

## Deep plug-and-play prior for low-rank tensor completion<sup>☆</sup>

Xi-Le Zhao<sup>a</sup>, Wen-Hao Xu<sup>a</sup>, Tai-Xiang Jiang<sup>b,\*</sup>, Yao Wang<sup>c,d</sup>, Michael K. Ng<sup>e</sup>

<sup>a</sup>School of Mathematical Sciences/Research Center for Image and Vision Computing, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China

<sup>b</sup>FinTech Innovation Center, Financial Intelligence and Financial Engineering Research Key Laboratory of Sichuan Province, School of Economic Information Engineering, Southwest University of Finance and Economics, Chengdu, Sichuan 611130, China

<sup>c</sup>School of Management, Xi'an Jiaotong University, Xi'an 710049, China

<sup>d</sup>State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China

<sup>e</sup>Department of Mathematics, The University of Hong Kong, Pokfulam, Hong Kong

### ARTICLE INFO

#### Article history:

Received 1 November 2019

Revised 24 January 2020

Accepted 2 March 2020

Available online xxx

Communicated by Dr. Xin Luo

#### Keywords:

Tensor completion

Tensor nuclear norm

Denoising neural network

Alternating direction method of multipliers

Plug-and-play framework

### ABSTRACT

Multi-dimensional images, such as color images and multi-spectral images (MSIs), are highly correlated and contain abundant spatial and spectral information. However, real-world multi-dimensional images are usually corrupted by missing entries. By integrating deterministic low-rankness prior to the data-driven deep prior, we suggest a novel regularized tensor completion model for multi-dimensional image completion. In the objective function, we adopt the newly emerged tensor nuclear norm (TNN) to characterize the global low-rankness prior of multi-dimensional images. We also formulate an implicit regularizer by plugging a denoising neural network (termed as deep denoiser), which is convinced to express the deep image prior learned from a large number of natural images. The resulting model can be solved by the alternating directional method of multipliers algorithm under the plug-and-play (PnP) framework. Experimental results on color images, videos, and MSIs demonstrate that the proposed method can recover both the global structure and fine details very well and achieve superior performance over competing methods in terms of quality metrics and visual effects.

© 2020 Elsevier B.V. All rights reserved.

### 1. Introduction

The image completion problem aims to estimate missing entries from the partially observed entries, which is a fundamental problem in low-level computer vision and computational imaging [1–8]. By exploiting the redundancy of natural images, the image completion problem can be formulated as the following low-rank matrix completion (LRMC) model:

$$\arg \min_{\mathbf{X}} \text{rank}(\mathbf{X}) \quad \text{s.t. } \mathcal{P}_{\Omega}(\mathbf{X}) = \mathcal{P}_{\Omega}(\mathbf{O}), \quad (1)$$

where  $\mathbf{X}$  is the underlying matrix,  $\mathbf{O}$  is the observed incomplete matrix,  $\Omega$  is the index set of the observed entries, and  $\mathcal{P}_{\Omega}(\cdot)$  is the

projection function that keeps the entries in  $\Omega$  while sets others be zeros. For grayscale images, the low-rankness is characterized by the matrix rank or its convex envelope, i.e., the nuclear norm. For multi-dimensional images, the image elements are usually reordered into a matrix, which is known as matricization or unfolding.

However, matricization will inevitably destroy the intrinsic structure of multi-dimensional images. As the higher-order extension of the matrix, the tensor can provide a more natural and elegant representation for multi-dimensional images. Thus we can formulate multi-dimensional images completion problem as the following low-rank tensor completion (LRTC) model:

$$\arg \min_{\mathcal{X}} \text{rank}(\mathcal{X}) \quad \text{s.t. } \mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{O}), \quad (2)$$

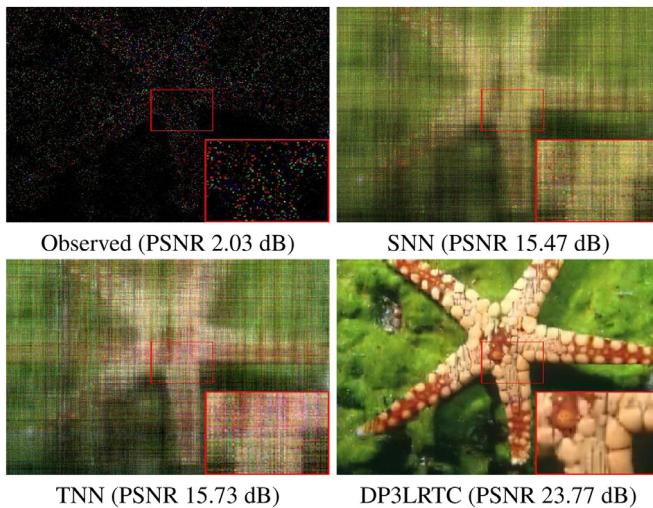
where  $\mathcal{X}$  is the underlying tensor and  $\mathcal{O}$  is the observed incomplete tensor (as shown in the top-left of Fig. 1). Here, we generally consider the element-wise sampling in the model (2), which can be easily extended to the structural sampling, e.g., inpainting [1] and demosaicing [9].

Different from the matrix case, there is no unique definition for the tensor rank. How to define the tensor rank is a fundamental problem. Many research efforts have been devoted to this hot

\* This work is supported by the National Natural Science Foundation of China (61876203, 61772003, and 11971374), the Fundamental Research Funds for the Central Universities (JBK2001011), HKRGC GRF (12306616, 12200317, 12300218, and 12300519), HKU Grant (104005583), China Postdoctoral Science Foundation (2017M610628 and 2018T111031), and the State Key Laboratory of Robotics (2019-006).

<sup>a</sup> Corresponding author.

E-mail addresses: [xlzhao122003@163.com](mailto:xlzhao122003@163.com) (X.-L. Zhao), [seanxwh@gmail.com](mailto:seanxwh@gmail.com) (W.-H. Xu), [taixiangjiang@gmail.com](mailto:taixiangjiang@gmail.com) (T.-X. Jiang), [yao.s.wang@gmail.com](mailto:yao.s.wang@gmail.com) (Y. Wang), [mng@maths.hku.hk](mailto:mng@maths.hku.hk) (M.K. Ng).



**Fig. 1.** The recovered color images *Starfish* by SNN, TNN, and the proposed DP3LRTC with the sampling rate 5%.

topic [10–13], such as the CANDECOMP/PARAFAC (CP) rank and the Tucker rank. The CP rank is defined based on the CP decomposition, where an  $n$ -th order tensor is decomposed as the sum of the rank-one tensors [14,15], i.e., the outer product of  $n$  vectors. The CP rank is defined as the minimal number of the rank-one tensors required to express the target tensor [16]. However, its related optimization is difficult, and even determining the CP rank of a given tensor is NP-hard [17]. The Tucker rank is based on the Tucker decomposition which decomposes a tensor into a core tensor multiplied by a matrix along each mode [18–21]. The Tucker rank is defined as the vector consisting of the ranks of unfolding matrices along different modes. The Tucker rank has been considered in the LRTC problem by minimizing its convex surrogate, i.e., the sum of the nuclear norm (SNN) [18], or the non-convex surrogates [22,23]. The top-right of Fig. 1 exhibits the recovered result by the SNN-based model. However, unfolding the tensor along each mode will also destroy the intrinsic structures of the tensor.

The tensor singular value decomposition (t-SVD), based on the tensor-tensor product (t-product), has been emerged as a powerful tool for preserving the intrinsic structures of the tensor [24,25]. Although the t-SVD is originally suggested for third-order tensors, it has been extended to  $n$ -th order tensors ( $n > 3$ ) [26,27]. Based on the t-SVD, the corresponding multi-rank and tubal-rank have received increasing attention. The tensor nuclear norm (TNN) [28–32] is suggested as a convex surrogate of the multi-rank. The TNN-based LRTC model is given as follows:

$$\arg \min_{\mathcal{X}} \|\mathcal{X}\|_{\text{TNN}} \quad \text{s.t.} \quad \mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{O}), \quad (3)$$

where  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ ,  $\|\mathcal{X}\|_{\text{TNN}} = \sum_{i=1}^{n_3} \|\tilde{\mathbf{X}}^{(i)}\|_*$ ,  $\tilde{\mathbf{X}}^{(i)}$  is the  $i$ -th frontal slice of  $\tilde{\mathcal{X}}$ , and  $\tilde{\mathcal{X}}$  is the tensor generated by performing discrete Fourier transformation (DFT) along the mode-3 fibers of  $\mathcal{X}$ , i.e.,  $\tilde{\mathcal{X}} = \text{fft}(\mathcal{X}, [], 3)$ . The TNN regularizer has the advantage of preserving the global structure of multi-dimensional images. We can observe from Fig. 1 that the recovered result by the TNN-based method (bottom-left) is slightly better than that by the SNN-based method (top-right).

Although promoting the low-rankness of the underlying tensor has shown its capability for multi-dimensional images completion, it suffer from two drawbacks. First, many real-world multi-dimensional images contain abundant fine details. This makes these multi-dimensional images are not strictly low-rank in mathematics. Second, for the element-wise sampling with low sampling rates or structural samplings, the rank of these observed tensors is already low. The low-rank regularization is insufficient for a

fancy recovery of the underlying tensor. This phenomenon can be observed in Fig. 1, where results by the SNN-based model and TNN-based model are of low-quality when the sampling rate is 5%.

As compensation, many LRTC methods introduce additional regularizers, which express other types of image priors, for a better solution of this ill-posed inverse problem. For example, the total variation (TV) regularizer, framelet regularizer, and nonlocal regularizer have received considerable attention [33–36]. In particular, the TV regularizer was incorporated into the TNN-based LRTC model by exploiting the local smoothness [37]. Recently, deep learning-based methods have been developed to learn data-driven priors from a large number of natural images [38–40] and have shown promising performance on extensive application. Therefore, we consider to leverage the prior information learned by deep learning-based methods. On account of that convolutional neural networks (CNNs) are successfully applied for the recovery of 2D natural images (or color images), we consider to regularize the spatial slices of the multi-dimensional images by CNNs, which are expected to preserve the image details well. Meanwhile, for low sampling rate cases, this data-driven term would compensate the recovery of the underlying tensor.

In this work, we attempt to simultaneously exploit the respective strengths of the deterministic low-rankness prior and the data-driven deep image prior. By integrating the TNN regularizer with an implicit regularizer, which express the deep image prior, we suggest the novel LRTC model as follows:

$$\arg \min_{\mathcal{X}} \|\mathcal{X}\|_{\text{TNN}} + \lambda \Phi(\mathcal{X}) \quad \text{s.t.} \quad \mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{O}), \quad (4)$$

where  $\Phi(\mathcal{X})$  is an implicit regularizer by plugging the denoising CNNs (termed as deep denoisers). In (4), the two regularizers are organically combined and benefit from each other. On the one hand, the implicit regularizer can bring in deep image priors to each spatial slice of the multi-dimensional images, well characterizing the fine details which can hardly be captured by the TNN regularizer. On the other hand, TNN can enforce the inner global correlations of the multi-dimensional images, avoiding the lackness of inherent relations that the slices, which are separately handled by the deep denoiser.  $\|\mathcal{X}\|_{\text{TNN}}$  and  $\Phi(\mathcal{X})$  respectively regularize the coarse structure and the fine details, being complementary to each other. To efficiently solve the proposed model, we develop the alternating direction method of multipliers (ADMM) [41] under the highly flexible Plug-and-Play (PnP) framework [42–44], which allows us to plug in the state-of-the-art deep denoisers. We term our method as **deep plug-and-play prior for low-rank tensor completion** (DP3LRTC). From the bottom-right of Fig. 1, we can observe that the deterministic low-rankness prior and data-driven deep image prior contribute to the superior performance of the proposed DP3LRTC.

The rest of this paper is organized as follows. Section 2 presents minimal preliminaries necessary for the subsequent discussion. Section 3 gives the corresponding solving algorithm to tackle the proposed model. Section 4 gives the experimental results and discusses the details about DP3LRTC. Section 5 concludes this paper.

## 2. Preliminaries

In this section, we introduce minimal and necessary preliminaries of the modular parts, i.e., TNN and PnP, for the subsequent discussion.

### 2.1. Notations

In this subsection, we briefly introduce the basic notations. We denote vectors as bold lowercase letters (e.g.,  $\mathbf{x}$ ), matrices as uppercase letters (e.g.,  $\mathbf{X}$ ), and tensors as calligraphic letters (e.g.,  $\mathcal{X}$ ). For a third-order tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , we denote its  $(i, j, k)$ -th

element as  $\mathcal{X}(i, j, k)$  or  $\mathcal{X}_{i,j,k}$  with the MATLAB notation. Its  $(i, j)$ -th mode-1, mode-2, and mode-3 fibers are denoted as  $\mathcal{X}(:, i, j)$ ,  $\mathcal{X}(i, :, j)$ , and  $\mathcal{X}(i, j, :)$ , respectively. We use  $\mathcal{X}(i, :, :, :)$ ,  $\mathcal{X}(:, i, :, :)$ , and  $\mathcal{X}(:, :, i)$  to denote the  $i$ -th horizontal, lateral, and frontal slices of  $\mathcal{X}$ , respectively. More compactly, we also use  $\mathbf{X}^{(i)}$  to represent the  $i$ -th frontal slices  $\mathcal{X}(:, :, i)$ . The Frobenius norm of  $\mathcal{X}$  is defined as  $\|\mathcal{X}\|_F := \sqrt{\sum_{i,j,k} |\mathcal{X}(i, j, k)|^2}$ .

## 2.2. T-SVD and TNN

Recently, Kilmer have introduced the novel tensor-tensor product (termed as (t-product) as follows:

**Definition 1** (t-product [25]). Let  $\mathcal{A}$  be  $n_1 \times n_2 \times n_3$  and  $\mathcal{B}$  be  $n_2 \times n_4 \times n_3$ . Then the tensor-tensor product is the  $n_1 \times n_4 \times n_3$  tensor  $\mathcal{C} = \mathcal{A} * \mathcal{B}$  whose  $(i, j)$ -th fiber is given by

$$\mathcal{C}(i, j, :) = \sum_{k=1}^{n_2} \mathcal{B}(i, k, :) \star \mathcal{C}(k, j, :), \quad (5)$$

where “ $\star$ ” denotes circular convolution of two vectors.

**Definition 2** (special tensors [25]). The **conjugate transpose** of a third-order tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , denote as  $\mathcal{X}^H$ , is the tensor obtained by conjugate transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through  $n_3$ . The **identity tensor**  $\mathcal{I} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is the tensor whose first frontal slice is the identity matrix, and the other frontal slices are all zeros. A third-order tensor  $\mathcal{Q}$  is **orthogonal** if  $\mathcal{Q} * \mathcal{Q}^H = \mathcal{Q}^H * \mathcal{Q} = \mathcal{I}$ .

Based on t-product, the tensor singular value decomposition (t-SVD) is suggested.

**Theorem 1** (t-SVD [25]). Let  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  be a third-order tensor, then it can be decomposed as

$$\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^H, \quad (6)$$

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 the orthogonal tensors, and  $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  is a f-diagonal tensor whose frontal slices are diagonal matrices.

The t-SVD can be efficiently obtained by computing a series of matrix SVDs in the Fourier domain; see [Algorithm 1](#) for more details.

Based on t-SVD, we have the corresponding definitions of tensor multi-rank and tubal-rank.

**Definition 3** (tensor tubal-rank and multi-rank [28]). Let  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  be a third-order tensor, the tensor multi-rank, denoted as  $\text{rank}_m(\mathcal{X}) \in \mathbb{R}^{n_3}$ , is a vector whose  $i$ -th element is the rank of the  $i$ -th frontal slice of  $\tilde{\mathcal{X}}$ , where  $\tilde{\mathcal{X}} = \text{fft}(\mathcal{X}, [], 3)$ . The tubal-rank of  $\mathcal{X}$ , denote as  $\text{rank}_t(\mathcal{X})$ , is defined as the number of non-zero tubes of  $\mathcal{S}$ , where  $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^H$ .

The relationship between the tubal-rank and multi-rank is  $\text{rank}_t(\mathcal{X}) = \max(\text{rank}_m(\mathcal{X}))$ . The convex surrogate of the multi-rank is suggested as follows:

---

### Algorithm 1 The t-SVD for a third-order tensor.

---

**Input:**  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ .

- 1:  $\tilde{\mathcal{X}} \leftarrow \text{fft}(\mathcal{X}, [], 3)$ .
- 2: **for**  $i = 1$  to  $n_3$  **do**
- 3:    $[U, S, V] = \text{svd}(\tilde{\mathcal{X}}^{(i)})$ .
- 4:    $\tilde{U}^{(i)} \leftarrow U$ ;  $\tilde{S}^{(i)} \leftarrow S$ ;  $\tilde{V}^{(i)} \leftarrow V$ .
- 5: **end for**
- 6:  $\mathcal{U} \leftarrow \text{ifft}(\tilde{U}, [], 3)$ .
- 7:  $\mathcal{S} \leftarrow \text{ifft}(\tilde{S}, [], 3)$ .
- 8:  $\mathcal{V} \leftarrow \text{ifft}(\tilde{V}, [], 3)$ .

**Output:**  $\mathcal{U}, \mathcal{S}, \mathcal{V}$ .

---

**Definition 4** (TNN [28]). The tensor nuclear norm of a tensor  $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ , denoted as  $\|\mathcal{X}\|_{\text{TNN}}$ , is defined as the sum of singular values of all the frontal slices of  $\tilde{\mathcal{X}}$ , i.e.,

$$\|\mathcal{X}\|_{\text{TNN}} := \sum_{i=1}^{n_3} \|\tilde{\mathcal{X}}^{(i)}\|_*, \quad (7)$$

where  $\tilde{\mathcal{X}}^{(i)}$  is the  $i$ -th frontal slice of  $\tilde{\mathcal{X}}$  and  $\tilde{\mathcal{X}} = \text{fft}(\mathcal{X}, [], 3)$ .

## 2.3. Plug-and-Play (PnP) framework

PnP is a highly flexible framework [42–44] that leverages the power of state-of-the-art denoisers in ADMM or other proximal algorithms. After variable splitting technique, the optimization problem is decoupled into easier subproblems, one of which is the proximal operator of regularization. The proximal operator of regularization  $\text{prox}_\Phi: \mathbb{R}^n \rightarrow \mathbb{R}^n$  is defined as

$$\text{prox}_\Phi(y) = \arg \min_x \left\{ \Phi(x) + \frac{\rho}{2} \|x - y\|^2 \right\}, \quad (8)$$

which maps the input  $y$  to the minimizer of (8). Under the PnP framework, the proximal operator of regularization is replaced by the denoising algorithm (termed as denoiser), which maps the noisy image to the clean image. Here the denoiser serves as an implicit regularizer to express image priors. Regularization by Denoising (RED) [45,46], which uses the denoising engine in defining the regularization of the inverse problem, is suggested as an alternative framework.

In the past decades, the hand-crafted denoisers, which exploit the deterministic priors of the nature images, have been dominantly used, e.g., TV denoiser [47], framelet denoiser [48], BM3D denoiser [49], WNNM denoiser [50], and ITS denoiser [51]. Recently, deep learning-based denoisers have been rapidly developed to learn data-driven image priors from a large number of natural images [38–40] and have shown promising performance. The state-of-the-art denoisers can be flexibly plugged as a modular part under the PnP framework. The PnP framework has shown great empirical success on diverse applications including denoising [1,52], restoration [53], tomography, super-resolution, and fusion.

However, even the convergence of algorithms under the PnP framework is still an open problem. The convergence of ADMM under the PnP framework has been considered from the fixed-point viewpoint in [44]. Very recently, the convergence of ADMM under the PnP framework with properly trained denoisers has been discussed in [54].

## 3. The proposed model and algorithm

As previously discussed, given an incomplete observation  $\mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$  with the observed index set  $\Omega$ , our DP3LRTC model is formulated as:

$$\arg \min_{\mathcal{X}} \|\mathcal{X}\|_{\text{TNN}} + \lambda \Phi(\mathcal{X}) \quad \text{s.t.} \quad \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{O}). \quad (9)$$

In our model,  $\Phi(\mathcal{X})$  is an implicit regularizer, whose related subproblem will be tackled by denoising CNNs within the PnP framework. Here we make some remarks about our model. When using CNNs, a straightforward idea is to design a network structure and conduct end-to-end training. However, this may lack flexibility and generalization ability for different kinds of data and various sampling scenarios. Therefore, in our model, we directly plug in a 2D denoising CNN, which is trained on the natural image dataset, to solve the  $\Phi(\mathcal{X})$  related subproblem. Specifically, the denoising CNN hereof acts on the spatial slices of the multi-dimensional images, for example bands of multi-spectral images (MSIs) and frames of videos, to preserve the spatial details. Meanwhile, the whole structure of the data is captured by the low-rank term

$\|\mathcal{X}\|_{\text{TNN}}$ , which would dominate the correlation along the temporal or spectral direction. Thus, our model can deal with different types of multi-dimensional images and various sampling scenarios in a flexible and economical manner.

Now, we can develop the ADMM [41] to tackle the minimization problem (11) under the PnP framework. First, we denote the indicator function as

$$1_{\mathbb{S}}(\mathcal{X}) = \begin{cases} 0, & \text{if } \mathcal{X} \in \mathbb{S}, \\ \infty, & \text{otherwise,} \end{cases} \quad (10)$$

where  $\mathbb{S} := \{\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{O})\}$  and introduce two auxiliary variables  $\mathcal{Y}$  and  $\mathcal{Z}$ . Then, we reformulate (4) as a constrained optimization problem, i.e.,

$$\arg \min_{\mathcal{X}, \mathcal{Y}, \mathcal{Z}} \|\mathcal{Y}\|_{\text{TNN}} + \lambda \Phi(\mathcal{Z}) + 1_{\mathbb{S}}(\mathcal{X}) \quad (11)$$

s.t.  $\mathcal{Y} = \mathcal{X}, \mathcal{Z} = \mathcal{X}$ .

The augmented Lagrangian function of (11) is

$$\begin{aligned} & \|\mathcal{Y}\|_{\text{TNN}} + \lambda \Phi(\mathcal{Z}) + 1_{\mathbb{S}}(\mathcal{X}) + \langle \mathcal{X} - \mathcal{Y}, \Lambda_1 \rangle + \frac{\beta}{2} \|\mathcal{X} - \mathcal{Y}\|_F^2 \\ & + \langle \mathcal{X} - \mathcal{Z}, \Lambda_2 \rangle + \frac{\beta}{2} \|\mathcal{X} - \mathcal{Z}\|_F^2, \end{aligned} \quad (12)$$

where  $\Lambda_1$  and  $\Lambda_2$  are the Lagrangian multipliers and  $\beta$  is a non-negative penalty parameter.

According to the ADMM framework, the solution of (4) can be obtained by solving a sequence of subproblems.

In Step 1, we need to solve the  $[\mathcal{Y}, \mathcal{Z}]$ -subproblem. Since the variables  $\mathcal{Y}$  and  $\mathcal{Z}$  are decoupled, their optimal solutions can be calculated separately.

1) The  $\mathcal{Y}$ -subproblem is

$$\arg \min_{\mathcal{Y}} \|\mathcal{Y}\|_{\text{TNN}} + \frac{\beta}{2} \|\mathcal{X}^l - \mathcal{Y} + \Lambda_1^l/\beta\|_F^2. \quad (13)$$

Let the t-SVD of  $(\mathcal{X}^l + \Lambda_1^l/\beta)$  be  $\mathcal{U} * \mathcal{S} * \mathcal{V}^H$ , the closed-form solution of the  $\mathcal{Y}$ -subproblem can be exactly calculated via tensor singular value thresholding (t-SVT) [55,56] as

$$\mathcal{Y}^{l+1} = \mathcal{U} * \mathcal{D} * \mathcal{V}^H, \quad (14)$$

where  $\mathcal{D}$  is an  $f$ -diagonal tensor whose each frontal slice in the Fourier domain is  $\tilde{\mathcal{D}}(i, i, k) = \max\{\tilde{\mathcal{S}}(i, i, k) - \frac{1}{\beta}, 0\}$ . The complexity of computing  $\mathcal{Y}$  is  $O(n_3 \min(n_1^2 n_2, n_1 n_2^2) + n_1 n_2 n_3 \log(n_3))$ .

2) The  $\mathcal{Z}$ -subproblem is

$$\arg \min_{\mathcal{Z}} \lambda \Phi(\mathcal{Z}) + \frac{\beta}{2} \|\mathcal{X}^l - \mathcal{Z} + \Lambda_2^l/\beta\|_F^2. \quad (15)$$

Let  $\sigma = \sqrt{\lambda/\beta}$ , then (15) can be rewritten as

$$\text{prox}_{\Phi}(\mathcal{Z}) = \arg \min_{\mathcal{Z}} \Phi(\mathcal{Z}) + \frac{1}{2\sigma^2} \|\mathcal{Z} - \mathcal{X}^l - \Lambda_2^l/\beta\|_F^2. \quad (16)$$

Under the PnP framework, the proximal operator of regularization  $\text{prox}_{\Phi} : \mathbb{R}^{n_1 \times n_2 \times n_3} \rightarrow \mathbb{R}^{n_1 \times n_2 \times n_3}$  is replaced by the deep learning-based denoiser, which maps the noisy image to the clean image. Here we consider the fast and flexible denoising convolutional neural network, namely FFDNet [39], as the deep denoiser. Feeding  $\mathcal{X}^l + \Lambda_2^l/\beta$  into the denoiser FFDNet, we obtain the solution of the  $\mathcal{Z}$ -subproblem as

$$\mathcal{Z}^{l+1} = \text{FFDNet}(\mathcal{X}^l + \Lambda_2^l/\beta, \sigma). \quad (17)$$

In FFDNet, the parameter  $\sigma$  is related to the noise level, but here  $\sigma$  is related to the error level between the estimation and ground truth. We feed the color images into the pre-trained FFDNet for color images and the spatial slices of grayscale videos and MSIs into the pre-trained FFDNet for grayscale images. The complexity of computing  $\mathcal{Z}$  is  $O(n_1 n_2 n_3 n_l n_f n_k)$ , where  $n_l$  is the number of layers,  $n_f$  is the number of features, and  $n_k$  is the number of kernel pixels.

### Algorithm 2 The ADMM algorithm for solving (4).

**Input:** the observed tensor  $\mathcal{O}$ , the observed index set  $\Omega$ , the parameters  $\beta$  and  $\sigma$ , and the maximum iteration number  $l_{\max}$ .

1: **Initialization:** Let  $\mathcal{Y}$ ,  $\mathcal{Z}$ , and  $\mathcal{X}$  be observed tensors, let  $\Lambda_1$  and  $\Lambda_2$  be zero tensors.

2: **while** not converged and  $l \leq l_{\max}$  **do**

3:     Updating  $\mathcal{Y}$  via (14),

4:     Updating  $\mathcal{Z}$  via (17),

5:     Updating  $\mathcal{X}$  via (19),

6:     Updating multipliers  $\Lambda_1$  and  $\Lambda_2$  via (20).

7: **end while**

**Output:** the recovered tensor  $\mathcal{X}$ .

In Step 2, we need to solve the  $\mathcal{X}$ -subproblem:

$$\begin{aligned} & \arg \min_{\mathcal{X}} 1_{\mathbb{S}}(\mathcal{X}) + \frac{\beta}{2} \|\mathcal{X} - \mathcal{Y}^{l+1} + \Lambda_1^l/\beta\|_F^2 + \frac{\beta}{2} \|\mathcal{X} - \mathcal{Z}^{l+1} \\ & + \Lambda_2^l/\beta\|_F^2. \end{aligned} \quad (18)$$

By minimizing the  $\mathcal{X}$ -subproblem, we have  $1_{\mathbb{S}}(\mathcal{X}) = 0$  i.e.,  $\mathcal{X} \in \mathbb{S}$ . Thus, the closed-form solution of  $\mathcal{X}$ -subproblem is given as follows:

$$\begin{cases} \mathcal{P}_{\Omega}(\mathcal{X}^{l+1}) = \mathcal{P}_{\Omega}(\mathcal{O}), \\ \mathcal{P}_{\Omega^c}(\mathcal{X}^{l+1}) = \mathcal{P}_{\Omega^c}\left(\frac{\beta \mathcal{Y}^{l+1} + \beta \mathcal{Z}^{l+1} - \Lambda_1^l - \Lambda_2^l}{2\beta}\right), \end{cases} \quad (19)$$

where  $\Omega^c$  denotes the complementary set of  $\Omega$ . The complexity of computing  $\mathcal{X}$  is  $O(n_1 n_2 n_3)$ .

In Step 3, we update the multipliers  $\Lambda_1$  and  $\Lambda_2$  as follows:

$$\begin{cases} \Lambda_1^{l+1} = \Lambda_1^l + \beta(\mathcal{X}^{l+1} - \mathcal{Y}^{l+1}), \\ \Lambda_2^{l+1} = \Lambda_2^l + \beta(\mathcal{X}^{l+1} - \mathcal{Z}^{l+1}). \end{cases} \quad (20)$$

The complexity of updating  $\Lambda_1$  and  $\Lambda_2$  is  $O(n_1 n_2 n_3)$ .

Finally, the ADMM algorithm is summarized in Algorithm 2. The computational complexity of the ADMM algorithm is  $O(n_3 \min(n_1^2 n_2, n_1 n_2^2) + n_1 n_2 n_3 \log(n_3) + n_1 n_2 n_3 n_l n_f n_k)$ . The main part of the ADMM algorithm is implemented on CPU while the pre-trained FFDNet is evaluated on GPU.

## 4. Numerical experiments

In this section, the performance of DP3LRTC will be comprehensively evaluated on color images, grayscale videos, and MSIs. Although the FFDNet denoiser is pre-trained for grayscale images or color images, the DP3LRTC can be well generalized to videos and MSIs. The DP3LRTC is compared with SNN [18], TNN [28], and TNN-3DTV [37].

The peak signal to noise ratio (PSNR) and the structural similarity index (SSIM) [57] are chosen as the quality metrics. We report the mean PSNR and mean SSIM of all bands (or frames). The relative change (RelCha) is adopted as the stopping criterion of all methods, which is defined as

$$\text{RelCha} = \frac{\|\mathcal{X}^{l+1} - \mathcal{X}^l\|_F}{\|\mathcal{X}^l\|_F}.$$

In all experiments, when  $\text{RelCha}$  is smaller than the tolerance  $10^{-4}$ , we stop the iterations. The parameters of different methods are manually selected from the candidate set  $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2\}$  to obtain the highest PSNR value. In this paper, we mainly consider the element-wise sampling, i.e., entries are randomly sampled for each band (or frame). For color images, we consider the element-wise sampling, the tubal sampling, i.e., entries are sampled along all RGB channels for

**Table 1**

Quantitative comparison of the results by SNN [18], TNN [28], TNN-3DTV [37], and DP3LRTC on color images. The **best** and second best values are highlighted in bold and underlined, respectively.

Image	SR	PSNR				SSIM				Time (s)			
		SNN	TNN	TNN-3DTV	DP3LRTC	SNN	TNN	TNN-3DTV	DP3LRTC	SNN	TNN	TNN-3DTV	DP3LRTC
<i>Starfish</i> 321 × 481 × 3	10%	18.48	19.47	<u>22.59</u>	<b>27.43</b>	0.3617	0.3007	<u>0.6345</u>	<b>0.8270</b>	<u>6</u>	<u>8</u>	38	17
	20%	22.27	22.55	<u>24.85</u>	<b>31.25</b>	0.5476	0.5052	<u>0.7519</u>	<b>0.9143</b>	<u>6</u>	<u>8</u>	38	17
	30%	24.66	25.71	<u>26.52</u>	<b>34.38</b>	0.6883	0.6685	<u>0.8217</u>	<b>0.9533</b>	<u>5</u>	<u>9</u>	40	15
<i>Airplane</i> 512 × 512 × 3	10%	20.91	21.94	<u>23.46</u>	<b>28.48</b>	0.6706	0.6623	<u>0.8414</u>	<b>0.9471</b>	<u>10</u>	<u>17</u>	71	54
	20%	24.81	25.17	<u>25.83</u>	<b>30.57</b>	0.8279	0.8242	<u>0.9220</u>	<b>0.9709</b>	<u>12</u>	<u>19</u>	76	33
	30%	27.20	27.90	<u>28.45</u>	<b>31.36</b>	0.9060	0.9014	<u>0.9531</u>	<b>0.9806</b>	<u>10</u>	<u>18</u>	75	30
<i>Baboon</i> 512 × 512 × 3	10%	17.43	17.83	<u>19.56</u>	<b>21.68</b>	0.4077	0.3959	<u>0.5835</u>	<b>0.7798</b>	<u>11</u>	<u>17</u>	71	30
	20%	19.34	20.07	<u>20.67</u>	<b>23.44</b>	0.5974	0.6005	<u>0.7271</u>	<b>0.8433</b>	<u>9</u>	<u>18</u>	77	34
	30%	21.17	21.66	<u>21.93</u>	<b>24.06</b>	0.7308	0.7346	<u>0.8083</u>	<b>0.9118</b>	<u>8</u>	<u>19</u>	80	60
<i>Fruits</i> 512 × 512 × 3	10%	20.73	20.72	<u>24.81</u>	<b>31.47</b>	0.6046	0.5646	<u>0.8363</u>	<b>0.9362</b>	<u>10</u>	<u>17</u>	72	54
	20%	24.23	24.23	<u>27.31</u>	<b>34.90</b>	0.7777	0.7510	<u>0.9122</u>	<b>0.9621</b>	<u>10</u>	<u>18</u>	75	57
	30%	26.90	26.99	<u>29.22</u>	<b>36.48</b>	0.8689	0.8541	<u>0.9434</u>	<b>0.9723</b>	<u>9</u>	<u>19</u>	77	44
<i>Lena</i> 512 × 512 × 3	10%	21.43	21.89	<u>25.96</u>	<b>30.93</b>	0.6415	0.6177	<u>0.8396</u>	<b>0.9241</b>	<u>10</u>	<u>17</u>	71	54
	20%	24.98	25.68	<u>28.41</u>	<b>32.92</b>	0.8034	0.7888	<u>0.9069</u>	<b>0.9423</b>	<u>10</u>	<u>19</u>	77	34
	30%	27.71	28.06	<u>30.07</u>	<b>33.46</b>	0.8844	0.8719	<u>0.9380</u>	<b>0.9575</b>	<u>10</u>	<u>18</u>	75	54
<i>Watch</i> 768 × 1024 × 3	10%	22.47	23.01	<u>26.27</u>	<b>33.96</b>	0.7128	0.7490	<u>0.8863</u>	<b>0.9825</b>	<u>39</u>	<u>58</u>	250	143
	20%	25.64	26.61	<u>28.46</u>	<b>38.18</b>	0.8641	0.8923	<u>0.9466</u>	<b>0.9888</b>	<u>38</u>	<u>62</u>	255	141
	30%	28.37	29.70	<u>30.31</u>	<b>40.89</b>	0.9332	0.9502	<u>0.9709</u>	<b>0.9971</b>	<u>36</u>	<u>62</u>	262	140
<i>Opera</i> 586 × 695 × 3	10%	24.23	25.05	<u>25.56</u>	<b>30.51</b>	0.7499	0.7486	<u>0.8075</u>	<b>0.9300</b>	<u>17</u>	<u>28</u>	121	87
	20%	27.55	28.24	<u>29.13</u>	<b>33.57</b>	0.8649	0.8734	<u>0.9132</u>	<b>0.9628</b>	<u>15</u>	<u>29</u>	124	89
	30%	29.80	30.89	<u>31.14</u>	<b>35.90</b>	0.9188	0.9323	<u>0.9447</u>	<b>0.9812</b>	<u>15</u>	<u>34</u>	124	48
<i>Water</i> 768 × 1024 × 3	10%	20.20	21.00	<u>22.57</u>	<b>26.17</b>	0.5860	0.6126	<u>0.7756</u>	<b>0.9223</b>	<u>38</u>	<u>64</u>	254	180
	20%	22.75	23.37	<u>24.29</u>	<b>28.74</b>	0.7790	0.6685	<u>0.8822</u>	<b>0.9472</b>	<u>34</u>	<u>60</u>	256	103
	30%	24.67	25.76	<u>25.88</u>	<b>30.65</b>	0.8767	0.8922	<u>0.9302</u>	<b>0.9806</b>	<u>31</u>	<u>61</u>	262	142
Average	10%	20.74	21.36	<u>23.85</u>	<b>28.83</b>	0.5919	0.5814	<u>0.7756</u>	<b>0.9061</b>	<u>18</u>	<u>28</u>	119	81
	20%	23.95	24.49	<u>26.12</u>	<b>31.70</b>	0.7578	0.7380	<u>0.8703</u>	<b>0.9415</b>	<u>17</u>	<u>29</u>	122	68
	30%	26.31	27.08	<u>27.94</u>	<b>33.40</b>	0.8509	0.8507	<u>0.9138</u>	<b>0.9668</b>	<u>16</u>	<u>30</u>	124	72

the spatial location, and the Bayer sampling, i.e., each  $2 \times 2 \times 3$  cell contains two green pixels, one blue pixel, and one red pixel.

All the experiments are implemented on Windows 10 and Matlab (R2018a) with an Intel(R) Core(TM) i5-4590 CPU at 3.30 GHz, 16 GB RAM, and NVIDIA GeForce GTX 1060 6GB.

#### 4.1. Color image completion

In this subsection, 8 color images<sup>1</sup> are selected for the color images completion experiments. Each image is rescaled to [0,1], and the observed elements are element-wisely sampled with different sampling rates (SR). For color images, we use the pre-trained FFDNet on color images as the deep denoiser.

Table 1 presents the PSNR values, SSIM values, and running time of recovered results by different methods. From Table 1, it can be observed that the TNN-based method obtains better performance than SNN. Since TNN-3DTV introduces additional prior knowledge, it outperforms the TNN-based method and achieves the second-best PSNR and SSIM values. The proposed DP3LRTC obtains the highest PSNR and SSIM values. Meanwhile, it is noteworthy that the average margins between the results by DP3LRTC and TNN-3DTV are more than 5.34 dB in PSNR and 0.08 in SSIM.

For visualization, we exhibit the recovered images with the sampling rate 30% in Fig. 2. From Fig. 2, we can find that the SNN-based method and TNN-based method are able to recover the general structure of the color images, while the details are unclear and the results remain artifacts, which can be seen in the enlarged areas. The results by TNN-3DTV are relatively better. However, we can observe the over-smoothed phenomena in the enlarged areas. The recovered results by the proposed method are of high quality, being highly clear and very similar to the original images.

Now, we consider the structural samplings, i.e., inpainting and demosaicing, which are more difficult than the element-wise sampling. In image inpainting, we consider the tubal sampling (i.e., entries are sampled along all RGB channels for the spatial location) with three masks: letters, graffiti, and grid. In demosaicing, we consider the Bayer pattern sampling (i.e., each two-by-two cell contains two green, one blue, and one red). Fig. 3 shows the inpainting and demosaicing results by different methods on color images: *Fruits*, *Lena*, *Sailboat*, and *Baboon*. It is easy to observe that the proposed method is superior to competing methods in inpainting and demosaicing experiments.

#### 4.2. Video completion

In this subsection, 5 grayscale videos<sup>2</sup> with different sizes are selected. For grayscale videos, we feed the spatial slices into the pre-trained FFDNet on grayscale images as the deep denoiser. Table 2 lists the PSNR values, SSIM values, and running time of the recovered videos by different methods with different sampling rates. It is easy to observe that DP3LRTC obtains the results with the highest quality metrics. For visualization, we show the 20-th frame of the recovered videos with the sampling rate 10% in Fig. 4. We can see that the results recovered by DP3LRTC are of higher quality than the results obtained by compared methods.

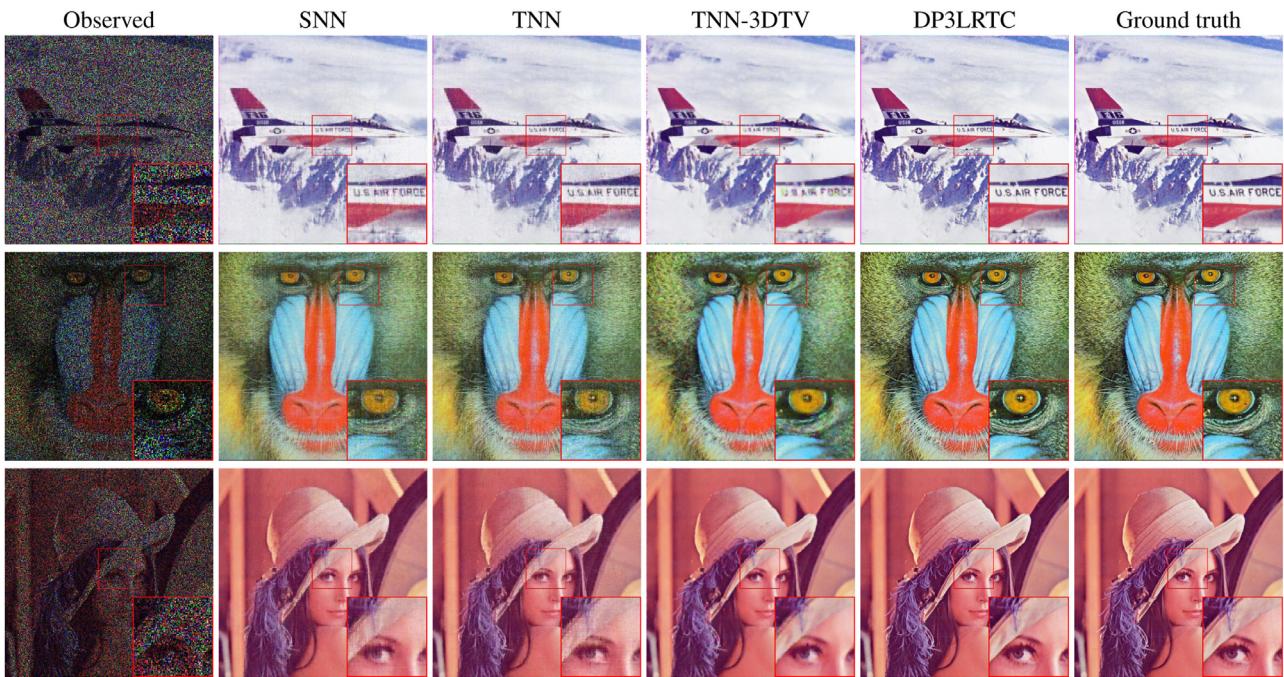
#### 4.3. MSI Completion

In this subsection, we test 7 MSIs from the CAVE dataset<sup>3</sup> which are of size  $256 \times 256 \times 31$  with the wavelengths in the range of 400 ~ 700 nm at an interval of 10 nm. For MSIs, we feed the spatial

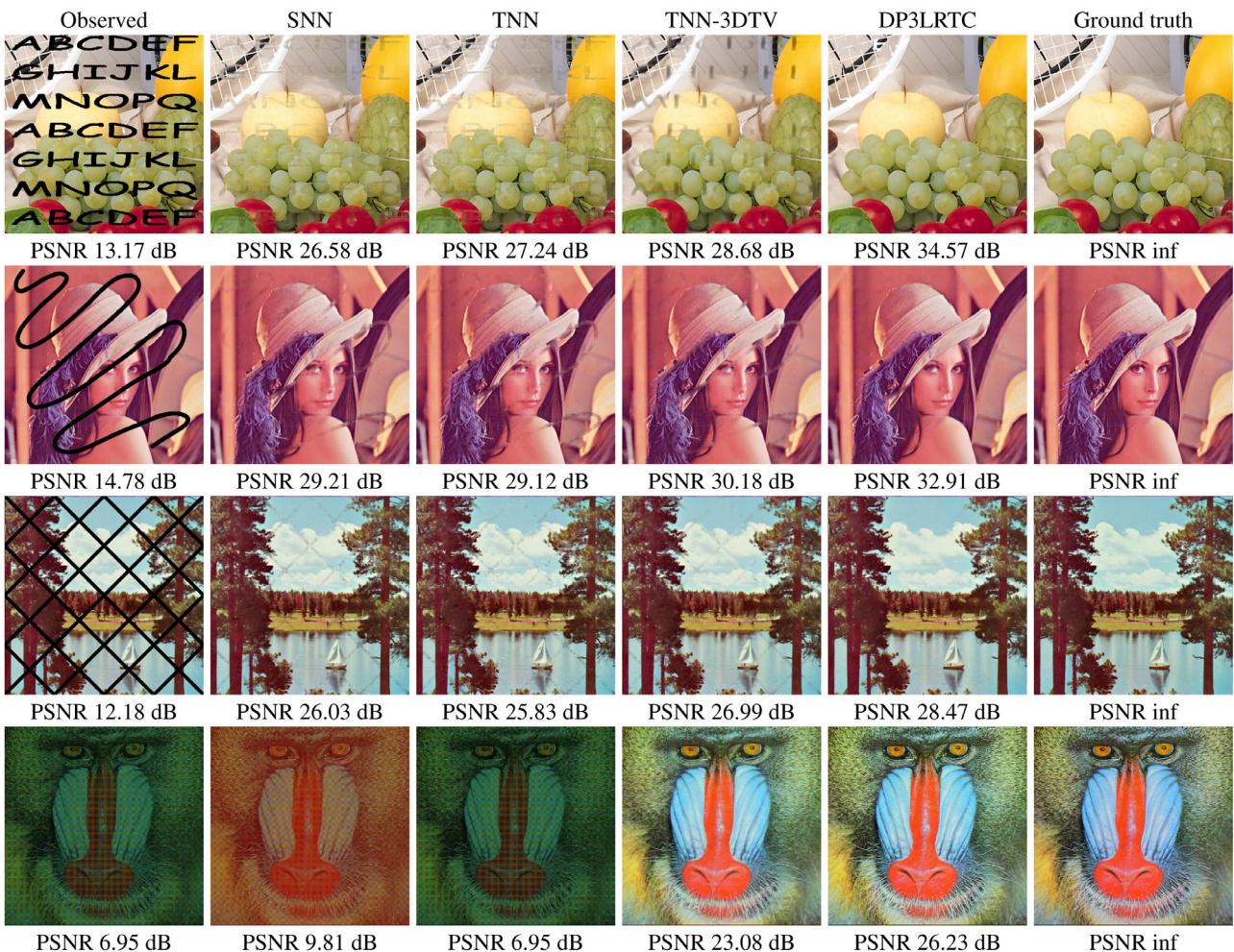
<sup>1</sup> Available at <http://sipi.usc.edu/database/database.php>.

<sup>2</sup> Available at <http://trace.eas.asu.edu/yuv/>.

<sup>3</sup> Available at <http://www.cs.columbia.edu/CAVE/databases/multispectral/>.



**Fig. 2.** The recovered color images by SNN [18], TNN [28], TNN-3DTV [37], and DP3LRTC with the sampling rate 30%.

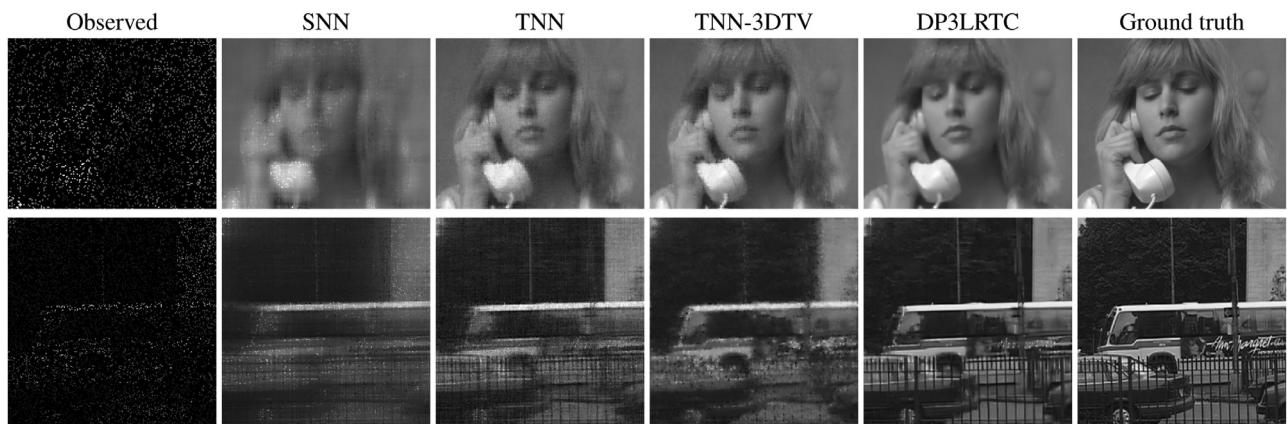


**Fig. 3.** Inpainting (top three rows) and demosaicing (bottom row) results by SNN [18], TNN [28], TNN-3DTV [37], and DP3LRTC on color images.

**Table 2**

Quantitative comparison of the results by SNN [18], TNN [28], TNN-3DTV [37], and DP3LRTC on videos. The **best** and second best values are highlighted in bold and underlined, respectively.

Video	SR	PSNR				SSIM				Time (s)			
		SNN	TNN	TNN-3DTV	DP3LRTC	SNN	TNN	TNN-3DTV	DP3LRTC	SNN	TNN	TNN-3DTV	DP3LRTC
Akiyo 144 × 176 × 30	5%	20.18	28.33	28.76	<b>30.59</b>	0.5976	0.8630	0.8949	<b>0.9294</b>	5	20	56	54
	10%	23.64	31.16	<u>31.97</u>	<b>33.33</b>	0.7340	0.9272	<u>0.9452</u>	<b>0.9623</b>	3	<u>20</u>	56	51
	20%	27.54	34.82	<u>36.06</u>	<b>36.96</b>	0.8642	0.9674	<u>0.9777</u>	<b>0.9816</b>	3	<u>20</u>	57	52
Suzie 144 × 176 × 30	5%	20.84	26.37	<u>27.39</u>	<b>29.37</b>	0.5944	0.7203	<u>0.7989</u>	<b>0.8373</b>	4	<u>19</u>	56	51
	10%	24.40	28.41	<u>29.27</u>	<b>31.80</b>	0.7046	0.7976	<u>0.8513</u>	<b>0.8899</b>	3	<u>18</u>	56	51
	20%	28.12	31.14	<u>31.95</u>	<b>34.75</b>	0.8189	0.8739	0.9123	<b>0.9358</b>	2	<u>18</u>	57	52
Container 144 × 176 × 30	5%	19.81	26.27	<u>26.63</u>	<b>26.87</b>	0.6446	0.8305	<u>0.8611</u>	<b>0.8695</b>	4	<u>19</u>	56	51
	10%	22.26	29.53	<u>30.05</u>	<b>30.44</b>	0.7413	0.9023	<u>0.9272</u>	<b>0.9230</b>	3	<u>19</u>	56	51
	20%	25.56	33.65	<u>34.69</u>	<b>34.82</b>	0.8495	0.9553	<u>0.9677</u>	<b>0.9654</b>	2	<u>19</u>	57	52
News 144 × 176 × 30	5%	18.34	26.65	<u>27.20</u>	<b>28.29</b>	0.5610	0.8182	<u>0.8738</u>	<b>0.9023</b>	4	<u>20</u>	56	51
	10%	21.47	30.31	<u>30.56</u>	<b>31.96</b>	0.6955	0.9106	<u>0.9285</u>	<b>0.9473</b>	3	<u>20</u>	57	51
	20%	25.00	33.82	<u>34.21</u>	<b>35.55</b>	0.8256	0.9554	<u>0.9665</u>	<b>0.9740</b>	3	<u>21</u>	57	52
Bus 256 × 256 × 30	5%	15.79	18.30	<u>18.36</u>	<b>20.78</b>	0.3300	0.3331	<u>0.3973</u>	<b>0.5932</b>	11	<u>52</u>	185	145
	10%	17.70	19.61	<u>19.66</u>	<b>22.83</b>	0.4174	0.4421	<u>0.5083</u>	<b>0.7222</b>	9	<u>50</u>	185	146
	20%	19.86	21.74	<u>21.91</u>	<b>25.48</b>	0.5560	0.5993	<u>0.6778</u>	<b>0.8359</b>	5	<u>50</u>	187	148
Average	5%	18.99	25.18	<u>25.67</u>	<b>27.18</b>	0.5455	0.7130	<u>0.7652</u>	<b>0.8263</b>	6	<u>26</u>	82	70
	10%	21.89	27.80	<u>28.30</u>	<b>30.07</b>	0.6586	0.7960	<u>0.8321</u>	<b>0.8889</b>	4	<u>25</u>	82	70
	20%	25.22	31.03	<u>31.76</u>	<b>33.51</b>	0.7828	0.8703	<u>0.9004</u>	<b>0.9385</b>	3	<u>26</u>	83	71



**Fig. 4.** The 20-th frames of the recovered videos by SNN, TNN, TNN-3DTV, and DP3LRTC with the sampling rate 10%.

slices into the FFDNet denoiser pre-trained on grayscale images as the PnP denoiser.

**Table 3** exhibits the PSNR values, SSIM values, and running time of all the results by different methods. The DP3LRTC achieves the highest PSNR and SSIM values, while the TNN-3DTV gets the second best. For visual inspection, **Fig. 5** displays the recovered MSIs (pseudo-color images composed of the 1st, 2nd, and 31-st bands) by different methods with the sampling rate 10%. From **Fig. 5**, it can be observed that the results recovered by DP3LRTC are of the highest quality. In **Fig. 6**, we display the selected mode-3 fibers of recovered MSIs *Balloons* by SNN, TNN, TNN-3DTV, and DP3LRTC with the sampling rate 10%. We can observe that the selected mode-3 fiber of the recovered MSI by DP3LRTC is closer to the ground truth as compared with those by SNN, TNN, and TNN-3DTV.

#### 4.4. Discussions

##### 4.4.1. Convergence behaviour

The numerical experiments have shown the great empirical success of DP3LRTC. However, it is still an open question whether the ADMM algorithm under the PnP framework has a good convergence behavior. In **Fig. 7**, we displayed the relative change curves of the ADMM algorithm (in the logarithmic scale) with respect to

iterations on different videos. We can evidently observe the numerical convergence of the ADMM algorithm.

##### 4.4.2. Contributions of different terms

To evaluate the distinct contributions of the low-rankness prior and deep image prior, we test the performance of TNN (only the TNN regularizer), DPR (only the implicit deep regularizer), and DP3LRTC (both the TNN and implicit deep regularizers). In **Fig. 8**, we exhibit the recovered results by TNN, DPR, and DP3LRTC on color image *Airplane* and MSI *Flowers*, together with PSNR values. In **Fig. 8**, the color image recovered result by DPR are clearer than that by TNN, being comparable to the result by DP3LRTC. This shows the remarkable ability of the deep denoiser to capture the spatial fine structures. For the recovery of the MSI, which consists of 31 spectral bands, TNN and DP3LRTC outperform DPR. This shows that the TNN regularizer is good at utilizing the spectral redundancy. Therefore, we can confirm that two regularizers are indispensable to DP3LRTC and benefit from each other. Together, they contribute to the superior performance of the proposed DP3LRTC.

To take a little more in-depth look at the mechanism of each term, we conduct experiments on all 7 MSIs in **Section 4.3** with different sampling rates. The averaged PSNR and SSIM values are reported in **Table 4**. Generally, the metric SSIM evaluates the

**Table 3**

Quantitative comparison of the results by SNN [18], TNN [28], TNN-3DTV [37], and DP3LRTC on MSIs. The **best** and second best values are highlighted in bold and underlined, respectively.

MSI	SR	PSNR				SSIM				Time (s)			
		SNN	TNN	TNN-3DTV	DP3LRTC	SNN	TNN	TNN-3DTV	DP3LRTC	SNN	TNN	TNN-3DTV	DP3LRTC
Balloons 256 × 256 × 31	5%	24.29	31.35	32.66	<b>39.04</b>	0.8232	0.8670	0.9381	<b>0.9843</b>	<b>8</b>	58	191	144
	10%	31.27	35.63	<u>37.66</u>	<b>42.98</b>	0.9242	0.9380	<u>0.9735</u>	<b>0.9937</b>	<u>7</u>	<u>61</u>	200	155
	20%	37.34	41.11	<u>42.77</u>	<b>47.54</b>	0.9737	0.9803	<u>0.9904</u>	<b>0.9975</b>	<u>7</u>	<u>59</u>	192	147
Beads 256 × 256 × 31	5%	16.50	19.86	<u>21.27</u>	<b>22.53</b>	0.3259	0.4569	<u>0.6506</u>	<b>0.6912</b>	<b>9</b>	<u>60</u>	199	147
	10%	17.99	22.92	<u>23.95</u>	<b>26.08</b>	0.4191	0.6512	<u>0.7907</u>	<b>0.8369</b>	<u>7</u>	<u>61</u>	198	153
	20%	21.34	27.61	<u>27.78</u>	<b>30.20</b>	0.6404	0.8420	<u>0.9024</u>	<b>0.9319</b>	<u>7</u>	<u>58</u>	197	152
Cd 256 × 256 × 31	5%	22.19	26.51	<u>29.35</u>	<b>30.22</b>	0.8497	0.8029	<u>0.9251</u>	<b>0.9597</b>	<b>10</b>	<u>62</u>	191	144
	10%	25.98	29.54	<u>31.92</u>	<b>35.30</b>	0.9018	0.8871	<u>0.9550</u>	<b>0.9790</b>	<b>10</b>	<u>69</u>	196	157
	20%	30.85	33.22	<u>36.04</u>	<b>39.35</b>	0.9498	0.9447	<u>0.9791</u>	<b>0.9890</b>	<b>8</b>	<u>64</u>	193	150
Clay 256 × 256 × 31	5%	28.83	33.62	<u>34.25</u>	<b>39.89</b>	0.9224	0.9052	<u>0.9487</u>	<b>0.9680</b>	<b>10</b>	<u>59</u>	194	145
	10%	35.78	38.22	<u>39.66</u>	<b>43.80</b>	0.9726	0.9657	<u>0.9819</u>	<b>0.9907</b>	<b>9</b>	<u>62</u>	193	144
	20%	41.58	43.69	<u>44.69</u>	<b>48.02</b>	0.9895	0.9813	<u>0.9890</u>	<b>0.9952</b>	<b>9</b>	<u>62</u>	191	147
Face 256 × 256 × 31	5%	25.64	32.32	<u>32.70</u>	<b>36.82</b>	0.8346	0.8981	<u>0.9381</u>	<b>0.9663</b>	<b>9</b>	<u>61</u>	198	146
	10%	30.51	36.50	<u>37.52</u>	<b>39.66</b>	0.9125	0.9548	<u>0.9736</u>	<b>0.9829</b>	<b>8</b>	<u>60</u>	192	145
	20%	36.20	41.33	<u>42.00</u>	<b>42.96</b>	0.9664	0.9842	<u>0.9778</u>	<b>0.9916</b>	<b>8</b>	<u>61</u>	192	147
Feathers 256 × 256 × 31	5%	20.45	27.55	<u>28.23</u>	<b>31.14</b>	0.6565	0.7694	<u>0.8712</u>	<b>0.9375</b>	<b>7</b>	<u>58</u>	196	151
	10%	24.58	31.19	<u>31.83</u>	<b>34.59</b>	0.7925	0.8717	<u>0.9308</u>	<b>0.9656</b>	<b>8</b>	<u>58</u>	193	147
	20%	29.30	35.86	<u>36.73</u>	<b>38.68</b>	0.8995	0.9481	<u>0.9726</u>	<b>0.9838</b>	<b>8</b>	<u>60</u>	193	150
Flowers 256 × 256 × 31	5%	20.47	26.75	<u>27.90</u>	<b>30.34</b>	0.6293	0.7153	<u>0.8288</u>	<b>0.8894</b>	<b>10</b>	<u>60</u>	196	148
	10%	24.88	30.36	<u>31.37</u>	<b>33.70</b>	0.7605	0.8357	<u>0.9025</u>	<b>0.9419</b>	<b>7</b>	<u>58</u>	192	147
	20%	29.49	35.33	<u>36.25</u>	<b>37.66</b>	0.8804	0.9342	<u>0.9621</u>	<b>0.9745</b>	<b>7</b>	<u>60</u>	194	149
Average	5%	22.62	28.28	<u>29.48</u>	<b>32.85</b>	0.7202	0.7735	<u>0.8715</u>	<b>0.9138</b>	<b>9</b>	<u>60</u>	195	146
	10%	27.28	32.05	<u>33.42</u>	<b>36.59</b>	0.8119	0.8720	<u>0.9297</u>	<b>0.9558</b>	<b>8</b>	<u>61</u>	195	150
	20%	32.30	36.88	<u>38.04</u>	<b>40.63</b>	0.9000	0.9450	<u>0.9676</u>	<b>0.9805</b>	<b>8</b>	<u>60</u>	193	149

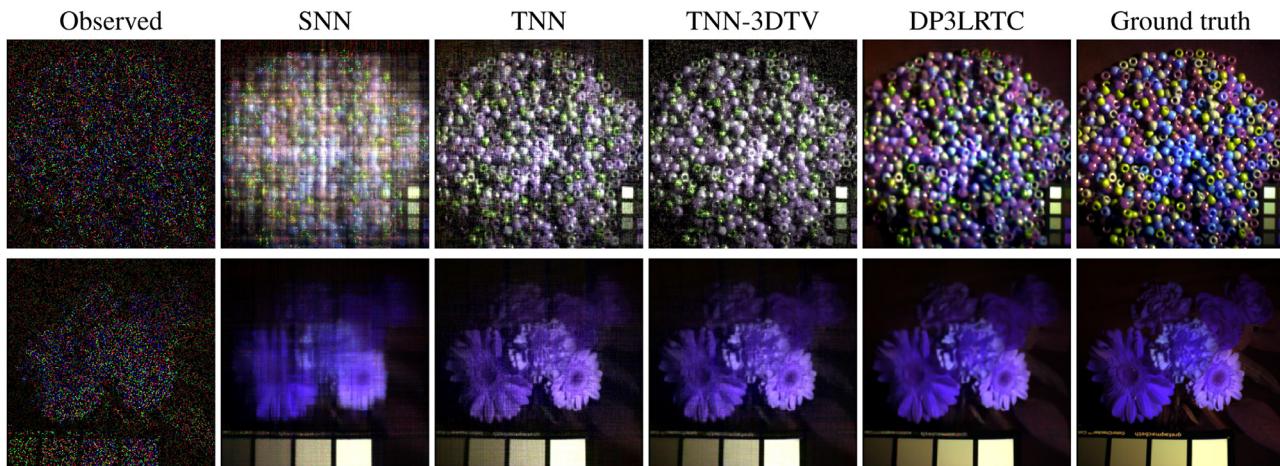


Fig. 5. The recovered MSIs (pseudo-color images composed of the 1st, 2nd, and 31-st bands) by SNN, TNN, TNN-3DTV, and DP3LRTC with the sampling rate 10%.

**Table 4**

Quantitative comparison of the results by TNN, DPR, and DP3LRTC on MSIs. The **best** values are highlighted in bold.

MSIs	SR	PSNR			SSIM		
		TNN	DPR	DP3LRTC	TNN	DPR	DP3LRTC
Average	5%	28.28	24.80	<b>32.85</b>	0.7735	0.8631	<b>0.9138</b>
	10%	32.05	30.64	<b>36.59</b>	0.8720	0.9111	<b>0.9558</b>
	20%	36.88	36.24	<b>40.63</b>	0.9443	0.9516	<b>0.9805</b>

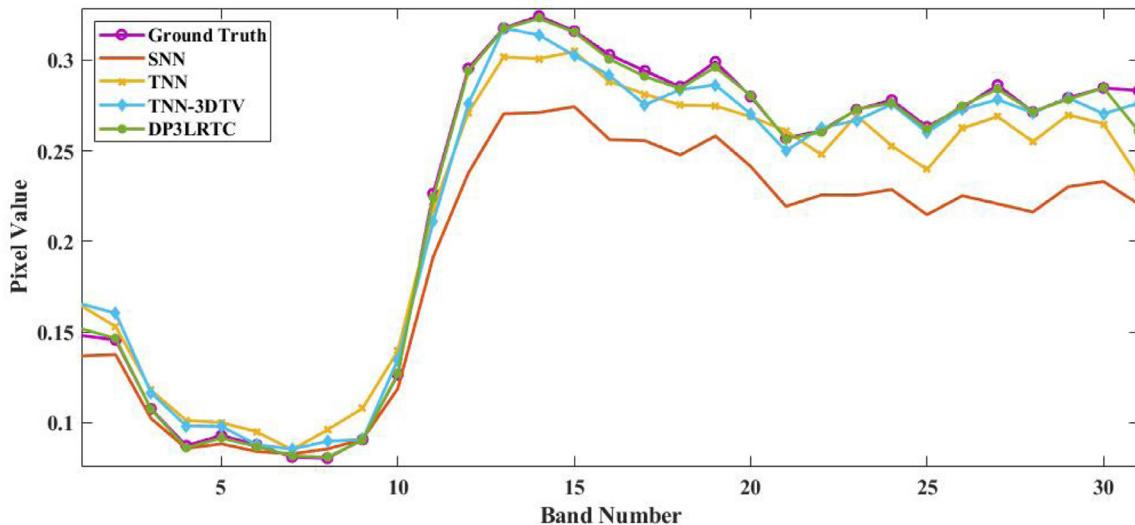
local structure similarity [57] and the metric PSNR reflects the overall recovery quality. The PSNR values of results by TNN are higher than those by DPR, while DPR achieves higher SSIM values. This phenomenon also reveals that the TNN regularizer is more suitable for characterizing the global information and the CNN denoiser is appropriate for recovering the fine structures. As

the sampling rate increases, the SSIM margin between TNN and DP3LRTC decreases from 0.140 to 0.036 while the PSNR margin is always around 4 dB. This shows that for low sampling rates, recovering the local structure becomes more difficult than capturing the global information.

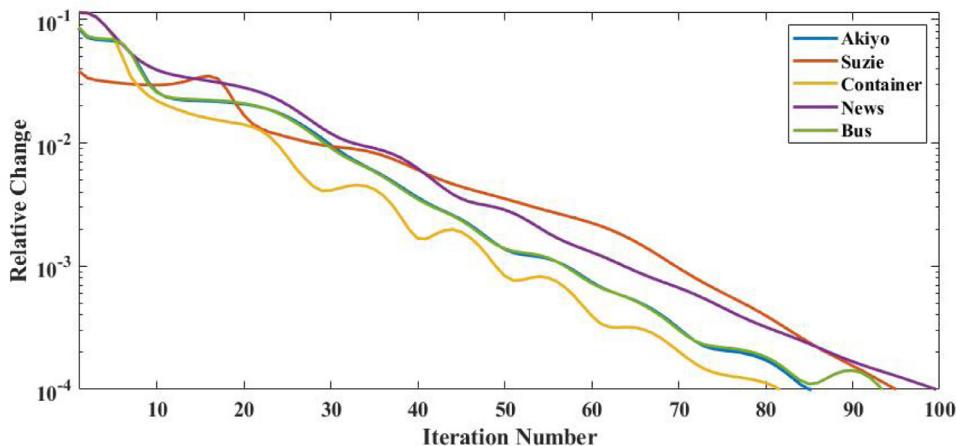
Moreover, we study the influence of parameters  $\beta$  and  $\sigma$ . In Fig. 9, we show the PSNR and SSIM values of the recovered color images *Starfish* by DP3LRTC with respect to  $\beta$  and  $\sigma$ . We can see that DP3LRTC delivers the best performance when  $\beta \in \{10^{-1}, 1\}$  and  $\sigma \in \{10^{-2}, 10^{-1}, 1\}$ .

#### 4.4.3. Low-matrix-rank and low-tensor-rank

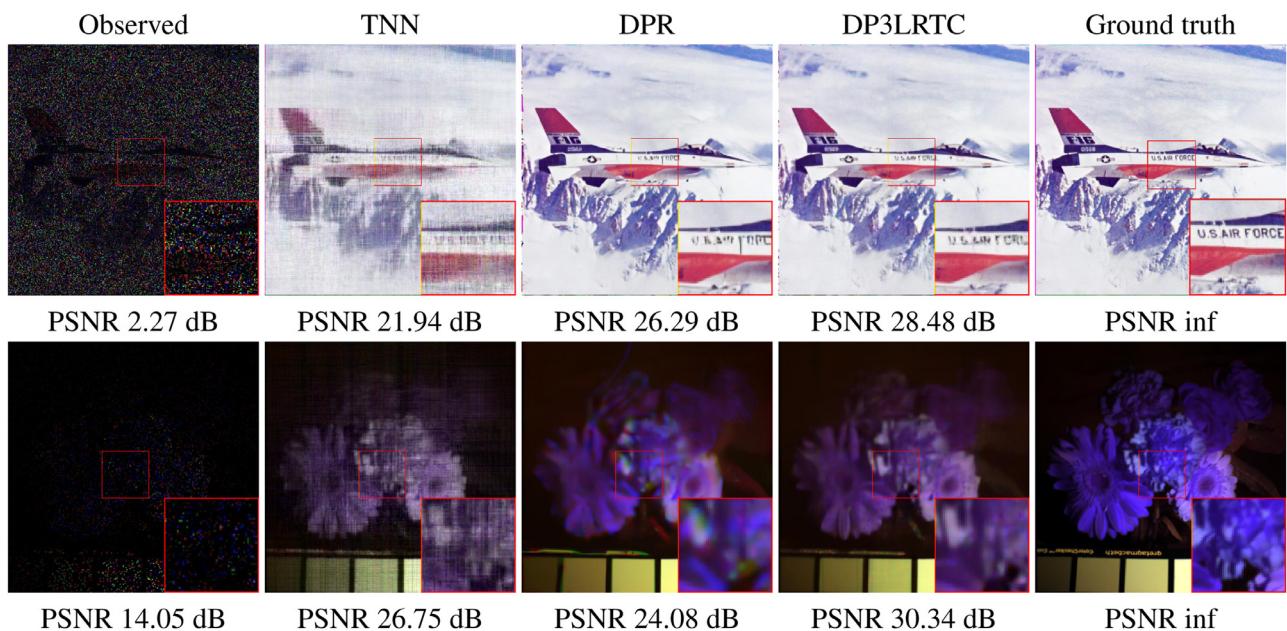
To illustrate the superiority of the tensor-rank-based model, we reported the quantitative comparison of the recovered results by our DP3LRTC and DP3LRC, in which the TNN is replaced by the matrix nuclear norm of the matricization of multi-dimensional



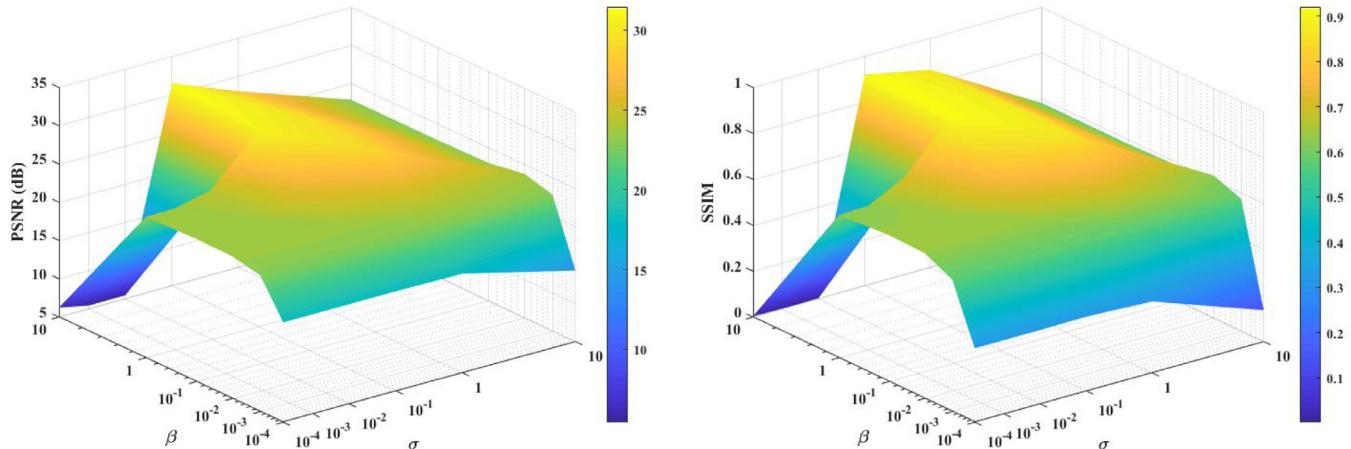
**Fig. 6.** The selected mode-3 fibers of recovered MSIs *Balloons* by SNN [18], TNN [28], TNN-3DTV [37], and DP3LRTC with the sampling rate 10%.



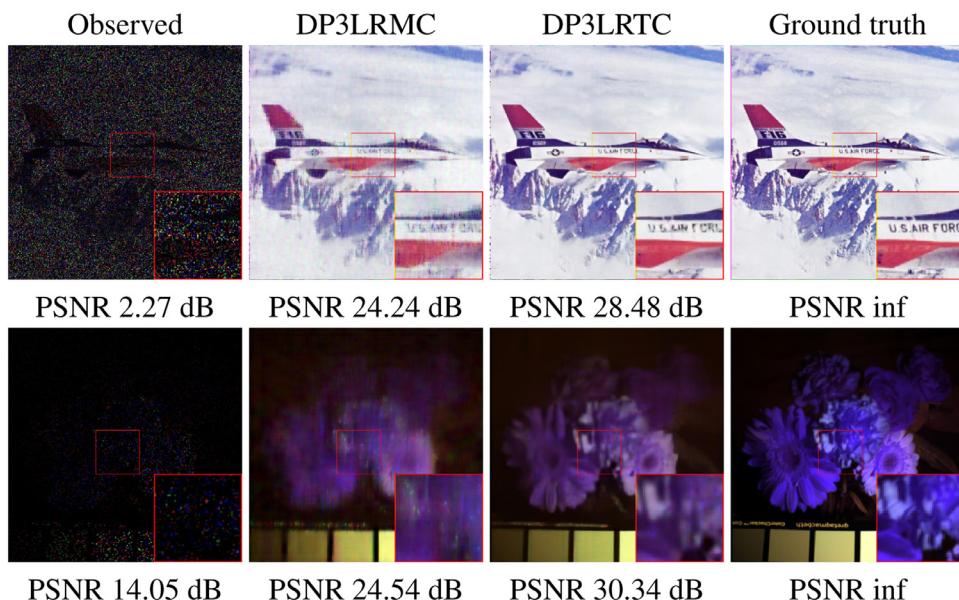
**Fig. 7.** The relative change curves of the ADMM algorithm on videos *Akiyo*, *Suzie*, *Container*, *News*, and *Bus* with the sampling rate 10%.



**Fig. 8.** First row: the recovered color images *Airplane* by TNN, DPR, and DP3LRTC with the sampling rate 10%. Second row: the recovered MSIs *Flowers* (pseudo-color images composed of the 1st, 2nd, and 31-st bands) by TNN, DPR, and DP3LRTC with the sampling rate 5%.



**Fig. 9.** The PSNR and SSIM values of the results recovered by DP3LRTC with respect to  $\beta$  and  $\sigma$  on the color image *Starfish* with the sampling rate 20%.



**Fig. 10.** First row: the recovered color images *Airplane* results by DP3LRMC and DP3LRTC with the sampling rate 10%. Second row: the recovered MSIs *Flowers* (pseudo-color images composed of the 1st, 2nd, and 31-th bands) by DP3LRMC and DP3LRTC with the sampling rate 5%.

**Table 5**  
Quantitative comparison of the results by DP3LRMC and DP3LRTC on color images and MSIs. The **best** are highlighted in bold.

	SR	PSNR		SSIM	
		DP3LRMC	DP3LRTC	DP3LRMC	DP3LRTC
Color images	10%	23.89	<b>28.83</b>	0.7630	<b>0.9061</b>
	20%	26.76	<b>31.70</b>	0.8719	<b>0.9415</b>
	30%	29.07	<b>33.40</b>	0.9308	<b>0.9668</b>
MSIs	5%	25.76	<b>32.85</b>	0.6948	<b>0.9138</b>
	10%	28.75	<b>36.59</b>	0.7863	<b>0.9558</b>
	20%	32.27	<b>40.63</b>	0.8706	<b>0.9805</b>

images (where the rows are the vectorized spatial images and the columns are the spectral channels), on color images and MSIs in **Table 5**. The setup of experiments on color images and MSIs are the same as **Section 4.1** and **Section 4.3**, respectively. Similarly, we only report the average metrics of the results. For better visual inspection, we further displayed the recovered results by DP3LRMC and DP3LRTC in **Fig. 10**. From **Table 5** and **Fig. 10**, we confirmed

that the tensor-rank-based model DP3LRTC is evidently better than the matrix-rank-based model DP3LRMC.

#### 4.4.4. Beyond TNN, FFDNet, and third-order tensors

In this work, we suggested a user-friendly LRTC framework which integrates low-rankness prior and the deep image prior. **Table 6** shows the quantitative comparison of the results obtained by different combinations of low-rankness metrics and denoisers. The setup is the same as the color image completion with the element-wise sampling in **Section 4.1**. For saving space, we only report the average metrics of the results of 8 color images. In **Table 6**, “SNN-DPR” denotes the combination of SNN and the FFDNet denoiser, while “TNN-BM3D” indicates that TNN is combined with the BM3D denoiser [49]. From **Table 6**, we can generally observe that the combination of TNN and the FFDNet denoiser is best among all combinations. Thus, we choose the combination of TNN and the FFDNet denoiser as the representative combination of the DP3LRTC framework.

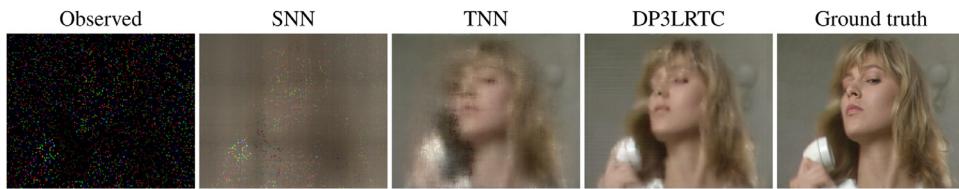
Next, we extend our DP3LRTC framework to high-order tensors ( $n \geq 3$ ) by replacing TNN with its high-order extension, i.e., the weighted sum of tensor nuclear norm (WSTNN) [27]. We

**Table 6**Quantitative comparison of the recovered results by SNN-DPR, TNN-BM3D, and DP3LRTC on color images. The **best** are highlighted in bold.

Color image	SR	PSNR			SSIM		
		SNN-DPR	TNN-BM3D	DP3LRTC	SNN-DPR	TNN-BM3D	DP3LRTC
Average	10%	28.55	27.80	<b>28.83</b>	0.9013	0.8754	<b>0.9061</b>
	20%	31.24	30.91	<b>31.70</b>	<b>0.9461</b>	0.9382	0.9415
	30%	32.59	33.25	<b>33.40</b>	0.9597	0.9641	<b>0.9668</b>

**Table 7**Quantitative comparison of the results by different methods on the color video *Suzie*. The **best** are highlighted in bold.

Color Video	SR	PSNR			SSIM		
		SNN	WSTNN	DP3LRTC	SNN	WSTNN	DP3LRTC
Suzie	1%	7.44	23.26	<b>24.96</b>	0.0372	0.6967	<b>0.7371</b>
	5%	17.97	28.24	<b>30.73</b>	0.5148	0.8384	<b>0.8891</b>
	10%	21.22	30.94	<b>33.47</b>	0.6139	0.8943	<b>0.9285</b>

Fig. 11. The 50-th frames of recovered color videos *Suzie* by SNN, TNN, and DP3LRTC with the sampling rate 5%.

conduct the experiment on a fourth-order tensor, i.e., the color video "Suzie"<sup>4</sup> of size  $144 \times 176 \times 3 \times 100$ . For color videos, we feed each frame of size  $144 \times 176 \times 3$  into the pre-trained FFDNet on color images as the deep denoiser. The quantitative comparison of the recovered results by SNN, WSTNN, and DP3LRTC on color video *Suzie* is reported in Table 7. For better visual inspection, we further displayed the recovered results by SNN, WSTNN, and DP3LRTC in Fig. 11. Finally, we can conclude that the DP3LRTC framework can be easily extended to handle the high-order tensor completion problem.

## 5. Conclusions

In this paper, we proposed a novel tensor completion model, in which the TNN regularizer is utilized to catch the global information and a data-driven implicit regularizer is used to express the local information. The proposed model simultaneously combines the model-based optimization method with the deep learning-based method, in consideration of the global structure and fine details preservation. The ADMM based algorithm is developed to tackle the proposed model. Extensive numerical experiments on different types of multi-dimensional images illustrate the superiority of the proposed method for tensor completion problem.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## CRediT authorship contribution statement

**Xi-Le Zhao:** Methodology, Investigation, Writing - original draft.  
**Wen-Hao Xu:** Data curation, Software, Investigation, Visualization.  
**Tai-Xiang Jiang:** Conceptualization, Writing - review & editing, Methodology.  
**Yao Wang:** Writing - review & editing, Investigation, Validation.  
**Michael K. Ng:** Supervision, Writing - review & editing.

## References

- [1] L. Zhuang, J.M. Bioucas-Dias, Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 11 (3) (2018) 730–742.
- [2] Y. Wang, J. Peng, Q. Zhao, Y. Leung, X.-L. Zhao, D. Meng, Hyperspectral image restoration via total variation regularized low-rank tensor decomposition, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 11 (4) (2018) 1227–1243.
- [3] Y. Chang, L. Yan, H. Fang, C. Luo, Anisotropic spectral-spatial total variation model for multispectral remote sensing image despeckling, *IEEE Trans. Image Process.* 24 (6) (2015) 1852–1866.
- [4] J.-H. Yang, X.-L. Zhao, T.-H. Ma, Y. Chen, T.-Z. Huang, M. Ding, Remote sensing images despeckling using unidirectional hybrid total variation and nonconvex low-rank regularization, *J. Comput. Appl. Math.* 363 (2020) 124–144.
- [5] Q. Zhao, D. Meng, X. Kong, Q. Xie, W. Cao, Y. Wang, Z. Xu, A novel sparsity measure for tensor recovery, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 271–279.
- [6] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, Q. Zhu, A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method, *IEEE Trans. Neural Netw. Learn. Syst.* 27 (3) (2015) 579–592.
- [7] X. Luo, M. Zhou, S. Li, L. Hu, M. Shang, Non-negativity constrained missing data estimation for high-dimensional and sparse matrices from industrial applications, *IEEE Trans. Cybern.* (2019), doi:10.1109/TCYB.2019.2894283.
- [8] S. Li, M. Zhou, X. Luo, Modified primal-dual neural networks for motion control of redundant manipulators with dynamic rejection of harmonic noises, *IEEE Trans. Neural Netw. Learn. Syst.* 29 (10) (2017) 4791–4801.
- [9] L. Zhuang, J.M. Bioucas-Dias, Hy-Demosaicing: Hyperspectral blind reconstruction from spectral subsampling, in: Proceedings of the IEEE International Geoscience and Remote Sensing Symposium, 2018, pp. 4015–4018.
- [10] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, *SIAM Rev.* 51 (3) (2009) 455–500.
- [11] Y. Wang, D. Meng, M. Yuan, Sparse recovery: from vectors to tensors, *Natl. Sci. Rev.* 5 (5) (2017) 756–767.
- [12] Z. Long, Y. Liu, L. Chen, C. Zhu, Low rank tensor completion for multiway visual data, *Signal Process.* 155 (2019) 301–316.
- [13] N.D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E.E. Papalexakis, C. Faloutsos, Tensor decomposition for signal processing and machine learning, *IEEE Trans. Signal Process.* 65 (13) (2017) 3551–3582.
- [14] M. Che, A. Cichocki, Y. Wei, Neural networks for computing best rank-one approximations of tensors and its applications, *Neurocomputing* 267 (2017) 114–133.
- [15] X. Wang, M. Che, Y. Wei, Neural networks based approach solving multi-linear systems with m-tensors, *Neurocomputing* 351 (2019) 33–42.
- [16] Q. Zhao, L. Zhang, A. Cichocki, Bayesian CP factorization of incomplete tensors with automatic rank determination, *IEEE Trans. Pattern Anal. Mach. Intell.* 37 (9) (2015) 1751–1763.
- [17] C.J. Hillar, L.-H. Lim, Most tensor problems are NP-hard, *J. ACM* 60 (6) (2013) 1–39.

<sup>4</sup> Available at <http://trace.eas.asu.edu/yuv/>.

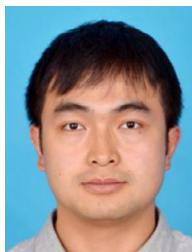
- [18] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (1) (2013) 208–220.
- [19] S. Gandy, B. Recht, I. Yamada, Tensor completion and low-n-rank tensor recovery via convex optimization, *Inverse Probl.* 27 (2) (2011) 025010.
- [20] Q. Xie, Q. Zhao, D. Meng, Z. Xu, Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery, *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (8) (2018) 1888–1902.
- [21] H. Tan, B. Cheng, J. Feng, G. Feng, W. Wang, Y.-J. Zhang, Low-n-rank tensor recovery based on multi-linear augmented lagrange multiplier method, *Neurocomputing* 119 (2013) 144–152.
- [22] T.-Y. Ji, T.-Z. Huang, X.-L. Zhao, T.-H. Ma, L.-J. Deng, A non-convex tensor rank approximation for tensor completion, *Appl. Math. Model.* 48 (2017) 410–422.
- [23] W. Cao, Y. Wang, C. Yang, X. Chang, Z. Han, Z. Xu, Folded-concave penalization approaches to tensor completion, *Neurocomputing* 152 (2015) 261–273.
- [24] M.E. Kilmer, C.D. Martin, Factorization strategies for third-order tensors, *Linear Algebra Appl.* 435 (3) (2011) 641–658.
- [25] M.E. Kilmer, K. Braman, N. Hao, R.C. Hoover, Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging, *SIAM J. Matrix Anal. Appl.* 34 (1) (2013) 148–172.
- [26] C.D. Martin, R. Shafrazi, B. LaRue, An order-p tensor factorization with applications in imaging, *SIAM J. Sci. Comput.* 35 (1) (2013) A474–A490.
- [27] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-Y. Ji, T.-H. Ma, Tensor n-tubal rank and its convex relaxation for low-rank tensor recovery, *arXiv:1812.00688*(2018).
- [28] Z.-M. Zhang, G. Ely, S. Aeron, N. Hao, M. Kilmer, Novel methods for multi-linear 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, 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] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm, *J. Comput. Appl. Math.* (2020), doi: 10.1016/j.cam.2019.112680.
- [31] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, T.-X. Jiang, T.-H. Ma, T.-Y. Ji, Mixed noise removal in hyperspectral image via low-fibered-rank regularization, *IEEE Trans. Geosci. Remote Sens.* 58 (1) (2020) 734–749.
- [32] A. Wang, Z. Lai, Z. Jin, Noisy low-tubal-rank tensor completion, *Neurocomputing* 330 (2019) 267–279.
- [33] X.-T. Li, Y. Ye, X. Xu, Low-rank tensor completion with total variation for visual data inpainting, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 2210–2216.
- [34] M. Ding, T.-Z. Huang, T.-Y. Ji, T.-Y. Ji, X.-L. Zhao, J.-H. Yang, Low-rank tensor completion using matrix factorization based on tensor train rank and total variation, *J. Sci. Comput.* 81 (2) (2019) 941–964.
- [35] X.-T. Li, X.-L. Zhao, T.-X. Jiang, Y.-B. Zheng, T.-Y. Ji, T.-Z. Huang, Low-rank tensor completion via combined non-local self-similarity and low-rank regularization, *Neurocomputing* 367 (2019) 1–12.
- [36] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, T.-Y. Ji, L.-J. Deng, Matrix factorization for low-rank tensor completion using framelet prior, *Inf. Sci. (Ny)* 436 (2018) 403–417.
- [37] F. Jiang, X.-Y. Liu, H. Liu, R. Shen, Anisotropic total variation regularized low-rank tensor completion based on tensor nuclear norm for color image inpainting, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2018, pp. 1363–1367.
- [38] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, Beyond a gaussian denoiser: residual learning of deep CNN for image denoising, *IEEE Trans. Image Process.* 26 (7) (2017) 3142–3155.
- [39] K. Zhang, W. Zuo, L. Zhang, FFDNet: Toward a fast and flexible solution for CN-N-based image denoising, *IEEE Trans. Image Process.* 27 (9) (2018) 4608–4622.
- [40] L. Zhang, W. Zuo, Image restoration: from sparse and low-rank priors to deep priors, *IEEE Signal Process Mag* 34 (5) (2017) 172–179.
- [41] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends® Mach. Learn.* 3 (1) (2011) 1–122.
- [42] S. Sreehari, S.V. Venkatakrishnan, B. Wohlberg, G.T. Buzzard, L.F. Drummy, J.P. Simmons, C.A. Bouman, Plug-and-play priors for bright field electron tomography and sparse interpolation, *IEEE Trans. Comput. Imaging* 2 (4) (2016) 408–423.
- [43] S.V. Venkatakrishnan, C.A. Bouman, B. Wohlberg, Plug-and-play priors for model based reconstruction, in: Proceedings of the IEEE Global Conference on Signal and Information Processing, 2013, pp. 945–948.
- [44] S.H. Chan, X. Wang, O.A. Elgendy, Plug-and-play ADMM for image restoration: fixed-point convergence and applications, *IEEE Trans. Comput. Imaging* 3 (1) (2016) 84–98.
- [45] Y. Romano, M. Elad, P. Milanfar, The little engine that could: regularization by denoising (RED), *SIAM J. Imaging Sci.* 10 (4) (2017) 1804–1844.
- [46] E.T. Reehorst, P. Schniter, Regularization by denoising: clarifications and new interpretations, *IEEE Trans. Comput. Imaging* 5 (1) (2018) 52–67.
- [47] L.I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D* 60 (1–4) (1992) 259–268.
- [48] J.-F. Cai, B. Dong, S. Osher, Z. Shen, Image restoration: total variation, wavelet frames, and beyond, *J. Am. Math. Soc.* 25 (4) (2012) 1033–1089.
- [49] K. Dabov, A. Foi, K. Egiazarian, Image denoising by sparse 3d transform-domain collaborative filtering, in: Proceedings of the 15th European Signal Processing Conference, 2007, pp. 145–149.
- [50] S. Gu, L. Zhang, W. Zuo, X. Feng, Weighted nuclear norm minimization with application to image denoising, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2862–2869.
- [51] Q. Xie, Q. Zhao, D. Meng, Z. Xu, S. Gu, W. Zuo, L. Zhang, Multispectral images denoising by intrinsic tensor sparsity regularization, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1692–1700.
- [52] W. He, Q. Yao, C. Li, N. Yokoya, Q. Zhao, Non-local meets global: an integrated paradigm for hyperspectral denoising, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6861–6870.
- [53] K. Zhang, W. Zuo, S. Gu, L. Zhang, Learning deep CNN denoiser prior for image restoration, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3929–3938.
- [54] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, W. Yin, Plug-and-play methods provably converge with properly trained denoisers, in: Proceedings of the International Conference on Machine Learning, 2019, pp. 5546–5557.
- [55] J.-F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, *SIAM J. Optim.* 20 (4) (2010) 1956–1982.
- [56] C. Lu, J. Feng, W. Liu, Z. Lin, S. Yan, Tensor robust principal component analysis with a new tensor nuclear norm, *IEEE Trans. Pattern Anal. Mach. Intell.* (2019), doi:10.1109/TPAMI.2019.2891760.
- [57] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (4) (2004) 600–612.



**Xi-Le Zhao** received the M.S. and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2009 and 2012. He is currently a Professor with the School of Mathematical Sciences, UESTC. His research interests are model-driven and data-driven methods for image processing problems.



**Wen-Hao Xu** received the B.S. degrees in mathematics and applied mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2017. He is currently pursuing the M.S. degree with the School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, China. His research interests are multi dimensional image processing.



**Tai-Xiang Jiang** received the B.S., Ph.D. degrees in mathematics and applied mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2013. He is currently a lecturer with the School of Economic Information Engineering, Southwestern University of Finance and Economics. His research interests include sparse and low-rank modeling, tensor decomposition and multi-dimensional image processing.



**Yao Wang** received the Ph.D. degree in applied mathematics from Xian Jiaotong University, Xian, China, in 2014. He is currently an Associate Professor with the School of Management, Xian Jiaotong University. His current research interests include statistical signal processing, high-dimensional data analysis, and machine learning.



**Michael K. Ng** is the Director of Research Division for Mathematical and Statistical Science, and Chair Professor of Department of Mathematics, the University of Hong Kong. His research areas are data science, scientific computing, and numerical linear algebra.