Double Transformed Tubal Nuclear Norm Minimization for Tensor Completion

TIAN Jialue[1], *ZHU Yulian*[2*], *LIU Jiahui*[1]

1. College of Computer Science and Technology/College of Artificial Intelligence, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P.R. China;
2. Fundamental Experimental Teaching Department, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P.R. China

(Received 8 May 2022; revised 23 July 2022; accepted 28 August 2022)

Abstract: Non-convex methods play a critical role in low-rank tensor completion for their approximation to tensor rank is tighter than that of convex methods. But they usually cost much more time for calculating singular values of large tensors. In this paper, we propose a double transformed tubal nuclear norm (DTTNN) to replace the rank norm penalty in low rank tensor completion (LRTC) tasks. DTTNN turns the original non-convex penalty of a large tensor into two convex penalties of much smaller tensors, and it is shown to be an equivalent transformation. Therefore, DTTNN could take advantage of non-convex envelopes while saving time. Experimental results on color image and video inpainting tasks verify the effectiveness of DTTNN compared with state-of-the-art methods.

Key words: double transformed tubal nuclear norm; low tubal-rank; non-convex optimization; tensor factorization; tensor completion

CLC number: TN911.73 **Document code:** A **Article ID:** 1005-1120(2022)S-0166-09

0 Introduction

Low rank tensor completion (LRTC) aims at completing the whole tensor through mining hidden information from visual positions for further tasks, such as color image inpainting, video recovery, hyperspectral image reconstruction, recommender system, etc. LRTC constraints low rank property on the target tensor. However, unlike matrix, tensor rank doesn't have a unified definition. CP-rank, tucker rank, and tubal rank are the popular definitions used widely. Although there are various tensor rank definitions, the same as matrix rank, they are all NP-hard to solve directly. Hence, researchers usually convert to find convex or concave envelopes of the rank norm. Zhang et al. found a convex envelope named tensor nuclear norm (TNN)[1] based on tensor singular value decomposition (T-SVD), which is similar to matrix nuclear norm, and achieve great success in tensor completion tasks. Other researchers further extended TNN to concave cases, such as Wu et al.[2] applied Laplace function to tensor singular values, Shi et al.[3] used LogDet function to replace nuclear norm function, Kong et al.[4] extended nuclear norm to Schatten-q norm with $0<q<1$, etc. But these methods often correspond with high time-consumption, because of the T-SVD operation on large tensors. To solve this problem, we transfer to factorization methods. In matrix case, Shang et al.[5] proposed a low rank matrix completion model based on matrix factorization, which factorizes the original large matrix into two small matrices. Hence, the original SVD on large matrix is converted to two much smaller matrices, and this operation saves a lot of time. Inspired by this success application, we consider extending this approach to tensor case.

*Corresponding author, E-mail address: lianyi_1999@nuaa.edu.cn.

How to cite this article: TIAN Jialue, ZHU Yulian, LIU Jiahui. Double transformed tubal nuclear norm minimization for tensor completion[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2022, 39(S): 166-174.

http://dx.doi.org/10.16356/j.1005-1120.2022.S.022

1 Notations, Preliminaries and Related Work

1.1 Notations and preliminaries

In this paper, we mainly focus on the field of 3-order complex tensors denoted as $\mathbb{C}^{n_1\times n_2\times n_3}$. Table 1 demonstrates the commonly used symbols. Tensors and matrices are denoted by handwritten letters and italic capital letters respectively.

Definition 1 $\boldsymbol{\Phi}$-product[6] The $\boldsymbol{\Phi}$-product of $\boldsymbol{\mathcal{A}}\in\mathbb{C}^{n_1\times n_2\times n_3}$ and $\boldsymbol{\mathcal{B}}\in\mathbb{C}^{n_2\times n_4\times n_3}$ is a tensor $\boldsymbol{\mathcal{C}}\in\mathbb{C}^{n_1\times n_4\times n_3}$, which is given by

$$\boldsymbol{\mathcal{C}}=\boldsymbol{\mathcal{A}}\diamond_{\boldsymbol{\Phi}}\boldsymbol{\mathcal{B}}=\left(\text{fold}\left(\text{bdiag}\left(\boldsymbol{\mathcal{A}}_{\boldsymbol{\Phi}}\right)\text{bdiag}\left(\boldsymbol{\mathcal{B}}_{\boldsymbol{\Phi}}\right)\right)\right)_{\boldsymbol{\Phi}^{\mathrm{H}}}=\left(\text{fold}\left(\hat{A}_{\boldsymbol{\Phi}}\hat{B}_{\boldsymbol{\Phi}}\right)\right)_{\boldsymbol{\Phi}^{\mathrm{H}}} \tag{1}$$

Definition 2 Unitary tensor A tensor $\boldsymbol{\mathcal{A}}\in\mathbb{C}^{n_1\times n_2\times n_3}$ is called unitary tensor if satisfying $\boldsymbol{\mathcal{A}}\diamond_{\boldsymbol{\Phi}}\boldsymbol{\mathcal{A}}^{\mathrm{H}}=\boldsymbol{\mathcal{A}}^{\mathrm{H}}\diamond_{\boldsymbol{\Phi}}\boldsymbol{\mathcal{A}}=\boldsymbol{\mathcal{I}}$, where $\boldsymbol{\mathcal{I}}$ is the identity tensor.

Definition 3 Transformed tubal nuclear norm (TTNN)[6] The transformed tubal nuclear norm of a tensor $\boldsymbol{\mathcal{A}}\in\mathbb{C}^{n_1\times n_2\times n_3}$, denoted as $\|\boldsymbol{\mathcal{A}}\|_{\text{TTNN}}$, is the sum of the nuclear norm of all the frontal slices of $\boldsymbol{\mathcal{A}}_{\boldsymbol{\Phi}}$

$$\|\boldsymbol{\mathcal{A}}\|_{\text{TTNN}}=\sum_{i=1}^{n_3}\|\boldsymbol{\mathcal{A}}_{\boldsymbol{\Phi}}{}^{i}\|_{*} \tag{2}$$

Theorem 1 Tensor singular value decomposition (T-SVD)[6] Suppose that $\boldsymbol{\mathcal{X}}\in\mathbb{C}^{n_1\times n_2\times n_3}$, then $\boldsymbol{\mathcal{X}}$ can be factorized as follows

$$\boldsymbol{\mathcal{X}}=\boldsymbol{\mathcal{U}}\diamond_{\boldsymbol{\Phi}}\boldsymbol{\mathcal{S}}\diamond_{\boldsymbol{\Phi}}\boldsymbol{\mathcal{V}}^{\mathrm{H}} \tag{3}$$

where $\boldsymbol{\mathcal{U}}\in\mathbb{C}^{n_1\times n_1\times n_3}$, $\boldsymbol{\mathcal{V}}\in\mathbb{C}^{n_2\times n_2\times n_3}$ are unitary tensors, and $\boldsymbol{\mathcal{S}}\in\mathbb{C}^{n_1\times n_2\times n_3}$ is a diagonal tensor (the frontal slices of its transformed form $\boldsymbol{\mathcal{S}}_{\boldsymbol{\Phi}}$ are diagonal matrices). The proof is constructed and the computation of T-SVD is shown in the Algorithm of T-SVD as follows.

Algorithm of T-SVD[6]

Input: $\boldsymbol{\mathcal{X}}\in\mathbb{C}^{n_1\times n_2\times n_3}$, $\boldsymbol{\Phi}\in\mathbb{C}^{n_3\times n_3}$

(1) Compute transformed tensor $\boldsymbol{\mathcal{X}}_{\boldsymbol{\Phi}}$

(2) for $i=1,2,\cdots,n_3$ do

(3) $[\boldsymbol{U},\boldsymbol{S},\boldsymbol{V}]=\text{SVD}(\boldsymbol{\mathcal{X}}_{\boldsymbol{\Phi}}{}^{i})$;

Table 1 Basic notations

Notation	Field	Description
$\boldsymbol{\mathcal{A}}$	$\mathbb{C}^{n_1\times n_2\times n_3}$	A 3-order tensor
$\mathcal{A}_{ijk}$	$\mathbb{C}$	The (i,j,k)-th entry of $\boldsymbol{\mathcal{A}}$
$\langle\boldsymbol{\mathcal{A}},\boldsymbol{\mathcal{B}}\rangle$	$\mathbb{C}$	The inner product between $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$, $\langle\boldsymbol{\mathcal{A}},\boldsymbol{\mathcal{B}}\rangle=\sum_{i,j,k}\left(\boldsymbol{\mathcal{A}}_{ijk}^{\mathrm{H}}\boldsymbol{\mathcal{B}}_{ijk}\right)$
$\|\boldsymbol{\mathcal{A}}\|_{\mathrm{F}}$	$\mathbb{C}$	The Frobenius norm of $\boldsymbol{\mathcal{A}}$, $\|\boldsymbol{\mathcal{A}}\|_{\mathrm{F}}=\sqrt{\langle\boldsymbol{\mathcal{A}},\boldsymbol{\mathcal{A}}\rangle}$
$\boldsymbol{\mathcal{A}}^{i}$	$\mathbb{C}^{n_1\times n_2}$	The i-th frontal slice $\left(\boldsymbol{\mathcal{A}}(:,:,i)\right)$ of $\boldsymbol{\mathcal{A}}$
$\hat{A}=\text{bdiag}(\boldsymbol{\mathcal{A}})$	$\mathbb{C}^{n_1n_3\times n_2n_3}$	A block diagonal matrix with each diagonal element being $\boldsymbol{\mathcal{A}}^{i}$
$\text{fold}\left(\hat{A}\right)=\boldsymbol{\mathcal{A}}$	$\mathbb{C}^{n_1\times n_2\times n_3}$	The inverse operation of bdiag$(\boldsymbol{\mathcal{A}})$
$A=\text{unfold}(\boldsymbol{\mathcal{A}})$	$\mathbb{C}^{n_3\times n_1n_2}$	The unfolded tensor of $\boldsymbol{\mathcal{A}}$ in the third dimension with the columns of A being the tubes $\left(\boldsymbol{\mathcal{A}}(i,j,:)\right)$ of $\boldsymbol{\mathcal{A}}$
$\text{tensor}(A)=\boldsymbol{\mathcal{A}}$	$\mathbb{C}^{n_1\times n_2\times n_3}$	The inverse operation of unfold$(\boldsymbol{\mathcal{A}})$
$\boldsymbol{\Phi}$	$\mathbb{C}^{n_3\times n_3}$	A unitary transform matrix with $\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{H}}=\boldsymbol{\Phi}^{\mathrm{H}}\boldsymbol{\Phi}=I$
$\boldsymbol{\mathcal{A}}_{\boldsymbol{\Phi}}=\text{tensor}(\boldsymbol{\Phi}A)$	$\mathbb{C}^{n_1\times n_2\times n_3}$	The transformed tensor was obtained by multiplying $\boldsymbol{\Phi}$ to the third dimension of $\boldsymbol{\mathcal{A}}$, and $\boldsymbol{\mathcal{A}}=(\boldsymbol{\mathcal{A}}_{\boldsymbol{\Phi}})_{\boldsymbol{\Phi}^{\mathrm{H}}}$
$\boldsymbol{\mathcal{A}}^{\mathrm{H}}$	$\mathbb{C}^{n_2\times n_1\times n_3}$	The tensor conjugate transpose obtained by $\text{fold}\left(\hat{A}_{\boldsymbol{\Phi}}^{\mathrm{H}}\right)$
$\boldsymbol{\mathcal{I}}$	$\mathbb{C}^{n_1\times n_2\times n_3}$	The identity tensor with each $\boldsymbol{\mathcal{I}}_{\boldsymbol{\Phi}}{}^{i}$ being identity matrix
$\|\cdot\|_{*}$	$\mathbb{C}$	The matrix nuclear norm

(4) $\boldsymbol{\mathcal{U}}_{\boldsymbol{\Phi}}^{\,i}=U$, $\boldsymbol{\mathcal{S}}_{\boldsymbol{\Phi}}^{\,i}=S$, $\boldsymbol{\mathcal{V}}_{\boldsymbol{\Phi}}^{\,i}=V$;

(5) end for

(6) $\boldsymbol{\mathcal{U}}=(\boldsymbol{\mathcal{U}}_{\boldsymbol{\Phi}})_{\boldsymbol{\Phi}^{\mathrm{H}}}$, $\boldsymbol{\mathcal{S}}=(\boldsymbol{\mathcal{S}}_{\boldsymbol{\Phi}})_{\boldsymbol{\Phi}^{\mathrm{H}}}$, $\boldsymbol{\mathcal{V}}=(\boldsymbol{\mathcal{V}}_{\boldsymbol{\Phi}})_{\boldsymbol{\Phi}^{\mathrm{H}}}$;

Output:

$\boldsymbol{\mathcal{U}}\in\mathbb{C}^{n_1\times n_1\times n_3}$, $\boldsymbol{\mathcal{S}}\in\mathbb{C}^{n_1\times n_2\times n_3}$, $\boldsymbol{\mathcal{V}}\in\mathbb{C}^{n_2\times n_2\times n_3}$.

Definition 4 Tensor tubal multi-rank, tubal rank and sum rank, m-rank, t-rank, and s-rank[7] Given a tensor $\boldsymbol{\mathcal{X}}\in\mathbb{C}^{n_1\times n_2\times n_3}$, m-rank($\boldsymbol{\mathcal{X}}$) is a vector with its i-th entry being the rank of the i-th frontal slice of $\boldsymbol{\mathcal{X}}_{\boldsymbol{\Phi}}$. t-rank($\boldsymbol{\mathcal{X}}$) is defined as the max value of frontal slices rank of $\boldsymbol{\mathcal{X}}_{\boldsymbol{\Phi}}$ or described by the diagonal tensor $\boldsymbol{\mathcal{S}}$ calculated in T-SVD. s-rank($\boldsymbol{\mathcal{X}}$) is the rank sum of all frontal slices of $\boldsymbol{\mathcal{X}}_{\boldsymbol{\Phi}}$. m-rank = $[\,\mathrm{rank}(\boldsymbol{\mathcal{X}}_{\boldsymbol{\Phi}}^{\,1}),\ \mathrm{rank}(\boldsymbol{\mathcal{X}}_{\boldsymbol{\Phi}}^{\,2}),\ \cdots,\ \mathrm{rank}(\boldsymbol{\mathcal{X}}_{\boldsymbol{\Phi}}^{\,n_3})\,]$, t-rank($\boldsymbol{\mathcal{X}}$) $=\#\{i:\boldsymbol{\mathcal{S}}(i,i,:)\neq 0\}=\max\left(\mathrm{rank}(\boldsymbol{\mathcal{X}}_{\boldsymbol{\Phi}}^{\,j})\right)$, s-rank($\boldsymbol{\mathcal{X}}$) $=\sum_{j=1}^{n_3}\mathrm{rank}(\boldsymbol{\mathcal{X}}_{\boldsymbol{\Phi}}^{\,j})$, where $i=1,2,\cdots,\min(n_1,n_2)$; $j=1,2,\cdots,n_3$ and rank($\cdot$) represents the matrix rank.

1.2 Related work

LRTC is proposed under the assumption that the data has a low rank attribute[8]. Given a tensor $\boldsymbol{\mathcal{D}}\in\mathbb{C}^{n_1\times n_2\times n_3}$, the low-rank tensor completion model is described as follows

$$\min_{\boldsymbol{\mathcal{L}}}\ \mathrm{rank}(\boldsymbol{\mathcal{L}})$$
$$\text{s.t. }\ \mathcal{P}_{\Omega}(\boldsymbol{\mathcal{L}})=\mathcal{P}_{\Omega}(\boldsymbol{\mathcal{D}})$$

where $\boldsymbol{\mathcal{L}}$ is the completed result, $\boldsymbol{\mathcal{D}}$ the observed tensor and $\mathcal{P}_{\Omega}(\cdot)$ the projection operation, in which the entries in Ω indicate the observed values and missing values otherwise. However, directly solving the rank norm minimization problem is NP-hard. Therefore, the rank norm penalty is usually released to its envelopes. Using the tensor tubal rank defined in Definition 4, Song et al.[6] proved that transformed tensor nuclear norm (TTNN) is a convex envelope of the sum of the elements of the tensor tubal multi-rank over a unit ball of the tensor spectral norm[6]. And they proposed the LRTC model based on TTNN

$$\min_{\boldsymbol{\mathcal{L}}}\|\boldsymbol{\mathcal{L}}\|_{\mathrm{TTNN}}=\min_{\boldsymbol{\mathcal{L}}}\sum_{i=1}^{n_3}\|\boldsymbol{\mathcal{L}}_{\boldsymbol{\Phi}}^{\,i}\|_{*}=\min_{\boldsymbol{\mathcal{L}}}\sum_{i}^{n_3}\sum_{j}^{\min(n_1,n_2)}\sigma_j(\boldsymbol{\mathcal{L}}_{\boldsymbol{\Phi}}^{\,i})$$
$$\text{s.t. }\ \mathcal{P}_{\Omega}(\boldsymbol{\mathcal{L}})=\mathcal{P}_{\Omega}(\boldsymbol{\mathcal{D}}) \tag{4}$$

It's worth noting that TTNN is under a unitary transform matrix $\boldsymbol{\Phi}$, when $\boldsymbol{\Phi}$ is a discrete Fourier transform (DFT) matrix, TTNN is converted to the TNN[1].

When the tensor only has two dimensions, LRTC is degraded into low rank matrix completion (LRMC). To solve the highly time-consuming problem in the matrix case, Shang et al.[5] proposed a matrix factorization method: Double nuclear norm (D-N), and applied it to the LRMC problem

$$\min_{U,V}\frac{1}{2}\left(\|U\|_{*}+\|V\|_{*}\right)$$
$$\text{s.t. }\ \mathcal{P}_{\Omega}(L)=\mathcal{P}_{\Omega}(D),\ L=UV^{\mathrm{H}} \tag{5}$$

which motivates us to extend this factorization mind to the tensor case.

2 Double Transformed Tubal Nuclear Norm

Since the TTNN noted in Section 1.2 is the convex envelope of tensor sum rank, in this section, we extend TTNN to non-convex envelope fields. We first define the transformed tensor Schatten-q norm (TT-S_q) under the transform matrix $\boldsymbol{\Phi}$ distinguishing from the DFT matrix used in Ref. [4]. After that, inspired by the D-N[5] in the matrix case, we propose a DTTNN and prove its equivalence with TT-$S_{\frac{1}{2}}$ under minimum optimization problem.

Definition 5 TT-S_q Given a tensor $\boldsymbol{\mathcal{X}}\in\mathbb{C}^{n_1\times n_2\times n_3}$ with t-rank($\boldsymbol{\mathcal{X}}$) $=r\leqslant\min(n_1,n_2)$, TT-S_q ($0<q<\infty$) is defined as

$$\|\boldsymbol{\mathcal{X}}\|_{\mathrm{TT\text{-}S_q}}:=\left(\sum_{i}^{n_3}\sum_{j}^{\min(n_1,n_2)}\sigma_j^{q}(\boldsymbol{\mathcal{X}}_{\boldsymbol{\Phi}}^{\,i})\right)^{\frac{1}{q}} \tag{6}$$

where $\sigma_j(\boldsymbol{\mathcal{X}}_{\boldsymbol{\Phi}}^{\,i})$ denotes the j-th singular value of $\boldsymbol{\mathcal{X}}_{\boldsymbol{\Phi}}^{\,i}$, $\boldsymbol{\Phi}$ the any unitary matrices, including discrete Fourier transform matrix[4], wavelet transform matrix[6], cosine transform matrix[9], and data-based transform matrix[6].

Definition 6 DTTNN Given a tensor $\mathcal{X}\in\mathbb{C}^{n_1\times n_2\times n_3}$ with t-rank$(\mathcal{X})=r\leqslant d$ (d is the estimated tubal rank), we decompose it into two factor tensors $\mathcal{U}\in\mathbb{C}^{n_1\times d\times n_3}$ and $\mathcal{V}\in\mathbb{C}^{n_2\times d\times n_3}$ such that $\mathcal{X}=\mathcal{U}\diamond_{\Phi}\mathcal{V}^{\mathrm{H}}$. Then the DTTNN is defined as

$$\|\mathcal{X}\|_{\mathrm{DTTNN}}:=\min_{\mathcal{U},\mathcal{V}}\frac{1}{4}\left(\|\mathcal{U}\|_{\mathrm{TTNN}}+\|\mathcal{V}\|_{\mathrm{TTNN}}\right)^2 \quad (7)$$

Theorem 2 DTTNN is equivalent to the TT-Schatten-1/2 norm. i.e.

$$\|\mathcal{X}\|_{\mathrm{DTTNN}}=\|\mathcal{X}\|_{\mathrm{TT\text{-}S_{1/2}}} \quad (8)$$

To prove Theorem 2, we first demonstrate the following Lemma.

Lemma 1 Given two tensors $\mathcal{U}\in\mathbb{C}^{n_1\times d\times n_3}$ and $\mathcal{V}\in\mathbb{C}^{n_2\times d\times n_3}$ and let $q>0$, then the following inequality holds for the decreasing ordered singular values of each frontal slice of $\mathcal{X}=\mathcal{U}\diamond_{\Phi}\mathcal{V}^{\mathrm{H}}$, $\mathcal{U}$ and $\mathcal{V}$.

$$\sum_{i}^{n_3}\sum_{j}^{\min(n_1,n_2,d)}\sigma_j^q(\mathcal{X}_{\Phi}^{\,i})\leqslant\sum_{i}^{n_3}\sum_{j}^{\min(n_1,n_2,d)}\sigma_j^q(\mathcal{U}_{\Phi}^{\,i})\,\sigma_j^q(\mathcal{V}_{\Phi}^{\,i}) \quad (9)$$

Proof

$$\sum_{i}^{n_3}\sum_{j}^{\min(n_1,n_2,d)}\sigma_j^q(\mathcal{X}_{\Phi}^{\,i})=\sum_{k}^{n_3\min(n_1,n_2,d)}\sigma_k^q\left(\hat{X}_{\Phi}\right)=\sum_{k}^{n_3\min(n_1,n_2,d)}\sigma_k^q\left(\hat{U}_{\Phi}\hat{V}_{\Phi}^{\mathrm{H}}\right)\leqslant\sum_{k}^{n_3\min(n_1,n_2,d)}\sigma_k^q\left(\hat{U}_{\Phi}\right)\sigma_k^q\left(\hat{V}_{\Phi}^{\mathrm{H}}\right)=\sum_{i}^{n_3}\sum_{j}^{\min(n_1,n_2,d)}\sigma_j^q\left(\hat{U}_{\Phi}^{\,i}\right)\sigma_j^q\left(\hat{V}_{\Phi}^{\,i}\right)$$

where the inequality in the proof follows from Ref. [10] (Theorem 3.3.14).

Proof of Theorem 2 Since $\mathcal{X}=\mathcal{U}\diamond_{\Phi}\mathcal{V}^{\mathrm{H}}$, where $\mathcal{U}\in\mathbb{C}^{n_1\times d\times n_3}$ and $\mathcal{V}\in\mathbb{C}^{n_2\times d\times n_3}$, we have

$$\|\mathcal{X}\|_{TT-S_{1/2}}^{1/2}=\sum_{i}^{n_3}\sum_{j}^{\min(n_1,n_2,d)}\sigma_j^{\frac{1}{2}}(\mathcal{X}_{\Phi}^{\,i})\leqslant\sum_{i}^{n_3}\sum_{j}^{\min(n_1,n_2,d)}\sigma_j^{\frac{1}{2}}(\mathcal{U}_{\Phi}^{\,i})\,\sigma_j^{\frac{1}{2}}(\mathcal{V}_{\Phi}^{\,i})\leqslant\left(\sum_{i}^{n_3}\sum_{j}^{\min(n_1,d)}\sigma_j(\mathcal{U}_{\Phi}^{\,i})\right)^{\frac{1}{2}}\left(\sum_{i}^{n_3}\sum_{j}^{\min(n_2,d)}\sigma_j(\mathcal{V}_{\Phi}^{\,i})\right)^{\frac{1}{2}}\leqslant\frac{1}{2}\left(\sum_{i}^{n_3}\sum_{j}^{\min(n_1,d)}\sigma_j(\mathcal{U}_{\Phi}^{\,i})+\sum_{i}^{n_3}\sum_{j}^{\min(n_2,d)}\sigma_j(\mathcal{V}_{\Phi}^{\,i})\right)=\frac{1}{2}\left(\sum_{i=1}^{n_3}\|\mathcal{U}_{\Phi}^{\,i}\|_{*}+\sum_{i=1}^{n_3}\|\mathcal{V}_{\Phi}^{\,i}\|_{*}\right)=\frac{1}{2}\left(\|\mathcal{U}\|_{\mathrm{TTNN}}+\|\mathcal{V}\|_{\mathrm{TTNN}}\right)$$

where the first inequality follows from Lemma 1 as $q=1/2$, the second inequality follows from the well-known Holder's inequality[11] and the third inequality follows from the Jensen's inequality[12].

Next, when $\mathcal{U}_{\star}=\mathcal{P}\diamond_{\Phi}\mathcal{D}^{1/2}$, $\mathcal{V}_{\star}=\mathcal{R}\diamond_{\Phi}\mathcal{D}^{1/2}$, and the SVD of $\mathcal{X}$ being $\mathcal{X}=\mathcal{U}_{\star}\diamond_{\Phi}\mathcal{V}_{\star}^{\mathrm{H}}=\mathcal{P}\diamond_{\Phi}\mathcal{D}\diamond_{\Phi}\mathcal{R}^{\mathrm{H}}$, we have $\|\mathcal{X}\|_{\mathrm{TT\text{-}S_{1/2}}}^{1/2}=\frac{1}{2}\left(\|\mathcal{U}_{\star}\|_{\mathrm{TTNN}}+\|\mathcal{V}_{\star}\|_{\mathrm{TTNN}}\right)$.

Hence, we can conclude that

$$\|\mathcal{X}\|_{\mathrm{DTTNN}}=\|\mathcal{X}\|_{\mathrm{TT\text{-}S_{1/2}}} \quad (10)$$

3 Problem Formulation and Optimization

3.1 Problem formulation

By Theorem 2, we use DTTNN, which is equivalent to TT-$S_{1/2}$ and tighter envelope than TTNN for tensor rank defined in Definition 3, to substitute the tensor rank norm penalty in LRTC. Then we present the following tensor completion model

$$\begin{aligned}&\min_{\mathcal{U},\mathcal{V},\mathcal{L}}\frac{1}{2}\left(\|\mathcal{U}\|_{\mathrm{TTNN}}+\|\mathcal{V}\|_{\mathrm{TTNN}}\right)+\\&\quad\frac{\lambda}{2}\|\mathcal{P}_{\Omega}(\mathcal{L})-\mathcal{P}_{\Omega}(\mathcal{D})\|_{\mathrm{F}}^2\\&\quad\text{s.t.}\quad\mathcal{L}=\mathcal{U}\diamond_{\Phi}\mathcal{V}^{\mathrm{H}}\end{aligned} \quad (11)$$

where $\mathcal{P}_{\Omega}(\cdot)$ represents the tensor observed entries and λ the balance coefficient.

3.2 Optimization algorithm

We use the framework of ADMM[13] to solve Eq.(1). By introducing the auxiliary variables $\dot{\mathcal{U}}$ and $\dot{\mathcal{V}}$, Eq.(1) converts to solving the following augmented Lagrangian function

$$\begin{aligned}L_{\mu}(\mathcal{U},\mathcal{V},\dot{\mathcal{U}},\dot{\mathcal{V}},\mathcal{L},\mathcal{Y}_i)=&\frac{1}{2}\left(\|\dot{\mathcal{U}}\|_{\mathrm{TTNN}}+\|\dot{\mathcal{V}}\|_{\mathrm{TTNN}}\right)+\frac{\lambda}{2}\|\mathcal{P}_{\Omega}(\mathcal{L})-\mathcal{P}_{\Omega}(\mathcal{D})\|_{\mathrm{F}}^2+\\&\langle\mathcal{Y}_1,\mathcal{U}-\dot{\mathcal{U}}\rangle+\langle\mathcal{Y}_2,\mathcal{V}-\dot{\mathcal{V}}\rangle+\\&\langle\mathcal{Y}_3,\mathcal{L}-\mathcal{U}\diamond_{\Phi}\mathcal{V}^{\mathrm{H}}\rangle+\frac{\mu}{2}\left(\|\mathcal{U}-\dot{\mathcal{U}}\|_{\mathrm{F}}^2+\right.\\&\left.\|\mathcal{V}-\dot{\mathcal{V}}\|_{\mathrm{F}}^2+\|\mathcal{L}-\mathcal{U}\diamond_{\Phi}\mathcal{V}^{\mathrm{H}}\|_{\mathrm{F}}^2\right)\end{aligned} \quad (12)$$

where $\mu>0$ is the penalty parameter and $\mathcal{Y}_i$ are La-

grangian multipliers.

3.2.1 Updating $\mathcal{U}_{k+1}$ and $\mathcal{V}_{k+1}$

By fixing other variables, $\mathcal{U}_{k+1}$ and $\mathcal{V}_{k+1}$ could be calculated by solving the following two optimization problems

$$\min_{\mathcal{U}} \left\| \mathcal{U} - \dot{\mathcal{U}}_k + \mathcal{Y}_1^k/\mu_k \right\|_F^2 + \left\| \mathcal{L}_k - \mathcal{U} \diamond_{\Phi} \mathcal{V}_k^{\mathrm{H}} + \mathcal{Y}_3^k/\mu_k \right\|_F^2 \tag{13}$$

$$\min_{\mathcal{V}} \left\| \mathcal{V} - \dot{\mathcal{V}}_k + \mathcal{Y}_2^k/\mu_k \right\|_F^2 + \left\| \mathcal{L}_k - \mathcal{U}_{k+1} \diamond_{\Phi} \mathcal{V}^{\mathrm{H}} + \mathcal{Y}_3^k/\mu_k \right\|_F^2 \tag{14}$$

It is obvious that Eq.(13) and Eq.(14) are similar optimization problems, for simplicity we only show the procedure of solving Eq.(13). Eq.(13) is actually equivalent to the following Eq.(15) according to the property of $\|\cdot\|_F$

$$\min_{\mathcal{U}_{\Phi}} \left\| \mathcal{U}_{\Phi} - \dot{\mathcal{U}}_{k\Phi} + \left(\mathcal{Y}_1^k/\mu_k\right)_{\Phi} \right\|_F^2 + \left\| \mathcal{L}_{k\Phi} - \left(\mathcal{U} \diamond_{\Phi} \mathcal{V}_k^{\mathrm{H}}\right)_{\Phi} + \left(\mathcal{Y}_3^k/\mu_k\right)_{\Phi} \right\|_F^2 \tag{15}$$

and block diagonalizing the tensor doesn't change the value of $\|\cdot\|_F$, hence we can convert Eq.(15) to Eq.(16)

$$\min_{\hat{U}_{\Phi}} \left\| \hat{T}_1 + \hat{U}_{\Phi} \right\|_F^2 + \left\| \hat{T}_2 - \hat{U}_{\Phi} \hat{V}_{k\Phi}^{\mathrm{H}} \right\|_F^2$$

$$\hat{T}_1 = \mathrm{bdiag}\left(\left(\mathcal{Y}_1^k/\mu_k\right)_{\Phi} - \dot{\mathcal{U}}_{k\Phi} \right) \tag{16}$$

$$\hat{T}_2 = \mathrm{bdiag}\left(\mathcal{L}_{k\Phi} + \left(\mathcal{Y}_3^k/\mu_k\right)_{\Phi} \right)$$

Therefore, $\hat{U}_{\Phi}$ can be directly obtained by solving the least square Eq.(16), and the optimal solution is given by

$$\hat{U}_{k+1\Phi} = \left(\hat{T}_2 \hat{V}_{k\Phi}^{\mathrm{H}} + \hat{T}_1 \right) \left(-I_{3d} + \hat{V}_{k\Phi}^{\mathrm{H}} \hat{V}_{k\Phi} \right)^{-1} \tag{17}$$

and $\mathcal{U}_{k+1}$ can be gotten as

$$\mathcal{U}_{k+1} = \left(\mathrm{fold}\left(\hat{U}_{k+1\Phi} \right) \right)_{\Phi^{\mathrm{H}}} \tag{18}$$

Similarly, $\mathcal{V}_{k+1}$ is

$$\hat{V}_{k+1\Phi} = \left(\hat{T}_2^{\mathrm{H}} \hat{U}_{k+1\Phi} + \hat{T}_3 \right) \left(-I_{3d} + \hat{U}_{k+1\Phi}^{\mathrm{H}} \hat{U}_{k+1\Phi} \right)^{-1}$$

$$\mathcal{V}_{k+1} = \left(\mathrm{fold}\left(\hat{V}_{k+1\Phi} \right) \right)_{\Phi^{\mathrm{H}}} \tag{19}$$

where $\hat{T}_3 = \mathrm{bdiag}\left(\left(\mathcal{Y}_2^k/\mu_k\right)_{\Phi} - \dot{\mathcal{V}}_{k\Phi} \right)$.

3.2.2 Updating $\dot{\mathcal{U}}_{k+1}$ and $\dot{\mathcal{V}}_{k+1}$

To update $\dot{\mathcal{U}}_{k+1}$ and $\dot{\mathcal{V}}_{k+1}$, we fix other variables and obtain the following optimization problems

$$\min_{\dot{\mathcal{U}}} \frac{1}{2} \left\| \dot{\mathcal{U}} \right\|_{\mathrm{TTNN}} + \frac{\mu_k}{2} \left\| \mathcal{U}_{k+1} - \dot{\mathcal{U}} + \mathcal{Y}_1^k/\mu_k \right\|_F^2 \tag{20}$$

$$\min_{\dot{\mathcal{V}}} \frac{1}{2} \left\| \dot{\mathcal{V}} \right\|_{\mathrm{TTNN}} + \frac{\mu_k}{2} \left\| \mathcal{V}_{k+1} - \dot{\mathcal{V}} + \mathcal{Y}_2^k/\mu_k \right\|_F^2 \tag{21}$$

which are equivalent to

$$\min_{\dot{\mathcal{U}}} \sum_{i=1}^{n_3} \left(\frac{1}{2\mu_k} \left\| \dot{\mathcal{U}}_{\Phi}{}^{i} \right\|_* + \frac{1}{2} \left\| \dot{\mathcal{U}}_{\Phi}{}^{i} - \mathcal{U}_{k+1\Phi}{}^{i} - \left(\mathcal{Y}_1^k/\mu_k\right)_{\Phi}{}^{i} \right\|_F^2 \right) \tag{22}$$

$$\min_{\dot{\mathcal{V}}} \sum_{i=1}^{n_3} \left(\frac{1}{2\mu_k} \left\| \dot{\mathcal{V}}_{\Phi}{}^{i} \right\|_* + \frac{1}{2} \left\| \dot{\mathcal{V}}_{\Phi}{}^{i} - \mathcal{V}_{k+1\Phi}{}^{i} - \left(\mathcal{Y}_2^k/\mu_k\right)_{\Phi}{}^{i} \right\|_F^2 \right) \tag{23}$$

By Ref.[14], the optimal solutions of Eq.(22) and Eq.(23) are

$$\dot{\mathcal{U}}_{k+1\Phi}{}^{i} = \mathcal{P}_{\Phi}{}^{i} \mathcal{Q}_{2\mu_k\Phi}{}^{i} \left(\mathcal{R}_{\Phi}{}^{i}\right)^{\mathrm{H}} \tag{24}$$

$$\dot{\mathcal{V}}_{k+1\Phi}{}^{i} = \mathcal{A}_{k+1\Phi}{}^{i} \mathcal{B}_{2\mu_k\Phi}{}^{i} \left(\mathcal{C}_{\Phi}{}^{i}\right)^{\mathrm{H}} \tag{25}$$

where $\mathcal{P}_{\Phi}{}^{i} \mathcal{Q}_{\Phi}{}^{i} \left(\mathcal{R}_{\Phi}{}^{i}\right)^{\mathrm{H}}$ is the singular value decomposition of $\mathcal{U}_{k+1\Phi}{}^{i} + \left(\mathcal{Y}_1^k/\mu_k\right)_{\Phi}{}^{i}$ and $\mathcal{A}_{k+1\Phi}{}^{i} \mathcal{B}_{\Phi}{}^{i} \left(\mathcal{C}_{\Phi}{}^{i}\right)^{\mathrm{H}}$ is the singular value decomposition of $\mathcal{V}_{k+1\Phi}{}^{i} + \left(\mathcal{Y}_2^k/\mu_k\right)_{\Phi}{}^{i}$. $\mathcal{Q}_{2\mu_k\Phi}{}^{i} = \max\left\{ \mathcal{Q}_{\Phi}{}^{i} - 2\mu_k,\ 0 \right\}$ and $\mathcal{B}_{2\mu_k\Phi}{}^{i} = \max\left\{ \mathcal{B}_{\Phi}{}^{i} - 2\mu_k,\ 0 \right\}$.

3.2.3 Updating $\mathcal{L}_{k+1}$

Fixing other variables, we then get the optimization problem in reference to $\mathcal{L}_{k+1}$

$$\min_{\mathcal{L}} \frac{\lambda}{2} \left\| \mathcal{P}_{\Omega}(\mathcal{L}) - \mathcal{P}_{\Omega}(\mathcal{D}) \right\|_F^2 + \frac{\mu_k}{2} \left\| \mathcal{L} - \mathcal{U}_{k+1} \diamond_{\Phi} \mathcal{V}_{k+1}^{\mathrm{H}} + \mathcal{Y}_3^k/\mu_k \right\|_F^2 \tag{26}$$

It is a least square problem as well with limitation $\mathcal{P}_{\Omega}(\cdot)$ on $\mathcal{L} - \mathcal{D}$, we can directly write the optimal solution as

$$\mathcal{L}_{k+1} = \mathcal{P}_{\Omega}\left(\frac{1}{\lambda + \mu_k} \left(\lambda D + \mu_k \mathcal{U}_{k+1} \diamond_{\Phi} \mathcal{V}_{k+1}^{\mathrm{H}} - \mathcal{Y}_3^k \right) \right) + \mathcal{P}_{\Omega^{\perp}}\left(\mathcal{U}_{k+1} \diamond_{\Phi} \mathcal{V}_{k+1}^{\mathrm{H}} - \mathcal{Y}_3^k/\mu_k \right) \tag{27}$$

where $\Omega^{\perp}$ is the complement set of Ω.

3.2.4 Updating Lagrange multiplier $\mathcal{Y}_i^{k+1}$ and μ_{k+1}

The multiplier $\mathcal{Y}_i^{k+1}$ are directly obtained by

$$\mathcal{Y}_1^{k+1} = \mathcal{Y}_1^k + \mu_k \left(\mathcal{U}_{k+1} - \dot{\mathcal{U}}_{k+1} \right)$$

$$\mathcal{Y}_2^{k+1} = \mathcal{Y}_2^k + \mu_k \left(\mathcal{V}_{k+1} - \dot{\mathcal{V}}_{k+1} \right) \tag{28}$$

$$\mathcal{Y}_3^{k+1} = \mathcal{Y}_3^k + \mu_k \left(\mathcal{L}_{k+1} - \mathcal{U}_{k+1} \diamond_{\Phi} \mathcal{V}_{k+1}{}^{\mathrm{H}} \right)$$

and the penalty parameter μ_{k+1} is

$$\mu_{k+1} = \min\left(\rho\mu_k,\ \mu_{\max} \right) \tag{29}$$

3.2.5 Convergence judgment

We think the algorithm is convergent when the max value of

$$\left\| \mathcal{U}_{k+1} - \dot{\mathcal{U}}_{k+1} \right\|_{\mathrm{F}} / \| \mathcal{D} \|_{\mathrm{F}}$$

$$\left\| \mathcal{V}_{k+1} - \dot{\mathcal{V}}_{k+1} \right\|_{\mathrm{F}} / \| \mathcal{D} \|_{\mathrm{F}}$$

$$\left\| \mathcal{L}_{k+1} - \mathcal{U}_{k+1} \diamond_{\boldsymbol{\Phi}} \mathcal{V}_{k+1}^{\mathrm{H}} \right\|_{\mathrm{F}} / \| \mathcal{D} \|_{\mathrm{F}}$$

is smaller than the stop parameter ϵ.

We conclude the whole procedure in Algorithm 1.

Algorithm 1 ADMM for solving Eq.(11)

Input: $\mathcal{P}_{\Omega}(\mathcal{D}) \in \mathbb{C}^{n_1 \times n_2 \times n_3}$, the estimated rank d and λ

Initialize: μ_0, $\rho > 1$, $k = 0$, and ϵ.

(1) while not converged do

(2) Update $\mathcal{U}_{k+1}$ and $\mathcal{V}_{k+1}$ by Eq.(8) and Eq.(10)

(3) Compute $\dot{\mathcal{U}}_{k+1}$ and $\dot{\mathcal{V}}_{k+1}$ via the SVT[14]

(4) Update $\mathcal{L}_{k+1}$ by Eq.(27)

(5) Update the multipliers $\mathcal{Y}_i^{k+1}$ by Eq.(28)

(6) Update μ_{k+1} by Eq.(29)

(7) $k \leftarrow k+1$

(8) end while

Output: $\mathcal{U}_{k+1}$, $\mathcal{V}_{k+1}$ and $\mathcal{L}_{k+1}$.

4 Experimental Results

We evaluate the performance of DTTNN on image and video inpainting tasks and compare our methods with some start-of-the-art methods including D-N[5], TTNN[6], t-Schatten-q[4] and PSTNN[15], where D-N is the matrix completion method based on matrix factorization, TTNN the tensor completion method using data-based transformed tubal nuclear norm, t-Schatten-q the tensor completion method based on tensor factorization in the frequency domain and PSTNN the non-convex method by truncating smaller singular values. It is worth noting that when $\boldsymbol{\Phi}$ is DFT matrix, our method is equal to t-Schatten-q[4].

For the satisfactory performance of the data-based transform matrix $\boldsymbol{\Phi}$ verified by quantities of experiments in Refs.[6, 16], we use the data-based transform matrix in DTTNN and TTNN. It can be calculated by unfolding tensor data $\mathcal{X}$ into matrix $\boldsymbol{X}$ along the third-dimension, then take the singular value decomposition of the unfolded matrix $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{H}}$ and $\boldsymbol{U}^{\mathrm{H}}$ is the data-based transform matrix.

In our methods, basic parameters are set as follows: $\lambda = 1/\sqrt{\max(n_1, n_2) n_3}$, $\mu_0 = 1 \times 10^{-4}$, $\rho = 1.05$, $\epsilon = 1 \times 10^{-4}$, $d = [30, 50, 100]$ corresponds to missing rate (MR) MR = [95%, 85%, 70%] in image inpainting task and $\lambda = 5/\sqrt{\max(n_1, n_2) n_3}$, $\mu_0 = 1 \times 10^{-4}$, $\rho = 1.2$, $\epsilon = 1 \times 10^{-4}$, $d = [30, 50, 50]$ corresponds to MR = [90%, 80%, 70%] in video inpainting task. Besides, the parameters in other methods are consistent with their proposed papers. When it comes to matrix completion method D-N, we apply it to each frontal slice of color images. We use linear interpolation to pre-complete the input images. Peak signal-to-noise ratio (PSNR) (higher is better)[6] and running time are used to quantify the performance of different methods. PSNR is defined as follows

$$\mathrm{PSNR} = 10\log_{10} \frac{n_1 n_2 n_3 (\mathcal{X}_{\max} - \mathcal{X}_{\min})^2}{\| \mathcal{X} - \mathcal{X}_0 \|_{\mathrm{F}}^2} \quad (30)$$

where $\mathcal{X}$ is the recovered solution, and $\mathcal{X}_0$ the ground-truth tensor. $\mathcal{X}_{\max}$ and $\mathcal{X}_{\min}$ are the maximal and the minimal entries of $\mathcal{X}_0$, respectively.

4.1 Image inpainting

We use the six classic color images (shown in Fig.1.) to evaluate the performance of our proposed methods. From left to right, Image (1) 300×300×3, Image (2) 300×300×3, Image (3) 300×300×3, Image (4) 300×300×3, Image (5) 278×410×3, Image (6) 210×350×3. In the experiments, the input images are generated by setting randomly some pixels as missing entries, then applying different methods to recover them, respectively. PSNR and running time are given in Table 2, and the 1st place is bolded among tensor methods. All experiments are run in MATLAB R2021b under Windows 10 on a personal computer with a 2.60 GHz CPU and 16 GB memory.

(1) (2) (3) (4) (5) (6)

Fig.1 Six color images

Table 2 Average quantitative assessment index (PSNR/Time(s)) of different methods on the six color images

Image	MR=95%				
	D-N	TTNN	PSTNN	t-Schatten-1/2	DTTNN
(1)	18.51/0.78	20.02/9.156	17.95/4.22	20.72/3.01	**21.19/1.79**
(2)	17.99/0.70	18.90/8.64	18.94/4.19	19.13/3.38	**19.40/2.27**
(3)	19.72/0.78	20.35/8.17	**20.82**/4.09	20.43/3.59	20.38/**2.33**
(4)	14.30/0.67	16.55/8.68	16.26/4.17	17.68/3.43	**17.72/2.16**
(5)	19.49/0.69	20.09/10.41	19.30/4.81	20.94/3.78	**21.22/2.39**
(6)	23.46/0.52	23.72/6.51	23.41/2.83	23.21/3.00	**24.18/1.83**
Image	MR=85%				
	D-N	TTNN	PSTNN	t-Schatten-1/2	DTTNN
(1)	24.58/0.94	25.96/8.10	24.48/4.31	26.62/3.14	**27.02/2.07**
(2)	21.32/0.88	21.95/7.74	22.39/4.20	22.43/3.43	**23.00/2.05**
(3)	22.33/0.87	22.83/7.60	**23.37**/4.34	23.04/3.73	23.30/**2.11**
(4)	18.47/0.86	20.34/7.78	21.51/4.22	23.32/3.41	**23.65/2.14**
(5)	22.38/1.03	23.83/9.28	23.77/4.84	24.05/4.18	**24.59/2.54**
(6)	27.964/0.67	28.25/5.96	27.62/**2.77**	28.62/3.10	**29.00**/3.32
Image	MR=70%				
	D-N	TTNN	PSTNN	t-Schatten-1/2	DTTNN
(1)	26.36/1.23	30.04/7.94	31.36/4.26	31.47/5.23	**32.01/3.34**
(2)	22.44/1.22	25.10/7.63	26.57/4.28	26.51/5.32	**26.71/3.47**
(3)	23.41/1.21	25.53/7.58	**26.63/4.26**	26.44/5.47	26.23/4.70
(4)	19.03/1.43	24.78/7.83	28.81/4.07	28.90/5.22	**29.15/3.64**
(5)	23.63/1.45	26.17/9.24	**27.89**/4.77	27.37/6.44	27.67/**4.06**
(6)	29.86/1.03	**33.33**/5.72	33.07/**2.82**	33.27/5.19	33.14/5.13

As is shown in Table 2, D-N is a matrix method and others are tensor methods. It is obvious to see those tensor methods outperform D-N under PSNR which verifies the advantage of tensor methods. Hence, it is effective and necessary to deal with tensor data directly. In most cases, our DTTNN saves a great quantity of time than TTNN, PSTNN, and t-Schatten-1/2 when MR = [95%, 85%]. DTTNN also achieves competitive results when MR = 70%. This verifies the effectiveness of DTTNN. Besides, Fig.2 (from left to right, original image, input image with MR= 85%, D-N, TTNN, PSTNN, t-Schatten-1/2, DTTNN) presents the completion result of all methods when the input images have 85% random missing pixels. Actually, the visual effects don't vary a lot between different methods. But we have to emphasize that DTTNN takes less time to get this result especially when MR is very low. This advantage will be enlarged when dealing with larger tensors in the video inpainting tasks.

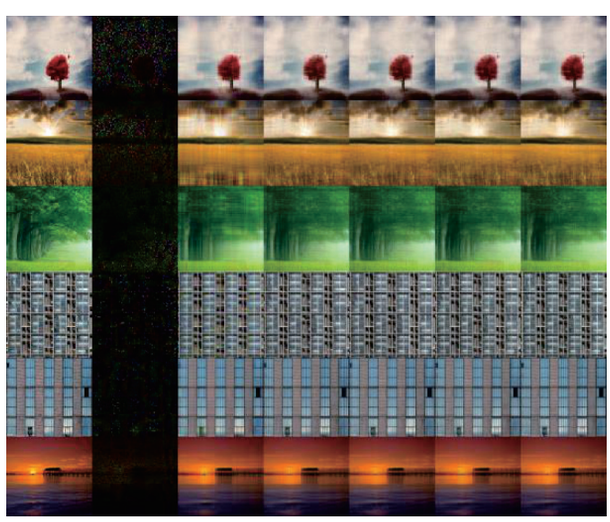

Fig.2 Completion results of color images

4.2 Video inpainting

In this part, we use the first 100 frames of three videos Suzie, Carphone, Container① to verify the performance of DTTNN. All videos are of size $144 \times 176 \times 100$.

Table 3 shows the average PSNR and running time of different methods. From Table 3, it is obviously seen that matrix method D-N does not perform

①http://trace.eas.asu.edu/yuv/

well although it runs fast. This again verifies the necessity of tensor methods. In all tensor methods, our DTTNN achieves the highest score in most cases and the running time is second only to D-N. DTTNN even saves more than twice as much time as TTNN and PSTNN. Additionally, the difference between DTTNN and t-Schatten-1/2 is larger than that in image cases. And this shows the advantage of DTTNN in dealing with large tensors. Fig.3 (from left to right, original image, input image with MR=90%, D-N, TTNN, PSTNN, t-Schatten-1/2, DTTNN) demonstrates the visual results of different methods when MR=90%. We can see that DTTNN achieves better visual effects compared with other methods.

Table 3 Average quantitative assessment index (PSNR/Time(s)) of different methods on the three videos

Video	MR=90%				
	D-N	TTNN	PSTNN	t-Schatten-1/2	DTTNN
Suzie	19.08/9.68	26.38/73.65	26.10/32.48	26.23/11.22	**27.26/11.03**
Carphone	16.20/7.00	26.65/78.06	26.10/32.46	25.87/10.93	**27.40/10.73**
Container	17.49/6.61	28.33/83.34	**31.27**/31.69	29.16/10.63	30.34/**10.22**
Video	MR=80%				
	D-N	TTNN	PSTNN	t-Schatten-1/2	DTTNN
Suzie	22.31/7.93	29.04/71.24	28.56/32.94	28.93/16.55	**29.81/16.17**
Carphone	19.57/8.65	29.41/72.09	28.61/32.59	28.69/16.06	**30.28/15.52**
Container	19.28/7.99	31.96/83.07	35.20/31.73	35.28/15.22	**36.71/14.09**
Video	MR=70%				
	D-N	TTNN	PSTNN	t-Schatten-1/2	DTTNN
Suzie	24.23/9.02	31.11/70.71	30.80/31.78	30.73/16.98	**31.86/16.25**
Carphone	20.92/9.36	31.54/69.02	30.74/31.91	30.19/16.54	**32.13/15.90**
Container	20.29/9.53	35.24/82.72	**39.69**/32.22	37.225/15.81	39.68/**14.16**

Fig.3 Completion results of the 20th frame of videos

5 Conclusions

In this article, we propose a non-convex tensor completion model. We first define a new TT-S_q and use it to replace the rank norm penalty in the LRTC problem. Then we prove the equivalence between TT-$S_{1/2}$ and DTTNN. Hence the non-convex problem is converted into a problem that each block is convex and can be solved under the multi-block ADMM framework. Finally, we verify the performance of DTTNN on color image inpainting.

Since the model proposed above is only applicable for suitable value q, we consider expanding the equivalence between DTTNN with TT-$S_{1/2}$ to any q values as our future work, and applying them to more LRTC tasks.

References

[1] ZHANG Zemin, ELY G, AERON S, et al. Novel methods for multilinear data completion and de-noising based on tensor-SVD[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. [S.l.]: IEEE, 2014: 3842-3849.

[2] XU Wenhao, ZHAO Xile, JI Tengyu, et al. Laplace function based nonconvex surrogate for low-rank tensor completion[J]. Signal Processing: Image Communication, 2019, 73: 62-69.

[3] SHI Chengfei, HUANG Zhengdong, WAN Li, et al. Low-rank tensor completion based on log-det rank approximation and matrix factorization[J]. Journal of Scientific Computing, 2019, 80(3): 1888-1912.

[4] KONG Hao, XIE Xingyu, LIN Zhouchen. *t*-Schatten-p norm for low-rank tensor recovery[J]. IEEE Journal of Selected Topics in Signal Processing, 2018, 12(6): 1405-1419.

[5] SHANG Fanhua, CHENG J, LIU Yuanyuan, et al. Bilinear factor matrix norm minimization for robust PCA: Algorithms and applications[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 40(9): 2066-2080.

[6] SONG Guangjing, NG M K, ZHANG Xiongjun. Robust tensor completion using transformed tensor singu-

lar value decomposition[J]. Numerical Linear Algebra with Applications, 2020, 27(3): e2299.

[7] LU Canyi, FENG Jiashi, CHEN Yudong, et al. Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. [S.l.]: IEEE, 2016: 5249-5257.

[8] HU Zhanxuan, NIE Feiping, WANG Rong, et al. Low rank regularization: A review[J]. Neural Networks, 2021, 136: 218-232.

[9] KERNFELD E, KILMER M, AERON S. Tensor-tensor products with invertible linear transforms[J]. Linear Algebra and Its Applications, 2015, 485: 545-570.

[10] SHANG Fanhua, LIU Yuanyuan, SHANG Fanjie, et al. A unified scalable equivalent formulation for schatten quasi-norms[J]. Mathematics, 2020, 8(8): 1325.

[11] YANG Xiaojing. Holder's inequality[J]. Applied Mathematics Letters, 2003, 16(6): 897-903.

[12] NEEDHAM T. A visual explanation of Jensen's inequality[J]. American Mathematical Monthly, 1993, 100(8): 768-71.

[13] WANG Fenghui, CAO Wenfei, Xu Zongben. Convergence of multi-block Bregman ADMM for nonconvex composite problems[J]. Science China Information Sciences, 2018, 61(12): 1-12.

[14] CAI Jianfeng, CANDÈS E J, SHEN Zuowei. A Singular value thresholding algorithm for matrix completion[J]. SIAM Journal on Optimization, 2010, 20(4): 1956-1982.

[15] JIANG Taixiang, HUANG Tingzhu, ZHAO Xile, et al. Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm[J]. Journal of Computational and Applied Mathematics, 2020, 372: 112680.

[16] ZENG Chao, NG M K. Decompositions of third-order tensors: HOSVD, T-SVD, and Beyond[J]. Numerical Linear Algebra with Applications, 2020, 27(3): e2290.

Acknowledgement This work was financially supported by the National Nautral Science Foundation of China (No. 61703206).

Authors Mr. **TIAN Jialue** received the B.S. degree from Yangzhou University in 2020. He is pursuing the M.S. degree in Nanjing University of Aeronautics and Astronautics, Nanjing, China. His current research is focused on tensor completion and pattern recognition.

Dr. **ZHU Yulian** received her M.S. and Ph.D. degree in Computer Application from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2004 and 2010, respectively. Since 2004, she has been working at the Computer Center, Nanjing University of Aeronautics and Astronautics. Her research interests include artificial intelligence, machine learning and image processing.

Author contributions Mr. **TIAN Jialue** conducted the experiment, interpreted the results and wrote the original manuscript. Dr. **ZHU Yulian** designed the study, revised and polished the manuscript. Ms. **LIU Jiahui** contributed to the formula derivation and results analysis. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: LIU Yandong)

基于双变换核范数最小化的张量补全

田甲略[1], 朱玉莲[2], 刘佳慧[1]

(1.南京航空航天大学计算机科学与技术学院/人工智能学院,南京 211106,中国;
2.南京航空航天大学公共实验教学部,南京 211106,中国)

摘要:非凸方法在低秩张张量补全中发挥着重要的作用,相比于凸方法,非凸方法是对张量秩的更紧估计。但是这类方法因为需要计算大张量的奇异值而非常耗时。本文提出了一种双变换核范数(Double transformed tubal neclear norm, DTTNN)作为秩范数的非凸松弛用于求解低秩张量补全(Low rank tensor completion, LRTC)问题。DTTNN将对大张量的非凸惩罚等价地转化为对两个小张量的凸惩罚。因此,DTTNN可以有效节省非凸方法的求解时间。彩色图像修复和视频修复的实验结果验证了DTTNN方法的有效性。

关键词:双张量核范数;低管秩;非凸优化;张量分解;张量补全