11]. The presented 2D sparse model is able to facilitate 2D signal processing. However,we notice that the sparse coding and dictionary learning algorithms presented for 2D sparse representation are not capable enough for MD signals. On the one hand, the 2D sparse coding problem is recast to 1D one by converting the 2D signal to a long vector via the Kronecker product during dictionary learning [26]. On the other hand, the Riemannian conjugate gradient algorithm and the manifold-based dictionary learning in [14] are quite complex to compute for high order data.

MD signal (tensor) A tensor $\mathcal{X}$ can be represented by a series of known Kroneker dictionaries $\{\mathbf{D}_n\}_{n=1}^N$ and the core tensor $\mathcal{B}$ with the multi-way sparsity or block sparsity constraint, denoted as the Tucker model $\mathcal{X} = \mathcal{B} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \cdots \times_N \mathbf{D}_N$ [5]. Given the fixed Kronecker dictionaries (such as DCT, DWT, and DFT), the Kroneker-OMP and N-BOMP algorithms are also presented in [5] to recovery MD signals possessing Kronecker structure and block-sparsity. However, the Kronecker-OMP and N-OMP algorithm is relatively complicated due to the Kronecker product operation. In addition, some other algorithms are

presented to approximate tensor based on tensor decomposition, such as PARAFAC [20] and tensor factorization under Tucker model [40], or tensor low-rank approximation [19], such as LRTA [30], HOSVD [28], and TensorDL [24]. However, they only decompose and approximate the tensor itself rather than model the tensor.

Different from all previous tensor-based methods, in this paper, we not only propose the tenser sparse model for MD signal representation, but also present the corresponding sparse coding and dictionary learning algorithms. Moreover, since no Kronecker product is used, our proposed scheme is much light-weighted in terms of computational and memory costs for MD signal processing.

3. MD Tensor Sparse Representation

In this section, we present our TenSR model for MD signal representation followed by the corresponding sparse coding and dictionary learning algorithms.

3.1. Notations

For easy understanding, we would like to first introduce some notations used in this paper. A tensor of order N is denoted as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$. The l_0, l_1 and l_F norms of a N- order tensor $\mathcal{X}$ are denoted as $\|\mathcal{X}\|_0 = \#\{\mathcal{X}(i_1, i_2, \cdots, i_N) \neq 0\}$, $\|\mathcal{X}\|_1 = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} |\mathcal{X}(i_1, i_2, \cdots, i_N)|$, and $\|\mathcal{X}\|_F = (\sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} \mathcal{X}(i_1, i_2, \cdots, i_N)^2)^{1/2}$, respectively, where $\mathcal{X}(i_1, i_2, \cdots, i_N)$ is the $(i_1, i_2, \cdots, i_N)$-element of $\mathcal{X}$. n-mode vectors are obtained by fixing every index but the one in the mode n. The n-mode unfolding matrix $\mathcal{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N}$ is defined by arranging all the n-mode vectors as columns of a matrix. Following the formulation of tensor multiplication in [2, 17], we denote the n-mode product of tensor $\mathcal{X}$ and matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$ as $\mathcal{X} \times_n \mathbf{U}$, which is also a N-order tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$, whose entries $\mathcal{Y}(i_1, \cdots, i_{n-1}, j_n, i_{n+1}, \cdots, i_N)$ are computed by $\sum_{i_n=1}^{I_n} \mathcal{X}(i_1, i_2, \cdots, i_N)\mathbf{U}(j_n, i_n)$. The inner product of two same-sized tensors $\mathcal{X}$ and $\mathcal{Y}$ is the sum of the products of their entries, i.e., $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} \mathcal{X}(i_1, i_2, \cdots, i_N)\mathcal{Y}(i_1, i_2, \cdots, i_N)$ Operator $\otimes$ represents the Kronecker product. The vectorization of $\mathcal{X}$ is $\mathbf{x}$.

3.2. The Proposed TenSR model

Let a N-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ denotes a MD signal. In order to approximate the structure and exploit the correlations in every dimension in the MD signal $\mathcal{X}$, we propose the MD TenSR model as

$$\mathcal{X} = \mathcal{B} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \cdots \times_N \mathbf{D}_N, \ \|\mathcal{B}\|_0 \leq K, \quad (1)$$

which formulates the tensor $\mathcal{X}$ as a n-mode product of a $N-$order sparse tensor $\mathcal{B}$ and a series of matrix $\mathbf{D}_n \in \mathbb{R}^{I_n \times M_n}, I_n \leq M_n$. Here $\mathbf{D}_n$ is defined as the n-th dimensional dictionary (or dictionary at mode n) and K is the sparsity denoting the number of the non-zero entries in $\mathcal{B}$. It is seen that there is a formal resemblance between the Tucker model in [5] and our TenSR model in (1); they are in fact quite different. [5] uses Tucker model to only approximate a tensor itself based on given Kronecker dictionaries. Our TenSR model, on the other hand, explores the features and structures of tensors in different dimensions by adaptive MD dictionaries rather than determined analytical dictionaries to model MD signals (tensors).

The dictionaries $\mathbf{D}_n$ in (1) can be learned by unfolding the tensor $\mathcal{X}$ in n-mode, resulting the equivalent unfolded matrix representation [2, 17]

$$\mathcal{X}_{(n)} = \mathbf{D}_n \mathcal{B}_{(n)} (\mathbf{D}_N \cdots \otimes \mathbf{D}_{n+1} \otimes \mathbf{D}_{n-1} \cdots \otimes \mathbf{D}_1)^T. \quad (2)$$

Let $\mathcal{A}_{(n)} = \mathcal{B}_{(n)} (\mathbf{D}_N \otimes \cdots \otimes \mathbf{D}_{n+1} \otimes \mathbf{D}_{n-1} \otimes \cdots \otimes \mathbf{D}_1)^T$. Then, the dictionary learning problem can be solved on the basis of $\mathcal{X}_{(n)} = \mathbf{D}_n \mathcal{A}_{(n)}$, where $\mathbf{D}_n$ is the dictionary of $\mathcal{X}_{(n)}$ that reflects the correlation of $\mathcal{X}$ in the n-th dimension and $\mathcal{A}_{(n)}$ is the corresponding representation on the n-th dimension. However, as mentioned before, the high complexity of the computation of $\mathcal{A}_{(n)}$ by kronecker product will prevent the dictionary learning method from high order data processing. Therefore, a dedicated MD dictionary learning method should be studied for our TenSR model for MD signal representation.

Given learned dictionaries $\{\mathbf{D}_n\}_{n=1}^N$, the TenSR model (1) can be easily converted to the traditional 1D sparse model as

$$\mathbf{x} = \mathbf{D}\mathbf{b}, \ \|\mathbf{b}\|_0 \leq K. \quad (3)$$

where $\mathbf{D} = \mathbf{D}_N \otimes \mathbf{D}_{N-1} \otimes \cdots \otimes \mathbf{D}_1$, $\mathbf{D} \in \mathbb{R}^{I \times M}$, $I = \prod_{n=1}^N I_n$, and $M = \prod_{n=1}^N M_n$. $\mathbf{x}$ and $\mathbf{b}$ are the vectorization of $\mathcal{X}$ and $\mathcal{B}$, respectively. However, with the increase of the dimension of MD signal, the size of the dictionary $\mathbf{D}$ will be exponentially expanded. No matter how large the dictionary is, the correlations that inherently exist in each domain of MD signals are ignored due to the vectorization in 1D sparse model. The sparse coding method presented in [5] may also be adopted here but still with Kronecker product. Thus a new sparse coding method is presented in the following subsection to solve these problems.

3.3. MD sparse coding

In this subsection, we discuss how to calculate the sparse coefficient $\mathcal{B}$ given dictionaries $\{\mathbf{D}_n\}_{n=1}^N$ under our TenSR model.

Given all the sparse dictionaries $\{\mathbf{D}_n\}_{n=1}^N$, calculating $\mathcal{B}$ from $\mathcal{X}$ is a MD sparse coding problem formulated as

$$\min_{\mathcal{B}} \frac{1}{2} \|\mathcal{X} - \mathcal{B} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \cdots \times_N \mathbf{D}_N\|_F^2 + \lambda \|\mathcal{B}\|_0, \quad (4)$$

Algorithm 1 Tensor-based Iterative Shrinkage Thresholding

Initialization: Set $\mathcal{C}_1 = \mathcal{B}_0 \in \mathbb{R}^{M_1 \times M_2 \cdots \times M_N}$, $t_1 = 1$
For $k = 1 : num$ **do**
  Set $L^k = \eta^k \prod_{n=1}^{N} \|\mathbf{D}_n^T \mathbf{D}_n\|_2$
  Compute $\nabla f(\mathcal{C}^k)$ via Eq.(9)
  Compute $\mathcal{B}^k$ via $P_{\lambda/L_k}(\mathcal{C}_k - \frac{1}{L}{}_k \nabla f(\mathcal{C}_k))$
  $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
  $\mathcal{C}_{k+1} = \mathcal{B}_k + \frac{t_k - 1}{t_{k+1}}(\mathcal{B}_k - \mathcal{B}_{k-1})$
End For
Output: Sparse coefficient $\mathcal{B}$

where λ is a parameter to balance the fidelity and sparsity. The non-convex l_0 constraint in (4) can be relaxed to the l_1 norm to yield a convex optimization problem as

$$\min_{\mathcal{B}} \frac{1}{2}\|\mathcal{X} - \mathcal{B} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \cdots \times_N \mathbf{D}_N\|_F^2 + \lambda \|\mathcal{B}\|_1. \quad (5)$$

Rather than using 1D sparse coding or Kronecker product based methods, we propose a new sparse coding algorithm − Tensor-based Iterative Shrinkage Thresholding Algorithm (TISTA) to solve (4) as well as (5) directly. We first rewrite the objective function w.r.t $\mathcal{B}$ in (4) or (5) as

$$\min_{\mathcal{B}} f(\mathcal{B}) + \lambda g(\mathcal{B}), \quad (6)$$

where $f(\mathcal{B})$ stands for the data-fitting term $\frac{1}{2}\|\mathcal{X} - \mathcal{B} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \cdots \times_N \mathbf{D}_N\|_F^2$ and $g(\mathcal{B})$ stands for the sparsity constraint term $\|\mathcal{B}\|_1$ or $\|\mathcal{B}\|_0$. We take a iterative shrinkage algorithm to solve the non-smooth regularized problem (6), which can be rewritten as a linearized function f around the previous estimate $\mathcal{B}_{k-1}$ with the proximal regularization [15] and the non-smooth regularization. Thus, at the k-th iteration, $\mathcal{B}_k$ can be updated by

$$\begin{aligned}\mathcal{B}_k = \operatorname*{argmin}_{\mathcal{B}} & f(\mathcal{B}_{k-1}) + \langle \nabla f(\mathcal{B}_{k-1}), \mathcal{B} - \mathcal{B}_{k-1} \rangle \\ & + \frac{L_k}{2}\|\mathcal{B} - \mathcal{B}_{k-1}\|_F^2 + \lambda g(\mathcal{B}),\end{aligned} \quad (7)$$

where $L_k > 0$ is a Lipschitz constant [15] and $\nabla f(\mathcal{B})$ is a gradient defined on the tensor-field. Then this problem can be rewritten as,

$$\mathcal{B}_k = \operatorname*{argmin}_{\mathcal{B}} \frac{1}{2}\|\mathcal{B} - (\mathcal{B}_{k-1} - \frac{1}{L_k}\nabla f(\mathcal{B}_{k-1}))\|_F^2 + \frac{\lambda}{L_k} g(\mathcal{B}). \quad (8)$$

To solve (8), we first deduce $\nabla f(\mathcal{B})$ with respect to the data fidelity term $\frac{1}{2}\|\mathcal{X} - \mathcal{B} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \cdots \times_N \mathbf{D}_N\|_F^2$ in our TenSR satisfies

$$\begin{aligned}-\nabla f(\mathcal{B}) = & \mathcal{X} \times_1 \mathbf{D}_1^T \times_2 \mathbf{D}_2^T \cdots \times_N \mathbf{D}_N^T \\ & - \mathcal{B} \times_1 \mathbf{D}_1^T \mathbf{D}_1 \times_2 \mathbf{D}_2^T \mathbf{D}_2 \cdots \times_N \mathbf{D}_N^T \mathbf{D}_N.\end{aligned} \quad (9)$$

We argue that our solution via (9) is equivalent to the corresponding 1D sparse coding by proximal method. Note that the problems (4) and (5) can be converted to the corresponding 1D sparse coding problems as

$$\min_{\mathbf{b}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_0. \quad (10)$$

$$\min_{\mathbf{b}} \frac{1}{2}\|\mathbf{x} - \mathbf{D}\mathbf{b}\|_2^2 + \lambda \|\mathbf{b}\|_1. \quad (11)$$

By using the formula of tensor and kronecker product in [2, 17], the right term in (9) can be converted as

$$\begin{aligned}&(\mathbf{D}_N^T \otimes \mathbf{D}_{N-1}^T \otimes \cdots \otimes \mathbf{D}_1^T)\mathbf{x} \\ &-(\mathbf{D}_N^T \mathbf{D}_N \otimes \mathbf{D}_{N-1}^T \mathbf{D}_{N-1} \cdots \otimes \mathbf{D}_1^T \mathbf{D}_1)\mathbf{b},\end{aligned} \quad (12)$$

which can be derived to

$$\mathbf{D}^T\mathbf{x} - \mathbf{D}^T\mathbf{D}\mathbf{b}, \quad (13)$$

with the equal term $(\mathbf{D}_N \otimes \mathbf{D}_{N-1} \otimes \cdots \otimes \mathbf{D}_1)^T\mathbf{x} - (\mathbf{D}_N^T \otimes \mathbf{D}_{N-1}^T \cdots \otimes \mathbf{D}_1^T)(\mathbf{D}_N \otimes \mathbf{D}_{N-1} \cdots \otimes \mathbf{D}_1)\mathbf{b}$. It can be observed that $-(\mathbf{D}^T\mathbf{x} - \mathbf{D}^T\mathbf{D}\mathbf{b})$ in (9) is equivalent to the gradient of the data-fitting term $\frac{1}{2}\|\mathbf{x} - \mathbf{D}\mathbf{b}\|_2^2$ at vector $\mathbf{b}$ in 1D sparse coding problems (10) and (11), thus making these two solutions equivalent.

We then discuss how to determine the Lipschitz constant L_k in (8). We assume f is a smooth convex function of the type $C^{1,1}$. That is, for every $\mathcal{B}, \mathcal{C} \in \mathbb{R}^{M_1 \times M_2 \cdots \times M_N}$, f is continuously differentiable with Lipschitz continuous gradient $L(f)$ satisfying

$$\|\nabla f(\mathcal{B}) - \nabla f(\mathcal{C})\| \leq L(f)\|\mathcal{B} - \mathcal{C}\|. \quad (14)$$

where $\|\cdot\|$ denotes the l_F norm on N-order tensor and $L(f)$ is the Lipschitz constant of ∇f. Substitute (9) to (14), we have

$$\begin{aligned}&\|\nabla f(\mathcal{B}) - \nabla f(\mathcal{C})\|_F \\ &= \|(\mathcal{B} - \mathcal{C}) \times_1 \mathbf{D}_1^T \mathbf{D}_1 \times_2 \mathbf{D}_2^T \mathbf{D}_2 \cdots \times_N \mathbf{D}_N^T \mathbf{D}_N\|_F \\ &= \|(\mathbf{D}_N^T \mathbf{D}_N \otimes \mathbf{D}_{N-1}^T \mathbf{D}_{N-1} \cdots \otimes \mathbf{D}_1^T \mathbf{D}_1)(\mathbf{b} - \mathbf{c})\|_2 \\ &\leq \|\mathbf{D}_N^T \mathbf{D}_N \otimes \mathbf{D}_{N-1}^T \mathbf{D}_{N-1} \cdots \otimes \mathbf{D}_1^T \mathbf{D}_1\|_2 \|\mathbf{b} - \mathbf{c}\|_2 \\ &= \|\mathbf{D}_N^T \mathbf{D}_N\|_2 \|\mathbf{D}_{N-1}^T \mathbf{D}_{N-1}\|_2 \cdots \|\mathbf{D}_1^T \mathbf{D}_1\|_2 \|\mathbf{b} - \mathbf{c}\|_2 \\ &= \|\mathbf{D}_N^T \mathbf{D}_N\|_2 \|\mathbf{D}_{N-1}^T \mathbf{D}_{N-1}\|_2 \cdots \|\mathbf{D}_1^T \mathbf{D}_1\|_2 \|\mathcal{B} - \mathcal{C}\|_F.\end{aligned} \quad (15)$$

Thus the smallest Lipschitz constant of the gradient ∇f is $L(f) = \prod_{n=1}^{N} \|\mathbf{D}_n^T \mathbf{D}_n\|_2$. Then, in our iteration process, $L_k = \eta^k \prod_{n=1}^{N} \|\mathbf{D}_n^T \mathbf{D}_n\|_2$ with $\eta \geq 1$, i.e. $L_k \geq L(f)$.

Finally, we present the solution of (8) with different sparsity constraint $g(\mathcal{B})$. With the regularization term $g(\mathcal{B}) = \|\mathcal{B}\|_1$, the proximal operator $P_\tau(\cdot)$ for solving (8) is the (elementwise) soft-thresholding operator. Thus, the unique solution of (8) is $S_{\lambda/L_k}(\mathcal{B}_{k-1} - \frac{1}{L}{}_k \nabla f(\mathcal{B}_{k-1}))$, where $S_\tau(\cdot)$ is the soft-thresholding operator $S_\tau(\cdot) \longmapsto sign(\cdot)\max(|\cdot| - \tau, 0)$. In case of the l_0 norm sparsity constraint, i.e., $g(\mathcal{B}) = \|\mathcal{B}\|_0$, the solution of (8) is

Algorithm 2 Tensor-based Dictionary Learning

Input: Training Set $\mathcal{I}$, number of iteration num
Initialization: Set the dictionary $\{\mathbf{D}_n\}_{n=1}^N$.
For $l = 1 \ : \ num$
Sparse coding Step:
Compute $\mathcal{J}$ via Eq.(18) according to Algorithm 1.
Dictionary Update Step:
$\mathcal{A} = \mathcal{J} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \cdots \mathbf{D}_{n-1} \times_{n+1} \mathbf{D}_{n+1} \cdots \times_N \mathbf{D}_N$
Get $\mathcal{A}_{(n)}$ and Update $\mathbf{D}_n$ via Eq.(19).
End For
Output: Learned dictionaries $\{\mathbf{D}_n\}_{n=1}^N$

$H_{\lambda/L_k}(\mathcal{B}_{k-1} - \frac{1}{L_k}\nabla f(\mathcal{B}_{k-1}))$, where $H_\tau(\cdot)$ is the hard-thresholding operator $H_\tau(\cdot) \longmapsto \max(\cdot - \tau, 0)$. For convenience of algorithm description, we denote $P_{\lambda/L_k}(\mathcal{B}_{k-1} - \frac{1}{L_k}\nabla f(\mathcal{B}_{k-1}))$ as the solution of (8) with either l_1 norm or l_0 norm.

We further speed up the convergence of the iterative shrinkage algorithm by employing the iterative shrinkage operator at the point $\mathcal{C}_k$ where

$$\mathcal{C}_k = \mathcal{B}_{k-1} + \zeta_k(\mathcal{B}_{k-1} - \mathcal{B}_{k-2}) \tag{16}$$

and $\zeta_k > 0$ is a suitable step size, rather than at the point $\mathcal{B}_{k-1}$. We also set $\zeta_k = (t_k - 1)/t_{k+1}$ where $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$ [3] and this extrapolation significantly accelerates the proximal gradient method for convex composite problem [35]. **Algorithm 1** summarizes our proposed Tensor-based Iterative Shrinkage Thresholding Algorithm for MD sparse coding.

3.4. MD dictionary learning

Given a set of training samples $\mathcal{I} = (\mathcal{X}^1, \mathcal{X}^2, \cdots, \mathcal{X}^S) \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N \times S}$, where S denotes the number of N-order tensors $\mathcal{X}^j \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$, we formulate our MD dictionary learning problem as

$$\begin{aligned} &\min_{\{\mathbf{D}_n\}_{n=1}^N, \mathcal{J}} \frac{1}{2}\|\mathcal{I} - \mathcal{J} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \cdots \times_N \mathbf{D}_N\|_F^2 + \lambda\|\mathcal{J}\|_1 \\ &s.t. \ \|\mathbf{D}_n(:, r)\|_2^2 = 1, \ 1 \le j \le S, \ 1 \le n \le N, \ 1 \le r \le M_n, \end{aligned} \tag{17}$$

where $\mathcal{J} = (\mathcal{B}^1, \mathcal{B}^2, \cdots, \mathcal{B}^S) \in \mathbb{R}^{M_1 \times M_2 \cdots \times M_N \times S}$ denotes the set of sparse coefficients of all the training samples $\mathcal{I}$, λ is a parameter to balance the fidelity and sparsity, and $\{\mathbf{D}_n\}_{n=1}^N$ are targeted MD separate dictionaries.

The problem (17) can be solved by using a two-phase block-coordinate-relaxation approach via sparse coding and dictionary updating. The process repeats until certain stop criterion is satisfied, *e.g.* the relative error of the objective function at adjacent two iteration is below some predetermined level ϵ. **Algorithm 2** summarizes our Tensor-based Dictionary Learning method.

Sparse coding aims to approximate the sparse coefficient $\mathcal{J}$ of the training set $\mathcal{I}$ with fixed $\{\mathbf{D}_i\}_{i=1}^N$ by solving

$$\min_{\mathcal{J}} \frac{1}{2}\|\mathcal{I} - \mathcal{J} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \cdots \times_N \mathbf{D}_N\|_F^2 + \lambda\|\mathcal{J}\|_1. \tag{18}$$

We are able to directly solve (18) by the MD sparse coding algorithm described in Sec. 3.3, rather than solving S independent MD sparse coding optimization problems with respect to each N-order signal $\mathcal{X}^j$ [1, 26]. In addition, we can divide all the samples to different subsets and solve the sparse coding problem for each subset in parallel to generate the final sparse coefficient $\mathcal{J}$. Thus our sparse coding process runs much faster than the other related solutions.

Dictionary update tries to update $\{\mathbf{D}_n\}_{n=1}^N$ using the computed sparse coefficients $\mathcal{J}$. The optimization procedures for $\{\mathbf{D}_n\}_{n=1}^N$ are similar. Without loss of generality, we take the updating of $\mathbf{D}_n$ as an example to present our dictionary update method. Due to the interchangeability of n-mode product in our TenSR model (1), each tensor $\mathcal{X}^j$ satisfies $\mathcal{X}^j = \mathcal{A}^j \times_n \mathbf{D}_n$ with $\mathcal{A}^j = \mathcal{B}^j \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \cdots \times_{n-1} \mathbf{D}_{n-1} \times_{n+1} \mathbf{D}_{n+1} \cdots \times_N \mathbf{D}_N$, thus $\mathcal{A}_{(n)}^j$ can be easily obtained by unfolding the tensor $\mathcal{A}^j$ rather than in the way by kronecker product mentioned in Sec. 3.2. Therefore, we first calculate $\mathcal{A} \in \mathbb{R}^{M_1 \times M_2 \cdots \times M_{n-1} \times I_n \times M_{n+1} \times \cdots \times M_N \times S}$[1] by $\mathcal{J} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \cdots \times_{n-1} \mathbf{D}_{n-1} \times_{n+1} \mathbf{D}_{n+1} \cdots \times_N \mathbf{D}_N$ to make sure $\mathcal{I} \approx \mathcal{A} \times_n \mathbf{D}_n$, and then unfold $\mathcal{A}$ in n-mode to obtain $\mathcal{A}_{(n)}$ to guarantee $\mathcal{I}_{(n)} \approx \mathbf{D}_n \mathcal{A}_{(n)}$. Thus, $\mathbf{D}_n$ can be updated by

$$\begin{aligned} &\widehat{\mathbf{D}}_n = \underset{\mathbf{D}_n}{\operatorname{argmin}} \|\mathcal{I}_{(n)} - \mathbf{D}_n \mathcal{A}_{(n)}\|_F^2, \\ &s.t. \ \|\mathbf{D}_n(:, r)\|_2^2 = 1, \ 1 \le r \le M_n. \end{aligned} \tag{19}$$

It is a quadratically constrained quadratic programming (QCQP) problem, where $\mathcal{I}_{(n)} \in \mathcal{R}^{I_n \times H_n}$ and $\mathcal{A}_{(n)} \in \mathcal{R}^{M_n \times H_n}$ are the mode-n unfolding matrix of $\mathcal{I}$ and $\mathcal{A}$, respectively. Here $H_n = I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N S$. The problem (19) can be resolved via the Lagrange dual [18]. The Lagrangian $\mathcal{L}$ here is $\mathcal{L}(\mathbf{D}_n, \boldsymbol{\lambda}) = trace((\mathcal{I}_{(n)} - \mathbf{D}_n \mathcal{A}_{(n)})^T (\mathcal{I}_{(n)} - \mathbf{D}_n \mathcal{A}_{(n)}) + \sum_{j=1}^{M_n} \lambda_j (\sum_{i=1}^{I_n} \mathbf{D}_n(i,j)^2 - 1))$, where each $\lambda_j \ge 0$ is a dual variable. Thus, the Lagrange dual function $\mathcal{D}(\boldsymbol{\lambda}) = \min_{\mathbf{D}_n} \mathcal{L}(\mathbf{D}_n, \boldsymbol{\lambda})$ can be optimized by the Newton's method or conjugate gradient. After maximizing $\mathcal{D}(\boldsymbol{\lambda})$, we obtain the optimal bases $\mathbf{D}_n^T = (\mathcal{A}_{(n)} \mathcal{A}_{(n)}^T + \boldsymbol{\Lambda})^{-1} (\mathcal{I}_{(n)} \mathcal{A}_{(n)}^T)^T$, where $\boldsymbol{\Lambda} = diag(\boldsymbol{\lambda})$. Compared with [40] and [26], the new way of computing $\mathcal{A}_{(n)}$ without Kronecker product can greatly reduce the computation complexity of our dictionary updating.

3.5. Complexity Analysis

In this subsection, we discuss the complexity as well as the memory usage of our sparse coding and dictionary

[1] $\mathcal{A}$ is a function of n, however the subscript n is omitted for brevity.

Table 1. Complexity Analysis of Sparse Coding (SC) and Dictionary Update (DU) for MD and 1D Sparse Model

		Operation	Complexity in Detail	Complexity
SC	1D	$\mathbf{D}^T\mathbf{x} - \mathbf{D}^T\mathbf{D}\mathbf{b}$	$O(IM + IM + MIM + MM)$	$O(IM^2)$
	MD	$\nabla f(\mathcal{B})$	$O(\sum_{n=1}^{N}(\prod_{i=1}^{n} M_i \prod_{j=n}^{N} I_j + M_n I_n M_n + M_n M))$	$O(\sum_{n=1}^{N} M_n M)$
DU	1D	$\min_{\mathbf{D}} \|\mathbf{I} - \mathbf{D}\mathbf{B}\|_F^2$	$O(MSM + M^3 + ISM + MMI)$	$O(M^2 S)$
	MD	$\mathcal{A}_{(n)}$ by kronecker product	$O(IM/(M_n I_n) + IM/M_n S)$	O($\sum_{n=1}^{N} \sum_{\substack{k=1 \\ k\neq n}}^{N} \prod_{i=1}^{k} M_i \prod_{j=k}^{N} I_j I_n S$)*
		$\mathcal{A}_{(n)}$ by n-mode product	$O(\sum_{\substack{k=1 \\ k\neq n}}^{N} (\prod_{i=1}^{k} M_i \prod_{j=k}^{N} I_j I_n S))$	
		$\min_{\mathbf{D}_n} \|\mathcal{I}_{(n)} - \mathbf{D}_n \mathcal{A}_{(n)}\|_F^2$	$O(M_n^2 H_n)^+$	

$^+$ $H_n = I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N S$

* The n-mode product method for $\mathcal{A}_{(n)}$ is less complicated than the Kronecker Product one. Thus, here only summarize the complexity of $\mathcal{A}_{(n)}$ by n-mode product and $\min_{\mathbf{D}_n} \|\mathcal{I}_{(n)} - \mathbf{D}_n \mathcal{A}_{(n)}\|_F^2$.

Table 2. Time Complexity of SC and DU, and Memory Usage of Dictionary for MD and 1D Sparse Model

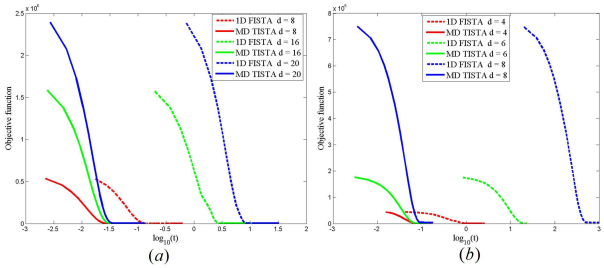
	Time Complexity		Memory
	SC	DU	
1D	$O(c^{2N} d^{3N})$	$O(c^{2N} d^{2N} S)$	$\prod_{n=1}^{N} M_n I_n$
MD	$O(N c^{N+1} d^{N+1})$	$O(N c^N d^{N+2} S)$	$\sum_{n=1}^{N} M_n I_n$

learning algorithms with regard to those of conventional 1D counterparts.

We first analyze complexities of the main components of MD and 1D sparse coding (SC) and dictionary updating (DU) algorithms and summarized in Table 1. In terms of SC, Table 1 shows the complexity of calculating $\nabla f(\mathcal{B})$ and $\mathbf{D}^T\mathbf{x} - \mathbf{D}^T\mathbf{D}\mathbf{b}$, which cost most of time in SC step at each iteration. For a N-order signal $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$, the MD sparse coefficient $\mathcal{B} \in \mathbb{R}^{M_1 \times M_2 \cdots \times M_N}$ is computed with fixed dictionaries $\{\mathbf{D}_n\}_{n=1}^N$, where $\mathbf{D}_n \in \mathbb{R}^{I_n \times M_n}$. Correspondingly, the 1D sparse coefficient $\mathbf{b} \in \mathbb{R}^M$ is sparse approximated by the 1D dictionary $\mathbf{D} \in \mathbb{R}^{I \times M}$ and $\mathbf{x} \in \mathbb{R}^I$, where $I = \prod_{n=1}^{N} I_n$, and $M = \prod_{n=1}^{N} M_n$.

In terms of DU, given a set of training samples $\mathcal{I} = (\mathcal{X}^1, \mathcal{X}^2, \cdots, \mathcal{X}^S) \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N \times S}$, we learn MD dictionaries $\{\mathbf{D}_n\}_{n=1}^N$, where $\mathbf{D}_n \in \mathbb{R}^{I_n \times M_n}$. In order to update $\mathbf{D}_n$ via (19), we need calculate $\mathcal{A}_{(n)}$ in our scheme. In fact, $\mathcal{A}_{(n)}$ can be computed in two ways, a) $n-$mode product which directly unfolds the tensor $\mathcal{A} = \mathcal{J} \times_1 \mathbf{D}_1 \times_2 \mathbf{D}_2 \cdots \times_{n-1} \mathbf{D}_{n-1} \times_{n+1} \mathbf{D}_{n+1} \cdots \times_N \mathbf{D}_N$, and b) Kronecker product, $\mathcal{A}_{(n)} = [\mathcal{A}_{(n)}^1, \mathcal{A}_{(n)}^2, \cdots, \mathcal{A}_{(n)}^S]$ where $\mathcal{A}_{(n)}^j = \mathcal{B}_{(n)}^j (\mathbf{D}_N \cdots \otimes \mathbf{D}_{n+1} \otimes \mathbf{D}_{n-1} \cdots \otimes \mathbf{D}_1)^T$ [40]. The complexity of these two ways are all given in Table 1. Clearly, our $n-$mode product method is less complicated than the Kronecker product one. For 1D dictionary learning, the correspondingly 1D training set is $\mathbf{I} = [\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^S] \in \mathbb{R}^{I \times S}$, thus the 1D dictionaries $\mathbf{D} \in \mathbb{R}^{I \times M}$ is updated by $\min_{\mathbf{D}} \|\mathbf{I} - \mathbf{D}\mathbf{B}\|_F^2$, where $\mathbf{B} = [\mathbf{b}^1, \mathbf{b}^2, \cdots, \mathbf{b}^S] \in \mathbb{R}^{M \times S}$.

Table 2 summarizes the total time complexity of SC and DU for 1D and MD sparse model. Without loss of generality, we assume $I_n = d$ and M_n is c times of I_n, denoted as $M_n = cd$, where c reflects the redundancy rate of dictionary $\mathbf{D}_n$. We can observe that our proposed MD sparse coding and dictionary learning algorithms will greatly reduce the time complexity especially for high order signals. In addition, the memory usage of our MD model is also significantly less than that of the 1D model.



(a) (b)

Figure 1. Convergence rates of sparse coding algorithms 1D FISTA[3] and our MD TISTA. $Y-$label is objective function value of (5), $X-$label is the computational time in the $log_{10}(t)$ coordinate. (a) shows the case of 2D patch ($N = 2$) of size $d \times d$ and (b) is the one of 3D cube ($N = 3$) of size $d \times d \times d$, respectively.

4. Experimental Results

We demonstrate the effectiveness of our TenSR model by first discussing the convergence of our dictionary learning and sparse coding algorithms in the Simulation Experiment and then evaluate the performance on 3D Multispectral Image (MSI) Denoising.

4.1. Simulation Experiment

Fig. 1 shows the convergence rate of our sparse coding algorithm Tensor-based Iterative Shrinkage Thresholding Algorithm (TISTA) with regard to that of the classic 1D sparse coding method FISTA [3]. Two sets of convergences curves are shown in Fig 1 (a) and (b) for 2D patch ($N = 2$) at sizes $d \times d$ ($d = 8, 16, 20$) and 3D cube ($N = 3$) at sizes $d \times d \times d$ ($d = 4, 6, 8$), respectively. The dictionaries used in both simulations are Overcomplete DCT (ODCT) dictionaries $\{\mathbf{D}_n\}_{n=1}^N$, where $\mathbf{D}_n \in \mathbb{R}^{d \times cd}, c = 2$ (definitions of parameters can be found in subsection 3.5). This figure shows that the reconstruction precisions determined by (5) of these two methods are similar whereas the convergence

Table 3. Time complexity (in seconds) of recovering three sets of sampling cubes of 1D FISTA as well as our TISTA. Here Single, Batch, and All denote that the reconstruction are performed sequentially, in batch of 500, and altogether, respectively.

	FISTA	TISTA		
Cube Size (cubes)	Single	Single	Batch	All
$12 \times 12 \times 31(1758)$	15674	247.7	16.9	16.1
$16 \times 16 \times 31(3888)$	35912	556.2	36.4	35.4
$32 \times 32 \times 31(21168)$	193490	3038.7	200.7	189.0

times (in logarithmic coordinates) are quite different. Our TISTA method converge much more rapidly. The higher the dimension as well as data size, the higher the acceleration of our sparse coding algorithm.

We further evaluate the time efficiency of our TISTA for recovering a series of MD signals in comparison with that of 1D FISTA in Table 3. In this simulation, we sample cubes of size $5 \times 5 \times 5$ from a 3D sub-MSI of size $L \times W \times H (12 \times 12 \times 31, 16 \times 16 \times 31$, and $32 \times 32 \times 31)$. ODCT dictionaries $\{\mathbf{D}_n\}_{n=1}^3$, where $\mathbf{D}_n \in \mathbb{R}^{5\times 10}$, are used for the reconstructions. As illustrated in Fig. 1, TISTA and FISTA are similar in precision at each iteration. We thus measure the time efficiencies of these two methods by the running time of reconstructing a same number of sampled cubes at iteration $num = 50$ and $\lambda = 1$. As shown in Table 3, three set of time complexities are provided for TISTA when the cubes are recovered sequentially (Single), in batch of 500 (Batch), and altogether (All), respectively. We provide the complexity comparisons in sequential in both Fig. 1 and Tab. 3, respectively. It is clear that our sparse coding is much fast in this case. Moreover, our scheme supports parallel naturally and can be easily speeded up as shown in Tab. 3.

The convergence of the presented MD dictionary learning algorithm is evaluated in Fig. 3. Here we train three dictionaries $\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3$ of size 5×10 from $40,000$ cubes, which are of size $5 \times 5 \times 5$ randomly sampled from the 3D Multi-spectral images 'beads' [38]. The learned 3D dictionaries $\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3$ are illustrated Fig. 2. These two figures show that our dictionary learning method is able to capture the feature at each dimension along with the convergence property.

4.2. Multi-spectral Image Denoising

In this subsection, we evaluate the performance of our TenSR model using 3D real-word examples − MSI images in Columbia MSI Database [38][2]. The denoising problem which has been widely studied in sparse representation is used as the target application. We add Gaussian white noise to these images at different noise levels $\sigma = 5, 10, 20, 30, 50$. In our TenSR-based denoising method, the 3D dictionaries $\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3$ of size 5×10 are

[2]The dataset contains 32 real-world scenes at a spatial resolution of 512×512 and a spectral resolution of 31 ranging from 400nm to 700nm.

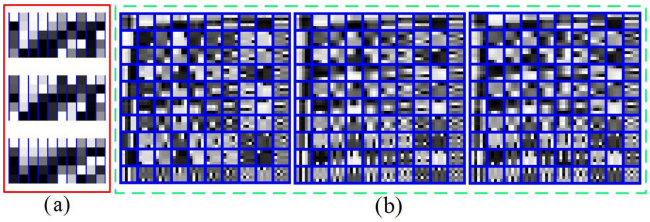


Figure 2. Exemplified dictionary in our TenSR model. (a) Learned dictionaries $\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3$ using TenSR model and (b) The Kroneker product $\mathbf{D}$ of learned dictionaries in (a) of arbitrary dimensions, where each column of $\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3$ is an atom of each dimension of the cube and each square of $\mathbf{D}$ is an atom of size 5×5.

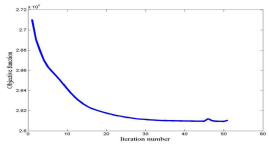


Figure 3. Convergent Analysis. The X-label is the iteration number and the Y-label is the objective function of Eq.(17). It is shown the our Tensor-based dictionary learning algorithm is convergent.

initialized by ODCT and trained iteratively ($\leq$ 50 times) in the same configuration of Fig. 3. Then we use the learned dictionaries to denoise the MSI images, with overlap of 3 pixels between adjacent cubes of size $5 \times 5 \times 5$. Parameters in our scheme are $\lambda = 9, 20, 45, 70, 160$ for $\sigma = 5, 10, 20, 30, 50$, respectively.

Table 4 shows the comparison results in terms of average PSNR and SSIM. There are 6 state-of-the-art MSI denoising methods are involved, including tensor dictionary learning (Tensor-DL) method [24], BM4D method [21], PARAFAC method [20], low-rank tensor approximation (LRTA) method [30], band-wise KSVD (BwK-SVD) method [10] and 3D-cube KSVD (3DK-SVD) method [10][3]. We further classify these methods in two categories (1) without any extra constraint, *e.g.* nonlocal similarity, and (2) with additional prior like nonlocal similarity. As shown in Table 4, our current solution belongs to category (1). Our scheme outperforms most of methods and is comparable with LRTA in category (1). Due to lack of additional constraint, all the algorithms in category (1) achieves lower PSNR and SSIM values than those in category (2). Fig. 4 shows one exemplified visual result of a portion of the MSI image 'cloth' at the 420nm band with Gaussian noise at $\sigma = 10$.

We would like to point out that the size of our dictionary is the smallest among those of all sparse-based test methods including Bw-KSVD [10], 3DK-SVD [10], and Tensor-DL [24]. The total size of dictionary of our method is $3 \times 5 \times 10$ as we learned three small dictionaries of size 5×10. The total size of learned dictionaries of Bw-KSVD is $64 \times 128 \times 31$, where a dictionary of size 64×128 is trained for each band image of all the 31 frame images. A dictio-

[3]We thank all the authors of [24, 21, 20, 30, 10] for providing their source codes in their websites.

Table 4. Average PSNR and SSIM results of the different methods for different noise levels on the set of test multispectral images. (1) Methods without nonlocal similarity, (2) Methods with nonlocal similarity and additional priority.

	method	$\sigma = 5$		$\sigma = 10$		$\sigma = 20$		$\sigma = 30$		$\sigma = 50$	
(1)	PARAFAC [20]	32.77	0.8368	32.72	0.8344	32.48	0.8235	32.15	0.8052	30.22	0.7051
	BwK-SVD [10]	37.79	0.8873	34.11	0.7854	30.99	0.6571	29.34	0.5727	27.35	0.4614
	3DK-SVD [10]	39.47	0.9199	36.33	0.8612	33.47	0.7927	31.80	0.7457	29.63	0.6761
	LRTA [30]	43.69	0.9664	40.56	0.9421	37.29	0.9018	35.29	0.8661	32.71	0.8030
	TenSR	43.74	0.9750	39.05	0.9264	35.01	0.8473	33.31	0.7837	31.38	0.7778
(2)	BM4D [21]	47.72	0.9894	44.33	0.9792	40.70	0.9560	38.46	0.9289	35.55	0.8687
	TensorDL [24]	47.29	0.9896	44.05	0.9800	40.57	0.9638	38.53	0.9482	35.86	0.9139

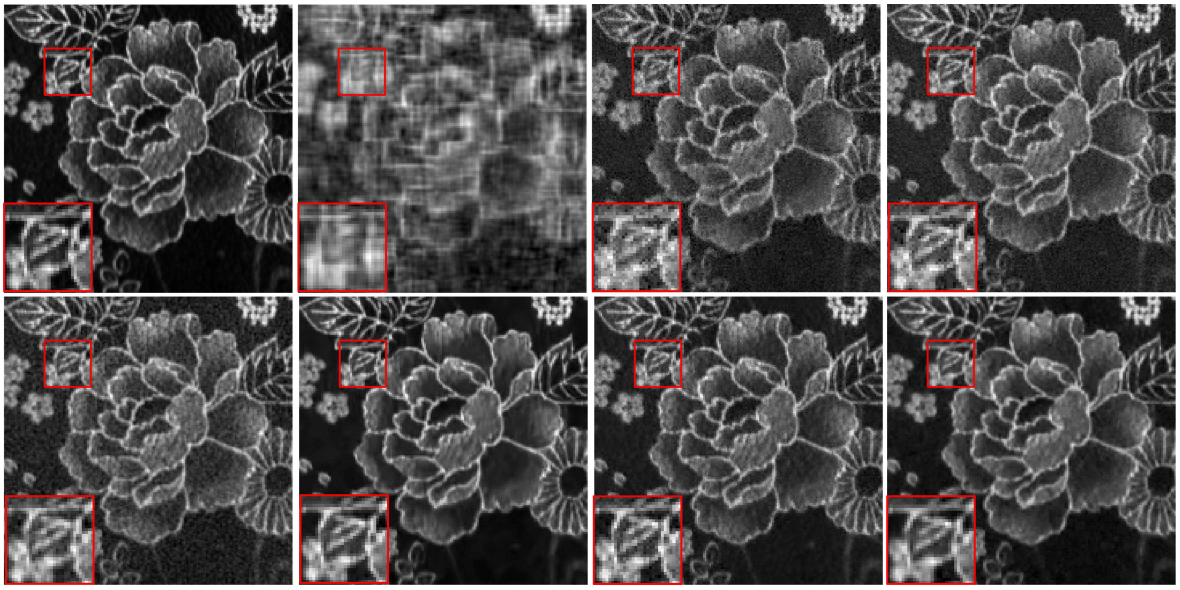


Figure 4. Visual comparison of reconstruction results by different methods on 'cloth' in dataset [38]. From left to right: Original image at 420nm band, PARAFAC [20], BwK-SVD [10], 3DK-SVD [10], LRTA [30], BM4D [21], TensorDL [24], and Ours.

nary of size 448×500 is learned in 3DK-SVD [10] with each cube size $8 \times 8 \times 7$. The Tensor-DL [24] has a dictionary of size $8 \times 8 \times 31 \times 648$ by first building 162 groups of 3D band patches with each cube of size $8 \times 8 \times 31$ and then obtaining a dictionary with 4 atoms for each groups via Tucker decomposition. The total dictionary size of BwKSVD [10], 3DK-SVD [10], and Tensor-DL [24] are **1693**, **1493**, and **8570** times of our TenSR model, respectively. We believe that if we integrate nonlocal similarity to our model and train dictionaries for each group of MD signals, our performance for denoising will be significantly improved.

5. Conclusion

In this paper, we propose the first tensor sparse model TenSR to capture features and explore the correlations inherent in MD signals. We also propose the corresponding formulations as well as algorithms for calculating MD sparse coefficients and learning MD dictionaries. The proposed MD sparse coding algorithm by proximal method reduces the time complexity significantly so as to facilitate the dictionary learning and the recovery problem for high order data. The presented dictionary learning method is capable of approximating structures in each dimension via a series of adaptive separable structure dictionaries. We further analyze the properties of the TenSR model in terms of convergence as well as complexity. The presented TenSR model is applied to 3D multi-spectral image denoising and achieves competitive performance with the state-of-the-art related methods but with much lower time complexity and memory cost.

On the other hand, as shown in the denoising results, the performance of our current solution is not as good as the ones with additional prior, *e.g.* nonlocal similarity. We would like to further improve the performance of our algorithm by introducing prior, *e.g.* non-local similarities, in our future work. Moreover, in our current TenSR model, we assign each dimension a dictionary which may not be adaptive enough. For dimensions who have similar structures or strong correlations, we may support much flexible combinations of dimensions in our future dictionary learning algorithm.

Acknowledgements This work was supported by the NSFC (61390510, 61370118, 61472018) and PHR (IHLB).

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.*, 54(11):4311–4322, 2006.

[2] B. W. Bader and T. G. Kolda. Algorithm 862: Matlab tensor classes for fast algorithm prototyping. *ACM Trans. Math. Software*, 32(4):635–653, 2006.

[3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imag. Sci.*, 2(1):183–202, 2009.

[4] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev.*, 51(1):34–81, 2009.

[5] C. F. Caiafa and A. Cichocki. Computing sparse representations of multidimensional signals using kronecker bases. *Neural Comput.*, 25(1):186–220, 2013.

[6] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Process. Mag.*, 32(2):145–163, March 2015.

[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process*, 16(8):2080–2095, Aug 2007.

[8] W. Dong, L. Zhang, G. Shi, and X. Li. Nonlocally Centralized Sparse Representation for Image Restoration. *IEEE Trans. Image Process.*, 22(4):1620–1630, April 2013.

[9] M. Elad. Sparse and redundant representations - from theory to applications in signal and image processing. *Springer*, 2010.

[10] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process*, 15(12):3736–3745, 2006.

[11] Y. Fang, J. Wu, and B. Huang. 2D sparse signal recovery via 2d orthogonal matching pursuit. *Sci. China. Inf. Sci.*, 55(4):889–897, 2012.

[12] A. Ghaffari, M. Babaie-Zadeh, and C. Jutten. Sparse Decomposition of Two Dimensional Signals. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pages 3157–3160, 2009.

[13] T. Guha and R. Ward. Learning sparse representations for human action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(8):1576–1588, 2012.

[14] S. Hawe, M. Seibert, and M. Kleinsteuber. Separable dictionary learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 438–445, 2013.

[15] R. Jenatton, J. Mairal, F. R. Bach, and G. R. Obozinski. Proximal methods for sparse hierarchical dictionary learning. In *Proc. 27th Annu, Int. Conf. Mach. Learn*, pages 487–494, 2010.

[16] Z. Jiang, Z. Lin, and L. Davis. Learning a discriminative dictionary for sparse coding via label consistent k-svd. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1697–1704, June 2011.

[17] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, 2009.

[18] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Proc. Adv. Neural Inf. Process. Syst.*, pages 801–808. 2007.

[19] J. Liu, P. Musialski, P. Wonka, and J. Ye. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(1):208–220, Jan 2013.

[20] X. Liu, S. Bourennane, and C. Fossati. Denoising of hyperspectral images using the parafac model and statistical performance analysis. *IEEE Trans. Geosci. and Remote Sensing*, 50(10):3717–3724, Oct 2012.

[21] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi. Nonlocal transform-domain filter for volumetric data denoising and reconstruction. *IEEE Trans. Image Process*, 22(1):119–133, Jan 2013.

[22] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2272–2279, Sept 2009.

[23] Y. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proc. 27th Asilomar Conf. Signals, Syst. and Comput.*, pages 40–44 vol.1, 1993.

[24] Y. Peng, D. Meng, Z. Xu, C. Gao, Y. Yang, and B. Zhang. Decomposable nonlocal tensor dictionary learning for multispectral image denoising. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2949–2956, June 2014.

[25] N. Qi, Y. Shi, X. Sun, W. Ding, and B. Yin. Single image super-resolution via 2d sparse representation. In *Proc. IEEE Int. Conf. Multimedia Expo.*, pages 1–6, June 2015.

[26] N. Qi, Y. Shi, X. Sun, J. Wang, and B. Yin. Two dimensional synthesis sparse model. In *Proc. IEEE Int. Conf. Multimedia Expo.*, pages 1–6, 2013.

[27] M. H. Quynh Nguyen, Antoine Gautier. A flexible tensor block coordinate ascent scheme for hypergraph matching. *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[28] A. Rajwade, A. Rangarajan, and A. Banerjee. Image denoising using the higher order singular value decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(4):849–862, April 2013.

[29] V. M. N. P. Ravishankar Sivalingam, Daniel Boley. Tensor sparse coding for positive definite matrices. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014.

[30] N. Renard, S. Bourennane, and J. Blanc-Talon. Denoising and dimensionality reduction using multilinear tools for hyperspectral images. *IEE Geosci. Remote Sensing Letters*, 5(2):138–142, April 2008.

[31] A. S. Tamir Hazan, Simon Polak. Sparse image coding using a 3d non-negative tensor factorization. *Proc. IEEE. Int. Conf. Comput. Vis.*, 1:50–57, 2005.

[32] R. Tibshirani. Regression Shrinkage and Selection Via the Lasso. *J. Roy. Statist. Soc.: Series B*, pages 267–288, 1996.

[33] S. Wang, D. Zhang, Y. Liang, and Q. Pan. Semi-Coupled Dictionary Learning with Applications to Image Super-Resolution and Photo-Sketch Synthesis. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2216–2223, June 2012.

[34] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(2):210–227, 2009.

[35] Y. Xu and W. Yin. A fast patch-dictionary method for whole image recovery. *arXiv preprint arXiv:1408.3740*, 2014.

[36] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang. Coupled dictionary training for image super-resolution. *IEEE Trans. Image Process.*, 21(8):3467–3478, Aug 2012.

[37] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Trans. Image Process*, 19(11):2861–2873, 2010.

[38] F. Yasuma, T. Mitsunaga, D. Iso, and S. Nayar. Generalized Assorted Pixel Camera: Postcapture Control of Resolution, Dynamic Range, and Spectrum. *IEEE Trans. Image Process*, 19(9):2241–2253, Sept 2010.

[39] D. Zhang, M. Yang, and X. Feng. Sparse representation or collaborative representation:which helps face recognition? In *Proc. IEEE. Int. Conf. Comput. Vis.*, pages 471–478, 2011.

[40] S. Zubair and W. Wang. Tensor dictionary learning with sparse tucker decomposition. In *Proc. Int. Conf. Digital Signal Process.*, pages 1–6, 2013.