

A Low-Rank Total-Variation Regularized Tensor Completion Algorithm

Liangchen Song¹, Bo Du²(✉), Lefei Zhang², and Liangpei Zhang¹

¹ State Key Laboratory of Information Engineering in Survey, Mapping, and Remote Sensing, Wuhan University, Wuhan 430079, People's Republic of China

² School of Computer, Wuhan University, Wuhan, Hubei, China
gunspace@163.com

Abstract. In this paper, we first model visual data as a tensor and then impose both low-rank and total-variation constraint to complete the tensor. More specifically, we adopt a novel tensor-tensor production framework (also known as t-product) and its theory of low-rank based completion. By using the concept of t-product, it is the first time that we extend classic Total-Variation (TV) to a t-product and $l_{1,1,2}$ norm based constraint on the gradient of visual data. After proposing our model, we derive a iterative solver based on alternating direction method of multipliers (ADMM). We show the effectiveness of our method and compare our method with state-of-art algorithms in the experimental section.

Keywords: Low-rank tensor completion · Tensor nuclear norm
Total-variation · Visual data completion

1 Introduction

Recent years has witnessed the development of acquisition techniques, and the visual data tends to contain more and more information and thus should be treated as a complex high-dimensional data, i.e. tensors. For example, a hyperspectral image or multispectral image can be represented as a third-order tensor since the spatial information has two dimensions and the spectral information takes one dimension. Using a tensor to model visual data can handle the complex structure better and has become a hot topic in computer vision community recently, for instance, face recognition [15], color image and video in-painting [7], hyperspectral image processing [16,17], gait recognition [12].

Visual data may have missing values in the acquisition process due to mechanical failure or man-induced factors. If we process the multi-way array data as a tensor instead of splitting it into matrices, estimating missing values for visual data is also known as tensor completion problem. Motivated by the successful achievements of low-rank matrix completion methods [8,10], many tensor completion problems could also be solved through imposing low-rank constraint.

Decomposition and subspace methods has been widely studied [13,14] and a common way to impose low-rank constraint is to decompose tensors and get its factors first, then by limiting the sizes of factors we attain low-rank property. CP decomposition and Tucker decomposition are two classic decomposition methods to compute the factors of a tensor [5]. However, both the two methods mentioned above have crucial parameters that are supposed to be determined by users. In this paper, we adopt a novel tensor-tensor product framework [1], and a singular value decomposition formula for tensors [3,4], which is parameter-free. The recently proposed singular value decomposition formula for tensors is also referred to as t-SVD, and t-SVD has been proved effective by many influential papers [18].

When trying to restore degraded visual data, it is reasonable to utilize the local smoothness property of visual data as prior knowledge or regularization. A common constraint is Total-Variation (TV) norm, which is computed as the l_1 of the gradient magnitude, and this norm has been qualified to be effective to preserve edges and piecewise structures. If we take the low-rank assumption as a global constraint and TV norm as a local prior, then it is natural to combine this two constraints together to estimate missing values in visual data. So more recently, some related methods that take tensor and TV into consideration are proposed [2,6,11]. [2] aims to design a norm that considers both the inhomogeneity and the multi-directionality of responses to derivative-like filters. [11] proposes an image super-resolution method that integrates both low-rank matrix completion and TV regularization. [6] proposes integrating TV into low-rank tensor completion, but their methods use the same rank definition as [7] or Tucker decomposition, and as mentioned before, the parameters are hard to deal with. Moreover, TV is l_1 norm on matrix but we are processing visual tensors, so we need to extend TV to tensor cases. Also, we employ a new tensor norm namely $l_{1,1,2}$ which stems from multi-task learning [9].

In this paper, we seek to design a parameter-free method for low-rank based completion procedure, which means parameters only exist in TV regularization. To achieve this goal, simply using t-SVD framework is not enough, since t-SVD is based on t-product and a free module so we need to reinvent TV with t-product system. Our main contributions are as follows:

1. Using the concept of t-product, we design difference tensors \mathcal{A} and \mathcal{B} that when multiplied by a visual tensor \mathcal{X} takes the gradient of \mathcal{X} , i.e. $\nabla\mathcal{X}$.
2. Motivated by $l_{1,1,2}$ norm which is originally designed to model sparse noise in visual data [18], we use $l_{1,1,2}$ norm to ensure sparsity of the gradient magnitude, then traditional TV is extended to tensor cases.

2 Notations and Preliminaries

In this section, we introduce some notations of tensor algebra and give the basic definitions used in the rest of the paper.

Scalars are denoted by lowercase letters, e.g., a . Vectors are denoted by bold-face lowercase letters, e.g., \mathbf{a} . Matrices are denoted by capital letters, e.g., A .

Higher-order tensors are denoted by boldface Euler script letters, e.g., \mathcal{A} . For a third-order tensor $\mathcal{A} \in \mathbb{R}^{r \times s \times t}$, the i th frontal slice is denoted A_i . In terms of MATLAB indexing notation, we have $A_i = \mathcal{A}(:, :, i)$.

The definition of new tensor multiplication strategy [1] begins with converting $\mathcal{A} \in \mathbb{R}^{r \times s \times t}$ into a block circulant matrix. Then

$$\text{bcirc}(\mathcal{A}) = \begin{pmatrix} A_1 & A_n & A_{n-1} & \cdots & A_2 \\ A_2 & A_1 & A_n & \cdots & A_3 \\ A_3 & A_2 & A_1 & \cdots & A_4 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_n & A_{n-1} & A_{n-2} & \cdots & A_1 \end{pmatrix}$$

is a block circulant matrix of size $rt \times st$. And the `unfold` command rearrange the frontal slices of \mathcal{A} :

$$\text{unfold}(\mathcal{A}) = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{pmatrix}, \quad \text{fold}(\text{unfold}(\mathcal{A})) = \mathcal{A}$$

Then we have the following new definition of tensor-tensor multiplication.

Definition 1 (t-product). Let $\mathcal{A} \in \mathbb{R}^{r \times s \times t}$ and $\mathcal{B} \in \mathbb{R}^{s \times p \times t}$. Then the t -product $\mathcal{A} * \mathcal{B}$ is a $r \times p \times t$ tensor

$$\mathcal{A} * \mathcal{B} = \text{fold}(\text{bcirc}(\mathcal{A}) \cdot \text{unfold}(\mathcal{B})) \quad (1)$$

An important property of the block circulant matrix is the observation that a block circulant matrix can be block diagonalized in the Fourier domain [3]. Before moving on to the definition of t-SVD, we need some more definitions from [4].

Definition 2 (Tensor Transpose). Let $\mathcal{A} \in \mathbb{R}^{r \times s \times t}$, then $\mathcal{A}^T \in \mathbb{R}^{s \times r \times t}$ and

$$\mathcal{A}^T = \text{fold}([A_1, A_n, A_{n-1}, \dots, A_2]^T)$$

Definition 3 (Identity Tensor). The identity tensor $\mathcal{I} \in \mathbb{R}^{m \times m \times n}$ is the tensor whose first frontal slice is the $m \times m$ identity matrix, and whose other frontal slices are all zeros.

Definition 4 (Orthogonal Tensor). A tensor $\mathcal{Q} \in \mathbb{R}^{m \times m \times n}$ is orthogonal if $\mathcal{Q}^T * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^T = \mathcal{I}$.

Definition 5 (f-diagonal Tensor). A tensor is called f -diagonal if each of its frontal slices is a diagonal matrix.

Definition 6 (Inverse Tensor). A tensor $\mathcal{A} \in \mathbb{R}^{m \times m \times n}$ has an inverse tensor \mathcal{B} if

$$\mathcal{A} * \mathcal{B} = \mathcal{I} \quad \text{and} \quad \mathcal{B} * \mathcal{A} = \mathcal{I}$$

Using these new definitions, we are able to derive a new decomposition method named t-SVD and an approximation theorem on this decomposition.

Theorem 1 (t-SVD). *Let $\mathcal{A} \in \mathbb{R}^{r \times s \times t}$, then the t-SVD of \mathcal{A} is*

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T \tag{2}$$

where $\mathcal{U} \in \mathbb{R}^{r \times r \times t}$, $\mathcal{V} \in \mathbb{R}^{s \times s \times t}$ are orthogonal and $\mathcal{S} \in \mathbb{R}^{r \times s \times t}$ is f-diagonal.

By using the idea of computing in the Fourier domain, we can efficiently compute the t-SVD factorization, and for more details see [3,4]. Then nuclear norm for \mathcal{A} is defined as $\|\mathcal{A}\|_* = \sum_{j=1}^t \sum_{i=1}^{\min(r,s)} |\hat{\mathcal{S}}(i, i, k)|$, where $\hat{\mathcal{S}}$ is the result of taking the Fourier transform along the third dimension of \mathcal{S} .

3 Proposed t-SVD-TV

In this section, we will first give the details of our model, and then derive a solver with alternating direction method of multipliers (ADMM). Since we combine t-SVD and TV together, we name our method as t-SVD-TV.

3.1 Low-Rank Regularization

Before discussing low-rank model, we need some notations for our problem first. Suppose there is a tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ representing some visual data with missing entries, and we use Ω indicating the set of indices of observations, \mathcal{X} denoting the desired recovery result, then we have

$$\mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{M})$$

where $\mathcal{P}_\Omega(\cdot)$ is the projector onto the known indices Ω . So we have the following vanilla model [18]:

$$\begin{aligned} \min_{\mathcal{X}} \quad & \|\mathcal{X}\|_* \\ \text{s.t.} \quad & \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{M}) \end{aligned} \tag{3}$$

Our method takes both low-rank and TV into consideration, which means we also have terms to ensure sparsity on gradient field. We use $\Psi(\cdot)$ as a function promoting sparsity, so our model is

$$\begin{aligned} \min_{\mathcal{X}} \quad & \|\mathcal{X}\|_* + \lambda_1 \Psi\left(\frac{\partial}{\partial_x} \mathcal{X}\right) + \lambda_2 \Psi\left(\frac{\partial}{\partial_y} \mathcal{X}\right) \\ \text{s.t.} \quad & \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{M}) \end{aligned} \tag{4}$$

where λ_1, λ_2 are tunable parameters. The forms of $\frac{\partial}{\partial_x} \mathcal{X}$, $\frac{\partial}{\partial_y} \mathcal{X}$ and $\Psi(\cdot)$ are given in the following texts.

3.2 TV Regularization

When we need to compute the gradient matrix of a 2D image M , a common way is to design a *difference matrix* A, B . If we assume vertical direction is x and horizontal direction is y , then from AM we get $\frac{\partial}{\partial x}M$ and from MB we get $\frac{\partial}{\partial y}M$. And for a 3D data \mathcal{X} we can derive similar means implemented by t-product system. Without loss of generality, we assume $\mathcal{X} \in \mathbb{R}^{n \times n \times k}$. By extending difference matrices to tensors, we define a difference tensor as follows:

$$\mathcal{A} = \text{fold} \begin{pmatrix} A \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \mathcal{B} = \text{fold} \begin{pmatrix} B \\ 0 \\ 0 \end{pmatrix}$$

where A and B are $n \times n$ difference matrix:

$$A = \frac{1}{2} \begin{pmatrix} -2 & 2 & 0 & \cdots & 0 & 0 \\ -1 & 0 & 1 & \cdots & 0 & 0 \\ 0 & -1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & -2 & 2 \end{pmatrix} \quad \text{and} \quad B = \frac{1}{2} \begin{pmatrix} -2 & -1 & 0 & \cdots & 0 & 0 \\ 2 & 0 & -1 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & -2 \\ 0 & 0 & 0 & \cdots & 1 & 2 \end{pmatrix}$$

For a visual tensor \mathcal{X} , we get its gradient tensor by multiplying with \mathcal{A}, \mathcal{B}

$$\frac{\partial}{\partial x} \mathcal{X} = \mathcal{A} * \mathcal{X}, \quad \frac{\partial}{\partial y} \mathcal{X} = \mathcal{X} * \mathcal{B} \quad (5)$$

Note that $\mathcal{A} * \mathcal{X}, \mathcal{X} * \mathcal{B}$ are third order tensors which have the same size as \mathcal{X} . In order to promote sparsity of the gradient, we use the $l_{1,1,2}$ norm for 3D tensors as penalty function $\Psi(\cdot)$. $l_{1,1,2}$ norm is introduced in [18] to model the sparse noise, and for a third order tensor \mathcal{G} , $\|\mathcal{G}\|_{1,1,2}$ is defined as $\sum_{i,j} \|\mathcal{G}(i, j, \cdot)\|_F$. Then our optimization problem (4) becomes

$$\begin{aligned} \min_{\mathcal{X}} \quad & \|\mathcal{X}\|_* + \lambda_1 \|\mathcal{A} * \mathcal{X}\|_{1,1,2} + \lambda_2 \|\mathcal{X} * \mathcal{B}\|_{1,1,2} \\ \text{s.t.} \quad & \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{M}) \end{aligned} \quad (6)$$

The terms in (6) are interdependent, so we adopt a widely used splitting scheme which is known as ADMM.

3.3 Optimization by ADMM

The first step of applying ADMM is introducing auxiliary variables. Specifically, let $\mathcal{S}, \mathcal{Y}, \mathcal{Z}_1, \mathcal{Z}_2$ have the same size as \mathcal{X} , then our optimization problem (6) becomes

$$\begin{aligned} \min_{\mathcal{X}} \quad & \|\mathcal{S}\|_* + \lambda_1 \|\mathcal{Y}_1\|_{1,1,2} + \lambda_2 \|\mathcal{Y}_2\|_{1,1,2} \\ \text{s.t.} \quad & \mathcal{Y}_1 = \mathcal{A} * \mathcal{Z}_1, \mathcal{Y}_2 = \mathcal{Z}_2 * \mathcal{B} \\ & \mathcal{S} = \mathcal{X}, \mathcal{Z}_1 = \mathcal{X}, \mathcal{Z}_2 = \mathcal{X}, \\ & \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{M}) \end{aligned} \quad (7)$$

So the augmented Lagrangian is

$$\begin{aligned}
 \mathcal{L} = & \|\mathcal{S}\|_* + \frac{\rho_1}{2} \left\| \mathcal{S} - \mathcal{X} + \frac{\mathcal{U}}{\rho_1} \right\|_F^2 \\
 & + \lambda_1 \|\mathcal{Y}_1\|_{1,1,2} + \frac{\rho_2}{2} \left\| \mathcal{Y}_1 - \mathcal{A} * \mathcal{Z}_1 + \frac{\mathcal{V}_1}{\rho_2} \right\|_F^2 \\
 & + \lambda_2 \|\mathcal{Y}_2\|_{1,1,2} + \frac{\rho_3}{2} \left\| \mathcal{Y}_2 - \mathcal{Z}_2 * \mathcal{B} + \frac{\mathcal{V}_2}{\rho_3} \right\|_F^2 \\
 & + \frac{\rho_4}{2} \left\| \mathcal{Z}_1 - \mathcal{X} + \frac{\mathcal{W}_1}{\rho_4} \right\|_F^2 + \frac{\rho_5}{2} \left\| \mathcal{Z}_2 - \mathcal{X} + \frac{\mathcal{W}_2}{\rho_5} \right\|_F^2 \\
 \text{s.t. } & \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{M})
 \end{aligned} \tag{8}$$

where tensors $\mathcal{U}, \mathcal{V}_1, \mathcal{V}_2, \mathcal{W}_1, \mathcal{W}_2$ are Lagrange multipliers and $\rho_i (i = 1, \dots, 5)$ are positive numbers. We solve (8) by alternatively minimize each variable, so (8) are turned into several subproblems:

Computing \mathcal{S} . The subproblem for minimize \mathcal{S} is

$$\mathcal{S}^{k+1} = \arg \min_{\mathcal{S}} \|\mathcal{S}\|_* + \frac{\rho_1}{2} \left\| \mathcal{S} - \mathcal{X}^k + \frac{\mathcal{U}^k}{\rho_1} \right\|_F^2 \tag{9}$$

Note that (9) can be solved by the singular value thresholding method in [18], so

$$\mathcal{S}^{k+1} := \arg \min_{\mathcal{S}} \|\mathcal{S}\|_* + \frac{\rho_1}{2} \left\| \mathcal{S} - \mathcal{X}^k + \frac{\mathcal{U}^k}{\rho_1} \right\|_F^2 \tag{10}$$

Computing \mathcal{Y}_1 and \mathcal{Y}_2 . The subproblem for minimize \mathcal{Y}_1 is

$$\mathcal{Y}_1^{k+1} = \arg \min_{\mathcal{Y}_1} \lambda_1 \|\mathcal{Y}_1\|_{1,1,2} + \frac{\rho_2}{2} \left\| \mathcal{Y}_1 - \mathcal{A} * \mathcal{Z}_1^k + \frac{\mathcal{V}_1^k}{\rho_2} \right\|_F^2 \tag{11}$$

The closed form solution to (11) is given by

$$\mathcal{Y}_1^{k+1} := \left(1 - \frac{\lambda_1}{\rho_2} \left\| \mathcal{A} * \mathcal{Z}_1^k + \frac{\mathcal{V}_1^k}{\rho_2} \right\|_F^{-1} \right)_+ \left(\mathcal{A} * \mathcal{Z}_1^k + \frac{\mathcal{V}_1^k}{\rho_2} \right) \tag{12}$$

where $(x)_+ = \max(x, 0)$. Updating strategy for \mathcal{Y}_2 is similar to \mathcal{Y}_1 , so

$$\mathcal{Y}_2^{k+1} := \left(1 - \frac{\lambda_2}{\rho_3} \left\| \mathcal{A} * \mathcal{Z}_1^k + \frac{\mathcal{V}_1^k}{\rho_3} \right\|_F^{-1} \right)_+ \left(\mathcal{Z}_2^k * \mathcal{B} + \frac{\mathcal{V}_2^k}{\rho_3} \right) \tag{13}$$

Input: an incomplete tensor \mathcal{M} , index of known entries Ω , iteration number N , parameters λ_1, λ_2 and $\rho_i (i = 1, \dots, 5)$

Output: a tensor \mathcal{X} after completion

- 1 Set $\mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{M})$, $\mathcal{P}_{\bar{\Omega}}(\mathcal{X}) = 0$, randomly initialize $\mathcal{S}, \mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Z}_1, \mathcal{Z}_2$;
- 2 **for** $n = 1$ to N **do**
- 3 Update \mathcal{S} by singular value thresholding [18];
- 4 Update $\mathcal{Y}_1, \mathcal{Y}_2$ by Eq.(12) and Eq.(13) respectively;
- 5 Update $\mathcal{Z}_1, \mathcal{Z}_2$ by Eq.(15) and Eq.(16) respectively;
- 6 Update \mathcal{X} by Eq.(18);
- 7 Update multipliers by

$$\begin{aligned} \mathcal{U}^{k+1} &:= \mathcal{U}^k + \rho_1(\mathcal{S}^k - \mathcal{X}^k) \\ \mathcal{V}_1^{k+1} &:= \mathcal{V}_1^k + \rho_2(\mathcal{Y}_1^k - \mathcal{A} * \mathcal{Z}_1^k) \\ \mathcal{V}_2^{k+1} &:= \mathcal{V}_2^k + \rho_3(\mathcal{Y}_2^k - \mathcal{Z}_2^k * \mathcal{B}) \\ \mathcal{W}_1^{k+1} &:= \mathcal{W}_1^k + \rho_4(\mathcal{Z}_1^k - \mathcal{X}^k) \\ \mathcal{W}_2^{k+1} &:= \mathcal{W}_2^k + \rho_5(\mathcal{Z}_2^k - \mathcal{X}^k) \end{aligned}$$
- 8 **end**
- 9 Return \mathcal{X} .

Algorithm 1. t-SVD-TV

Computing \mathcal{Z}_1 and \mathcal{Z}_2 . The subproblem for minimize \mathcal{Z}_1 is

$$\mathcal{Z}_1^{k+1} = \arg \min_{\mathcal{Z}_1} \frac{\rho_2}{2} \left\| \mathcal{Y}_1^k - \mathcal{A} * \mathcal{Z}_1 + \frac{\mathcal{V}_1^k}{\rho_2} \right\|_F^2 + \frac{\rho_4}{2} \left\| \mathcal{Z}_1 - \mathcal{X} + \frac{\mathcal{W}_1^k}{\rho_2} \right\|_F^2 \quad (14)$$

Then Update \mathcal{Z}_1 and \mathcal{Z}_2 by

$$\mathcal{Z}_1^{k+1} := (\rho_4 \mathcal{I} + \rho_2 \mathcal{A}^T * \mathcal{A})^{-1} * (\rho_4 \mathcal{I} - \mathcal{W}_1^k + \rho_2 \mathcal{A}^T * \mathcal{Y}_1^k + \mathcal{A}^T * \mathcal{W}_1^k) \quad (15)$$

Similarly, we have update rules for \mathcal{Z}_2 :

$$\mathcal{Z}_2^{k+1} := (\rho_5 \mathcal{I} - \mathcal{W}_2 + \rho_3 \mathcal{Y}_2^k * \mathcal{B}^T + \mathcal{V}_2^k * \mathcal{B}^T) * (\rho_5 \mathcal{I} + \rho_3 \mathcal{B} * \mathcal{B}^T)^{-1} \quad (16)$$

Note that the calculation of inverse tensors in (15) and (16) are not hard, since only the first front slices of $\rho_4 \mathcal{I} + \rho_2 \mathcal{A}^T * \mathcal{A}$ and $\rho_5 \mathcal{I} + \rho_3 \mathcal{B} * \mathcal{B}^T$ are non-zero.

Computing \mathcal{X} . The subproblem for \mathcal{X} is

$$\begin{aligned} \mathcal{X}^{k+1} = \arg \min_{\mathcal{X}} & \frac{\rho_1}{2} \left\| \mathcal{S}^k - \mathcal{X} + \frac{\mathcal{U}^k}{\rho_1} \right\|_F^2 + \frac{\rho_4}{2} \left\| \mathcal{Z}_1^k - \mathcal{X}^k + \frac{\mathcal{W}_1^k}{\rho_4} \right\|_F^2 \\ & + \frac{\rho_5}{2} \left\| \mathcal{Z}_2^k - \mathcal{X}^k + \frac{\mathcal{W}_2^k}{\rho_5} \right\|_F^2 \end{aligned} \quad (17)$$

So update \mathcal{X} by

$$\mathcal{P}_{\bar{\Omega}}(\mathcal{X}) := \left[\frac{1}{\rho_1 + \rho_4 + \rho_5} (\rho_1 \mathcal{X}^k + \mathcal{U}^k + \rho_4 \mathcal{Z}_1^k + \mathcal{W}_1^k + \rho_5 \mathcal{Z}_2^k + \mathcal{W}_2^k) \right]_{\bar{\Omega}} \quad (18)$$

$$\mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{M})$$

Updating Multipliers

$$\begin{aligned} \mathcal{U}^{k+1} &:= \mathcal{U}^k + \rho_1 (\mathcal{X}^k - \mathcal{X}^k) \\ \mathcal{V}_1^{k+1} &:= \mathcal{V}_1^k + \rho_2 (\mathcal{Y}^k - \mathcal{A} * \mathcal{Z}_1^k) \\ \mathcal{V}_2^{k+1} &:= \mathcal{V}_2^k + \rho_3 (\mathcal{Y}^k - \mathcal{Z}_2^k * \mathcal{B}) \\ \mathcal{W}_1^{k+1} &:= \mathcal{W}_1^k + \rho_4 (\mathcal{Z}_1^k - \mathcal{X}^k) \\ \mathcal{W}_2^{k+1} &:= \mathcal{W}_2^k + \rho_5 (\mathcal{Z}_2^k - \mathcal{X}^k) \end{aligned} \quad (19)$$

With these update formulae, we conclude the solver in Algorithm 1.

4 Experiments

In this section, we evaluate our methods on eight benchmark RGB-color images with different types of missing entries. The eight benchmark images are showed in Fig. 1. Each image is 256×256 and has 3 color channels, so it is a $256 \times 256 \times 3$ tensor. We compare our method (t-SVD-TV), with five state-of-art methods: HaLRTC [7], FBCP [19], t-SVD [18], LRTC-TV-I and LRTC-TV-II [6]. Relative Square Error (RSE) and Peak Signal to Noise Ratio (PSNR) are used to

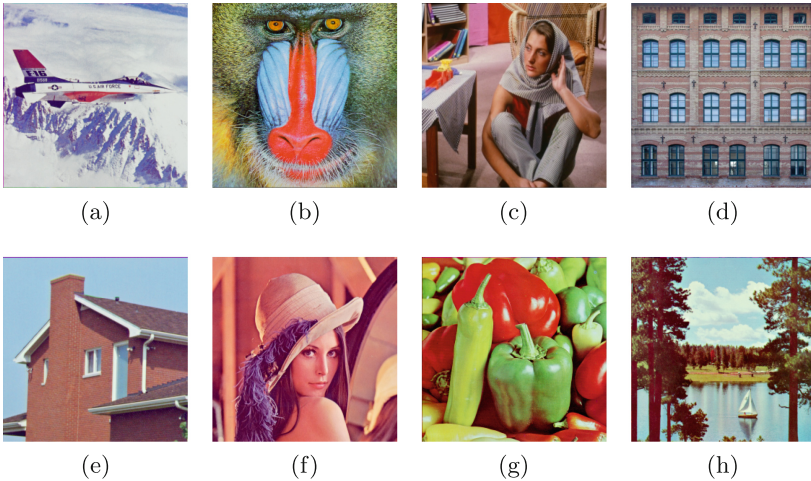


Fig. 1. RGB-color images used in our experiment. From left to right they are (a) Airplane (b) Baboon (c) Barbara (d) Facade (e) House (f) Lena (g) Peppers (h) Sailboat

assess the recovery result, and if we denote true data as \mathcal{T} , RSE and RSNR are defined as

$$\text{RSE} = \frac{\|\mathcal{X} - \mathcal{T}\|_F}{\|\mathcal{T}\|_F} \quad (20)$$

$$\text{PSNR} = 10 \log_{10} \frac{\mathcal{T}_{\max}^2}{\|\mathcal{X} - \mathcal{T}\|_F^2} \quad (21)$$

where \mathcal{T}_{\max} is the maximum value in \mathcal{T} . Better recovery result will have a smaller RSE and larger PSNR.

Parameter Settings. The key parameters in our methods are λ_1 and λ_2 . Since they are balancing the weights of vertical and horizontal gradient, and in general vertical and horizontal gradient are of the same importance, so we set $\lambda_1 = \lambda_2$. And by experience, we set $\lambda_1 = \lambda_2 = 0.01$ in our experiments. Other parameters such as $\rho_i (i = 1, \dots, 5)$ are concerned with the convergence property of the algorithm, and in our experiments we set $\rho_1 = \rho_2 = 0.001$, $\rho_3 = \rho_4 = \rho_5 = 0.1$.

Color Image Inpainting. We first compare our methods with the five state-of-art methods under different missing rates. We use Baboon in Fig. 1 to illustrate the different inpainting performances with random missing entries. As is shown in Fig. 2, our method performs well for both low and high missing rates while others have their limitations. For example, FBCP only performs well when missing rate is high while results of LRTC-TV-I and LRTC-TV-II are not so good with such a high missing rate.

Then we test our method with the eight images in Fig. 1 with 50% random missing entries. The results is shown in Table 1, from which we can see that our method outperforms others in most pictures and only the result on Peppers is an exception.

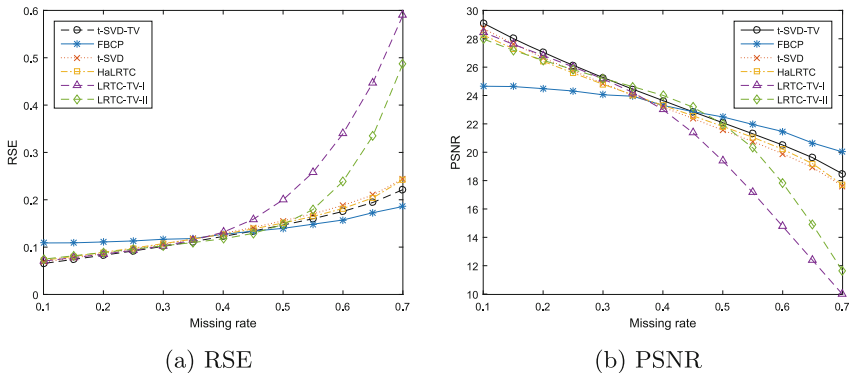
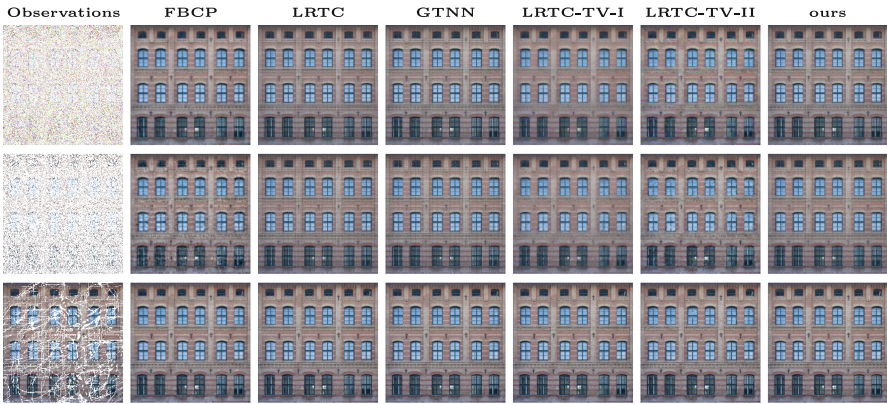


Fig. 2. Result of recovering Baboon with random missing entries

Table 1. Result of recovering different images with 50% missing entries

		Airplane	Baboon	Barbara	Facade	House	Lena	Peppers	Sailboat
PSNR	Ours	31.1781	25.2686	31.5562	35.0539	33.4708	31.6148	29.4887	27.9553
	FBCP	29.5188	24.0503	29.3032	29.7581	31.3138	30.6669	29.1078	27.0701
	t-SVD	30.4749	24.8544	30.8871	34.5542	32.5710	30.8560	28.6451	27.3461
	HaLRTC	30.2067	24.7912	29.7100	33.5470	32.0453	30.0387	29.2286	27.6037
	LRTC-TV-I	29.7039	25.1704	30.3483	30.9438	32.6083	30.9768	30.9704	27.9658
	LRTC-TV-II	29.6746	25.2043	29.8204	27.8308	32.4180	31.0287	31.2462	27.9656
RSE	Ours	0.0392	0.1016	0.0554	0.0342	0.0363	0.0480	0.0680	0.0756
	FBCP	0.0463	0.1166	0.0717	0.0630	0.0466	0.0528	0.0699	0.0819
	t-SVD	0.0423	0.1064	0.0599	0.0363	0.0403	0.0523	0.0748	0.0805
	HaLRTC	0.0440	0.1073	0.0685	0.0407	0.0430	0.0574	0.0703	0.0796
	LRTC-TV-I	0.0463	0.1028	0.0635	0.0550	0.0404	0.0514	0.0573	0.0767
	LRTC-TV-II	0.0463	0.1026	0.0675	0.0787	0.0417	0.0511	0.0559	0.0766

**Fig. 3.** Comparison of inpainting image Facade. The first line is random missing entries with rate 30%. The second line is random missing pixels with rate 30%. The third line is simulated scratches.

Finally, we present the visual effect of the inpainting algorithms in Fig. 3. The first line is random missing entries with rate 30%. The second line is random missing pixels with rate 30%. The third line is simulated scratches. And we can see that our method performs well in local details while preserving global structures.

5 Conclusions

In this paper, we aim to take both low-rank and total-variation constraint into consideration to complete visual data with missing entries. We propose a novel tensor named gradient tensor by using the t-product framework, then we fuse the novel tensor framework and classic TV together and we verify the effectiveness of our method by experiments. Our future work will focus on design a tensor framework directly that ensure global low-rank property and local smoothness.

Acknowledgements. This work was supported in part by the National Natural Science Foundation of China under Grants U1536204, 60473023, 61471274, and 41431175, China Postdoctoral Science Foundation under Grant No. 2015M580753.

References

1. Braman, K.: Third-order tensors as linear operators on a space of matrices. *Linear Algebra Appl.* **433**(7), 1241–1253 (2010)
2. Guo, X., Ma, Y.: Generalized tensor total variation minimization for visual data recovery. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015
3. Kilmer, M.E., Braman, K., Hao, N., Hoover, R.C.: Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging. *SIAM J. Matrix Anal. Appl.* **34**(1), 148–172 (2013)
4. Kilmer, M.E., Martin, C.D.: Factorization strategies for third-order tensors. *Linear Algebra Appl.* **435**(3), 641–658 (2011)
5. Kolda, T.G., Bader, B.W.: Tensor decompositions and applications. *SIAM Rev.* **51**(3), 455–500 (2009)
6. Li, X., Ye, Y., Xu, X.: Low-rank tensor completion with total variation for visual data inpainting. In: *Thirty-First AAAI Conference on Artificial Intelligence* (2017)
7. Liu, J., Musialski, P., Wonka, P., Ye, J.: Tensor completion for estimating missing values in visual data. *IEEE Trans. Patt. Anal. Mach. Intell.* **35**(1), 208–220 (2013)
8. Liu, T., Tao, D.: On the performance of manhattan nonnegative matrix factorization. *IEEE Trans. Neural Netw. Learn. Syst.* **27**(9), 1851 (2016)
9. Liu, T., Tao, D., Song, M., Maybank, S.J.: Algorithm-dependent generalization bounds for multi-task learning. *IEEE Trans. Patt. Anal. Mach. Intell.* **39**(2), 227 (2017)
10. Recht, B.: A simpler approach to matrix completion. *J. Mach. Learn. Res.* **12**, 3413–3430 (2011)
11. Shi, F., Cheng, J., Wang, L., Yap, P.T., Shen, D.: LRTV: MR image super-resolution with low-rank and total variation regularizations. *IEEE Trans. Med. Imaging* **34**(12), 2459–2466 (2015)
12. Tao, D., Li, X., Wu, X., Maybank, S.J.: General tensor discriminant analysis and gabor features for gait recognition. *IEEE Trans. Patt. Anal. Mach. Intell.* **29**(10), 1700–1715 (2007)
13. Tao, D., Li, X., Wu, X., Maybank, S.J.: Geometric mean for subspace selection. *IEEE Trans. Patt. Anal. Mach. Intell.* **31**(2), 260–274 (2009)
14. Tao, D., Tang, X., Li, X., Wu, X.: Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. *IEEE Trans. Patt. Anal. Mach. Intell.* **28**(7), 1088–1099 (2006)
15. Yan, S., Xu, D., Yang, Q., Zhang, L., Tang, X., Zhang, H.J.: Multilinear discriminant analysis for face recognition. *IEEE Trans. Image Process.* **16**(1), 212–220 (2007)
16. Zhang, L., Zhang, L., Tao, D., Huang, X.: A multifeature tensor for remote-sensing target recognition. *IEEE Geosci. Remote Sens. Lett.* **8**(2), 374–378 (2011)
17. Zhang, L., Zhang, L., Tao, D., Huang, X.: Tensor discriminative locality alignment for hyperspectral image spectral-spatial feature extraction. *IEEE Trans. Geosci. Remote Sens.* **51**(1), 242–256 (2013)

18. Zhang, Z., Ely, G., Aeron, S., Hao, N., Kilmer, M.: Novel methods for multilinear data completion and de-noising based on Tensor-SVD. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3842–3849. IEEE (2014)
19. Zhao, Q., Zhang, L., Cichocki, A.: Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE Trans. Patt. Anal. Mach. Intell.* **37**(9), 1751–1763 (2015)