

# Framelet Representation of Tensor Nuclear Norm for Third-Order Tensor Completion

Tai-Xiang Jiang, Michael K. Ng, Xi-Le Zhao, and Ting-Zhu Huang

**Abstract**—The main aim of this paper is to develop a framelet representation of the tensor nuclear norm for third-order tensor recovery. In the literature, the tensor nuclear norm can be computed by using tensor singular value decomposition based on the discrete Fourier transform matrix, and tensor completion can be performed by the minimization of the tensor nuclear norm which is the relaxation of the sum of matrix ranks from all Fourier transformed matrix frontal slices. These Fourier transformed matrix frontal slices are obtained by applying the discrete Fourier transform on the tubes of the original tensor. In this paper, we propose to employ the framelet representation of each tube so that a framelet transformed tensor can be constructed. Because of framelet basis redundancy, the representation of each tube is sparsely represented. When the matrix slices of the original tensor are highly correlated, we expect the corresponding sum of matrix ranks from all framelet transformed matrix frontal slices would be small, and the resulting tensor completion can be performed much better. The proposed minimization model is convex and global minimizers can be obtained. Numerical results on several types of multi-dimensional data (videos, multispectral images, and magnetic resonance imaging data) have tested and shown that the proposed method outperformed the other testing methods.

**Index Terms**—Tensor nuclear norm, Framelet, Alternating direction method of multipliers (ADMM), Tensor completion, Tensor robust principal component analysis

## I. INTRODUCTION

As a high order extension of matrix, the tensor is an important data format for multi-dimensional data applications, such as color image and video processing [1], [2], [3], hyperspectral data recovery and fusion [4], [5], [6], personalized web search [7], [8], high-order web link analysis [9], magnetic resonance imaging (MRI) data recovery [10], and seismic data reconstruction [11]. Owing to the objective restrictions, for example, the imaging condition for the visual data acquiring and the limitation of the transmission bandwidth, the multi-dimensional data in many applications are incomplete or grossly corrupted. This motivates us to perform tensor completion [3] or tensor robust principal component analysis

This work is supported by the National Natural Science Foundation of China (61772003, 61876203, and 61702083), HKRGC GRF (12306616, 12200317, 12300218, and 12300519), HKU Grant (104005583), and China Postdoctoral Science Foundation (2017M610628 and 2018T111031).

T.-X. Jiang is with FinTech Innovation Center, Financial Intelligence and Financial Engineering Research Key Laboratory of Sichuan province, School of Economic Information Engineering, Southwestern University of Finance and Economics, Chengdu, Sichuan, China (e-mail: taixiangjiang@gmail.com).

Michael K. Ng is with Department of Mathematics, The University of Hong Kong, Pokfulam, Hong Kong (e-mail: mng@maths.hku.hk).

X.-L. Zhao, and Ting-Zhu Huang are with the Research Center for Image and Vision Computing, School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu 611731, P.R.China (e-mail: xlzhao122003@163.com; tingzhuang@126.com).

(RPCA) [12], in which how to characterize and utilize the internal structural information of these multidimensional data is of crucial importance.

For the matrix processing, low-rank models can effectively and efficiently handle two-dimensional data of various sources [13], [14]. Generalized from matrix format, a tensor is able to contain more essentially structural information, being a powerful tool for dealing with multi-modal and multi-relational data [15]. Unfortunately, it is not easy to directly extend the low-rankness from the matrix to tensors. More precisely, there is not an exact (or unique) definition for the tensor's rank. In the past decades, the most popular rank definitions are the CANDECOMP/PARAFAC (CP)-rank [16], [17] and the Tucker-rank [18], [19] (or denoted as “ $n$ -rank” in [20]). The CP-rank is based on the CP decomposition, however, computing the CP-rank of a given tensor is NP-hard [21]. The Tucker-rank is based on the Tucker decomposition, in which the tensor is unfolded along each mode unavoidably destroying the intrinsic structures of the tensor.

In this paper, we investigate the newly emerged tensor rank definitions, i.e., the tensor multi-rank and the tensor tubal-rank, which are computable and induced from the tensor singular value decomposition (t-SVD). The t-SVD is initially proposed by Braman *et al.* [22] and Kilmer *et al.* [23], based on the tensor-tensor product (denoted as t-prod), in which the third-order tensors are operated integrally avoiding the loss of information inherent in matricization or flattening of the tensor [24]. Meanwhile, the t-SVD has shown its superior performance in capturing the spatial-shifting correlation that is ubiquitous in real-world data [25], [22], [23]. Although the t-SVD is initially designed for third-order tensors, it has been extended to high order tensors with arbitrary dimensions [25], [26].

In [27], Kernfeld *et al.* note that the t-prod is based on a convolution-like operation, which can be implemented using the discrete Fourier transform (DFT). Then, given a third-order tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , its Fourier transformed (along the third mode) tensor is denoted as  $\hat{\mathcal{X}} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  and its tensor multi-rank is a vector with the  $i$ -th element equal to the rank of  $i$ -th frontal slice of  $\hat{\mathcal{X}}$  [28]. The tensor nuclear norm (TNN) of  $\mathcal{X}$  is subsequently defined and it equals to the sum of the nuclear norm of  $\hat{\mathcal{X}}$ 's frontal slices and is the relaxation of the sum of matrix ranks from all  $\hat{\mathcal{X}}$ 's slices. By minimizing the TNN, Zhang *et al.* [28] build the low-rank tensor completion model and provided theoretical performance bounds for third-order tensor recovery from limited sampling.

Lu *et al.* [29] utilize the TNN<sup>1</sup> for the tensor RPCA. Similar researches, which adopt the TNN for multi-dimensional data recovery, can be found in [30], [31], [32].

Other than the Fourier transform, Kernfeld *et al.* find that the t-prod, together with the tensor decomposition scheme, can be defined via any invertible transform, for instance, the discrete cosine transform (DCT). Namely, the t-prod can be implemented by the matrices' product after the invertible transformation along the third mode. Xu *et al.* [33] validate that, when minimizing the DCT based TNN for the tensor completion problem, the DCT is superior to the DFT in terms of the preservation of the head and the tail frontal slices, because of its mirror boundary condition. Corroborative results can be found in [34], [35], which demonstrates that any invertible linear transform can be applied to induce the TNN for the tensor completion task. Coincidentally, Song *et al.* [36] find that the corresponding transformed tubal-rank could be approximately smaller with an appropriate unitary transform, for instance, the Haar wavelet transform, and they prove that one can recover a low transformed tubal-rank tensor exactly with overwhelming probability provided that its transformed tubal rank is sufficiently small and its corrupted entries are reasonably sparse.

The tensor data recovery within the t-SVD framework can be viewed as finding a low-rank approximation in the transformed domain. Therefore, if the transformed tensor could be approximately lower-rank, minimizing the corresponding TNN, namely the TNN defined based on the transformation, would be more effective for the recovery [36]. In [36], [34], [35], the authors establish elegant theoretical results based on the unitary transform or the invertible linear transform. However, the requirement of the invertibility prevents their results from other non-invertible (or semi-invertible) transformations, which could bring in redundancy. We note that redundancy in the transformation is important as such transformed coefficients can contain information of missing data in the original domain, see for example the work by Cai *et al.* [37].

In this paper, we suggest to use the tight wavelet frame (framelet) as the transformation within the t-SVD framework. Because of framelet basis redundancy, the representation of each tube is sparsely represented. We expect when each matrix slices of the original tensor, the corresponding sum of matrix ranks from all framelet transformed matrix slices would be small. As an example, we illustrate this motivation by using magnetic resonance image (MRI) of size  $142 \times 178 \times 121$ , multispectral image (MSI) of size  $512 \times 512 \times 31$  and video data of size  $144 \times 176 \times 100$  to demonstrate their rank reduction via framelet transformation<sup>2</sup> to the Fourier transformation. Note that for real imaging data, each transformed matrix frontal slice is not an exact low-rank matrix, but it is close to a low-rank matrix. There are many small singular values of each transformed matrix frontal slice. We show in Table I that the mean value of the matrix ranks of  $\mathcal{X}(:, :, i)$  (the  $i$ -th transformed matrix frontal slice). Here we discard the singular values of transformed matrix frontal slice when they

TABLE I  
THE MEAN VALUE OF ALL THE TRUNCATED TRANSFORMED MATRIX SLICES RANKS BY USING THE FFT AND THE FRAMELET TRANSFORM FOR MRI, MSI AND VIDEO DATA SETS.

Data	Parameter $\epsilon$	FFT Multi-rank (mean value)	Framelet Multi-rank (mean value)	Reduction
MRI	0.02	101.0	77.8	23.3
	0.01	120.1	94.1	25.9
	0.005	131.9	108.9	23.0
Video	0.02	106.7	74.5	32.2
	0.01	122.7	92.2	30.5
	0.005	132.6	108.5	24.1
MSI	0.02	83.8	46.1	37.7
	0.01	132.8	77.8	55.0
	0.005	218.0	136.0	82.0

are smaller than the truncation parameter, and the truncated rank of transformed matrix slice is obtained. It is clear that the mean value of such truncated matrix ranks by using framelet transformation is lower than that by using the Fourier transformation. When a framelet transformed tensor is close to a low-rank tensor compared with the use of the Fourier transform, it is expected that the resulting tensor completion can be performed much better in practice. The framelet based TNN (F-TNN) minimization models are subsequently formulated for the low-rank tensor completion (LRTC) and tensor RPCA. The proposed minimization models are convex and global minimizers can be obtained via the alternating direction multipliers method (ADMM) [38] with a theoretical convergence guarantee. We conduct numerical experiments on various types of multi-dimensional imaging data and the results verify that our framelet based method outperforms the compared methods.

#### A. Contributions

The main contributions can be summarised as follows. **(i)** We suggest the framelet transform within the t-SVD framework and proposed a tensor completion model, which minimizes the framelet representation of the tensor nuclear norm. **(ii)** To tackle the non-invertible framelet transform based models, we develop alternating direction multipliers method (ADMM) based algorithms with guaranteed convergence, and we test our method on various types of multi-dimensional data. The outperformance of our method further corroborates the usage of framelet.

The outline of this paper is given as follows. In Section II, some preliminary background on tensors and the framelet is given. The main results, including the proposed model and algorithm, are presented in Section III. Experimental results are reported in Section IV. Finally, Section V draws some conclusions.

## II. PRELIMINARIES

This section provides the basic ingredients to induce the proposed method. We firstly give the basic tensor notations and then introduce the t-SVD framework, which has been proposed in [24], [23], [28], [29]. We restate them here at the readers' convenience. Next, the basics of framelet are briefly presented.

<sup>1</sup>In [29], the TNN is defined with a factor  $1/n_3$ .

<sup>2</sup>The piece-wise cubic B-spline is used to generate framelet system.



### A. Tensor Notations And Definitions

Generally, a third-order tensor is denoted as  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , and  $x_{i,j,k}$  is its  $(i, j, k)$ -th component. We use  $\mathcal{X}^{(k)}$  or  $\mathcal{X}(:, :, k)$  to denote the  $k$ -th frontal slice of a third-order tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ .

**Definition 2.1 (tensor mode-3 unfolding and folding [39]):** The mode-3 unfolding of a tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is denoted as a matrix  $\mathbf{X}_{(3)} \in \mathbb{R}^{n_3 \times n_1 n_2}$ , where the tensor's  $(i, j, k)$ -th element maps to the matrix's  $(k, l)$ -th element satisfying  $l = (j-1)n_1 + i$ . The mode-3 unfolding operator and its inverse are respectively denoted as  $\text{unfold}_3$  and  $\text{fold}_3$ , and they satisfy  $\mathcal{X} = \text{fold}_3(\text{unfold}_3(\mathcal{X})) = \text{fold}_3(\mathbf{X}_{(3)})$ .

**Definition 2.2 (mode-3 tensor-matrix product [39]):** The mode-3 tensor-matrix product of a tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  with a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n_3}$  is denoted by  $\mathcal{X} \times_3 \mathbf{A}$  and is of size  $n_1 \times n_2 \times m$ . Elementwise, we have

$$(\mathcal{X} \times_3 \mathbf{A})_{i,j,k} = \sum_{n=1}^{n_3} x_{i,j,n} \cdot a_{k,n}. \quad (1)$$

The mode-3 tensor-matrix product can also be expressed in terms of the mode-3 unfolding

$$\mathcal{Y} = (\mathcal{X} \times_3 \mathbf{A}) \Leftrightarrow \mathbf{Y}_{(3)} = \mathbf{A} \cdot \text{unfold}_3(\mathcal{X}).$$

The one-dimensional DFT on a vector  $\mathbf{x} \in \mathbb{R}^n$ , denoted as  $\bar{\mathbf{x}}$ , is given by  $\bar{\mathbf{x}} = \mathbf{F}_n \mathbf{x} \in \mathbb{C}^n$ , where  $\mathbf{F}_n \in \mathbb{C}^{n \times n}$  is the DFT matrix. In this paper, we use  $\hat{\mathcal{X}}$  to denote the transformed tensor by performing one-dimensional DFT along the mode-3 fibers (tubes) of  $\mathcal{X}$ . By using the DFT matrix  $\mathbf{F}_{n_3} \in \mathbb{C}^{n_3 \times n_3}$ , we have

$$\hat{\mathcal{X}} = \mathcal{X} \times_3 \mathbf{F}_{n_3} = \text{fold}_3(\mathbf{F}_{n_3} \text{unfold}_3(\mathcal{X})) \in \mathbb{C}^{n_1 \times n_2 \times n_3}.$$

**Definition 2.3 (tensor conjugate transpose [24]):** The conjugate transpose of a tensor  $\mathcal{A} \in \mathbb{C}^{n_2 \times n_1 \times n_3}$  is tensor  $\mathcal{A}^H \in \mathbb{C}^{n_1 \times n_2 \times n_3}$  obtained by conjugate transposing each of the frontal slice and then reversing the order of transposed frontal slices 2 through  $n_3$ , i.e.,  $(\mathcal{A}^H)^{(1)} = (\mathcal{A}^{(1)})^H$  and  $(\mathcal{A}^H)^{(i)} = (\mathcal{A}^{(n_3+2-i)})^H$  ( $i = 2, \dots, n_3$ ).

**Definition 2.4 (t-prod [24]):** The tensor-tensor-product (t-prod)  $\mathcal{C} = \mathcal{A} * \mathcal{B}$  of  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  and  $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$  is a tensor of size  $n_1 \times n_4 \times n_3$ , where the  $(i, j)$ -th tube  $\mathbf{c}_{ij}$  is given by

$$\mathbf{c}_{ij} = \mathcal{C}(i, j, :) = \sum_{k=1}^{n_2} \mathcal{A}(i, k, :) * \mathcal{B}(k, j, :) \quad (2)$$

where  $*$  denotes the circular convolution between two tubes of same size.

**Definition 2.5 (identity tensor [24]):** The identity tensor  $\mathcal{I} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$  is the tensor whose first frontal slice is the  $n_1 \times n_1$  identity matrix, and whose other frontal slices are all zeros.

**Definition 2.6 (orthogonal tensor [24]):** A tensor  $\mathcal{Q} \in \mathbb{C}^{n_1 \times n_1 \times n_3}$  is orthogonal if it satisfies

$$\mathcal{Q}^H * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^H = \mathcal{I}. \quad (3)$$

**Definition 2.7 (f-diagonal tensor [24]):** A tensor  $\mathcal{A}$  is called f-diagonal if each frontal slice  $\mathcal{A}^{(i)}$  is a diagonal matrix.

**Theorem 2.1 (t-SVD [24], [23]):** For  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , the t-SVD of  $\mathcal{A}$  is given by

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^H \quad (4)$$

where  $\mathcal{U} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$  and  $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$  are orthogonal tensors, and  $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is an f-diagonal tensor. The t-SVD is illustrated in Figure 1.

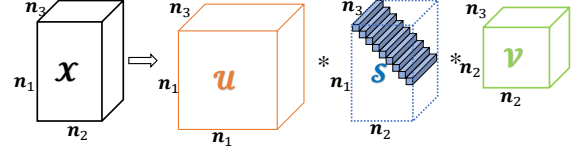


Fig. 1. The t-SVD of an  $n_1 \times n_2 \times n_3$  tensor.

**Definition 2.8 (tensor tubal-rank and multi-rank [28]):** The tubal-rank of a tensor  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , denoted as  $\text{rank}_t(\mathcal{A})$ , is defined to be the number of non-zero singular tubes of  $\mathcal{S}$ , where  $\mathcal{S}$  comes from the t-SVD of  $\mathcal{A}$ :  $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$ . That is

$$\text{rank}_t(\mathcal{A}) = \#\{i : \mathcal{S}(i, :, :) \neq 0\}. \quad (5)$$

The tensor multi-rank of  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is a vector, denoted as  $\text{rank}_r(\mathcal{A}) \in \mathbb{R}^{n_3}$ , with the  $i$ -th element equals to the rank of  $i$ -th frontal slice of  $\hat{\mathcal{A}}$ .

**Definition 2.9 (block diagonal form [28]):** Let  $\bar{\mathcal{A}}$  denote the block-diagonal matrix of the tensor  $\hat{\mathcal{A}}$  in the Fourier domain, i.e.,

$$\bar{\mathcal{A}} \triangleq \text{blockdiag}(\hat{\mathcal{A}}) \triangleq \begin{bmatrix} \hat{\mathcal{A}}^{(1)} & & & \\ & \hat{\mathcal{A}}^{(2)} & & \\ & & \ddots & \\ & & & \hat{\mathcal{A}}^{(n_3)} \end{bmatrix} \in \mathbb{C}^{n_1 n_3 \times n_2 n_3}, \quad (6)$$

where  $\hat{\mathcal{A}}^{(k)} = \hat{\mathcal{A}}(:, :, k)$  is the  $k$ -th slice of  $\hat{\mathcal{A}}$  for  $k = 1, 2, \dots, n_3$ .

It is not difficult to find that  $\bar{\mathcal{A}}^H = \bar{\mathcal{A}}^H$ , i.e., the block diagonal form of a tensor's conjugate transpose equals to the matrix conjugate transpose of the tensor's block diagonal form. Further more, for any tensor  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  and  $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ , we have

$$\mathcal{A} * \mathcal{B} = \mathcal{C} \Leftrightarrow \bar{\mathcal{A}} \cdot \bar{\mathcal{B}} = \bar{\mathcal{C}},$$

where  $\cdot$  is the matrix product.

**Definition 2.10 (tensor-nuclear-norm (TNN) [28]):** The tensor nuclear norm of a tensor  $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , denoted as  $\|\mathcal{A}\|_{\text{TNN}}$ , is defined as

$$\|\mathcal{A}\|_{\text{TNN}} \triangleq \|\bar{\mathcal{A}}\|_*, \quad (7)$$

where  $\|\cdot\|_*$  refers to the matrix nuclear norm. For a matrix  $\mathbf{X} \in \mathbb{C}^{m \times n}$ ,  $\|\mathbf{X}\|_* = \sum_i^{\min\{m,n\}} \sigma_i$ , where  $\sigma_i$  is the  $i$ -th singular value of  $\mathbf{X}$ . The TNN can be computed via the summation of the matrix nuclear norm of Fourier transformed tensor's slices, which are also the blocks of  $\bar{\mathcal{A}}$ . That is  $\|\mathcal{A}\|_{\text{TNN}} = \sum_{i=1}^{n_3} \|\hat{\mathcal{A}}^{(i)}\|_*$ .

We summary the frequent used notations in Table II-A.

TABLE II  
TENSOR NOTATIONS

Notation	Explanation
$\mathcal{X}, \mathbf{X}, \mathbf{x}, x$	Tensor, matrix, vector, scalar.
*	The tensor-tensor product or the circular convolution between vectors.
$\mathcal{X}(:, :, k)$ (or $\mathcal{X}^{(k)}$ )	The $k$ -th frontal slice of a third-order tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ .
$\text{fold}_3$ ( $\text{unfold}_3$ )	The fold (or unfold) operation along the third mode.
$\mathbf{X}_{(3)}$	The mode-3 unfolding of a tensor $\mathcal{X}$ .
$\hat{\mathcal{X}}$	The Fourier transformed (along the third mode) tensor.
$\text{rank}_r(\mathcal{A})$	The multi-rank of a tensor $\mathcal{A}$ and its $i$ -th element equals to $\text{rank}(\mathcal{A}^{(i)})$ .
$\ \mathcal{X}\ _{\text{TTN}}$	The tensor nuclear norm of a tensor $\mathcal{X}$ and it equals to the sum of the nuclear norms of $\mathcal{X}$ 's slices.

### B. Framelet

A tight frame is defined as a countable set  $X \subset L_2(\mathbb{R})$  with the property that  $\forall f \in L_2(\mathbb{R}), f = \sum_{g \in X} \langle f, g \rangle g$ . This is equivalent to that  $\forall f \in L_2(\mathbb{R})$ , we have

$$\|f\|_{L_2(\mathbb{R})}^2 = \sum_{g \in X} |\langle f, g \rangle|^2,$$

where  $\langle \cdot, \cdot \rangle$  is the inner product in  $L^2(\mathbb{R})$ , and  $\|\cdot\|_{L^2(\mathbb{R})} = \langle \cdot, \cdot \rangle^{\frac{1}{2}}$ .

For given  $\Psi := \{\psi_1, \psi_2, \dots, \psi_r\} \subset L^2(\mathbb{R})$ , the affine (or wavelet) system is defined by the collection of the dilations and the shifts of  $\Psi$  as  $X(\Psi) := \{\psi_{l,j,k} : 1 \leq l \leq r; j, k \in \mathbb{Z}\}$ , where  $\psi_{l,j,k} := 2^{j/2} \psi_l(2^j \cdot -k)$ . When  $X(\Psi)$  forms a tight frame of  $L^2(\mathbb{R})$ , it is called a tight wavelet frame, and  $\psi_l, l = 1, 2, \dots, r$  are called the (tight) framelets. In the numerical scheme of image processing, the framelet transform (decomposition operator) of a vector  $\mathbf{v} \in \mathbb{R}^n$  can be represented by a matrix  $\mathbf{W} \in \mathbb{R}^{wn \times n}$  is the framelet transform matrix constructed with  $n$  filters and  $l$  levels and  $w = (n-1)l + 1$ . The processes of generating such matrices have been detailed in many literatures such as [37], [40]. We omit them here for readability. Then the framelet transform of a discrete signal  $\mathbf{v} \in \mathbb{R}^n$ , can be written as  $\mathbf{u} = \mathbf{W}\mathbf{v} \in \mathbb{R}^{wn}$ . Besides, the unitary extension principle (UEP) [41] asserts that  $\mathbf{W}^T \mathbf{W}\mathbf{v} = \mathbf{v}$ , where  $\mathbf{W}^T$  indicates the inverse framelet transform. However,  $\mathbf{W}\mathbf{W}^T \mathbf{u} \neq \mathbf{u}$ .

### III. MAIN RESULTS

In this section, we replace the Fourier transform by the framelet transform. The starting point of our idea is that the framelet transform would bring in redundancy and the transformed data is of lower multi-rank. Then, we build the LRTC model and tensor RPCA model based on the framelet representation of the tensor nuclear norm and propose the ADMM based algorithms to optimize these models.

#### A. From DFT to The Framelet Transform

For a three way tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , owing to the circular convolution in Def. 2.4, its t-SVD can be efficiently computed via the DFT. Computing the one-dimensional DFT

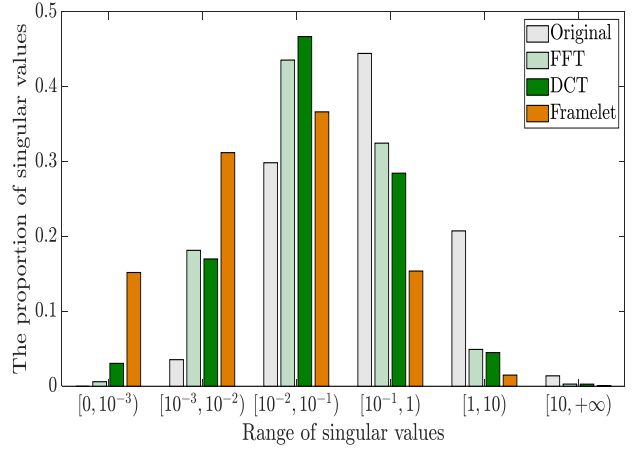


Fig. 2. The distribution of singular values. Here, the singular values are obtained by conducting SVD on each frontal slice of the original tensor data or the transformed tensors.

of a vector of length  $n$  by using the DFT matrix costs  $O(n^2)$ , and the computational cost can be reduced to  $O(n \log n)$  by employing the fast Fourier transform (FFT) technique [42]. Using the DFT matrix, for a tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , we can obtain its Fourier transformed tensor as

$$\hat{\mathcal{X}} = \text{fold}_3(\mathbf{F}_{n_3} \mathbf{X}_{(3)}) \in \mathbb{C}^{n_1 \times n_2 \times n_3},$$

where  $\mathbf{X}_{(3)}$  is the mode-3 unfolding of  $\mathcal{X}$

Next, we will adopt the framelet transform as a substitute for the Fourier transform, and give the definition of the framelet representation of the tensor nuclear norm. For simplicity, we denote the tensor after framelet transform along the third mode as

$$\mathcal{X}_{\mathbf{W}} = \text{fold}_3(\mathbf{W}\mathbf{X}_{(3)}) \in \mathbb{R}^{n_1 \times n_2 \times wn_3},$$

where  $\mathbf{W} \in \mathbb{R}^{wn_3 \times n_3}$  is the framelet transform matrix constructed with  $n$  filters and  $l$  levels and  $w = (n-1)l + 1$ . Considering the UEP property of the framelet transform, we have  $\mathcal{X} = \text{fold}_3(\mathbf{W}^T[\mathbf{X}_{\mathbf{W}}]_{(3)})$ , where  $[\mathbf{X}_{\mathbf{W}}]_{(3)} = \text{unfold}_3(\mathcal{X}_{\mathbf{W}})$ .

Recalling Def. 2.8, the tensor multi-rank is defined as a vector of the ranks of the frontal slices in the Fourier transform domain. Therefore, the framelet based multi-rank is defined in the same manner as follows.

**Definition 3.1 (Framelet based multi-rank):** The framelet based multi-rank of a tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is defined as a vector  $\mathbf{r}_w \in \mathbb{R}^{wn_3}$  with the  $i$ -th elements  $\mathbf{r}_w(i) = \text{rank}(\mathcal{X}_{\mathbf{W}}(:, :, i))$  for  $i = 1, 2, \dots, wn_3$ .

Here we have replaced the Fourier transform by the framelet and defined the framelet based multi-rank. As mentioned before, the framelet transformed tensor can be of lower (framelet based) multi-rank. To understand this in-depth, we give some empirically numerical analyses on the singular values of the frontal slices of the transformed tensors. Here, taking the video data news<sup>3</sup> as an example, the original video data is denoted as  $\mathcal{X} \in \mathbb{R}^{144 \times 176 \times 100}$  and its Fourier, DCT, and framelet transformed tensors are denoted as  $\hat{\mathcal{X}}$ ,  $\mathcal{X}_{\text{DCT}}$ <sup>4</sup>, and

<sup>3</sup>Data available at <http://trace.eas.asu.edu/yuv/>.

<sup>4</sup> $\mathcal{X}_{\text{DCT}}$  is obtained by replacing the DFT with DCT, being similar to  $\mathcal{X}_{\mathbf{W}}$ .

$\mathcal{X}_{\mathbf{W}}$ , respectively. In Figure 2, we exhibit the distributions of the singular values of the frontal slices of  $\mathcal{X}$ , the Fourier transformed tensors  $\hat{\mathcal{X}}$ , the DCT transformed tensor  $\mathcal{X}_{\text{DCT}}$ , and the framelet transformed tensors  $\mathcal{X}_{\mathbf{W}}$ <sup>5</sup>. In Figure 2, we show the proportion of the number of singular values of transformed matrix frontal slices in each magnitude interval. It can be found in the figure that a large proportion of the singular values of the framelet transformed data appears in the interval of  $[0, 10^{-2}]$  compared with the original video data, the Fourier transformed tensor  $\hat{\mathcal{X}}$ , and the DCT transformed tensor  $\mathcal{X}_{\text{DCT}}$ . This phenomenon brings in an advantage that the data can be better approximated with lower rank via the framelet representation. In Section IV, we will illustrate tensor completion and tensor RPCA can be obtained by using the framelet representation.

### B. Framelet Based TNN

Using the DFT matrix  $\mathbf{F}_{n_3}$ , the tensor nuclear norm in (7) of a tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  can be expressed as

$$\begin{aligned} \|\mathcal{X}\|_{\text{TNN}} &= \|\bar{\mathcal{X}}\|_* = \sum_{i=1}^{n_3} \|\hat{\mathcal{X}}^{(i)}\|_* \\ &= \sum_{i=1}^{n_3} \|\text{fold}_3(\mathbf{F}_{n_3} \mathbf{X}_{(3)})(:, :, i)\|_*, \end{aligned} \quad (8)$$

where  $\mathbf{X}_{(3)}$  is the mode-3 unfolding of  $\mathcal{X}$ .

**Definition 3.2 (Framelet based TNN (F-TNN)):** Similarly, the framelet representation of the tensor nuclear norm can be formulated as

$$\begin{aligned} \|\mathcal{X}\|_{\text{F-TNN}} &= \|\text{blockdiag}(\mathcal{X}_{\mathbf{W}})\|_* = \sum_{k=1}^{wn_3} \|\mathcal{X}_{\mathbf{W}}(:, :, k)\|_* \\ &= \sum_{k=1}^{wn_3} \|\text{fold}_3(\mathbf{W} \mathbf{X}_{(3)})(:, :, k)\|_*, \end{aligned} \quad (9)$$

where  $\mathbf{W} \in \mathbb{R}^{wn_3 \times n_3}$  is the framelet transform matrix.

It is not difficult to obtain that the F-TNN is a convex envelope of the  $\ell_1$  norm of the framelet based multi-rank.

### C. Tensor Completion via Minimizing F-TNN

Based on the proposed framelet based TNN, our tensor completion model, which is convex, is formulated as

$$\begin{aligned} \min_{\mathcal{X}} \quad & \|\mathcal{X}\|_{\text{F-TNN}} \\ \text{s.t.} \quad & \mathcal{X}_{\Omega} = \mathcal{O}, \end{aligned} \quad (10)$$

where  $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is the incomplete observed data, and  $\Omega$  is the set of indexes of the observed entries.  $\mathcal{X}_{\Omega} = \mathcal{O}$  constrains that the entries of  $\mathcal{X}$  should agree with  $\mathcal{O}$  in  $\Omega$ .

The next part gives the solving algorithm for our tensor completion model (10). Let

$$\mathcal{I}_{\Phi}(\mathcal{X}) = \begin{cases} 0, & \mathcal{X} \in \Phi, \\ \infty, & \text{otherwise,} \end{cases} \quad (11)$$

where  $\Phi := \{\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \mathcal{X}_{\Omega} = \mathcal{O}\}$ .

<sup>5</sup>The piece-wise cubic B-spline is used to generate framelet system.

Thus, the problem (10) can be rewritten as

$$\min_{\mathcal{X}} \quad \mathcal{I}_{\Phi}(\mathcal{X}) + \sum_{k=1}^{wn_3} \|\mathcal{X}_{\mathbf{W}}(:, :, k)\|_* \quad (12)$$

Then, the minimization problem (12) can be efficiently solved via ADMM [38].

After introducing the auxiliary variable  $\mathcal{V} \in \mathbb{R}^{n_1 \times n_2 \times wn_3}$ , the problem (12) can be rewritten as the following unconstrained problem

$$\begin{aligned} \min_{\mathcal{X}} \quad & \mathcal{I}_{\Phi}(\mathcal{X}) + \sum_{k=1}^{wn_3} \|\mathcal{V}(:, :, k)\|_* \\ \text{s.t.} \quad & \mathcal{V} = \mathcal{X}_{\mathbf{W}}. \end{aligned} \quad (13)$$

The augmented Lagrangian function of (13) is given by

$$\begin{aligned} L_{\beta}(\mathcal{X}, \mathcal{V}, \Lambda) &= \mathcal{I}_{\Phi}(\mathcal{X}) + \sum_{k=1}^{wn_3} \|\mathcal{V}(:, :, k)\|_* \\ &\quad + \frac{\beta}{2} \|\mathcal{X}_{\mathbf{W}} - \mathcal{V}\|_F^2 \end{aligned} \quad (14)$$

where  $\Lambda \in \mathbb{R}^{n_1 \times n_2 \times wn_3}$  is the Lagrangian multiplier,  $\beta$  is the penalty parameter for the violation of the linear constraints. In the scheme of the ADMM, we update each variable alternately.

**$\mathcal{V}$  sub-problem:** The  $\mathcal{V}$  at  $t$ -th iteration is

$$\mathcal{V}^{t+1} = \arg \min_{\mathcal{V}} \sum_{k=1}^{wn_3} \|\mathcal{V}(:, :, k)\|_* + \frac{\beta}{2} \|\mathcal{X}_{\mathbf{W}}^t - \mathcal{V}\|_F^2 \quad (15)$$

Then, (15) can be decomposed into  $wn_3$  subproblems and it is easy to obtain the closed form solution of these subproblems with the singular value thresholding (SVT) operator [43]. Hence, we update  $\mathcal{V}$  as

$$\mathcal{V}^{t+1}(:, :, k) = \text{SVT}_{\frac{1}{\beta}} \left( \mathcal{X}_{\mathbf{W}}^t(:, :, k) + \frac{\Lambda^t(:, :, k)}{\beta} \right), \quad (16)$$

where  $k = 1, 2, \dots, wn_3$ . The complexity of computing  $\mathcal{V}$  at each iteration is  $O(wn_1 n_2 n_3 \min(n_1 n_2))$ .

**$\mathcal{X}$  sub-problem:** For convenience, the subproblem of optimizing  $L_{\beta}$  with respect to  $\mathcal{X}$  at  $t$ -th iteration is written in the matrix format as (recalling that  $\mathcal{X}_{\mathbf{W}} = \text{fold}_3(\mathbf{W} \mathbf{X}_{(3)})$ )

$$\mathbf{X}^{t+1} = \arg \min_{\mathbf{X}} \mathcal{I}_{\Phi}(\mathcal{X}) + \frac{\beta}{2} \|\mathbf{W} \mathbf{X} - \mathbf{V}_{(3)}^{t+1} + \frac{\Lambda_{(3)}^t}{\beta}\|_F^2, \quad (17)$$

where  $\mathbf{V}_{(3)}^{t+1} = \text{unfold}_3(\mathcal{V}^{t+1})$  and  $\Lambda_{(3)}^t = \text{unfold}_3(\Lambda^t)$ . To optimize (17), we first solve the following equation

$$\mathbf{W}^{\top} \mathbf{W} \mathbf{X}_{(3)} = \mathbf{W}^{\top} \left( \mathbf{V}_{(3)}^{t+1} - \frac{\Lambda_{(3)}^t}{\beta} \right). \quad (18)$$

Thus, considering that  $\mathbf{W}^{\top} \mathbf{W} \mathbf{X}_{(3)} = \mathbf{X}_{(3)}$  (the UEP property of the framelet transformation), we have

$$\mathcal{X}^{t+1} = \mathcal{P}_{\Omega^c} \left( \text{fold}_3(\mathbf{W}^{\top} (\mathbf{V}_{(3)}^{t+1} - \frac{\Lambda_{(3)}^t}{\beta})) \right) + \mathcal{P}_{\Omega}(\mathcal{O}), \quad (19)$$

where  $\mathcal{P}_{\Omega}(\cdot)$  is the projection function that keeps the entries of  $\cdot$  in  $\Omega$  while making others be zeros, and  $\Omega^c$  denotes the complementary set of  $\Omega$ . Meanwhile, we have

$\mathcal{X}_{\mathbf{W}}^{t+1} = \text{fold}_3(\mathbf{W}\mathbf{X}_{(3)}^{t+1})$ . The complexity of computing  $\mathcal{X}$  is  $O(w n_1 n_2 n_3^2)$  at each iteration.

**Updating the multiplier:** The multiplier  $\Lambda$  can be updated by

$$\Lambda^{t+1} = \Lambda^t + \beta (\mathcal{X}_{\mathbf{W}}^{t+1} - \mathcal{V}^{t+1}). \quad (20)$$

Updating  $\Lambda$  costs  $O(w n_1 n_2 n_3)$  at each iteration.

Finally, our algorithm is summarized in Algorithm 1. The total complexity of Algorithm 1 at each iteration is  $O(w n_1 n_2 n_3 (n_3 + \min(n_1, n_2)))$ . The objective function of the proposed model in (10) is convex. Our algorithm fits the standard ADMM framework and its convergence is theoretically guaranteed [38].

---

**Algorithm 1** Tensor completion via minimizing F-TNN

---

**Input:** The observed tensor  $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ ; Lagrange parameter  $\beta$ ; convergence criteria  $\epsilon$ ; maximum iteration  $t_{\max}$ .

**Initialization:** The framelet transform matrix  $\mathbf{W}$ ;  $\mathcal{V}^{(0)} = \text{fold}_3(\mathbf{W}\mathcal{O}_{(3)})$ ;  $\mathcal{X}^{(0)} = \mathcal{O}$ ;  $t = 0$ .

- 1: **while** not converged and  $t < t_{\max}$  **do**
- 2:   Update  $\mathcal{V}^{t+1}$  via Eq. (16);
- 3:   Update  $\mathcal{X}^{t+1}$  via Eq. (19);
- 4:   Update  $\Lambda^{t+1}$  via Eq. (20);
- 5:   Check the convergence conditions  $\|\mathcal{V}^{k+1} - \mathcal{V}^k\|_{\infty} \leq \epsilon$  and  $\|\mathcal{X}^{k+1} - \mathcal{X}^k\|_{\infty} \leq \epsilon$ ;
- 6:    $t = t + 1$ .
- 7: **end while**

**Output:** The reconstructed tensor  $\mathcal{X}$ .

---

#### D. Tensor Robust Principal Components Analysis

As aforementioned, another typical tensor recovery problem is the tensor RPCA problem, which aims to recover the tensor from grossly corrupted observations. Adopting the F-TNN to characterize the low-rank part, our tensor RPCA model is formulated as

$$\begin{aligned} \min_{\mathcal{L}, \mathcal{S}} \quad & \|\mathcal{L}\|_{\text{F-TNN}} + \lambda \|\mathcal{E}\|_1 \\ \text{s.t.} \quad & \mathcal{L} + \mathcal{E} = \mathcal{O}, \end{aligned} \quad (21)$$

where  $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is the observed data,  $\mathcal{E}$  indicates the sparse part,  $\|\mathcal{E}\|_1 = \sum_{i,j,k} |\mathcal{E}_{i,j,k}|$ , and  $\lambda$  is a non-negative parameter.

For convenience, we introduce an auxiliary variable  $\mathcal{V} \in \mathbb{R}^{n_1 \times n_2 \times w n_3}$ , and reformulate (22) as

$$\begin{aligned} \min_{\mathcal{L}, \mathcal{S}, \mathcal{V}} \quad & \sum_{k=1}^{w n_3} \|\mathcal{V}(:, :, k)\|_* + \lambda \|\mathcal{E}\|_1 \\ \text{s.t.} \quad & \mathcal{L} + \mathcal{E} = \mathcal{O}, \quad \mathcal{V} = \mathcal{L}_{\mathbf{W}}, \end{aligned} \quad (22)$$

where  $\mathcal{L}_{\mathbf{W}} = \text{fold}_3(\mathbf{W}\mathcal{L}_{(3)}) \in \mathbb{R}^{n_1 \times n_2 \times w n_3}$  and  $\mathbf{W} \in \mathbb{R}^{w n_3 \times n_3}$  is the framelet transform matrix constructed with  $n$  filters and  $l$  levels ( $w = (n - 1)l + 1$ ).

Similarly, we adopt ADMM to solve (22). The augmented Lagrangian function of (22) is given as

$$\begin{aligned} L_{\beta}(\mathcal{L}, \mathcal{V}, \mathcal{E}, \Lambda) = & \sum_{k=1}^{w n_3} \|\mathcal{V}(:, :, k)\|_* + \frac{\beta}{2} \|\mathcal{L}_{\mathbf{W}} - \mathcal{V} + \frac{\Lambda_1}{\beta}\|_F^2 \\ & + \lambda \|\mathcal{E}\|_1 + \frac{\beta}{2} \|\mathcal{O} - \mathcal{L} - \mathcal{E} + \frac{\Lambda_2}{\beta}\|_F^2 \end{aligned} \quad (23)$$

where  $\Lambda_1 \in \mathbb{R}^{n_1 \times n_2 \times w n_3}$  and  $\Lambda_2 \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  are the Lagrangian multiplier, and  $\beta$  is a nonnegative parameter. In the scheme of the ADMM, we update each variable alternately as:

$$\begin{cases} \mathcal{V}^{t+1} = \arg \min_{\mathcal{V}} \sum_{k=1}^{w n_3} \|\mathcal{V}(:, :, k)\|_* + \frac{\beta}{2} \|\mathcal{L}_{\mathbf{W}}^t - \mathcal{V} + \frac{\Lambda_1^t}{\beta}\|_F^2, \\ \mathcal{L}^{t+1} = \arg \min_{\mathcal{L}} \frac{\beta}{2} \|\mathcal{L}_{\mathbf{W}} - \mathcal{V}^{t+1} + \frac{\Lambda_1^t}{\beta}\|_F^2 + \frac{\beta}{2} \|\mathcal{O} - \mathcal{L} - \mathcal{E}^t + \frac{\Lambda_2^t}{\beta}\|_F^2, \\ \mathcal{E}^{t+1} = \arg \min_{\mathcal{E}} \lambda \|\mathcal{E}\|_1 + \frac{\beta}{2} \|\mathcal{O} - \mathcal{L}^{t+1} - \mathcal{E} + \frac{\Lambda_2^t}{\beta}\|_F^2, \\ \Lambda_1^{t+1} = \Lambda_1^t + \beta (\mathcal{L}_{\mathbf{W}}^{t+1} - \mathcal{V}^{t+1}), \\ \Lambda_2^{t+1} = \Lambda_2^t + \beta (\mathcal{O} - \mathcal{L}^{t+1} - \mathcal{E}^{t+1}). \end{cases} \quad (24)$$

Specifically, the  $\mathcal{V}$  subproblem in (24) can be solved by

$$\mathcal{V}^{t+1}(:, :, k) = \text{SVT}_{\frac{1}{\beta}} \left( \mathcal{L}_{\mathbf{W}}^t(:, :, k) + \frac{\Lambda_1^t(:, :, k)}{\beta} \right), \quad (25)$$

for  $k = 1, 2, \dots, w n_3$ . The complexity of updating  $\mathcal{V}$  is  $O(w n_1 n_2 n_3 \min(n_1, n_2))$  at each iteration. The  $\mathcal{L}$  subproblem is a least square problem and its solution can be obtained as

$$\mathcal{L}^{t+1} = \frac{1}{2} \text{fold}_3 \left( \mathbf{W}^{\top} (\mathbf{V}_{(3)}^{t+1} - \frac{\Lambda_1^t_{(3)}}{\beta}) \right) + \frac{1}{2} \left( \mathcal{O} - \mathcal{E}^t + \frac{\Lambda_2^t}{\beta} \right). \quad (26)$$

At each iteration, computing  $\mathcal{L}$  costs  $O(w n_1 n_2 n_3^2)$ . The  $\mathcal{E}$  subproblem can be solved by

$$\mathcal{E}^{t+1} = \text{Soft}_{\frac{\lambda}{\beta}} \left( \mathcal{O} - \mathcal{L}^{t+1} + \frac{\Lambda_2^t}{\beta} \right), \quad (27)$$

where  $\text{Soft}_{\tau}(\cdot)$  is the tensor soft-thresholding operator, and  $\text{Soft}_{\tau}(\cdot) = \text{sign}(\cdot) \max(|\cdot| - \tau, 0)$ . Computing  $\mathcal{E}$  and updating the multipliers  $\Lambda_1$  cost  $O(w n_1 n_2 n_3)$  at each iteration. While the computation complexity of updating  $\Lambda_2$  is  $O(n_1 n_2 n_3)$ .

The pseudo-code of our algorithm for tensor RPCA is summarized in Algorithm 2. At each iteration of Algorithm 2, it costs  $O(w n_1 n_2 n_3 (n_3 + \min(n_1, n_2)))$ . Likewise, Algorithm 2 fits the standard ADMM framework and its convergence is theoretically guaranteed [38].

## IV. NUMERICAL EXPERIMENTS

In this section, to illustrate the performance of the proposed method, we will exhibit the tensor completion experimental results on three typical kinds of third-order data, i.e., the MRI data, the MSI data, and the video data. Meanwhile, we conduct Three numerical metrics, consisting of the peak signal-to-noise ratio (PSNR), the structural similarity index (SSIM) [44], and the feature similarity index (FSIM) [45] are selected to

---

**Algorithm 2** Tensor RPCA via minimizing F-TNN
 

---

**Input:** The observed tensor  $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ ; the Lagrange parameter  $\beta$ ; the parameter  $\lambda$ ; convergence criteria  $\epsilon$ ; maximum iteration  $t_{\max}$ .

**Initialization:** the framelet transform matrix  $\mathbf{W}$ ;  $\mathcal{V}^{(0)} = \text{fold}_3(\mathbf{W}\mathcal{O}_{(3)})$  and  $\mathcal{E}^{(0)} = \text{zeros}(n_1 \times n_2 \times n_3)$ ;  $t = 0$ .

- 1: **while** not converged and  $t < t_{\max}$  **do**
- 2:   Update  $\mathcal{V}^{t+1}$  via Eq. (25);
- 3:   Update  $\mathcal{L}^{t+1}$  via Eq. (26);
- 4:   Update  $\mathcal{E}^{t+1}$  via Eq. (27);
- 5:   Update  $\Lambda_1$  and  $\Lambda_2$  via Eq. (20);
- 6:   Check the convergence conditions  $\|\mathcal{V}^{k+1} - \mathcal{V}^k\|_{\infty} \leq \epsilon$ ,  $\|\mathcal{L}^{k+1} - \mathcal{L}^k\|_{\infty} \leq \epsilon$ , and  $\|\mathcal{E}^{k+1} - \mathcal{E}^k\|_{\infty} \leq \epsilon$ ;
- 7:    $t = t + 1$ .
- 8: **end while**

**Output:** The low-rank component  $\mathcal{L}$  and the sparse component  $\mathcal{E}$ .

---

quantitatively measure the reconstructed results. On account of that the data are third-order tensors, we report the mean values of PSNR, SSIM, and FISM of all the frontal slices.

**Experimental Settings:** We generated the framelet system via the piece-wise cubic B-spline. If not specified, the framelet decomposition level  $l$  is set as 4 ( $l = 2$  for the MSI data), and the Lagrangian penalty parameter  $\beta = 1$  for the tensor completion task and  $\beta = 5$  when dealing with the tensor RPCA problems. The maximum iteration  $t_{\max}$  and the convergence tolerance  $\epsilon$  are chosen as  $(t_{\max}, \epsilon) = (100, 10^{-2})$  for the tensor completion and  $(t_{\max}, \epsilon) = (200, 10^{-3})$  for the tensor RPCA. All the methods are implemented on the platform of Windows 10 and Matlab (R2017a) with an Intel(R) Core(TM) i5-4590 CPU at 3.30GHz and 16 GB RAM.

### A. Tensor Completion

We compare our F-TNN based tensor completion method with six methods, including a baseline low-rank matrix completion (LRMC) method [46], two Tucker-rank based methods HaLRTC [3] and TMac [47], a TNN based method [48], a non-convex method minimizing the partial sum of the TNN (PSTNN) [49], the DCT based TNN method (denoted as DCTNN) [34]. When employing LRMC, the input third-order tensor data is unfolded to a matrix along the third dimension.

1) **MRI Data:** We evaluate the performance of the proposed method and the compared methods on the MRI data<sup>6</sup>, which is of size  $142 \times 178 \times 121$ . As shown in Fig. 3, this is an MRI of the brain, which consists of abundant textures of the gray matter and the white matter. The sampling rates (SR) are set as 10%, 20%, and 30%.

Table III shows the quantitative assessments of the results recovered by different methods. From Table III, it can be found that the proposed method reaches the highest indices for different sampling rates. The results by TMac and DCTNN alternatively rank the second-best place. The margins between the results by our method and the second-best results are more

TABLE III

QUANTITATIVE COMPARISONS OF THE MRI DATA COMPLETION RESULTS BY LRMC [46], HaLRTC [3], TMac [47], TNN [48], PSTNN [49], DCTNN [34] AND THE PROPOSED METHOD. THE BEST VALUES AND THE SECOND BEST VALUES ARE RESPECTIVELY HIGHLIGHTED BY BOLDER FONTS AND UNDERLINES.

SR	Index	Observed	LRMC	HaLRTC	TMac	TNN	PSTNN	DCTNN	F-TNN
10%	PSNR	9.048	17.541	18.012	<u>24.866</u>	21.855	24.578	24.716	<b>26.104</b>
	SSIM	0.047	0.317	0.388	0.658	0.524	0.628	<u>0.659</u>	<b>0.759</b>
	FSIM	0.474	0.694	0.686	0.809	0.760	0.802	<u>0.817</u>	<b>0.862</b>
20%	PSNR	9.561	22.781	23.404	28.523	27.301	28.566	<u>28.595</u>	<b>30.207</b>
	SSIM	0.073	0.590	0.657	<u>0.835</u>	0.776	0.806	0.820	<b>0.886</b>
	FSIM	0.523	0.813	0.823	<u>0.896</u>	0.871	0.885	0.892	<b>0.925</b>
30%	PSNR	10.141	25.730	26.896	30.771	30.897	31.382	<u>31.547</u>	<b>33.142</b>
	SSIM	0.103	0.730	0.794	0.889	0.880	0.885	<u>0.896</u>	<b>0.936</b>
	FSIM	0.550	0.875	0.892	0.919	0.925	0.928	<u>0.935</u>	<b>0.956</b>

TABLE IV

THE AVERAGE PSNR, SSIM AND FSIM OF THE COMPLETION RESULTS ON 32 MSIs BY LRMC [46], HaLRTC [3], TMac [47], TNN [48], PSTNN [49], DCTNN [34] AND THE PROPOSED METHOD WITH DIFFERENT SAMPLING RATES. THE BEST VALUES AND THE SECOND BEST VALUES ARE RESPECTIVELY HIGHLIGHTED BY BOLDER FONTS AND UNDERLINES.

SR	Index	Observed	LRMC	HaLRTC	TMac	TNN	PSTNN	DCTNN	F-TNN
5%	PSNR	14.718	16.687	17.831	25.633	21.863	23.073	<u>32.068</u>	<b>33.536</b>
	SSIM	0.231	0.588	0.661	0.794	0.729	0.771	<u>0.909</u>	<b>0.930</b>
	FSIM	0.697	0.773	0.799	0.871	0.836	0.856	<u>0.940</u>	<b>0.955</b>
10%	PSNR	14.954	19.369	22.369	32.306	31.165	33.945	<u>37.870</u>	<b>38.415</b>
	SSIM	0.277	0.679	0.789	0.917	0.906	0.945	<u>0.974</u>	<b>0.977</b>
	FSIM	0.718	0.828	0.876	0.942	0.939	0.961	<u>0.981</u>	<b>0.984</b>
20%	PSNR	15.464	24.581	33.004	38.258	40.077	41.944	<u>42.675</u>	<b>43.557</b>
	SSIM	0.368	0.783	0.940	0.973	0.983	0.988	<u>0.992</u>	<b>0.993</b>
	FSIM	0.740	0.892	0.963	0.979	0.987	0.991	<u>0.994</u>	<b>0.995</b>

than 1.3dB considering the PSNR, and 0.03 for the SSIM and FSIM.

We illustrate one frontal slice of the results by different methods with different random sampling rates in Fig. 3. As shown in the top row of Fig. 3, when the sampling rate is 10%, the proposed method accurately reconstructs the MRI data, with a clear margin of the gray matter and the white matter. When the sampling rate is 30%, all the methods get good performances, and the white matter regions recovered by the proposed method and TMac are the visually best.

2) **MSI Data:** In this subsection, we evaluate the performance of our method and the compared methods on 32 MSIs<sup>7</sup> from the CAVE databases [50]. The size of the MSIs is  $512 \times 512 \times 31$ , where the spatial resolution is  $512 \times 512$  and the spectral resolution is 31. The sampling rates (SR) are set as 5%, 10%, and 20%<sup>8</sup>.

The average quantitative assessments of all the results by different methods are listed in Table IV. We can find that the proposed method achieves the best performance while DCTNN obtains the second best-metrics. When the sampling rate is 20%, TMac, TNN, PSTNN, DCTNN, and the proposed method all have good performances.

<sup>7</sup><http://www.cs.columbia.edu/CAVE/databases/multispectral/>.

<sup>8</sup>For the MSI data, when the sampling rate is higher than 20%, all the methods achieve very high performances and the results are very close to the ground truths. Therefore, we select the lower sampling rates to exhibit.

<sup>6</sup>[http://brainweb.bic.mni.mcgill.ca/brainweb/selection\\_normal.html](http://brainweb.bic.mni.mcgill.ca/brainweb/selection_normal.html).



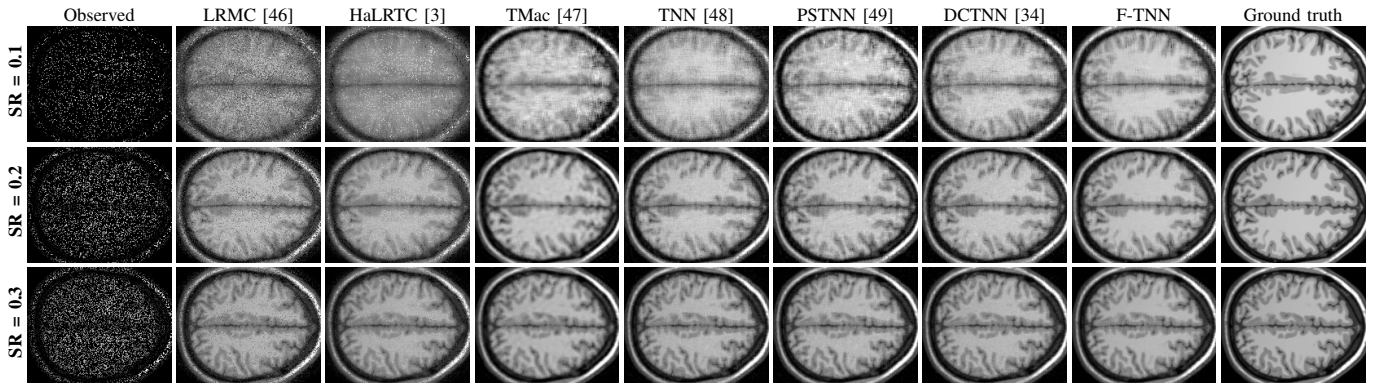


Fig. 3. The visual illustration of the results on the **MRI** data by different methods with different sampling rates (SR). From left to right are the frontal slices of observed incomplete data, results by different methods and the ground truth, respectively. From top to bottom are respectively corresponding to the 106-th slice, the 110-th slice and the 115-th slice.

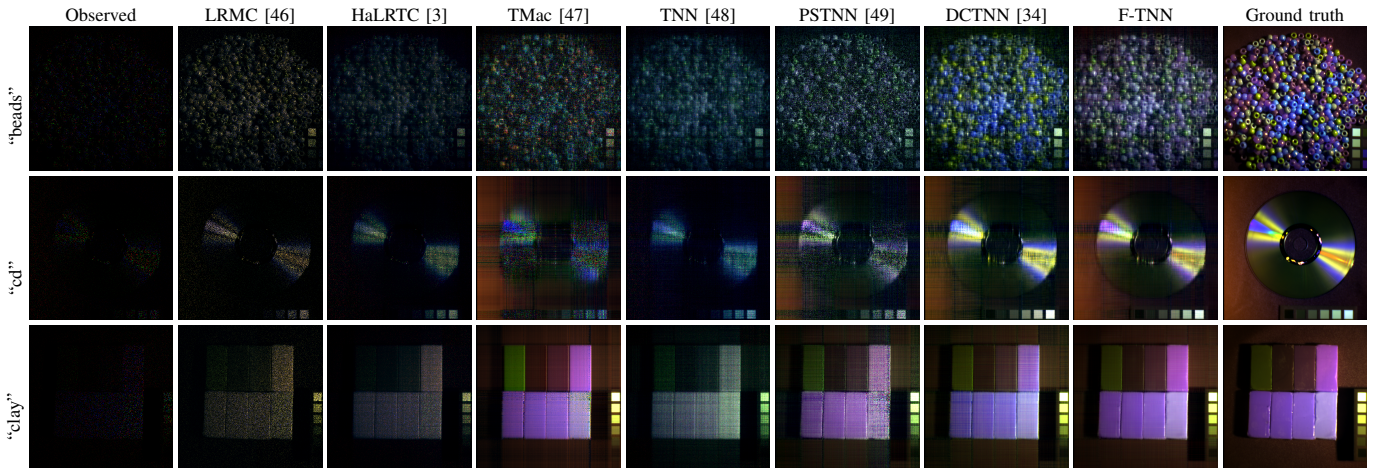


Fig. 4. The pseudo-color images (R-1 G-2 B-31) of the completion results on the **MSI** data “beads” (top row), “cd” (mid row), and “clay” (bottom row) by different methods, respectively, with the sampling rate = 0.05. From left to right are the observed incomplete data, results by different methods and the ground truth, respectively. For better visualization, the intensity of the pixels are adjusted.

The third dimension of the MSI represents the spectral information and facilitates a fine delivery of more faithful knowledge under real scenes [51]. Therefore, in Fig. 4, we illustrate the pseudo-color images (Red-1 Green-2 Blue-31) of the results on the MSI data “beads”, “cd”, and “clay”, with the sampling rate = 0.05. From the similarity of the color between the results and the ground truth, we can recognize the spectral distortion. From the first row of Fig. 4, we can see that, although DCTNN also obtains clear results on “beads” as our F-TNN, the result by DCTNN is spectrally distorted. TMac performs well on “clay”, however, undesirable artifacts can be found. The superior of the proposed F-TNN is visually obvious, considering the reconstruction of the image and preservation of spectral information.

3) *Video Data*: In this subsection, 9 videos<sup>9</sup> (respectively named “foreman”, “hall”, “carphone”, “highway”, “container”, “claire”, “news”, “coastguard” and “suzie”) with the size  $144 \times 176 \times 100$  are selected as the ground truth third-order data. The contents of these videos are different, consisting of humans, roads, rivers, cars, boats, bridges, walls and so on. The scenarios in some videos (such as “foreman”, “coast-

guard”, “suzie”, and “highway”) are more dynamic while in others are more static.

TABLE V  
THE AVERAGE PSNR, SSIM AND FSIM OF THE COMPLETION RESULTS ON 9 **videos** BY LRM [46], HaLR [3], TMac [47], TNN [48], PSTNN [49], DCTNN [34] AND THE PROPOSED METHOD WITH DIFFERENT SAMPLING RATES. THE **BEST** VALUES AND THE **SECOND BEST** VALUES ARE RESPECTIVELY HIGHLIGHTED BY BOLDER FONTS AND UNDERLINES.

SR	Index	Observed	LRM	HaLR	TMac	TNN	PSTNN	DCTNN	F-TNN
10%	PSNR	6.176	18.190	19.936	24.317	26.411	29.118	<u>29.246</u>	<b>30.654</b>
	SSIM	0.018	0.417	0.567	0.688	0.758	0.809	<u>0.819</u>	<b>0.880</b>
	FSIM	0.423	0.719	0.773	0.829	0.875	0.904	<u>0.909</u>	<b>0.931</b>
20%	PSNR	6.687	29.315	30.150	30.250	31.329	32.012	<u>32.259</u>	<b>33.568</b>
	SSIM	0.031	0.851	0.871	0.868	0.871	0.876	<u>0.881</u>	<b>0.927</b>
	FSIM	0.413	0.928	0.927	0.921	0.934	0.937	<u>0.940</u>	<b>0.957</b>
30%	PSNR	7.266	32.080	32.977	32.189	34.050	34.056	<u>34.434</u>	<b>35.820</b>
	SSIM	0.046	0.907	<u>0.917</u>	0.910	0.915	0.912	<u>0.915</u>	<b>0.951</b>
	FSIM	0.408	0.952	0.952	0.944	0.956	0.956	<u>0.958</u>	<b>0.971</b>

Table V lists the average MPSNR, MSSIM, and MFSIM on these 9 videos with different sampling rates. For different sampling rates, our F-TNN obtains the results with the best quantitative metrics. When the sampling rates are 10% and 20, the performances of PSTNN and DCTNN are comparable. The

<sup>9</sup><http://trace.eas.asu.edu/yuv/>.

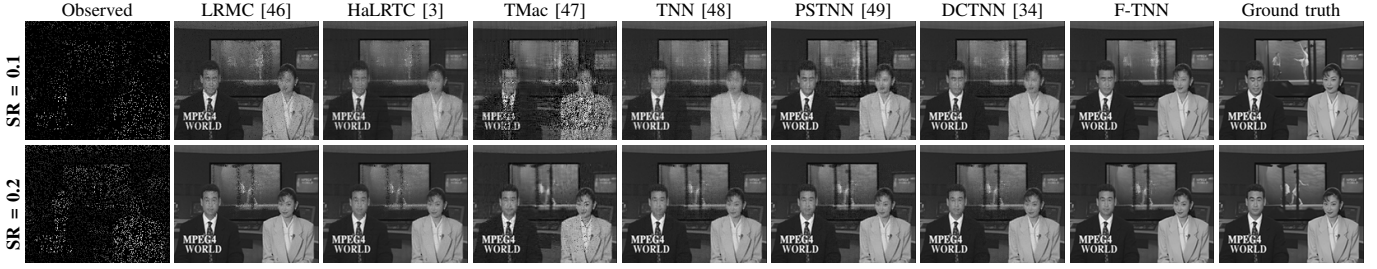


Fig. 5. The completion results on the video data “news” different methods with different sampling rates. From left to right are the observed incomplete data, results by different methods and the ground truth, respectively. From top to bottom are respectively the 15-th frame and the 67-th frame.

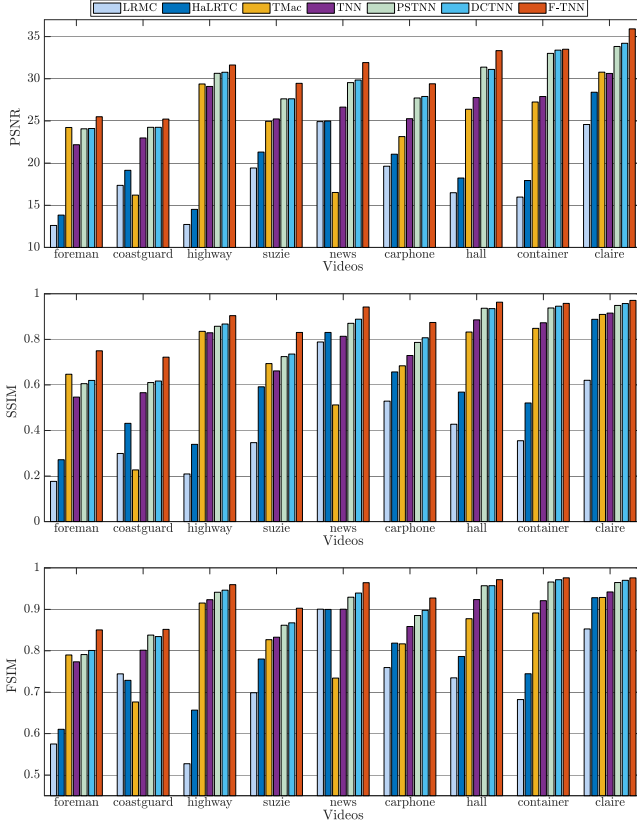


Fig. 6. The PSNR, SSIM, and FSIM of the results by different methods on all the video data with the sampling rate 10%.

DCTNN ranks second with the sampling rate of 30%. Fig. 5 exhibits the frames of the results on the videos, “news” with sampling rates 10% and 20%. The video “news” is captured by a static camera in a stationary scenery, and there are two dynamic parts, which are the two newscasters in the front position and a playing screen in the back, in this video. Thus, the scenario in this video contains both dynamic and static components. Most compared methods can reconstruct the static parts well while the proposed method obtains the best recovering performances on both the two newscasters (see their faces) and the dynamic screen.

To further illustrate the performance of all the methods on different videos, in Fig. 6 we exhibit the PSNR, SSIM, and FSIM on all the videos by all the methods when the sampling rate is 10%. From Fig. 6, it can be found that TMac is unstable with respect to different videos while other methods maintain

TABLE VI

THE SETTINGS OF THE PARAMETER  $\lambda$  FOR ALL THE METHODS, GIVEN THE OBSERVATION  $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ .

Method	Image recovery	Background subtraction
MRPCA	$1.5/\sqrt{n_1 n_2}$	$1/\sqrt{n_1 n_2}$
SNN	$3/\sqrt{n_1 n_2}$	$0.5/\sqrt{\max(n_1, n_2)n_3}$
TNN	$3/\sqrt{\max(n_1, n_2)n_3}$	$1/\sqrt{2 \max(n_1, n_2)n_3}$
DCTNN	$2/\sqrt{\max(n_1, n_2)n_3}$	$4/\sqrt{\max(n_1, n_2)n_3}$
F-TNN	$3/\sqrt{\max(n_1, n_3)n_2}$	$3/\sqrt{\max(n_1, n_2)n_3}$

better metrics when the video is more static. Although the scenario in “highway” is dynamic along the temporal direction, the contents in this video are not complicated. Therefore, many methods achieve good performances. It can be observed that the proposed method obtains the highest PSNR, SSIM, and FSIM on all the videos. This validates the robustness of our F-TNN.

### B. Tensor Robust Principal Component Analysis

In this section, we test our F-TNN based TRPCA methods on two problems, i.e., color images recovery from observations corrupted by the salt-and-pepper noise, and the background subtraction for surveillance videos. The compared methods consist of one matrix nuclear norm minimization based RPCA method (denoted as MRPCA) [14], a sum of the nuclear norm minimization based tensor RPCA method (denoted as “SNN”)[52], a TNN based tensor RPCA method [31], and a DCT transformed TNN based tensor RPCA method [35]. The  $\ell_1$  norm is used to characterize the sparse component by all the compared methods. The balance parameter  $\lambda$ , which is added to the  $\ell_1$  term, is manually selected for the best performances for all the methods. We list the settings of  $\lambda$  in Table VI. When implementing MRPCA, we unfold the observed data  $\mathcal{O}$  along the third mode and input  $\mathbf{O}_{(3)}$ . For the image recovery, since that the framelet transformation matrix  $\mathbf{W}$  requires the third dimension of the input data no less than 40, we shift the dimension of the input data no less than 40, we shift the dimension of the observed image as  $\hat{\mathcal{O}} \in \mathbb{R}^{n_2 \times n_3 \times n_1}$  via the Matlab command “shiftdim( $\cdot$ ,1)”.

1) *Color Image Recovery*: We select 4 images<sup>10</sup>, respectively named “airplane”, “watch”, “fruits”, and “baboon”, as ground truth clean images. Then, the salt-and-pepper noise

<sup>10</sup>The images named “airplane”, “fruits”, and “baboon” are of the size  $512 \times 512 \times 3$  and available at <http://sipi.usc.edu/database/database.php>, while the image “watch” of the size  $1024 \times 768 \times 3$  is available at <https://www.sitepoint.com/mastering-image-optimization-in-wordpress/>



TABLE VII

QUANTITATIVE COMPARISONS OF THE IMAGE RECOVERY RESULTS OF MRPCA [14], SNN [52], TNN [31], DCTNN [35], AND THE PROPOSED METHOD. THE BEST VALUES AND THE SECOND BEST VALUES ARE RESPECTIVELY HIGHLIGHTED BY BOLDER FONTS AND UNDERLINES.

$\rho$	Index	Observed	MRPCA	SNN	TNN	DCTNN	F-TNN
5%	PSNR	18.005	21.671	30.188	29.791	<u>31.735</u>	<b>33.846</b>
	SSIM	0.587	0.771	0.962	0.964	<u>0.979</u>	<b>0.987</b>
	FSIM	0.833	0.896	0.973	0.970	<u>0.982</u>	<b>0.990</b>
10%	PSNR	14.987	19.245	27.932	29.140	<u>30.807</u>	<b>31.937</b>
	SSIM	0.450	0.664	0.917	0.957	<u>0.971</u>	<b>0.975</b>
	FSIM	0.744	0.844	0.954	0.965	<u>0.977</u>	<b>0.984</b>

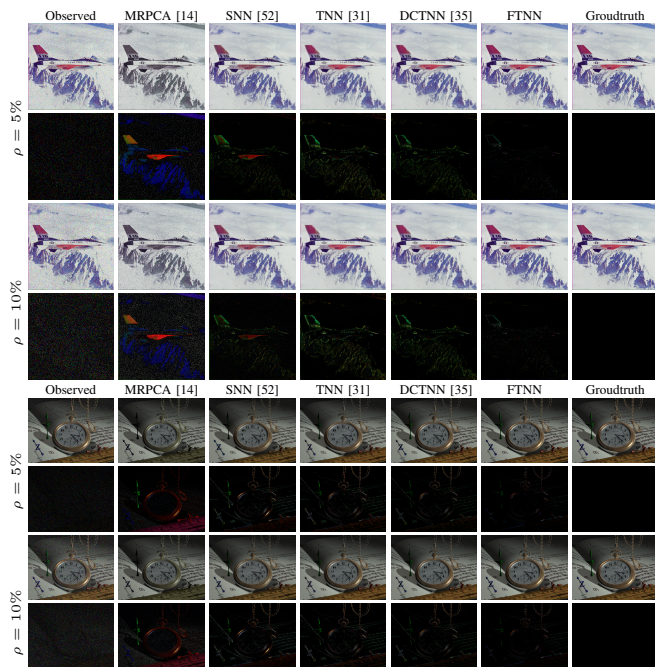


Fig. 7. The top four rows are the image recovery results and residual images on the image “airplane”, and the bottom 4 rows are corresponding to the image “watch”.

is added to these images, affecting  $\rho$  pixels. The parameter  $\rho$  varies from 5% to 10%. Table VII presents the averaged PSNR, SSIM, and FSIM values of the results by different methods for the color image recovery. We can find that the performance of our method is the best with different  $\rho_s$ . We exhibit the visual results on the images “airplane” and “watch” in Fig. 7. It can be obtained that all the tensor-based methods remove the salt-and-pepper noise while the performance of MRPCA is unsatisfactory. The residual images, which are absolute values of the difference between results and clean images, are magnified with a factor 2 for better visualization. From the residual images, we can see that our method preserves the structure and details of the color images well.

2) *Background subtraction*: Four video sequences, respectively named “Bootstrap1285”, “Escalator2805”, “ShoppingMall1535”, and “hall1368”, are selected from Li’s dataset<sup>11</sup>. After transforming the color frames to gray level ones, each video is of the size  $130 \times 160 \times 40$ . Results by all of the methods are displayed in Fig. 8. We can see that our method

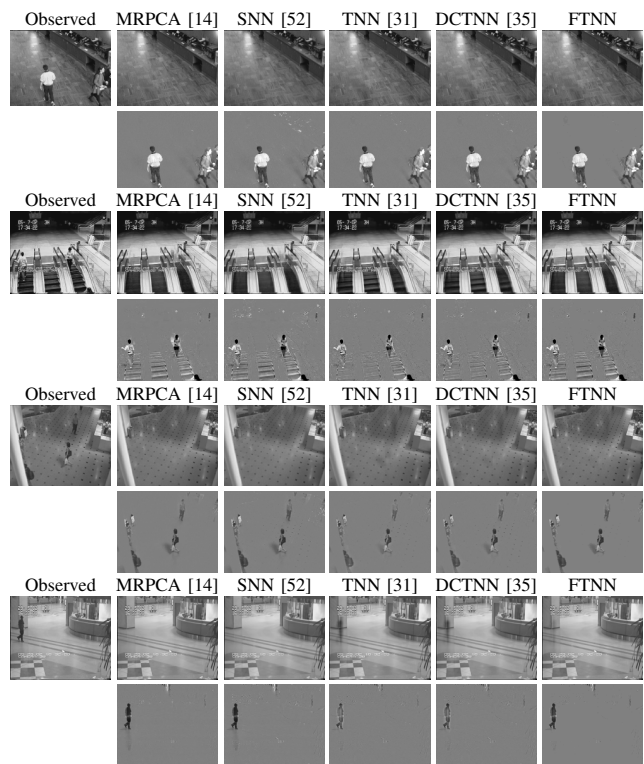


Fig. 8. Background subtraction results by different methods. The left column lists one frame of the observed video. From top to bottom are respectively separation results, i.e., the background and the foreground, of the video “Bootstrap1285”, “Escalator2805”, “ShoppingMall1535”, and “hall1368”. For better visualization, we add 0.5 to the foreground.

and MRPCA perform well for the videos “Bootstrap1285” and “ShoppingMall1535”, while some incorrectly extractions can be found in the foreground results by other three methods, the front desk in “Bootstrap1285” and the dot pattern of the ground in “ShoppingMall1535” for examples. For videos “Escalator2805” and “hall1368”, all the methods incorrectly extract contents of the background to the foreground, more or less. Overall, the foregrounds extracted by our method are the purest.

### C. Discussions

1) *Framelet setting*: In this part, taking the completion of MRI data (SR = 10%) as an example, we evaluate the performance of the proposed method with different Framelet transformation settings. Firstly, including the piece-wise cubic B-spline (denoted as “cubic”), we also adopted the Haar wavelet (denoted as “Haar”) and the piece-wise linear B-spline (denoted as “linear”) to generate the framelet transformation. Meanwhile, we also set the decomposition levels from 1 to 5. The quantitative metrics of the results obtained by the proposed method with different framelet settings are reported in Table VIII. From Table VIII, we can find that the piece-wise cubic B-spline is the best choice. As the decomposition level arise, the performance of the proposed method becomes better until level 5. Setting the level as 3 or 4 is a good choice.

2) *Convergency Behaviours*: Also, we take the completion of MRI data as an example to illustrate the convergency behaviours of our algorithm with respect to different sampling

<sup>11</sup>Data available at <http://vis-www.cs.umass.edu/narayana/castanza/I2Rdataset/>



TABLE VIII  
THE PSNR, SSIM AND FSIM OF THE RECOVERY RESULTS ON THE MRI DATA BY THE PROPOSED METHOD WITH DIFFERENT FRAMELET SETTINGS. THE BEST VALUES ARE HIGHLIGHTED BY BOLDER FONTS.

Filters	Index	Level = 1	Level = 2	Level = 3	Level = 4	level = 5
Haar	PSNR	21.176	23.327	24.183	24.366	24.372
	SSIM	0.537	0.647	0.680	0.685	0.685
	FSIM	0.755	0.801	0.817	0.821	0.821
Linear	PSNR	22.466	24.904	25.538	25.563	25.509
	SSIM	0.611	0.717	0.738	0.738	0.735
	FSIM	0.785	0.834	0.846	0.848	0.847
Cubic	PSNR	23.726	26.077	<b>26.287</b>	26.104	25.970
	SSIM	0.673	0.761	<b>0.765</b>	0.759	0.746
	FSIM	0.812	0.858	<b>0.863</b>	0.862	0.858

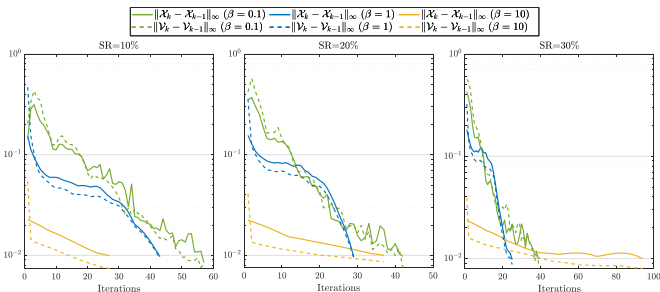


Fig. 9. The convergence behaviours of Algorithm 1, with respect to different sampling rates and different  $\beta$ .

rates and different parameters. In the framework of ADMM, the parameter  $\beta$ , which is brought in by the augmented Lagrangian function, mainly affects the convergency behaviour of our method. Thus, we test our algorithm with  $\beta = 10^{-1}, 1, 10$ . We plot  $\|\mathcal{V}^{k+1} - \mathcal{V}^k\|_\infty$  and  $\|\mathcal{X}^{k+1} - \mathcal{X}^k\|_\infty$  of each iteration in Fig. 9. It can be seen that when  $\beta = 10^{-1}$  and 1 our algorithm steadily converges. Although the behaviour of  $\|\mathcal{X}^{k+1} - \mathcal{X}^k\|_\infty$  is not that stable when  $\beta = 10$ , our algorithm also converges rapidly.

## V. CONCLUSIONS

In this paper, we propose to replace the Fourier transform by the framelet in the t-SVD framework. Then, we formulate the framelet representation of the tensor multi-rank and tensor nuclear norm. A low-rank tensor completion model and a tensor robust principal component analysis model are proposed by minimizing the framelet based tensor nuclear norm. We develop ADMM based algorithms to solve these convex models with guaranteed convergence. We compare the performance of the proposed method with state-of-the-art methods via numerical experiments on the magnetic resonance imaging data, videos, color images, and multispectral images. Our method outperforms many state-of-the-art methods quantitatively and visually.

## REFERENCES

- [1] A. Sobral and E.-h. Zahzah, "Matrix and tensor completion algorithms for background model initialization: A comparative evaluation," *Pattern Recognition Letters*, vol. 96, pp. 22–33, 2017.
- [2] T. Korah and C. Rasmussen, "Spatiotemporal inpainting for recovering texture maps of occluded building facades," *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2262–2271, 2007.

- [3] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2013.
- [4] J.-H. Yang, X.-L. Zhao, T.-H. Ma, Y. Chen, T.-Z. Huang, and M. Ding, "Remote sensing images destriping using unidirectional hybrid total variation and nonconvex low-rank regularization," *Journal of Computational and Applied Mathematics*, vol. 363, pp. 124–144, 2020.
- [5] R. Dian and S. Li, "Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization," *IEEE Transactions on Image Processing*, vol. 28, pp. 5135–5146, 2019.
- [6] L.-J. Deng, M. Feng, and X.-C. Tai, "The fusion of panchromatic and multispectral remote sensing images via tensor-based sparse modeling and hyper-laplacian prior," *Information Fusion*, vol. 52, pp. 76–89, 2019.
- [7] J.-T. Sun, H.-J. Zeng, H. Liu, Y. Lu, and Z. Chen, "CubeSVD: a novel approach to personalized web search," in *Proceedings of the International Conference on World Wide Web*, 2005, pp. 382–390.
- [8] D. A. Lima and G. M. Oliveira, "A cellular automata ant memory model of foraging in a swarm of robots," *Applied Mathematical Modelling*, vol. 47, pp. 551–572, 2017.
- [9] T. G. Kolda, B. W. Bader, and J. P. Kenny, "Higher-order web link analysis using multilinear algebra," in *Proceedings of the IEEE International Conference on Data Mining*, 2005, pp. 242–249.
- [10] V. N. Varghees, M. S. Manikandan, and R. Gini, "Adaptive MRI image denoising using total-variation and local noise estimation," in *Proceedings of the International Conference on Advances in Engineering, Science and Management (ICAESM)*, 2012, pp. 506–511.
- [11] N. Kreimer and M. D. Sacchi, "A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation," *Geophysics*, vol. 77, no. 3, pp. V113–V122, 2012.
- [12] J.-H. Yang, X.-L. Zhao, T.-Y. Ji, T.-H. Ma, and T.-Z. Huang, "Low-rank tensor train for tensor robust principal component analysis," *Applied Mathematics and Computation*, vol. 367, p. 124783, 2020.
- [13] E. Candès and B. Recht, "Exact matrix completion via convex optimization," *Communications of the ACM*, vol. 55, no. 6, pp. 111–119, 2012.
- [14] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.
- [15] Z. Song, D. Woodruff, and H. Zhang, "Sublinear time orthogonal tensor decomposition," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 793–801.
- [16] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011.
- [17] P. Tichavský, A.-H. Phan, and A. Cichocki, "Numerical CP decomposition of some difficult tensors," *Journal of Computational and Applied Mathematics*, vol. 317, pp. 362–370, 2017.
- [18] Y.-F. Li, K. Shang, and Z.-H. Huang, "Low Tucker rank tensor recovery via admm based on exact and inexact iteratively reweighted algorithms," *Journal of Computational and Applied Mathematics*, vol. 331, pp. 64–81, 2018.
- [19] X. Li, M. K. Ng, G. Cong, Y. Ye, and Q. Wu, "MR-NTD: manifold regularization nonnegative Tucker decomposition for tensor data dimension reduction and representation," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 8, pp. 1787–1800, 2017.
- [20] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, p. 025010, 2011.
- [21] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *Journal of the ACM (JACM)*, vol. 60, no. 6, p. 45, 2013.
- [22] K. Braman, "Third-order tensors as linear operators on a space of matrices," *Linear Algebra and its Applications*, vol. 433, no. 7, pp. 1241–1253, 2010.
- [23] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and its Applications*, vol. 435, no. 3, pp. 641–658, 2011.
- [24] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 148–172, 2013.
- [25] C. D. Martin, R. Shafer, and B. LaRue, "An order-p tensor factorization with applications in imaging," *SIAM Journal on Scientific Computing*, vol. 35, no. 1, pp. A474–A490, 2013.
- [26] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-H. Ma, and T.-Y. Ji, "Mixed noise removal in hyperspectral image via low-fibered-rank regularization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 1, pp. 734–749, 2019.

- [27] E. Kernfeld, M. Kilmer, and S. Aeron, "Tensor-tensor products with invertible linear transforms," *Linear Algebra and its Applications*, vol. 485, pp. 545–570, 2015.
- [28] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-SVD," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3842–3849.
- [29] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5249–5257.
- [30] J. Q. Jiang and M. K. Ng, "Robust low-tubal-rank tensor completion via convex optimization," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 10–16.
- [31] C. Lu, J. Feng, W. Liu, Z. Lin, S. Yan *et al.*, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE transactions on pattern analysis and machine intelligence*, 2019.
- [32] W. Hu, D. Tao, W. Zhang, Y. Xie, and Y. Yang, "The twist tensor nuclear norm for video completion," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 12, pp. 2961–2973, 2017.
- [33] W.-H. Xu, X.-L. Zhao, and M. Ng, "A fast algorithm for cosine transform based tensor singular value decomposition," *arXiv preprint arXiv:1902.03070*, 2019.
- [34] C. Lu, X. Peng, and Y. Wei, "Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5996–6004.
- [35] C. Lu and P. Zhou, "Exact recovery of tensor robust principal component analysis under linear transforms," *arXiv preprint arXiv:1907.08288*, 2019.
- [36] G. Song, M. K. Ng, and X. Zhang, "Robust tensor completion using transformed tensor SVD," *arXiv preprint arXiv:1907.01113*, 2019.
- [37] J.-F. Cai, R. H. Chan, and Z. Shen, "A framelet-based image inpainting algorithm," *Applied and Computational Harmonic Analysis*, vol. 24, no. 2, pp. 131–149, 2008.
- [38] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [39] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
- [40] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, T.-Y. Ji, and L.-J. Deng, "Matrix factorization for low-rank tensor completion using framelet prior," *Information Sciences*, vol. 436–437, pp. 403–417, 2017.
- [41] A. Ron and Z. Shen, "Affine systems in  $L_2(\mathbb{R}^d)$ : The analysis of the analysis operator," *Journal of Functional Analysis*, vol. 148, no. 2, pp. 408–447, 1997.
- [42] G. Bergland, "A guided tour of the fast fourier transform," *IEEE spectrum*, vol. 6, no. 7, pp. 41–52, 1969.
- [43] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.
- [44] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [45] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, 2011.
- [46] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, p. 717, 2009.
- [47] Y. Xu, R. Hao, W. Yin, and Z. Su, "Parallel matrix factorization for low-rank tensor completion," *Inverse Problems & Imaging*, vol. 9, no. 2, pp. 601–624, 2015.
- [48] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Transactions on Signal Processing*, vol. 65, no. 6, pp. 1511–1526, 2017.
- [49] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, and L.-J. Deng, "Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm," *Journal of Computational and Applied Mathematics*, p. 112680, 2019, <https://doi.org/10.1016/j.cam.2019.112680>.
- [50] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, "Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum," *IEEE transactions on image processing*, vol. 19, no. 9, pp. 2241–2253, 2010.
- [51] Q. Xie, Q. Zhao, D. Meng, Z. Xu, S. Gu, W. Zuo, and L. Zhang, "Multispectral images denoising by intrinsic tensor sparsity regularization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1692–1700.
- [52] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 1, pp. 225–253, 2014.