“Sparse + Low-Rank” Tensor Completion Approach for Recovering Images and Videos

Chenjian Pan\textsuperscript{1} and Chen Ling\textsuperscript{1} and Hongjin He\textsuperscript{2} and Liquan Qi\textsuperscript{3,4} and Yanwei Xu\textsuperscript{3}

\textsuperscript{1}Department of Mathematics, School of Science, Hangzhou Dianzi University, Hangzhou, 310018, China.
\textsuperscript{2} School of Mathematics and Statistics, Ningbo University, Ningbo 315211, China.
\textsuperscript{3} Theory Lab, Central Research Institute, 2012 Labs, Huawei Technologies Co., Ltd., Shatin, New Territory, Hong Kong, China.
\textsuperscript{4} Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China.

Recovering color images and videos from highly undersampled data is a fundamental and challenging task in face recognition and computer vision. By the multi-dimensional nature of color images and videos, in this paper, we propose a novel tensor completion approach, which is able to efficiently explore the sparsity of tensor data under the discrete cosine transform (DCT). Specifically, we introduce two “sparse + low-rank” tensor completion models as well as two implementable algorithms for finding their solutions. The first one is a DCT-based sparse plus weighted nuclear norm induced low-rank minimization model. The second one is a DCT-based sparse plus $p$-shrinkage mapping induced low-rank optimization model. Moreover, we accordingly propose two implementable augmented Lagrangian-based algorithms for solving the underlying optimization models. A series of numerical experiments including color image inpainting and video data recovery demonstrate that our proposed approach performs better than many existing state-of-the-art tensor completion methods, especially for the case when the ratio of missing data is high.

Index Terms—Tensor completion, $p$-shrinkage thresholding, weighted nuclear norm, discrete cosine transform, image inpainting.

I. INTRODUCTION

WITH the rapid developments of sensor technologies and video surveillance systems, the collected images and videos are naturally stored as multi-way arrays, which are also called tensors. As we know, the complete information of images and videos is very important to make a good decision for intelligent devices or data users. However, during the acquisition process, observed image and video data often contain missing entries (or pixels). In this situation, estimating these missing information of images and videos, which is also called tensor completion, is a fundamental and challenging problem in the communities of image processing and computer vision, e.g., see [1]–[4] and references therein.

It is well-known that tensor is a higher-order extension of matrix, so the tensor completion is also a natural generalization of matrix completion. Generally speaking, when the missing entries are sparse in an incomplete matrix, we could efficiently explore the local information around the unknown components to get a relatively ideal completion to the matrix, e.g., see [5]–[7]. However, when dealing with a highly undersampled matrix, it is incredibly difficult, even if not impossible, to accurately estimate these missing entries from a complete matrix without any global or prior information. Comparatively, the complex structure of tensors makes that recovering the missing entries from a highly undersampled tensor puts forward more theoretical and computational challenges at the interface of statistics and optimization, e.g., see [8]. In the past decades, it is well-documented that exploiting the inherent global information, e.g., low-rank and sparsity, of the data is able to greatly improve the estimation quality of highly undersampled matrices and tensors, e.g., see [9]–[13], to name just a few.

As a direct extension of the low-rank matrix completion, the canonical low-rank tensor completion (LRTC) model is expressed mathematically as

$\min_{\tilde{X}} \text{rank}(\tilde{X})$

s.t. $P_\Omega(\tilde{X}) = P_\Omega(H)$,

where both $\tilde{X}$ and $H$ are $N$-th order tensors, $\text{rank}(\cdot)$ denotes the rank function, $\Omega$ is the index set corresponding to the observed entries of the incomplete tensor $H$, and $P_\Omega(\cdot)$ is the linear operator that keeps known elements in $\Omega$ while setting the others to be zeros. It has been documented in [14], [15] that directly minimizing the rank function of tensors, even for matrices, is an NP-hard problem. Moreover, unlike the unique definition of rank for matrices, there are diverse definitions for tensors such as CANDECOMP/PARAFAC (CP) rank, Tucker rank and Tensor Train (TT) rank (see [16], [17]). Consequently, the diversity of tensor ranks often makes engineers difficult to choose an appropriate surrogate for characterizing the low-rankness of their problems.

Roughly speaking, most state-of-the-art tensor completion methods can be grouped into two categories. (i) The first type refers to the tensor nuclear norm minimization approach, which unfolds the underlying tensor as a series of matrices so that we could efficiently employ the matrix nuclear norm as an approximation to the rank of a tensor. Therefore, many efficient algorithms tailored for matrix completion could be naturally extended to tensor completion based on the so-called tensor nuclear norm definitions, e.g., see [9], [11], [18]–[23] and references therein. Mathematically, a seminal tensor nuclear norm minimization model introduced in [9] takes the
form

\[
\min_{\mathcal{X}} \sum_{n=1}^{N} \alpha_n \|X_{(n)}\|_{*},
\]

s.t. \( \mathcal{R}_{\Omega}(\mathcal{X}) = \mathcal{R}_{\Omega}(H), \) (2)

where \( \alpha_n \geq 0 \) \((n \in \{N\} := \{1, 2, \cdots, N\})\) can be regarded as weight parameters often satisfying \( \sum_{n=1}^{N} \alpha_n \approx 1 \), matrix \( X_{(n)} \) corresponds to the mode-\( n \) unfolding of tensor \( \mathcal{X} \) for every \( n \in \{N\}, \) and \( \| \cdot \|_{*} \) represents the well-known nuclear norm referring to the sum of all singular values of a matrix. Note that a vector consisting of the ranks of unfolding matrices \( X_{(n)} \)'s \((n = 1, 2, \cdots, N)\) is called the Tucker rank of the tensor \( \mathcal{X} \) (see [16]). Hence, the objective function in (2) is indeed a convexly weighted sum of the components of Tucker rank. Compared with the direct vectorization and matricization approaches for tensor data, model (2) efficiently exploits the multi-mode structure, thereby possibly achieving a better approximation to the low-rankness of tensor data. Moreover, these unfolding completion models (i.e., tensor nuclear norm minimization models) are beneficial for algorithmic design due to the convexity of tensor nuclear norm functions. Recently, the state-of-the-art alternating direction method of multipliers (ADMM) and Douglas-Rachford splitting method have been successfully applied to tensor completion, e.g., see [9], [10], [24]–[26]. (ii) The second type of tensor completion approach is based on tensor decomposition. Actually, we observe that many tensor nuclear norm based models utilize the so-called unfolding (or matricization) technique. However, Yuan and Zhang [27] showed that the matricization for higher-order tensors ignores the nature of tensors and potentially destroys some inherent properties of the data, thereby leading to suboptimal procedure, which encourages researchers to develop new completion methods from tensor decomposition perspectives. In the literature, some popular tensor decomposition techniques include CP decomposition, Tucker decomposition [16], T-SVD [28] and Tensor-Train decomposition [17]. Like the matrix decomposition, the aforementioned tensor decomposition forms have also been verified as powerful dimensionality reduction tools, which can promote the low-rankness of tensor data to some extent. Empirically, these tensor decomposition based completion models have received great successes in big data analysis, pattern recognition, and traffic data recovery, e.g., see [12], [13], [29]–[34] and references therein.

When regarding color images and videos as general tensors, the aforementioned tensor completion approaches are certainly applicable to images and video inpainting problems. However, it often fails to achieve an ideal restoration quality for highly undersampled image data since only the low-rank property is exploited in general tensor completion models (see Figs. 4 and 6). As we know, the total variation (TV) regularization (see [35]) is a widely used tool in the community of image processing to preserve sharp discontinuities (edges) of an image, while removing noise and other unwanted fine scale detail. Hence, researchers judiciously incorporated TV regularization into low-rank tensor completion models for the purpose of exploiting the inherent structure of images. The promising numerical performance can be found in recent works, e.g., [22], [33], [36]–[40]. From the mathematical modeling perspective, the underlying images usually are sparse under the TV transform. Consequently, we see that an \( \ell_1 \)-norm is used to promote the sparsity of an image in the TV transform domain, which demonstrates that combining the low-rank and TV-based sparse prior information simultaneously can greatly improve the restoration quality. However, as stated in [41], the traditional TV regularizer is still not perfect, since it only guarantees an estimation presenting a locally smooth visualization, while ignoring the global perception and structure. Such a drawback motivates us to find new sparse surrogates to recover more accurate data. Actually, in the image processing literature, the well-known discrete cosine transform (DCT) has been widely used for the most popular image compression standard JPEG, since DCT is powerful to obtain a (at least approximately) sparse representation of an image. For example, we consider a widely used gray video, i.e., sidewalk (see Fig. 1), which is a third order tensor of size \( 220 \times 352 \times 30 \). By performing DCT on each dimension of such a gray video, a large number of elements of approach to zeros, thereby leading to a high level sparsity when setting a truncated number (TN), e.g., 0.05 and 0.1, to drop all values less than such a TN. However, the video recovered directly from the truncated tensor is still close to the original one, which means that a small sample information is possible to recover an ideal (at least identifiable) one. Such an experimental observation also encourages us to consider a sparsity regularization on the coefficient tensor under the DCT procedure. Here, we refer the reader to [41]–[47] for recent applications of DCT on image recovery and some completion methods considering the property of the tensor data in transformed domains, e.g., see [18], [48]–[50]. Compared to the TV-based tensor completion model, to the best of our knowledge, DCT-based approaches received much less considerable attention on color images and videos inpainting from tensor completion angle. Hence, we are motivated to make a further study on showing the ability of the DCT technique for recovering images and videos from highly undersampled data.

![Fig. 1](image-url) Fig. 1. The first row show the visualization of the sparsity of the gray video sidewalk under DCT, where the white represents 0. The second row illustrates the comparisons of the 10-th frame of the original video and the frames reconstructed from the truncated tensor.

In this paper, we introduce two “sparse + low-rank” tensor completion approaches for recovering color images and videos. Our contribution is four-fold.

- Most nuclear norm based tensor completion models equally treat all singular values of the unfolding matrices.
For some real-world data sets, small singular values are often sensitive to the incomplete (or corrupted by noise) observed information, thereby possibly leading to a suboptimal low-rank tensor. In this situation, we will propose a minimization model, whose objective function is the sum of a DCT-based sparse regularization term and a weighted nuclear norm induced low-rank regularization term so that the singular values have different weights to enhance the global low-rank structure.

- In many unfolding approaches, the most popular low-rank surrogate of tensors is the nuclear norm of unfolding matrices. Recently, both theoretical and empirical results show that the so-called $p$-shrinkage thresholding algorithm outperforms the classical iterative soft thresholding algorithm induced by nuclear norm for low-rank and sparse recovery problems, e.g., see [51]–[53]. Accordingly, the second contribution of this paper is that we introduce a nonconvex optimization model, which minimizes the sum of a DCT-based sparse term and a $p$-shrinkage mapping induced low-rank term.

- We develop two implementable augmented Lagrangian-based algorithms for the proposed tensor completion models, where all subproblems of our algorithms have closed-form solutions.

- We conduct a series of experiments on color images and surveillance videos. Our computational results demonstrate that our proposed “sparse + low-rank” tensor completion approaches outperform many state-of-the-art model-driven tensor completion methods, especially for highly undersampled cases.

The structure of this paper is as follows. In Section II, we summarize some notations and recall some basic definitions including (weighted) nuclear norm of matrices, $p$-shrinkage mapping and proximal operators. In Section III, we will first introduce a unified “sparse + low-rank” tensor completion approach. Then, we split this section into two parts and propose two kinds of “sparse + low-rank” tensor completion models, respectively. Based on the augmented Lagrangian function, we also propose two implementable algorithms for the underlying tensor completion models. Numerically, in Section IV, we conduct the performance of our approaches on images and videos recovery from highly undersampled data. Finally, some concluding remarks are provided in Section V.

II. NOTATIONS AND PRELIMINARIES

In this section, we summarize some notations and definitions on $p$-shrinkage mapping and weighted nuclear norm that will be used throughout this paper.

Tensor is a multidimensional array, which is an extension of matrix. The space of all $N$-th order real tensors is denoted by $\mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, where the order of a tensor is also called way or mode. Given an $N$-th order tensor $A \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, we denote the $(i_1, i_2, \ldots, i_N)$-th component of $A$ by $a_{i_1i_2\ldots i_N}$. So the $N$-th order tensor $A$ is also denoted by $A = (a_{i_1i_2\ldots i_N})$. Throughout this paper, tensors of order $N \geq 3$ are denoted by calligraphical letters, e.g., $A, B, \ldots$. Generally, we use capital letters (e.g., $A, B, \ldots$), boldfaced lowercase letters (e.g., $a, b, \ldots$), and lowercase letters (e.g., $a, b, \ldots$) to denote matrices, vectors, and scalars, respectively. For any two $N$-th order real tensors $A = (a_{i_1i_2\ldots i_N})$ and $B = (b_{i_1i_2\ldots i_N})$, the inner product between $A$ and $B$ is given by

$$\langle A, B \rangle := \sum_{i_1,i_2,\ldots,i_N} a_{i_1i_2\ldots i_N} b_{i_1i_2\ldots i_N}.$$  

Consequently, the Frobenius norm of tensor $A$ associated with the above inner product is given by $\|A\|_F = \sqrt{\langle A, A \rangle}$. Given an $N$-th order tensor $A$, the mode-$n$ matricization (or unfolding) of $A$ is denoted by $A_{(n)}$, and the $(i_1, i_2, \ldots, i_N)$-th entry of tensor $A$ is mapped to the $(i_n, j)$-th entry of matrix $A_{(n)}$ in the lexicographical order, where

$$j = 1 + \sum_{1 \leq l \leq n, l \neq n} (i_l - 1) I_l \quad \text{with} \quad I_l = \prod_{1 \leq t \leq n, t \neq n} I_t.$$  

Given a matrix $A \in \mathbb{R}^{m \times n}$, we write its singular value decomposition (SVD) as $A = U \Lambda V^T$, where $U$ and $V$ are orthogonal matrices and $\Lambda$ is a diagonal matrix, i.e.,

$$\Lambda = \text{diag}(\sigma(A)) = \text{diag}(\sigma_1(A), \sigma_2(A), \ldots, \sigma_r(A))$$  

with $r := \min\{m, n\}$, and the diagonals $\sigma_i(A)$’s are singular values of $A$ satisfying $\sigma_1(A) \geq \sigma_2(A) \geq \ldots \geq \sigma_r(A)$ without loss of generality. With the above preparation, we first recall the definition of (weighted) nuclear norm of a matrix (see [43]).

**Definition 2.1.** For any given matrix $A \in \mathbb{R}^{m \times n}$, its nuclear norm is defined by

$$\|A\|_* = \sum_{i=1}^{r} \sigma_i(A)$$  

and the weighted nuclear norm is given by

$$\|A\|_{*,w} = \sum_{i=1}^{r} w_i \sigma_i(A)$$  

where $r := \min\{m, n\}$ and $\sigma_i(A)$ is the $i$-th largest singular value of $A$, and $w = (w_1, \ldots, w_r)$ is a weight vector, which, as suggested in [19], is given by

$$w_i = \delta / (\sigma_i(A) + \epsilon), \quad i = 1, 2, \ldots, r,$$

where $\delta \in \mathbb{R}_+$ is a constant and $\epsilon$ is a small positive number to avoid division by 0. In algorithmic implementation, we can iteratively update $w_i$’s.

Below, we recall the $p$-shrinkage mapping introduced in [52].

**Definition 2.2.** Given $a \in \mathbb{R}^n$, $\mu > 0$ and $p \leq 1$, the $p$-shrinkage thresholding operator is defined in component-wise by

$$\text{pshrink}(a, \mu, p) = \text{sign}(a) \odot \max \left\{ |a| - \mu |a|^{p-1}, 0 \right\},$$  

where ‘$\text{sign}(\cdot)$’ and ‘$| \cdot |’’ are the sign function and absolute value function in component-wise, respectively, and ‘$\odot$’ represents the component-wise product between two vectors. In particular, when setting $p = 1$, the $p$-shrinkage operator (3) immediately reduces to the well-known soft-thresholding.
which is denoted by

\[
\text{shrink}(a, \mu) = \text{sign}(a) \odot \max \{ |a| - \mu, 0 \}.
\]

Clearly, the smaller \( p \) is, the less \( \text{shrink}(a, \mu, p) \) shrinks large inputs. We refer the reader to [53] for an illustration to the philosophy of the \( p \)-shrinkage operator.

For any given \( a \in \mathbb{R}^n \), the proximal operator associated with \( \theta : \mathbb{R}^n \to \mathbb{R} \) about \( \tau > 0 \) is defined as

\[
\text{prox}_{\theta, \tau}(a) := \arg \min_{x \in \mathbb{R}^n} \left\{ \theta(x) + \frac{\tau}{2} \|x - a\|^2 \right\}.
\]

Consequently, as shown in [54, Theorem 4] (see also [55]), for any \( \mu > 0 \), \( A \in \mathbb{R}^{m \times n} \) and a weight vector \( w \), the global optimal solution to the following optimization problem

\[
\min_X \|X\|_p, w + \frac{1}{2\mu} \|X - A\|^2_F
\]

is given by the weighted SVD thresholding

\[
X^* = U\text{wshrink}(\Lambda, \mu, w)V^*,
\]

where \( A = UAV^T \) is the SVD of \( A \) and

\[
\text{wshrink}(\Lambda, \mu, w) = \max \{ \sigma(A) - \mu w, 0 \}.
\]

Moreover, it follows from [51] that the \( p \)-shrinkage mapping defined in Definition 2.2 can also be interpreted as the proximal operator of a penalty function \( \Phi^\mu_p(\cdot) : \mathbb{R}^n \to \mathbb{R} \), i.e.,

\[
\text{pshrink}(a, \mu, p) = \arg \min_{x \in \mathbb{R}^n} \left\{ \Phi^\mu_p(x) + \frac{1}{2\mu} \|x - a\|^2 \right\},
\]

where \( \Phi^\mu_p(x) := \sum_{i=1}^n \phi^\mu_p(x_i) \) with \( \phi^\mu_p(x_i) \) being even, concave, nondecreasing and continuous on \([0, \infty)\], differentiable on \((0, \infty)\), nondifferentiable at 0 with the subdifferential being \( \partial \phi^\mu_p(0) = [-1, 1] \). We refer the reader to [51, Theorem 1] (also [53], [56]) for more details. Notice that the \( p \)-shrinkage operator is available to matrix variables. Specifically, for the optimization problem

\[
\min_X \Phi^\mu_p(X) + \frac{1}{2\mu} \|X - A\|^2_F,
\]

we have its globally optimal solution given by

\[
X^* = U\text{pshrink}(\Lambda, \mu, p)V^T,
\]

where \( A = UAV^T \) is the SVD of \( A \) and

\[
\text{pshrink}(\Lambda, \mu, p) = \text{pshrink}(\sigma(A), \mu, p).
\]

III. Models and Algorithm

In this section, we aim to develop DCT-based sparse plus nonconvex functions induced low-rank tensor completion approaches to images and videos recovery. Mathematically, we first introduce a unified “sparse + low-rank” tensor completion model for \( N \)-th order tensors as follows:

\[
\min_X \text{rank}(X) + \lambda \|\mathcal{D}(X)\|_0 \quad \text{s.t.} \quad \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(\mathcal{H}),
\]

where \( \mathcal{D}(\cdot) \) denotes the multi-dimensional DCT operator satisfying orthogonality, \( \| \cdot \|_0 \) represents the so-called \( \ell_0 \)-norm to characterize the sparsity of DCT’s coefficients, and \( \lambda > 0 \) is a tuning parameter. Due to the appearances of rank function and \( \ell_0 \) norm, the optimization model (8) is a highly non-convex and NP-hard problem. Usually, we can find an approximate solution of (8) by employing the (non-) convex relaxations of rank function and \( \ell_0 \)-norm, respectively. Therefore, in this section, we first introduce a DCT-based sparse plus weighted nuclear norm (WNN) induced low-rank tensor completion model. Then, we propose a DCT-based sparse plus \( p \)-shrinkage mapping induced low-rank tensor completion model. Additionally, we will propose two implementable augmented Lagrangian-based splitting methods to solve the underlying models.

A. DCT-based sparse + WNN induced low-rank minimization model

As shown in the literature, directly minimizing the sum of the rank function and \( \ell_0 \)-norm of tensor \( X \) in (8) is not an easy task. So, how to approximate the rank of tensors is crucial for improving the completion quality. In this subsection, we use the sum of the weighted nuclear norms of the unfolding matrices to approximate the rank of \( X \) and employ the \( \ell_0 \)-norm to replace the \( \ell_0 \)-norm for promoting the sparsity of images under the DCT. Accordingly, we call the model DCT-based sparse + WNN induced low-rank minimization model, which is expressed as the form

\[
\min_X \sum_{i=1}^N \alpha_{i} \|X_{(i)}\|_{*, w} + \lambda \|\mathcal{D}(X)\|_1
\]

\[\text{s.t.} \quad \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(\mathcal{H}).\]

Apparently, two parts of the objective function in model (9) are nonsmooth, but both have promising structures so that their proximal operators enjoy explicit forms. To exploit the structure of model (9), we here first introduce auxiliary variables to separate the two nonsmooth terms in the objective function, thereby leading to a separable optimization model, i.e.,

\[
\begin{align*}
\min_X & \quad \sum_{i=1}^N \alpha_{i} \|Y_{(i)}\|_{*, w} + \lambda \|\mathcal{T}\|_1 \\
\text{s.t.} & \quad X = Y_{(i)} \quad \text{for all } i \in [N], \\
& \quad \mathcal{T} = \mathcal{D}(X), \\
& \quad \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(\mathcal{H}),
\end{align*}
\]

where \( Y_{(i)} \) represents the mode-\( i \) unfolding of tensor \( Y_i \) for \( i \in [N] \). Clearly, model (10) is an equality-constrained optimization problem, then its augmented Lagrangian function reads as

\[
\begin{align*}
\mathcal{L}(Y_i, X, \mathcal{T}, S_i, \mathcal{Q}) := & \sum_{i=1}^N \left( \alpha_i \|Y_{(i)}\|_{*, w} + \langle S_i, X - Y_{(i)} \rangle + \frac{\beta}{2} \|X - Y_{(i)}\|_F^2 \right) \\
& + \lambda \|\mathcal{T}\|_1 + \langle \mathcal{Q}, \mathcal{T} - \mathcal{D}(X) \rangle + \frac{\beta}{2} \|\mathcal{T} - \mathcal{D}(X)\|_F^2,
\end{align*}
\]

where \( S_i \) and \( \mathcal{Q} \) are Lagrangian multipliers associated to \( X_{(i)} = Y_{(i)} \) and \( \mathcal{T} = \mathcal{D}(X) \), respectively, and \( \beta \) is a positive penalty parameter. By (11), we follow the alternating spirit of ADMM to design an implementable algorithm. Theoretically, all primal variables can be updated in any order due to their
equal roles. However, their updating order is slightly sensitive to the numerical performance. Notice that the weight of the low-rank term is usually greater than the sparse part, we empirically first update \( \mathcal{Y} \) and then calculate \( \mathcal{X} \) due to newly introduced equality constraint. Thirdly, we update \( \mathcal{T} \) for the reason that we can immediately utilize the latest information of \( \mathcal{X} \). Finally, we update the Lagrangian multipliers \( \mathcal{S}_i \) and \( \mathcal{Q} \) simultaneously. Therefore, we accordingly update the variables for given \( \mathcal{X} \) of \( \mathcal{S}_i \) and \( \mathcal{Q} \).

With the above preparations, we formally summarize the updating schemes for model (10) in Algorithm 1.

### Algorithm 1 ADMM for Model (10)

**Input:** Initial starting points \( \mathcal{X}^0, \mathcal{Y}_i^0, \mathcal{T}, \mathcal{S}_i^0, \mathcal{Q}_0 \) and \( \beta_0 > 0 

1. Update \( \mathcal{Y}_i^{k+1} \) simultaneously via (12) for \( i = [N] \);
2. Update \( \mathcal{X}^{k+1} \) via (13);
3. Update \( \mathcal{T}^{k+1} \) via (14);
4. Update \( \mathcal{S}_i^{k+1} \) simultaneously via (15) for \( i = [N] \);
5. Update \( \mathcal{Q}^{k+1} \) via (16);
6. Update \( \beta_{k+1} = \beta_k \) with \( \rho > 1 \);
7. Until a termination criterion is fulfilled.

**Output:** \( \mathcal{X}^* \).

### B. DCT-based sparse + \( p \)-shrinkage mapping induced low-rank optimization model

In model (9), we employ the sum of the weighted nuclear norms of the unfolding matrices \( X_{ij} \) to approximate the low-rankness of the data. However, we do not know whether such an approximation is enough ideal for color images and videos tensor data. Recently, the so-called \( p \)-shrinkage algorithm is widely used in low-rank and sparse recovery problems. Hence, in this subsection, we further consider the combination of \( p \)-shrinkage low-rank penalty and DCT-driven sparse regularization, and propose the following tensor completion model:

\[
\min_{\mathcal{X}} \sum_{i=1}^{N} \alpha_i \Phi_p^u(\mathcal{X}_{ij}) + \lambda \| \mathcal{D}(\mathcal{X}) \|_1 \quad \text{s.t.} \quad \mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{H}),
\]

where \( p \leq 1 \) and \( \Phi_p^u(\cdot) \) is a nonconvex penalty function defined in (6). Compared to model (9), the only difference is that we here employ the \( \Phi_p^u(\mathcal{X}_{ij}) \) to replace the weighted nuclear norm term \( \|X_{ij}\|_{*,w} \). So, both (9) and (17) share the same structure except the low-rank-inducing term. For simplicity, we use the symbols in (10) again, then model (17) can be rewritten as the following separable optimization problem, i.e.,

\[
\min_{\mathcal{X}} \sum_{i=1}^{N} \alpha_i \Phi_p^u(\mathcal{X}_{ij}) + \lambda \| \mathcal{D}(\mathcal{X}) \|_1 \quad \text{s.t.} \quad \mathcal{X} = \mathcal{Y}_i, \quad \forall \quad i \in [N],
\]

\[
\mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{H}).
\]

Correspondingly, the augmented Lagrangian function for (18) reads as

\[
L(\mathcal{Y}_i, \mathcal{X}, \mathcal{T}, \mathcal{S}_i, \mathcal{Q}) := \sum_{i=1}^{N} \alpha_i \Phi_p^u(\mathcal{X}_{ij}) + (\mathcal{S}_i, \mathcal{X} - \mathcal{Y}_i) + \frac{\beta}{2} \| \mathcal{X} - \mathcal{Y}_i \|^2_F + \lambda \| \mathcal{T} \|_1 + (\mathcal{Q}, \mathcal{T} - \mathcal{D}(\mathcal{X})) + \frac{\beta}{2} \| \mathcal{T} - \mathcal{D}(\mathcal{X}) \|^2_F.
\]

Based upon (19), by invoking the update idea of Algorithm 1, i.e., \( \mathcal{Y}_i \rightarrow \mathcal{X} \rightarrow \mathcal{T} \rightarrow \mathcal{S}_i \rightarrow \mathcal{Q} \), we derive the detailed iterative schemes for solving model (18).
• Update $Y^{k+1}_i$ via
  
  $$Y^{k+1}_i = \arg\min_{Y_i} L(Y_i, X^k, T^k, S^k_i, Q^k)$$
  
  $$= \arg\min_{Y_i} \left\{ \alpha_i \phi_p(Y_{i(i)}(i)) \pi \|Y_i - \hat{Y}^k_i\|_F^2 \right\}$$
  
  $$= \text{fold} \left( \hat{U}_k \text{pshrink} \left( \tilde{X}_i, \frac{\alpha_i}{\beta_k}, p \right) \tilde{Y}^k_i \right), \quad (20)$$

  where 'pshrink(·, ·, ·)' is given by (3) and

  $$\hat{Y}^k_i = X^k + \frac{1}{\beta_k} S^k_i$$ and

  $$\hat{Y}^k_{i(i)} = \hat{U}_k \tilde{X}_i \tilde{Y}^k_i.$$

• We obtain $X^{k+1}$ via solving

  $$\min_{X} \{ L \left( Y^{k+1}_i, X^k, T^k, S^k_i, Q^k \right) \mid \mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(H) \},$$

  which can also be expressed explicitly by

  $$\left\{ \begin{array}{l}
  X_i^{k+1} = H_i \Omega \\
  X_i^{k+1} = \frac{1}{N+1} \hat{X}_i 
  \end{array} \right. \quad (21)$$

  where

  $$\hat{X}_i = \sum_{i=1}^N \left( Y^{k+1}_i - \frac{1}{\beta_k} S^k_i \right) + \beta_k^{-1} \left( T^k + \frac{1}{\beta_k} Q^k \right).$$

• Update $T^{k+1}$ via

  $$T^{k+1} = \arg\min_T L \left( Y^{k+1}_i, X^{k+1}, T^k, S^k_i, Q^k \right)$$

  $$= \text{shrink} \left( \mathcal{P}(X^{k+1}) \frac{1}{\beta_k} Q^k \right). \quad (22)$$

• Finally, update the two Lagrangian multipliers $S^{k+1}_i$ and $Q^{k+1}$ via (15) and (16), respectively.

  Formally, we summarize the iterative schemes for (18) in Algorithm 2.

**Algorithm 2 ADMM for Model (18).**

**Input:** Initial starting points $X^0, Y^0, T^0, S_0, Q_0$.

1. Update $Y^{k+1}_i$ simultaneously via (20) for $i = [N]$;
2. Update $X^{k+1}$ via (21);
3. Update $T^{k+1}$ via (22);
4. Update $S^{k+1}_i$ simultaneously via (15) for $i = [N]$;
5. Update $Q^{k+1}$ via (16);
6. Update $\beta_k+1 = \theta \beta_k$, $\theta > 1$;
7. Until a termination criterion is fulfilled.

**Output:** $X^*$.

**C. Convergence results**

In this subsection, we follow the way used in [57] to prove the convergence result of Algorithm 1 for problem (10) under some assumptions. Since Algorithm 2 share the almost same iterative schemes expect the $Y$-subproblem, we skip the convergence analysis of Algorithm 2 for model (18) for the conciseness of the paper.

**Lemma 3.1.** Suppose that both $\{S^k\} := \{S^k_1, \ldots, S^k_N\}$ and $\{Q^k\}$ are bounded, the sequences $\{Y^k\} := \{Y^k_1, \ldots, Y^k_N\}$ and $\{T^k\}$ produced by Algorithm 1 are bounded.

**Proof.** From the definition of $L(Y, X, T, S, Q)$, we have

$$L(Y^k, X^k, T^k, S^k, Q^k) - L(Y^k, X^k, T^k, S^k_{k-1}, Q^k_{k-1})$$

$$= \sum_{i=1}^N \left( \langle S^k_i - S^k_{k-1}, X^k - Y^k_i \rangle + \langle Q^k - Q^k_{k-1}, T^k - \mathcal{P}(X^k) \rangle \right)
+ \frac{\beta_k - \beta_{k-1}}{2} \left( \sum_{i=1}^N \|X^k - Y^k_i\|_F^2 + \|T^k - \mathcal{P}(X^k)\|_F^2 \right)$$

$$= \frac{1}{\beta_k-1} \left( \sum_{i=1}^N \|S^k_i - S^k_{k-1}\|_F^2 + \|Q^k - Q^k_{k-1}\|_F^2 \right)
+ \frac{\beta_k - \beta_{k-1}}{2\beta_k} \left( \sum_{i=1}^N \|S^k_i - S^k_{k-1}\|_F^2 + \|Q^k - Q^k_{k-1}\|_F^2 \right)$$

$$= \frac{\beta_k + \beta_{k-1}}{2\beta_k} \left( \sum_{i=1}^N \|S^k_i - S^k_{k-1}\|_F^2 + \|Q^k - Q^k_{k-1}\|_F^2 \right), \quad (23)$$

where the second equality follows from (15) and (16). Then, it follows from the iterative schemes of Algorithm 1 and (23) that

$$L(Y^{k+1}, X^{k+1}, T^{k+1}, S^k, Q^k)$$

$$\leq L(Y^k, X^k, T^k, S^k, Q^k)$$

$$+ \frac{\beta_k + \beta_{k-1}}{2\beta_k} \left( \sum_{i=1}^N \|S^k_i - S^k_{k-1}\|_F^2 + \|Q^k - Q^k_{k-1}\|_F^2 \right). \quad (24)$$

By invoking $\beta_k = \theta \beta_{k-1}$ with $\theta > 1$, we immediately prove

$$\sum_{i=1}^N \frac{\beta_k + \beta_{k-1}}{2\beta_k} \|S^k_i - S^k_{k-1}\|_F^2 + \|Q^k - Q^k_{k-1}\|_F^2 < \infty.$$ As a consequence of (24), the sequence $\{L(Y^k, X^k, T^k, S^k, Q^k)\}$ is upper-bounded due to the boundedness of $\{S^k\}$ and $\{Q^k\}$. Rearranging terms of $\{L(Y^k, X^k, T^k, S^k, Q^k)\}$ immediately yields

$$\sum_{i=1}^N \left( \alpha_i \|Y^k_{i(i)}\|_w + \lambda \|T^k\|_1 \right)$$

$$= L(Y^k, X^k, T^k, S^k, Q^k) - \beta_{k-1} \|Q^k - Q^k_{k-1}\|_F^2$$

$$- \frac{\beta_{k-1}}{2} \left( \sum_{i=1}^N \|S^k_i\|_2^2 - \|S^k_{k-1}\|_2^2 \right),$$

which, together with the boundedness of the sequences $\{L(Y^k, X^k, T^k, S^k, Q^k)\}$, $\{S^k\}$ and $\{Q^k\}$, implies that $\{Y^k_{i(i)}\}$ and $\{T^k\}$ for all $i \in [N]$ are bounded, since $\|\cdot\|_w$ and $\|\cdot\|_1$ are nonnegative. \qed

**Theorem 3.2.** Let $\{Y^k, X^k, T^k, S^k, Q^k\}$ be a sequence generated by Algorithm 1. Suppose that both $\{S^k\}$ and $\{Q^k\}$ are bounded, and further satisfy $\lim_{k \to \infty} \|S^k - S^k\|_F = 0$ and $\lim_{k \to \infty} \|Q^k - Q^k\|_F = 0$, respectively. Then,

(i) $\{X^k\}, \{Y^k\}, \{T^k\}$ are bounded Cauchy sequences.

(ii) any accumulation point $\{Y^\infty, X^\infty, T^\infty, S^\infty, Q^\infty\}$ satisfies the KKT condition of model (10).
Proof. For any \( i \in [N] \), it easily follows from the iterative scheme (15) and \( \lim_{k \to \infty} \beta_k = \infty \) that the sequence \( \{X^k\} \) is bounded due to Lemma 3.1. Moreover, by the properties of \( \{S^k\} \) and \( \{Q^k\} \), we have

\[
\begin{align*}
\lim_{k \to \infty} \|X^{k+1} - Y_{i(i)}^{k+1}\|_F &= 0, \\
\lim_{k \to \infty} \|T^{k+1} - D(X^{k+1})\|_F &= 0,
\end{align*}
\]

which further implies that \( \{Y^k\} \) and \( \{T^k\} \) approach feasible solutions.

By the boundedness of both \( \{Y^k\} \) and \( \{S^k\} \) and properties of \( \| \cdot \|_{*,w} \), it follows from (26) that \( \|X^k(i) - X^{k+1}(i)\|_F = O(\beta_k^{-1}) \), which together with \( \sum_{i=0}^{N} \beta_k^{-1} = \frac{1}{\beta_0(\beta_0-1)} < \infty \), implies that \( \{X^k\} \) is a Cauchy sequence, and it immediately has a limit point.

Let \( X^\infty, Y^\infty, T^\infty, S^\infty, Q^\infty \) be the limit points of \( \{X^k\}, \{Y^k\}, \{T^k\}, \{S^k\}, \{Q^k\} \), respectively. It first follows from (25) that

\[ X^\infty = Y^\infty, \quad (i = 1, \ldots, N) \quad \text{and} \quad T^\infty = D(X^\infty). \]

Rearranging (26) arrives at

\[
\begin{align*}
0 &\in \alpha_i \partial_C \|Y^k_{i, i} + 1\|_{*,w} - \beta_k \left( X^k_{i, i} - Y^k_{i, i} + 1 \right) \\
&= \alpha_i \partial_C \|Y^k_{i, i} + 1\|_{*,w} - \beta_k \left( X^k_{i, i} - Y^k_{i, i} + 1 \right).
\end{align*}
\]

Consequently, taking limit on (27) immediately yields \( S^\infty_{i, i} \in \alpha_i \partial_C \|X^\infty\|_{*,w} \).

By the first-order optimality condition of the \( X \)-subproblem, we have

\[
\begin{align*}
\sum_{i=1}^{N} \left( S^k_i + \beta_k(X^{k+1} - Y^{k+1}) \right) + \beta_k \left( X^{k+1} - D^{-1}(T^{k+1}) \right) - D^{-1}(Q^{k+1}) = 0,
\end{align*}
\]

Taking \( k \to \infty \) on the above two equalities leads to

\[ S^\infty_{i, i} = D^{-1}(Q^\infty) \quad \text{and} \quad \partial_X(X^\infty) = \partial_X(H). \]

Finally, the first-order optimality condition of the \( T \)-subproblem reads as

\[
\begin{align*}
0 &\in \partial \lambda \|T^{k+1}\|_1 + \beta_k \left( T^{k+1} - D(T^{k+1}) + \beta_k^{-1} Q^{k+1} \right) \\
&= \partial \lambda \|T^{k+1}\|_1 + Q^{k} + \beta_k \left( T^{k+1} - D(T^{k+1}) + \beta_k^{-1} Q^{k+1} \right) \\
&= \partial \lambda \|T^{k+1}\|_1 + Q^{k+1}.
\end{align*}
\]

Consequently, we have \( -Q^* \in \partial \lambda \|T^*\|_1 \) when \( k \to \infty \).

Therefore, we conclude that the accumulation point satisfies the KKT condition of model (10).

Hereafter, we only give the time complexity analysis of Algorithm 1 and omit the analysis of Algorithm 2 since both of them share the almost iterative scheme and the same SVD.

The main time cost of Algorithm 1 is consuming by performing SVD and multi-dimensional DCT. In each iteration, the complexity of subproblem \( Y_{i(i)}^{k+1} \) \( (i \in [N]) \) is \( O \left( \sum_{i=1}^{N} (I_i)^2 \times \prod_{j \neq i} I_j \right) \). The complexities of subproblem \( X^{k+1} \) and \( T^{k+1} \) are both \( O \left( \prod_{i=1}^{N} I_i \times \log \prod_{i=1}^{N} I_i \right) \). In addition, the complexities to update \( S_{i, i}^{k+1} \) \( (i \in [N]) \) and \( Q^{k+1} \) are \( O(\Omega(N I_i) I_i) \) and \( O(\Omega(I_i) I_i) \). So, the time complexity of each iteration is \( O \left( \sum_{i=1}^{N} (I_i)^2 \times \prod_{j \neq i} I_j + \prod_{i=1}^{N} I_i \times \log \prod_{i=1}^{N} I_i \right) \).

Thus, the total time complexity of Algorithm 1 is \( O \left( t \left( \sum_{i=1}^{N} (I_i)^2 \times \prod_{j \neq i} I_j + \prod_{i=1}^{N} I_i \times \log \prod_{i=1}^{N} I_i \right) \right) \), where \( t \) is the number of iterations.

IV. NUMERICAL EXPERIMENTS

In this section, we are concerned with the numerical performance of our approach proposed in Section III on images and videos recovery. Here, we will consider two kinds of images and videos data sets: 1) RGB images; 2) surveillance videos. All algorithms were implemented in MATLAB R2018b (64bit) and experiments were conducted on a laptop computer with Intel(R) Core(TM) i7-7500 CPU @2.70GHz and 8GB memory. Throughout this section, we denote Algorithms 1 and 2 by ‘DCT-WNN’ and ‘DCT-IpST’ for simplicity, respectively.

Moreover, we also compare the proposed algorithms with eight state-of-the-art tensor completion approaches as follows:

- IpST [53]: Iterative p-shrinkage thresholding algorithm for solving low Tucker rank tensor recovery problem, which only employs a nonconvex penalty function \( \Phi^p_\mu (\cdot) \) given in (6) to replace the traditional nuclear norm of unfolding matrices.

- TTNNL1 [44]: Using the truncated tensor nuclear norm for low-rank approximation, and a sparse regularization term combined with the 3-D DCT bases.

- F-TNN [48]: Framelet representation of tensor nuclear norm for third-order tensor completion.

- TNN-3DTV [39]: Anisotropic total variation regularized low-rank tensor completion based on tensor nuclear norm, which utilizes the 3D total variation regularization to exploit the structure of images.

- TCTF [12]: Tensor factorization for low-rank tensor completion, which uses the operation of T-product to decompose a tensor into two smaller tensors.

- WSTNN [58]: Tensor N-tubal rank and its convex relaxation for low-rank tensor recovery, which defines the weighted sum of the tensor nuclear norm.

- SMF-LRTC [59]: Low-rank tensor completion via smooth matrix factorization, which exploits the piecewise

1https://github.com/TaiXiangJiang/Framemelt-TNN
2https://panzhou.github.io/
3https://yubangzheng.github.io/ybz/
4https://github.com/zhaolede/Low-rank-tensor-completion-via-smooth-
matrix-factorization
smoothness prior of tensors by introducing smoothness constraints on the factor matrices. Here, we only use it for video experiments, because such a model is designed for video and hyperspectral image data sets.

- **DP3LRCT** [50]: Using tensor nuclear norm to characterize the global low-rankness prior and plugging a denoising neural network, which learned from a large number of natural images. It is a deep learning method.

For the fair comparison, we take

\[
\text{RelCha} = \frac{\|X^{k+1} - X^k\|_F}{\|X_{\text{true}}\|_F} \leq 10^{-4},
\]

(28)
as the stopping criterion for all methods, where \(X_{\text{true}}\) is the true tensor. Moreover, the following metrics are chosen to evaluate the recovery performance of the different algorithms.

- **Peak Signal-to-Noise Ratio (PSNR)** ([12]) is defined as:

\[
\text{PSNR} = 10 \log_{10} \left( \frac{\#X_{\text{true}} \cdot (\|X_{\text{true}}\|_F^2)}{\|X^* - X_{\text{true}}\|_F^2} \right).
\]

to measure the quality of the restored tensor data by an algorithm, where \((\#X_{\text{true}})\) denotes the number of elements of \(X_{\text{true}}\), \(\|X_{\text{true}}\|_F^2\) represents the largest element of \(X_{\text{true}}\), and \(X^*\) corresponds to the restored tensor.

- **Structural Similarity (SSIM)** ([60])\(^5\) is defined as:

\[
\text{SSIM} = \frac{(2\mu_X\mu_{X^*} + a_1)(2\sigma_{XX^*} + a_2)}{\mu_X^2 + \mu_{X^*}^2 + a_1(\sigma_X^2 + \sigma_{X^*}^2 + a_2)},
\]

where \(X\) and \(X^*\) denote the greyscale images for the original image and its recovered image, respectively; \(a_1\) and \(a_2\) are constants; \(\mu_X\) and \(\mu_{X^*}\) denote the average values, while \(\sigma_X\) and \(\sigma_{X^*}\) denote the standard deviation of \(X\) and \(X^*\), respectively; and \(\sigma_{XX^*}\) denote the covariance matrix between \(X\) and \(X^*\).

Since there are some parameters in models and algorithms, for both models (10) and (18), we set \((\alpha_1, \alpha_2, \alpha_3) = (1/3, 1/3, 10^{-3})\) for image inpainting and \((\alpha_1, \alpha_2, \alpha_3) = (1/3, 1/3, 1/3)\) for video inpainting, respectively. Moreover, we take \(\lambda = 0.05\) and \(\lambda = 10^{-2}\) for (10) and (18), respectively. For the algorithmic parameters, we set \(p = 0.2\) for DCT-IpST throughout the experiments. In addition, we set \(\beta = 1.2\) and \(\beta^0 = 10^{-5}\), respectively. All parameters of the other compared algorithms were taken as the default values used in the paper.

A. **RGB image inpainting**

In this subsection, we consider the RGB image inpainting problem, where RGB images namely include Red, Green, and Blue channels and the number of channels corresponds to the modality of a third-order tensor. Here, we conduct the numerical performance our approach on 16 widely used images\(^6\), which are summarized in Fig. 2. Here, the eight images in the first two rows of Fig. 2 are \(256 \times 256 \times 3\), and the remaining eight images are \(512 \times 768 \times 3\) except image \texttt{woman} of size \(768 \times 512 \times 3\). These images have no visual features such as low-rank and sparsity. However, as shown in Fig. 3, the sparse structure is apparent when applying DCT to these images, which also sufficiently supports that the main idea of this paper is reasonable.

![Fig. 2. The sixteen RGB images for experiments.](image)

![Fig. 3. Sparsity level of sixteen RGB images under DCT, where “sparsity level” is defined as the proportion of the number of zero elements in a tensor.](image)

Now, we are concerned with the inpainting performance of our approach on the cases where the pixels of images are missed in a random way. In Fig. 4, we first consider four scenarios on the sampling rate (‘sr’), i.e., \(sr = \{3\% , 5\% , 10\% , 20\% \}\), for five images. Clearly, except the deep learning method DP3LRTC, the recovered images by our methods, i.e., DCT-WNN and DCT-IpST, are better than the other compared model-driven methods for the scenarios where \(sr\)’s are lower than 10\% (see the first four columns of Fig. 4). To investigate the full sensitivity of these methods to \(sr\), we further consider five scenarios on \(sr\)’s from 3\% to 30\% for the images listed in Fig. 2, and plot PSNR values with respect to

\(^5\)MATLAB package: https://ece.uwaterloo.ca/~z70wang/research/ssim/

\(^6\)http://r0k.us/graphics/kodak/
sr’s in Fig. 5. It can be easily seen from Fig. 5 that both DCT-WNN and DCT-IpST outperform the other compared methods except the deep learning method DP3LRTC in terms of taking higher PSNR values for all scenarios with \( sr \leq 20\% \).

![Images](image1.png)

**Fig. 4.** Images recovered by the nine algorithms for the cases with random missing information. The top row corresponds to the observed incomplete images. From left to right: downsampled images with \( sr = 3\%, 3\%, 5\%, 10\% \) and \( 20\% \), respectively. From the second row to bottom: Images recovered by IpST, TTNNL1, F-TNN, TNN-3DTV, TCTF, WSTNN, DP3LRTC, DCT-IpST and DCT-WNN, respectively.

Below, we are interested in the numerical performance on structurally missing scenarios such as missing slices or shape/text mask (i.e., downsampling operator). Specifically, we consider five scenarios as shown by the first row of Fig. 6. Obviously, we see from the last two rows in Fig. 6 that our DCT-WNN and DCT-IpST have promising ability to recover the structurally missing images. Moreover, from the PSNR, SSIM, and computing time in seconds (TIME(s) for short) reported in Table I, we can see that the proposed DCT-WNN and DCT-IpST have better numerical performance than the compared methods except the deep learning method DP3LRTC. However, our DCT-WNN and DCT-IpST take less computing time than DP3LRTC to achieve comparable PSNR and SSIM values when all methods ran on a general personal computer without GPU acceleration. Those results efficiently demonstrate that the popular deep learning methods usually obtain better inpainting quality than the traditional model-driven approaches. However, model-driven approaches are still necessary for real-world applications due to their relatively low requirements on computer hardware.

![Images](image2.png)

**Fig. 6.** Images recovered by the nine algorithms for the cases with structurally missing information. From the second row to bottom: Images recovered by IpST, TTNNL1, F-TNN, TNN-3DTV, TCTF, WSTNN, DP3LRTC, DCT-IpST and DCT-WNN, respectively.

### B. Video recovery

The above color images are actually low-dimensional third-order tensors. In this subsection, we consider higher dimensional cases, i.e., surveillance video data sets, which are natural third-order and fourth-order tensors corresponding to grayscale and color videos, respectively. Here, we choose six grayscale videos including ‘airport’, ‘sidewalk’, ‘lunges’, ‘biking’, ‘baby’ and ‘basketball’ for experiments. Here, we add a ‘camera-shake’ operation to ‘airport’ and ‘sidewalk’ so that their backgrounds are not static. Typical frames in these videos are depicted in Fig. 7.

![Images](image3.png)

**Fig. 7.** Typical frames in six grayscale videos.

We also first check the sparsity level of videos data sets when applying DCT to videos. The results in Fig. 8 clearly demonstrate that videos also appear sparse structure under DCT, since the six video data sets have more than 60\% zeros.
Fig. 5. PSNR values with respect to different sampling rates for the sixteen RGB images in Fig. 2.

TABLE I

| Method            | caps-1 | caps-2 | girl-1 | girl-2 | motorbikes |
|-------------------|--------|--------|--------|--------|------------|
|                   | PSNR   | SSIM   | TIME   | PSNR   | SSIM   | TIME   | PSNR   | SSIM   | TIME   | PSNR   | SSIM   | TIME   |
| IPST              | 17.28  | 0.558  | 53.81  | 17.87  | 0.749  | 52.21  | 25.01  | 0.892  | 55.29  | 23.24  | 0.668  | 62.14  |
| TTTNL             | 24.07  | 0.711  | 55.37  | 26.86  | 0.857  | 55.31  | 31.22  | 0.942  | 53.25  | 23.70  | 0.724  | 56.93  |
| F-TNN             | 17.00  | 0.540  | 135.08 | 17.87  | 0.749  | 25.13  | 27.61  | 0.916  | 97.64  | 19.11  | 0.559  | 137.30 |
| TNN-3DTV          | 18.96  | 0.602  | 315.42 | 20.47  | 0.800  | 300.23 | 29.38  | 0.934  | 361.91 | 22.47  | 0.712  | 323.01 |
| TCTF              | 12.70  | 0.130  | 26.43  | 18.22  | 0.753  | 30.23  | 25.59  | 0.863  | 28.62  | 10.94  | 0.135  | 29.64  |
| WSTNN             | 19.27  | 0.639  | 167.36 | 34.41  | 0.965  | 154.31 | 29.05  | 0.881  | 157.83 | 16.34  | 0.576  | 168.18 |
| DP3LRTC           | 29.04  | 0.933  | 190.79 | 39.25  | 0.922  | 218.53 | 33.93  | 0.899  | 188.16 | 27.30  | 0.890  | 183.15 |
| DCT-IpST          | 28.08  | 0.842  | 102.59 | 35.84  | 0.973  | 100.13 | 31.79  | 0.946  | 99.67  | 26.10  | 0.821  | 101.77 |
| DCT-WNN           | 28.15  | 0.831  | 116.81 | 36.09  | 0.974  | 102.33 | 31.76  | 0.943  | 100.07 | 26.22  | 0.807  | 114.25 |

which means that DCT is an ideal technique for promoting sparsity of video data sets.

Now, we turn our attention to investigating the performance of our approach on incomplete videos data. In our experiments, we conduct four scenarios on the sampling rate, i.e., $sr = \{5\%, 10\%, 20\%, 30\%\}$, for the six video data sets. The PSNR, SSIM and TIME(s) values of the six videos obtained by the algorithms are summarized in Table II, which shows that both DCT-IpST and DCT-WNN perform better than the other compared methods except the deep learning method DP3LRTC for the low sampling rate cases (i.e., $sr \leq 20\%$). Here, it is noteworthy that the deep learning method DP3LRTC does not necessarily always achieve the best results. For example, our approach either DCT-WNN or DCT-IpST works better than DP3LRTC for “basketball” and “lunge” in some cases. More interestingly, our DCT-WNN and DCT-IpST always take much less computing time than DP3LRTC for video data sets to achieve comparable inpainting quality. In Fig. 9, we first display the destroyed (downsampled) frames (ten consecutive frames) are listed in the first row, and the recovered frames
by different algorithms are displayed from the second row to the bottom. From the inpainted frames, it is obvious that both DCT-WNN and DCT-IpST can successfully obtain relatively ideal videos even for highly undersampled cases.

C. Discussion

Parameters sensitivity: Notice that there are some model parameters in our models, which are very important for the numerical performance of the models. However, it is well-known that setting model parameters is an incredibly difficult task in practice. Generally speaking, we can take constants empirically according to existing experiments in the literature. Here, we will conduct the numerical sensitivity of our models’ parameters. Note that there is a triple of parameters in model (9), i.e., \( \{ (\alpha_1, \cdots, \alpha_N), \delta, \lambda \} \). According to [53], [58], we can easily determine the choice of \( \alpha_i \)'s. So, we below are concerned with the sensitivity of \( \delta \) and \( \lambda \). Similarly, for model (17), we focus on the sensitivity of \( p \) and \( \lambda \). In our experiments, we consider the “airport” video data set with \( sr = 30\% \). The parameter sensitivity results are summarized in Fig. 10. It can be easily seen from Fig. 10 that both models run relatively robust for \( \lambda \in [0, 0.1, 1] \), which also provides us some suggestions on the settings for the above experiments.

Model discussion: Notice that the objective functions of our models consists of two nonsmooth parts. Ones will be interested in what will happen if we remove one of the nonsmooth parts. To answer this question, we shall investigate the numerical performance of our models when removing the low-rank regularization term or the sparse term. Here, we also focus on the “airport” video data set with \( sr = \{ 5\%, 10\%, 20\%, 30\% \} \) in our experiments. In Fig. 11, “only low-rank term” and “only sparse term” correspond to our models without sparse term and without low-rank regularization, respectively. From the plots in Fig. 11, we observe that our models equipped with only one low-rank term works better than the cases with only one sparse term. However, the proposed models with sparse and low-rank terms have the best numerical performance, which also sufficiently supports the idea of this paper.

In summary, the above computational results show that the proposed DCT-WNN and DCT-IpST are powerful to complete different types of downsampled images and videos data sets. Although our DCT-WNN and DCT-IpST cannot obtain the same inpainting quality of the state-of-the-art deep learning method DP3LRTC, DCT-WNN and DCT-IpST run much faster for large-scale data sets than DP3LRTC on a general personal computer. From the computing time and computer hardware perspectives, those model-driven approaches still have their own advantages. Moreover, we believe that our models could be further improved if some deep learning techniques are embedded in. We leave it as one of our future concerns.

V. Conclusion

In this paper, we introduced a unified “sparse + low-rank” tensor completion approach, which includes two structured optimization models for recovering images and videos from highly undersampled data. By introducing auxiliary variables,
we gainfully separated the two nonsmooth terms appeared in the objective functions, thereby being of benefit for designing easily implementable algorithms for the underlying separable models. A series of computational experiments on RGB images and surveillance videos data sets demonstrated that our approach performs better than some state-of-the-art model-driven tensor completion approaches.

**Declaration of Competing Interest:** The authors declared that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Acknowledgments**

The authors are grateful to the editor and two anonymous referees for their close reading and valuable comments, which led to great improvements of the paper. They also would like to thank all authors of [12], [39], [44], [48], [50], [53], [58],
Fig. 9. Visualizations of recovered ‘sidewalk’ data sets by the ten algorithms. From left to right: ten consecutive observed frames with 20% information. From top to bottom: observed frames, and frames recovered by IpST, TTNL1, F-TNN, TNN-3DTV, TCTF, WSTNN, SMF-LRTC, DP3LRTC, DCT-IpST and DCT-WNN, respectively.

Fig. 10. Numerical sensitivity of model parameters $\lambda$, $\delta$, and $p$. The left figure corresponds to our first model (9). The right one corresponds to the second model (17).

Fig. 11. Numerical performance of our models with only one low-rank objective or with only one sparse regularization term, respectively.

[59] for sharing their code, especially thank Dr. Yufan Li and Mr. Zhiyuan Zhang for their kind help on our experiments. C. Ling and H. He were supported in part by National Natural Science Foundation of China (Nos. 11971138 and 11771113) and Natural Science Foundation of Zhejiang Province (Nos. LY19A010019 and LD19A010002).

REFERENCES
[1] N. Komodakis, “Image completion using global optimization,” in Computer Vision and Pattern Recognition, vol. 1, 2006, pp. 442–452.

[2] N. Hao, M. Kilmer, K. Braman, and R. Hoover, “Facial recognition using tensor-tensor decompositions,” SIAM J. Imaging Sci., vol. 6, no. 1, pp. 437–463, 2013.

[3] A. Criminisi, P. Perez, and K. Toyama, “Region filling and object removal by exemplar-based image inpainting,” IEEE Trans. Image Process., vol. 13, no. 9, pp. 1200–1212, 2004.

[4] X. Geng, K. Smith-Miles, Z. Zhou, and L. Wang, “Face image modeling by multilinear subspace analysis with missing values,” Proceedings of the 17th ACM international conference on Multimedia, 2009.

[5] A. Efros and T. Leung, “Texture synthesis by non-parametric sampling,” in Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, 1999, pp. 1033–1038.

[6] D. Coupier, A. Desolneux, and B. Ycart, “Image denoising by statistical
[57] T. Oh, Y. Tai, J. Bazin, H. Kim, and I. Kweon, “Partial sum minimization of singular values in robust PCA: Algorithm and applications,” IEEE Trans. Image Process., vol. 38, pp. 744–758, 2016.

[58] Y. Zheng, T. Huang, X. Zhao, T. Jiang, and T. Ma, “Tensor n-tubal rank and its convex relaxation for low-rank tensor recovery,” Information Sciences., vol. 532, pp. 170–189, 2020.

[59] Y. Zheng, T. Huang, T. Ji, X. Zhao, T. Jiang, and T. Ma, “Low-rank tensor completion via smooth matrix factorization,” Appl. Math. Model., vol. 70, pp. 677–695, 2019.

[60] Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, pp. 600-612, 2004.