

Article

Not peer-reviewed version

Low Tensor Rank Constrained Image Inpainting Using a Novel Arrangement Scheme

[Shuli Ma](#)^{*}, [Youchen Fan](#), [Shengliang Fang](#), Weichao Yang, Li Li

Posted Date: 24 October 2023

doi: 10.20944/preprints202310.1523.v1

Keywords: image inpainting; tensor decomposition; rearrangement scheme; unfolding matrix; alternating direction multiplier method



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Low Tensor Rank Constrained Image Inpainting Using a Novel Arrangement Scheme

Shuli Ma ^{1,*}, Youchen Fan ¹, Shengliang Fang ¹, Weichao Yang ² and Li Li ²

¹ School of Space Information, Space Engineering University, Beijing, 101400, China

² China Academy of Space Technology (Xi'an), Xi'an, Shanxi 710100, China

* Correspondence: shulima63@163.com

Abstract: Employing low tensor rank decompositions in image inpainting has attracted increasing attention. This paper exploits a novel tensor-augmentation schemes to transform an image (a low-order tensor) to a higher-order tensor without changing the total number of pixels. The developed augmentation schemes enhance the low-rankness of an image under three tensor decompositions: matrix SVD, tensor train (TT) decomposition, and tensor singular value decomposition (t-SVD). By exploiting the schemes, we solve the image inpainting problem with three low-rank constrained models which use the matrix rank, TT rank, and tubal rank as constrained priors respectively. The tensor tubal rank and tensor train multi-rank are developed from t-SVD and TT decomposition respectively. We exploit efficient ADMM algorithms for solving the three models. Experimental results demonstrate that our methods are effective for image inpainting and superior to numerous close methods.

Keywords: image inpainting; tensor decomposition; rearrangement scheme; unfolding matrix; alternating direction multiplier method

1. Introduction

Image inpainting refers to the process of completing missing entries or restoring damaged regions of an image. It is a typical ill-posed inverse problem, generally solved by exploiting the image priors [1,2], such as smoothness, sparsity, and low rankness. In recent years, tensor analysis including tensor low-rank decomposition and tensor completion, has attracted increasing attention [3–6]. A color image itself is an order-3 tensor, or it can be used to construct a high order (greater than 3) tensor, then the image inpainting problem becomes a tensor completion problem. A tensor is more challenging to analyze than a matrix due to the complicated nature of higher-order arrays [7]. We can constrain the low tensor rank to recover the missing pixels. The effectiveness relies on the tensor rank. The lower the tensor rank is, the better the recovery results are. Thus, finding ways to decrease the tensor rank is essential in the tensor completion problem. Unlike matrix rank, the definition of tensor rank is not unique, and relates to the tensor decomposition scheme.

Low tensor-rank completion methods can be categorized according to the tensor decomposition frameworks they use [8]. The traditional tensor decomposition tools include CANDECOMP/PARAFAC (CP), and Tucker decomposition [8,9]. The recently proposed decomposition frameworks include tensor singular value decomposition (t-SVD) [10–12], tensor train (TT) decomposition [13,14], tensor tree (TTR) decomposition [6,15] etc. As we know, CP rank is hard to estimate. Tucker rank is multi-rank, whose elements are the ranks of mode- n matrices which are highly unbalanced. TT rank is also multi-rank, whose elements are the ranks of TT matrices. For a high-order tensor, the most TT matrices are more balanced than the mode- n matrices. Since the matrix rank minimization is only efficient when the matrix is balanced, TT decomposition is more suitable for describing global information of high-order tensors than Tucker decomposition. T-SVD defines the tubal rank of the high order tensor, which can be easily estimated according to a fast Fourier-based method. The tubal rank has been shown more efficient than the matrix-rank and Tucker multi-rank in video applications [16–18]. TTR rank is essentially equivalent to Tucker multi-rank.

Many popular tensor-completion methods have applied the traditional CP or Tucker decomposition on color image inpainting. Some recent works exploited the sparse Tucker core tensor and nonnegative Tucker factor matrices for image restoration [7,19,20]. Some works constrained the low rankness of the mode- n matrix caused by decomposition of a color image for inpainting [21,22]. Since low-tensor-rank constraint cannot fully capture the local smooth and global sparsity priors of tensors, some works combine Tucker and total variation (TV). The SPCTV (smooth PARAFAC tensor completion & total variation) method [23] used the PD (PARAFAC decomposition, a derivation of Tucker decomposition) framework, and constrained the TV (total variation) on every factor matrix of PD respectively. Some works combined the constraints of the low rankness of every mode- n matrix and the TV regularization on every mode- n matrix for color image inpainting [24,25]. Some works proposed data restoration methods based on Bayesian tensor completion [26–29].

The afore-mentioned methods all take the color image as an order-3 tensor directly and haven't deeply explored the potential low-rank prior to a color image. Since TT decomposition is efficient for higher-order tensors, the TMac-TTKA method [30] first used the Ket augmentation (KA) scheme to permute the image to a high order data, then proposed the optimal models by enforcing low TT rankness. The KA scheme is proven to be efficient for improving the accuracy of color image/video inpainting and dynamic MR image reconstruction in TT rank based completion methods [30–33]. As far as we know, the KA scheme is the only one used to permute data into a high order data.

This paper aims to deeply explore the potential low-rank structure of the image and to find an efficient way to apply the SVD, t-SVD, and TT decomposition in the image inpainting problems. The contributions of our work are summarized as follows:

- First, we developed a novel rearrangement named as quarter augmentation (QA) scheme for permuting the image into three flexible forms of data. The first flexible QA scheme can permute an image into an unfolding matrix (with a low matrix rank structure). The second and the third flexible QA schemes can permute the color image into a balanced 3-order form of data (with low tubal rank structure) and a higher-order form of data (with low TT rank structure) respectively. Since those developed schemes are designed to exploit the internal structure similarity of the original data as much as possible, the rearranged data has the corresponding kind of low-rank structure.
- Second, based on the above QA scheme, we developed three image inpainting models that exploit the unfolding matrix rank, tensor tubal rank, and TT multi-rank of the rearranged data respectively for solving the image inpainting problem.
- Lastly, three efficient ADMM algorithms were developed for solving the above three models. Compared with numerous close image inpainting methods, the experimental results demonstrated the superior performance of our methods.

The remainder of this paper is organized as follows. In section II, we give the related work. In section III, we mainly introduce the proposed methods. Section IV the experimental results and analyses. The conclusion is given in section V.

2. Related work

In this section, we briefly introduce the KA scheme, the t-SVD decomposition, and tensor train decomposition. Notations and definitions are summarized in Table 1.

Table 1. Notations and definitions.

Symbols	Notations and definitions
fiber	A vector defined by fixing every index but one of a tensor.
slice	A matrix defined by fixing all but two indices of a tensor.
$\mathcal{A}(:, :, k)$	The k^{th} frontal slice of a 3-order tensor \mathcal{A} .
$\mathcal{A}_{(n)}$	Mode- n matrix, the result of unfolding tensor \mathcal{A} by reshaping its mode- n fibers to the columns of $\mathcal{A}_{(n)}$.

f-diagonal tensor	Order-3 tensor \mathcal{A} is called f-diagonal if each frontal slice $\mathcal{A}(:, :, k)$ is a diagonal matrix [10].
orthogonal tensor	Tensor \mathcal{A} with the size of $n \times n \times n_3$ is called orthogonal tensor if $\mathcal{A} * \mathcal{A}^H = \mathcal{I}$, where \mathcal{I} stands for identity tensor if the first frontal slice $\mathcal{I}^{(1)}$ is the $n \times n$ identity matrix and all other frontal slices $\mathcal{I}^{(k)} (k = 1, 2, \dots, n_3)$ are zero.

2.1. Ket augmentation

The Ket Augmentation (KA) scheme was originally introduced by Latorre in [34] for casting a grayscale image into the real ket state of a Hilbert space. Bengua etc. [30] used KA to reshape a low-order tensor e.g. a color image to a higher-order tensor and proved that KA is efficient in improving the accuracy of the recovered image in TT-based completion.

Figure 1 shows the operation of KA for an 8×8 matrix [31–34]. By the KA scheme, the 8×8 matrix can be turned into a 3-order tensor of size $4 \times 4 \times 4$. As well, the KA scheme can turn a 3-order tensor size of $x^N \times y^N \times N_3$ into an $(N+1)$ -order tensor with a size of $xy \times xy \times \dots \times xy \times N_3$.

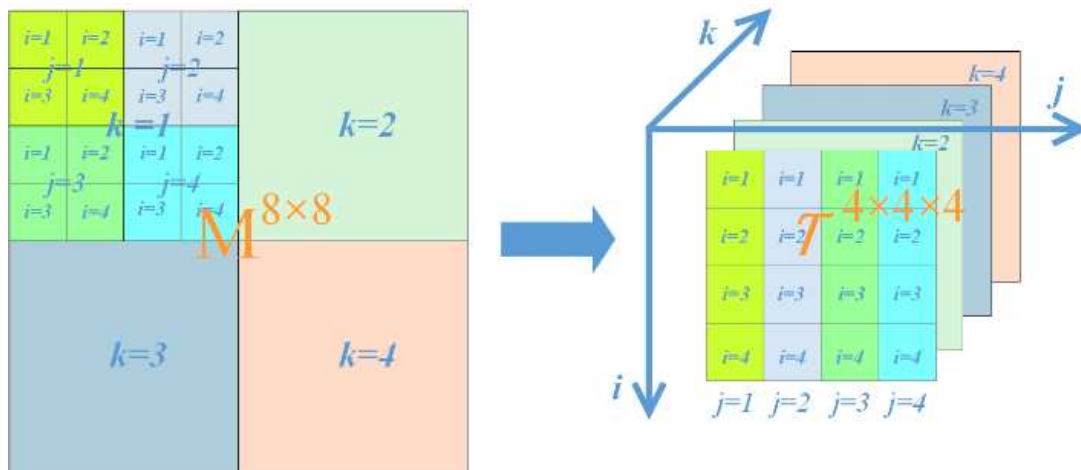


Figure 1. Example of KA for an 8×8 matrix. The order-2 M can be rearranged to a higher-order tensor T (order = 3) without changing the total number of entries.

2.2. T-SVD decomposition

Definition 1 t-product [35]. For $\mathcal{A} \in R^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in R^{n_2 \times n_4 \times n_3}$, the t-product $\mathcal{A} * \mathcal{B} = \mathcal{C}$ is a tensor of size $n_1 \times n_4 \times n_3$ where $\mathcal{C}(i, j, :)$ is given by $\sum_{k=1}^{n_3} \mathcal{A}(i, k, :) \circ \mathcal{B}(k, j, :)$. \circ denotes the circular convolution between the two vectors, and $i = 1, 2, \dots, n_1, j = 1, 2, \dots, n_4$.

The t-SVD of $\mathcal{A} \in R^{n_1 \times n_2 \times n_3}$ is given by

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

where \mathcal{U} and \mathcal{V} are orthogonal tensors of size $n_1 \times n_1 \times n_3$ and $n_2 \times n_2 \times n_3$ respectively. \mathcal{S} is a rectangular f-diagonal tensor of size $n_1 \times n_2 \times n_3$ and $*$ denote t-product [35], T denotes tensor transpose defined in [35].

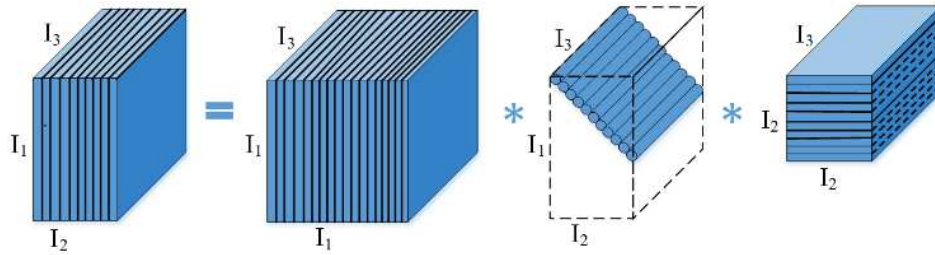


Figure 2. The t-SVD of a tensor of size $I_1 \times I_2 \times I_3$.

Figure 2 depicts the t-SVD of an order-3 tensor [36][4]. Tensor rank defined in t-SVD is tensor tubal rank, which is the number of nonzero singular tubes in \mathcal{S} . [36] proposed the fast Fourier-based method to calculate the tubal rank, and used tensor nuclear norm (TNN) as the convex relaxation of the tensor tubal rank.

$$\|\mathcal{A}\|_{TNN} = \|\text{blockdiag}(\bar{\mathcal{A}})\|_*$$

where $\bar{\mathcal{A}} = \text{fft}(\mathcal{A}, [], 3)$ is the tensor obtained by applying the 1D FFT along the third dimension of \mathcal{A} , $\|\cdot\|_*$ denotes nuclear norm, and

$$\text{blockdiag}(\bar{\mathcal{A}}) = \begin{bmatrix} \bar{\mathcal{A}}^{(1)} & & & \\ & \bar{\mathcal{A}}^{(2)} & & \\ & & \ddots & \\ & & & \bar{\mathcal{A}}^{(I_3)} \end{bmatrix}$$

2.3. Tensor train decomposition

$$\mathcal{A}(i_1, i_2, \dots, i_n, \dots, i_N) = \mathcal{U}_1(:, i_1, :) \mathcal{U}_2(:, i_2, :) \dots \mathcal{U}_n(:, i_n, :) \dots \mathcal{U}_N(:, i_N, :)$$

Given a tensor $\mathcal{A} \in R^{I_1 \times I_2 \times \dots \times I_N}$, tensor train (TT) decomposition [13,14] can decompose it to N order-3 tensors $\mathcal{U}_n \in R^{S_n \times I_n \times S_{n+1}}$, $n=1, \dots, N$. The tensor rank defined in TT decomposition is a multi-rank i.e. $(S_1, S_2, \dots, S_{N+1})$, which is combined with the second-dimensional size of each \mathcal{U}_n . The details of TT decomposition are shown in the following formula and Figure 3 [31–33].

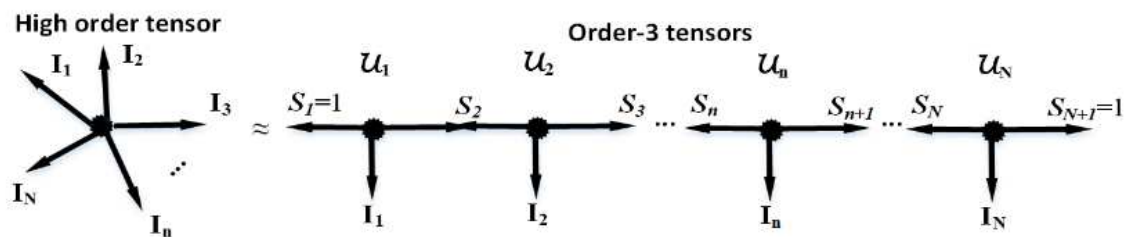


Figure 3. The tensor train decomposition of an order- N tensor with the size of $I_1 \times I_2 \times \dots \times I_n \times \dots \times I_N$.

The widely used way to find TT rank is to estimate the rank of each TT matrix [37] as the element of $(S_1, S_2, \dots, S_{N+1})$. The TT matrix $\mathcal{A}_{[n]}$ ($n=1, \dots, N-1$) with rank S_n is the mode- $(1, 2, \dots, n)$ matricization of

the tensor with the size of $m \times h$, where $m = \prod_{l=1}^n I_l$, $h = \prod_{l=n+1}^N I_l$.

3. Methods

3.1. Quarter Augmentation

To deeply explore the more efficient low-rank structure of an image, we develop a novel rearrangement scheme named as quarter augmentation (QA) scheme to turn a color image into other forms of data. The QA schemes can maintain the internal similarity of the original image in the rearranged data.

The basic QA scheme: For example, as shown in Figure 4 (a), M is a 2D matrix (8×8). We first extract the entries of M every other row and column to get four smaller matrices. Each smaller matrix with a size of 4×4 . Then we place these four smaller matrices along the third dimension in a designed order. Lastly, a 3D tensor of size $4 \times 4 \times 4$ is obtained from the 8×8 matrix M without changing the total number of entries. The entries in the four smaller matrices are labeled as the MATLAB notation $(:, :, 1)$, $(:, :, 2)$, $(:, :, 3)$ and $(:, :, 4)$ respectively. If M is smooth (most images satisfy), the four smaller matrices are similar in structure due to the adjacent entries.

Applying the basic QA scheme on the single Lena image, the Lena image can be divided into 4 smaller Lena images, and as shown in Figure 4 (b) the four smaller Lena images are similar to each other. In Figure 4 (c), the pixel values curves of the four smaller images have overlapped into one curve. We can say that the similarity of local image structure is mainly maintained by the basic QA scheme.

Under this basic QA scheme, three flexible QA schemes are proposed for permuting the image into three flexible forms of data. The three flexible QA schemes can enhance the low-rankness for an image by matrix SVD, tensor train decomposition, and tensor-SVD respectively. Then, by exploiting the flexible QA schemes, three low-rank constrained methods which use the TT rank, tubal rank, and matrix rank as constrained priors respectively are exploited for image inpainting.

The three flexible QA schemes and methods are described in detail in the following three sections.

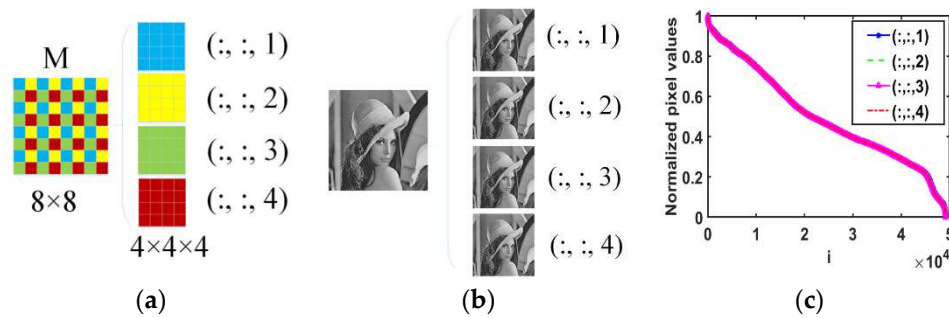


Figure 4. (a) Examples of the basic QA. By the basic QA scheme, the matrix M size of 8×8 can be turned into an order-3 tensor with a size of $4 \times 4 \times 4$. (b) By the basic QA, the Lena image can be divided into 4 small Lena images. (c) The four-pixel values curves of the four smaller Lena images have overlapped into one curve.

3.2. Method 1: The low unfolding matrix rank-based method

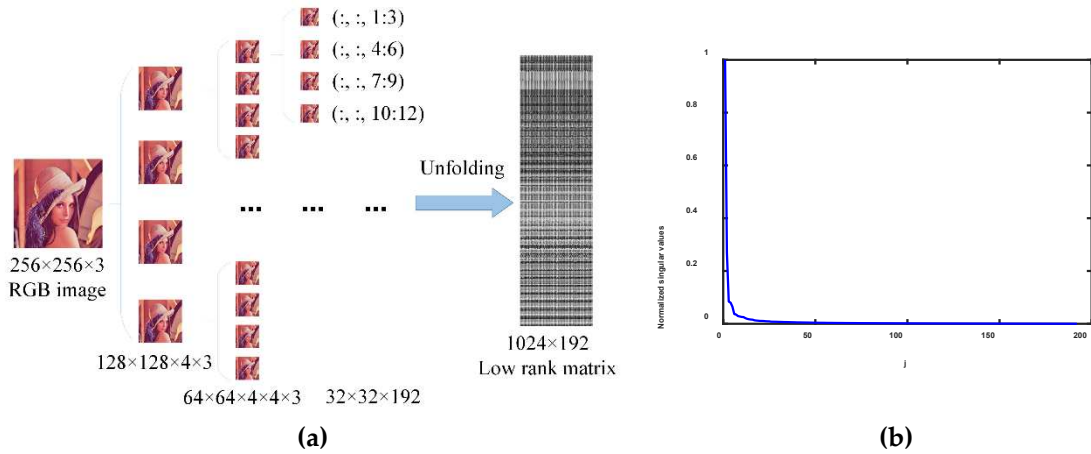


Figure 5. (a) The first flexible QA scheme to obtain the unfolding matrix. Take the Lena RGB image as an example, we first permute the image size of $256 \times 256 \times 3$ to order-3 tensor with the size of $32 \times 32 \times 192$ by the basic QA scheme. Then this order-3 tensor is reshaped into the unfolding matrix of size 1024×92 . (b) The singular values of this unfolding matrix.

The unfolding method is widely used to permute the order-3 video or dynamic magnetic resonance images into an unfolding matrix, and then exploit the low rankness of this matrix for data reconstruction [21,38]. The unfolding matrix has a low-rank structure because of the similarity of every slightly changed slice along the time dimension.

We try to dig out the potential low unfolding-matrix rankness of a color image by a flexible QA scheme, and we call this scheme the first flexible QA scheme.

Take a $256 \times 256 \times 3$ Lena image as an example, as shown in Figure 5 (a), we first permute the image into the 3-order tensor size of $32 \times 32 \times 192$ by the basic QA scheme, then unfold the similar slices of this 3-order. Lastly, the balanced¹ unfolding matrix size of 1024×192 is obtained. Since the slices (32×32) in the 3-order tensor are similar, the unfolding matrix is low rank, as shown in Figure 5 (b). In practice, the size of the designed unfolding matrix should be balanced such that the minimization of the unfolding matrix rank is efficient.

We exploit the low unfolding matrix rank in image inpainting and give the low unfolding matrix-rank-based model as follows.

$$\min_X \|M(\Phi_1 X)\|_* \quad \text{subject to} \quad X(i, j) = Y(i, j), \forall (i, j) \in \Omega \quad (1)$$

where X denotes the image to be recovered, Φ_1 denotes the operator of permuting the image into a suitable 3-order tensor by multiple basic QA schemes. M denotes the operator of the unfolding process, which unfolding every slice along the third dimension of the 3-order tensor $\Phi_1 X$. Ω is the position without painting, Y is the painted image with damaged entries at the positions Ω^c .

To reduce the computational complexity, in the model (1), the following SVD-free approach [39,40] is exploited to constrain the low rankness of the unfolding matrix $M(\Phi_1 X)$ instead of the nuclear norm.

$$\min_{U, V^H = M(\Phi_1 X)} \frac{1}{2} (\|U\|_F^2 + \|V\|_F^2) = \|M(\Phi_1 X)\|_* \quad (2)$$

¹ The context of ‘balanced’ is that the size changes from the unbalanced 256×3 to the more balanced size of 1024×192 .

Besides, since total variation (TV) has been proved as an effective constraint of smooth prior [41,42], incorporate model (1) with 2D TV to exploit the local smooth priors of visual image data. Then, the image inpainting model (1) turns to the following.

$$\begin{aligned} \min_{U,V,X} & \frac{1}{2} (\|U\|_F^2 + \|V\|_F^2) + \frac{\beta}{2} \|X\|_{TV} \\ \text{subject to} & X(i, j) = Y(i, j), \forall (i, j) \in \Omega, \quad M\Phi_1 X = UV^H \end{aligned} \quad (3)$$

where β is the regularization parameter.

We conduct the algorithm by alternating direction method of multipliers (ADMM) for solving the low unfolding matrix rank and TV-based image inpainting model (3). Firstly, introduce an auxiliary variable $Z = DX$, where D is the finite difference operator, and then rewrite (3) as the unconstrained convex optimization problem (4).

$$\begin{aligned} \min_{U,V,\Lambda,L,Z,X} & \tau_\Omega(X) + \frac{1}{2} (\|U\|_F^2 + \|V\|_F^2) + \frac{\rho_1}{2} \|M\Phi_1 X - UV^H + \Lambda\|_F^2 \\ & + \frac{\beta}{2} \|Z\|_1 + \frac{\beta\rho_2}{2} \|DX - Z + L\|_F^2 \end{aligned} \quad (4)$$

where $\tau_\Omega(X)$ denotes the indicator function:

$$\tau_\Omega(X) = \begin{cases} 0, & X \in \Omega \\ \infty, & \text{otherwise} \end{cases}$$

L and Λ are the Lagrangian multipliers for variables Z and UV^H respectively. The regularization parameter β is used to balance the low rankness and sparsity constraints (i.e. TV), the penalty parameters $\rho_1 > 0$ and $\rho_2 > 0$ generally affect the convergence of the algorithm. By applying ADMM, each sub-problem is performed at each iteration t as follows:

$$X^t = \arg \min_X \tau_\Omega(X) + \frac{\rho_1}{2} \|M\Phi_1 X - U^{t-1}V^{(t-1)H} + \Lambda^{t-1}\|_F^2 \quad (5)$$

$$U^t = \arg \min_U \|U\|_F^2 + \frac{\rho_1}{2} \|M\Phi_1 X^t - UV^{(t-1)H} + \Lambda^{t-1}\|_F^2 \quad (6)$$

$$V^t = \arg \min_V \|V\|_F^2 + \frac{\rho_1}{2} \|M\Phi_1 X^t - U^tV^H + \Lambda^{t-1}\|_F^2 \quad (7)$$

$$Z^t = \arg \min_Z \|Z\|_1 + \frac{\rho_2}{2} \|DX^t - Z + L^{t-1}\|_F^2 \quad (8)$$

$$\Lambda^t = \Lambda^{t-1} + M\Phi_1 X^t - U^tV^{(t)H} \quad (9)$$

$$L^t = L^{t-1} + DX^t - Z^t \quad (10)$$

The initial U and V can be determined by solving the following optimization problem using the LMaFit method [43].

$$\min_{U,V,X} \|UV^H - M\Phi_1 X\|_F^2 \quad \text{subject to} \quad X(i, j) = Y(i, j), \forall (i, j) \in \Omega \quad (11)$$

The whole algorithm for solving the model (3) is shown in Table 2.

Table 2. Algorithm 1.

Input: $Y, \Omega, \rho_1, \beta, \rho_2$, maximum number of iteration t_{\max} , convergence condition η_{tol} .
Initialization: initial $U^{(0)}, V^{(0)}$ by solving the matrix completion problem (11), $\Lambda^{(0)}, L^{(0)}, Z^{(0)}, t=0$.
While $t < t_{\max}$ and $\eta < \eta_{\max}$ do
The first flexible QA scheme: Turn an image into an order-N tensor $\Phi_1 X$, then unfold it.
Solve (5)-(10) for X^* , where $*$ represents the optimal solution.
Update $\eta_{t+1} = \frac{\ X_n^{t+1}(\cdot) - X_n^t(\cdot)\ _F}{\ X_n^t(\cdot)\ _F}, t = t + 1$.
End while
Output: X^* .

3.3. Method 2: The low tubal-rank-based method

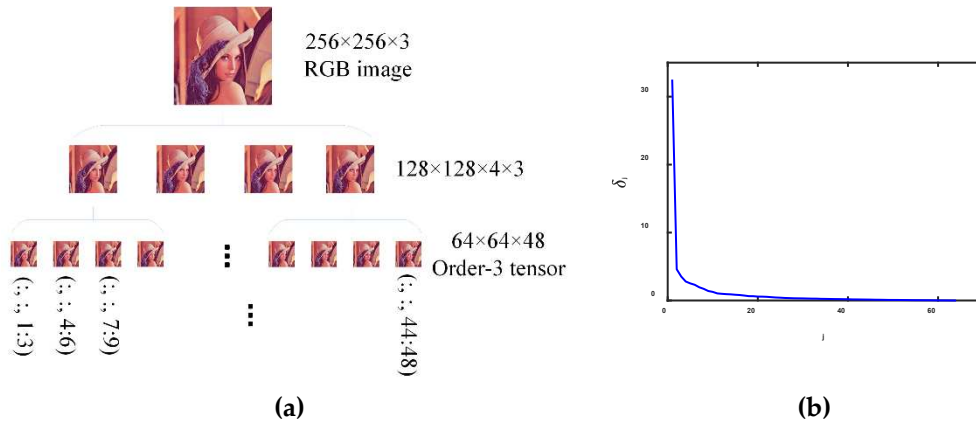


Figure 6. (a) The second flexible QA scheme to permute the image into a balanced order-3 tensor. Take the Lena image size of $256 \times 256 \times 3$ as an example, we obtain the balanced order-3 tensor size of $64 \times 64 \times 48$ by multiple QA schemes. This balanced order-3 tensor is more suitable for the t-SVD decomposition than the original image size of $256 \times 256 \times 3$. (b) The low tubal rankness of the balanced order-3 tensor.

Tensor-SVD decomposition has been efficiently used in the video image completion and dynamic MR image reconstruction problem [16,44–46]. Since the color image is highly unbalanced in the size of three dimensions, which is not suitable for the low tubal rank constraint, we exploit the second flexible QA scheme to deeply dig out the potential low tubal-rank prior information.

Considering that tubal rank minimizations are more efficient for the balanced tensor [10], we first turn the unbalanced image into the balanced order-3 data by the second flexible QA scheme.

Take the color image size of $256 \times 256 \times 3$ as an example, as shown in Figure 6 (a), we can obtain the order-4 tensor size of $128 \times 128 \times 4 \times 3$ by the basic QA schemes, and then multiplying the basic QA schemes we can obtain the order-4 tensor size of $64 \times 64 \times 4 \times 4 \times 3$. Lastly, we reshape the order-4 tensor into the balanced order-3 tensor size of $64 \times 64 \times 48$. Here, the context of ‘balanced’ is that the size changes from the unbalanced $256 \times 256 \times 3$ to the more balanced size of $64 \times 64 \times 48$. In practice, the size of the designed order-3 tensor should be as balanced as possible. We call the above the second flexible QA scheme.

In Figure 6 (b), we show the low tubal rankness of the balanced order-3 data (with the size of $n_1 \times n_2 \times n_3 = 64 \times 64 \times 48$ here) by plotting δ_j which is defined as follows.

$$\delta_j = \frac{1}{n_3} \sum_{i=1}^{n_3} T(i, i, j), \quad i = 1, 2, \dots, \min(n_1, n_2)$$

Then, TNN is used to enforce the tensor tubal rank in the image inpainting model as follows.

$$\min_X \|\Phi_2 X\|_{TNN} \quad \text{subject to } X(i, j) = Y(i, j), \forall (i, j) \in \Omega \quad (12)$$

where Φ_2 denotes the operator of permuting the color image into a more ‘balanced’ order-3 tensor by the second flexible QA scheme. Combining the low tubal rank and sparsity, we introduce auxiliary variables $\mathcal{B} = \Phi_2 X$, and $Z = DX$, then rewrite (12) as the following unconstrained convex optimization problem.

$$\begin{aligned} \min_X \tau_\Omega(X) + \sum_i^{I_3} \|\bar{\mathcal{Z}}^{(i)}\|_* + \frac{\rho}{2} \|\Phi_2 X - \mathcal{Z} + \Lambda\|_F^2 \\ + \frac{\beta}{2} \|Z\| + \frac{\beta \rho_2}{2} \|DX - Z + L\|_F^2 \end{aligned} \quad (13)$$

where I_3 is the third size of the 3-order tensor $\Phi_2 X$. We conduct the algorithm by ADMM for solving model (13) as shown in Table 3.

Table 3. Algorithm 2.

Input: $Y, \Omega, \rho_1, \beta, \rho_2$, the maximum number of iteration t_{\max} , convergence condition η_{tol} .

Initialization: $\Lambda^{(0)}, L^{(0)}, \mathcal{B}^{(0)}, Z^{(0)}, t=0$.

While $t < t_{\max}$ and $\eta < \eta_{\max}$ **do**

QA scheme: Turn an image into the balanced order-3 tensor $\Phi_2 X$.

Update $X^t = \arg \min_X \tau_\Omega(X) + \frac{\rho}{2} \|\Phi_2 X - \mathcal{B}^{t-1} + \Lambda^{t-1}\|_F^2$

Update $\bar{\mathcal{B}}^{(i)t} = \arg \min_{\bar{\mathcal{B}}^{(i)}} \|\bar{\mathcal{B}}^{(i)}\|_* + \frac{\rho}{2} \|\Phi_2 X^t - \mathcal{B} + \Lambda^{t-1}\|_F^2, \quad i=1, \dots, I_3$

Update $Z^t = \arg \min_{\bar{\mathcal{Z}}^{(i)}} \|Z\| + \frac{\rho}{2} \|DX^t - Z + L^{t-1}\|_F^2$

Update $\Lambda^t = \Lambda^{t-1} + \Phi_2 X^t - \mathcal{Z}^t, \quad L^t = L^{t-1} + DX^t - Z^t$

Update $\eta_{t+1} = \frac{\|X_n^{t+1}(\cdot) - X_n^t(\cdot)\|_F}{\|X_n^t(\cdot)\|_F}, \quad t = t + 1$.

End while

Output: X^* .

3.4. Method 3: the low TT-rank-based method

TT decomposition works better on higher-order tensors than Tucker decomposition. To fulfill TT decomposition efficiently, we first exploit the third flexible QA scheme to permute the 3-order image into a higher-order tensor. Based on the basic QA scheme, high-order tensors can be obtained flexibly.

The third flexible QA scheme is shown below. We take a 16×16 matrix as an example, as shown in Figure 7 (a). We first turn a 2-order matrix into a 3D tensor via the basic QA scheme and then repeat the basic QA to obtain the final 4D tensor with the size of $4 \times 4 \times 4 \times 4$. The entry comes from the i^{th} smaller matrix of the first basic QA scheme, and the j^{th} smaller matrix of the second basic QA scheme is labeled as the MATLAB notation $(::, i, j)$. By analogy, the third flexible QA scheme can

permute a matrix with the size of $4^P \times 4^Q$ to order- $\min\{P, Q\}$ tensor with the size of $\underbrace{4 \times 4 \times \dots \times 4}_{\min\{P, Q\}}$. An RGB image with the size of $4^P \times 4^Q \times 3$ can be permuted into an order- $\min\{P, Q\}+1$ tensor with the size of $\underbrace{4 \times 4 \times \dots \times 4 \times 3}_{1+\min\{P, Q\}}$. The third flexible QA scheme should ensure that the designed tensor has a higher order.

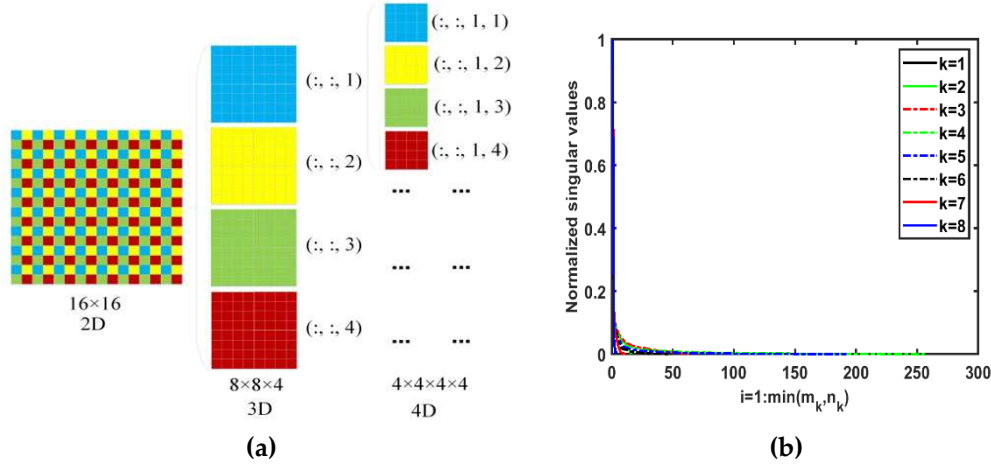


Figure 7. (a) Examples of the third flexible QA scheme. By the third flexible QA, the matrix size of 16×16 can be permuted into an order-4 tensor size of $4 \times 4 \times 4 \times 4$. (b) Singular values of TT matrices. We permute the order-3 Lena RGB image size of $256 \times 256 \times 3$ into an 8-order tensor with the size of $4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 3$ by the third flexible QA scheme. Then eight different TT matrices are obtained from this higher-order tensor. We labeled those TT matrices as $k=1, 2, \dots, 8$.

We permute the Lena image into a high-order tensor via the third flexible QA scheme, and then obtain the TT matrices of the augmented tensor. We name those TT matrices as QA-TT matrices, and their singular values are shown in Figure 7 (b), which demonstrates the low TT rankness of the rearranged tensor.

Then, we enforce the low TT rankness to improve the inpainting accuracy. The third model is as follows.

$$\min_X \sum_{n=1}^N \alpha_n \|\mathcal{T}_n \Phi_3 X\|_* \quad \text{subject to} \quad X(i, j) = Y(i, j), \forall (i, j) \in \Omega \quad (14)$$

where Φ_3 stands for the third flexible QA used to permute image X into a high-dimensional tensor. We name the tensor obtained by the third flexible QA scheme as a QA tensor. \mathcal{T}_n is the operator that converts a tensor into the n^{th} TT matrix, $n=1, 2, \dots, N$. The order of QA tensor is N . The inverse operators corresponding to Φ and \mathcal{T}_n are Φ^{-1} and \mathcal{T}_n^{-1} respectively. The weight α_n is given by:

$$\alpha_n = \frac{\theta_n}{\sum_{n=1}^{N-1} \theta_n} \quad \text{with} \quad \theta_n = \min\left(\prod_{l=1}^n I_l, \prod_{l=k+1}^N I_l\right) \quad (15)$$

where $I_1 \times I_2 \times \dots \times I_N$ is the size of the QA tensor.

Combining the low TT rank and sparsity constraints, we introduce auxiliary variables $Z=DX$ and $U_n V_n^H = \mathcal{T}_n \Phi_3 X$, rewrite (14) as the following unconstrained convex optimization problem, for all $n=1, \dots, N-1$.

$$\begin{aligned} \min_{U_n, V_n, \Lambda_n, L, Z, X} & \tau_\Omega(X) + \frac{1}{2} \sum_{n=1}^{N-1} \alpha_n (\|U_n\|_F^2 + \|V_n\|_F^2) + \frac{\beta}{2} \|Z\|_F^2 \\ & + \frac{\rho_1}{2} \sum_{n=1}^{N-1} \alpha_n \|\mathcal{T}_n \Phi_3 X - U_n V_n^H + \Lambda_n\|_F^2 + \frac{\beta \rho_2}{2} \|DX - Z + L\|_F^2 \end{aligned} \quad (16)$$

By applying ADMM, each sub-problem is performed at each iteration t . Lastly, we obtain X by $X^* = \sum_{n=1}^{N-1} \alpha_n X_n^*$, where X_n^* represents the optimal solution of the n^{th} subproblem. The whole algorithm for solving the model (16) is shown in Table 4.

Table 4. Algorithm 3.

Input: $Y, \Omega, \beta, \rho_1, \rho_2$, the maximum number of iteration t_{\max} , convergence condition η_{tol} .
Initialization: $U_n^{(0)}, V_n^{(0)}$ by the LMaFit method [43]; $\Lambda_n^{(0)}, L_n^{(0)}, Z^{(0)}$.
For $n=1$ to $N-1$ do
$t=0$.
While $t < t_{\max}$ and $\eta < \eta_{\max}$ do
QA scheme: permute image to order-N tensor $\Phi_3 X$.
Update $X_n^t = \arg \min_X \tau_{\Omega}(X) + \frac{\rho_1}{2} \left\ \mathcal{T}_n \Phi_3 X - U_n^{t-1} V_n^{(t-1)H} + \Lambda_n^{t-1} \right\ _F^2$
Update $U_n^t = \arg \min_{U_n} \ U_n\ _F^2 + \frac{\rho_1}{2} \left\ \mathcal{T}_n \Phi_3 X^t - U_n V_n^{(t-1)H} + \Lambda_n^{t-1} \right\ _F^2$
Update $V_n^t = \arg \min_{V_n} \ V_n\ _F^2 + \frac{\rho_1}{2} \left\ \mathcal{T}_n \Phi_3 X^t - U_n^t V_n^H + \Lambda_n^{t-1} \right\ _F^2$
Update $Z^t = \arg \min_Z \ Z\ _1 + \frac{\rho_2}{2} \ DX^t - Z + L^{t-1}\ _F^2$
Update $\Lambda_n^t = \Lambda_n^{t-1} + T_n \Phi_3 X^t - U_n^t V_n^{(t)H}$, $L^t = L^{t-1} + DX^t - Z^t$
Update $\eta_{t+1} = \frac{\ X_n^{t+1}(\cdot) - X_n^t(\cdot)\ _F}{\ X_n^t(\cdot)\ _F}$, $t = t + 1$.
End while
End for
Output: $X^* = \sum_{n=1}^{N-1} \alpha_n X_n^*$.

4. Experimental results and analyses

In this section, we conduct the above methods 1-3 for solving image inpainting problems. For simplicity, we denote methods 1-3 which only exploit low unfolding matrix rank, low tensor tubal rank, and low tensor train rankness as UnfoldingLR, TTLR, and tSVDLR methods respectively. The methods that enhance the low rank and total variation constraints simultaneously are denoted as UnfoldingLRTV, tSVDLRTV, and TTLRTV methods respectively. We denote the low matrix-rank completion method which is solved by the model (17) and ADMM algorithm as the MatrixLR method.

$$\min_X \text{rank}(X) \quad \text{subject to} \quad X(i, j) = Y(i, j), \forall (i, j) \in \Omega \quad (17)$$

We denote the method that only exploits sparsity in the gradient domain and is solved by the ADMM algorithm as the TV method. Besides, we conduct the following numerous close methods for comparison, some of their codes are available online.

STDC: the method exploited the images into three factor matrices and one core tensor for image inpainting [7,19,20]².

HaLRTC: the method constrained the low rankness of the three mode-n matrices caused by decomposition of a color image for inpainting and which was solved by the ADMM [21,22]³.

SPCTV⁴: the smooth PARAFAC tensor completion and total variation method [23], which used the PD (PARAFAC decomposition, a derivation of Tucker decomposition) framework and constrained the TV on every factor matrix of PD respectively.

LRTV: the methods combined the constraints of the low rankness of every mode-n matrix and the TV regularization on every mode-n matrix for color image inpainting [24,25]⁵.

² <http://mp.cs.nthu.edu.tw/module/publications/>

³ <https://www.cs.rochester.edu/u/jliu/publications.html>

⁴ <https://sites.google.com/site/yokotatsuya/home/software>

⁵ https://sites.google.com/site/yokotatsuya/home/software/lrtv_pds

FBCP: the inpainting methods based on Bayesian tensor completion [26–29]⁶.

All simulations were carried out on Windows 10 and MATLAB R2019a running on a PC with an Intel Core i7 CPU 2.8GHz and 16GB of memory. For a fair comparison, every method is conducted with its optimal parameters to ensure every method has the best performance. The reconstruction quality is quantified using the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM)⁷ [37]. The original color images (from the standard image database) and missing patterns used in the experiments are shown in Figure 8.

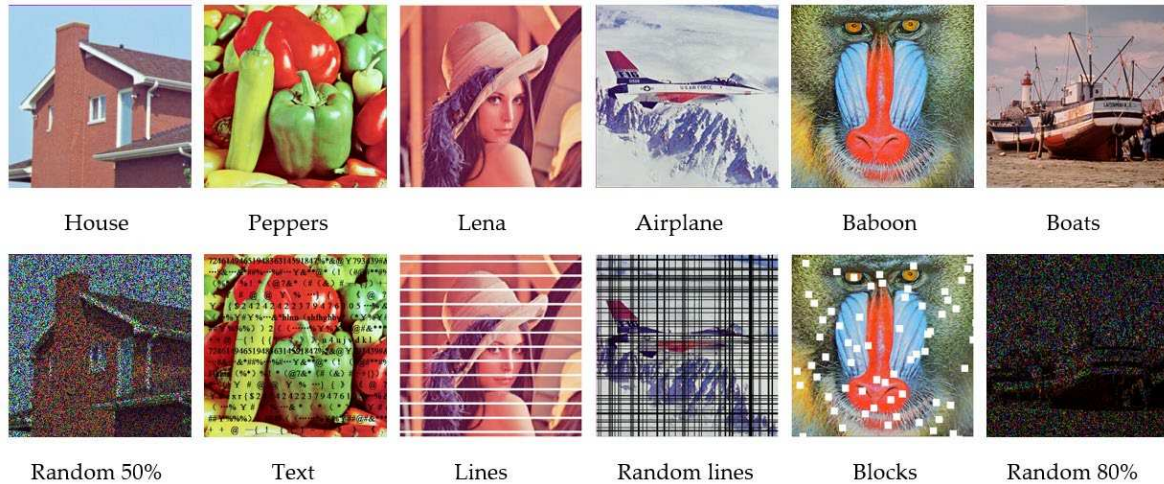


Figure 8. Original color images and missing patterns.

We set the maximum number of iterations $t_{\max}=100$ and convergence condition $\eta_{\text{tol}}=10^{-6}$ in all our methods (UnfoldingLRTV, tSVDLRTV, and TTLRTV). The pixel range of all the images is normalized to 0-1. In UnfoldingLR, tSVDLR, and TTLR methods, we set $\rho_1 = 0.04, 0.002$, and 0.6 respectively. In UnfoldingLRTV, Tsvdlrtv, and TTLRTV methods, we set the parameter set (ρ_1, β, ρ_2) as $(0.4, 0.004, 2)$, $(0.6, 0.07, 0.1)$, and $(0.7, 0.03, 0.1)$ respectively.

4.1. Analyses of the three flexible QA schemes

Next, we call the first, second, and third flexible QA schemes QA scheme briefly. The PSNRs (dB)/SSIMs of the UnfoldingLR, tSVDLR, and TTLR methods with and without the QA scheme are shown in Table 5. The red numerical values correspond to the worst results. We can see that, without the QA scheme, Lena and Airplane cannot be recovered. The UnfoldingLR, tSVDLR, and TTLR methods with the QA scheme have better numerical results than those without the QA scheme. In the low matrix-rank completion method (i.e. MatrixLR), no QA scheme is applied, i.e. the color image is dealt with as three-channel matrices directly.

Due to the support of the QA scheme, the low tensor-rank based methods (TTLR, tSVDLR, and UnfoldingLR) with the QA scheme provide better results than the traditional low matrix-rank completion method (i.e. MatrixLR method). So, the QA scheme is successful to be used as the first step to deeply explore the low tensor rank prior to an image.

⁶ <https://github.com/qbzha0/BCPF>

⁷ <http://www.ece.uwaterloo.ca/~z70wang/research/ssim/>



Figure 9. Comparison of KA and QA scheme under the corresponding TMac-TTKA method [21] and our TTLR method respectively. The first row lists the painted images with a random missing pattern and the missing ratio is 80%. The second row lists the recovered images by TMac-TTKA method. The last row lists the recovered images by LRTT method.

The KA scheme and the third flexible QA scheme both can rearrange an image into a high-order tensor. However, our QA scheme is different from the KA scheme used in [21]. The KA scheme maintains the local block similarity of the image, while the third flexible QA scheme uses adjacent pixels to maintain the global similarity of the image. We conduct the comparison of KA and the third flexible QA scheme under the corresponding TMac-TTKA [21] and TTLR methods. As shown in Figure 9, the small blocks are obvious in the recovered images by the TMac-TTKA method. The images recovered by the TTLR method preserve more details and without the obvious blocks.

4.2. Analyses of the methods exploiting both low rankness and sparsity

In this section, we analyze the recovery results of the methods both exploiting low rankness and sparsity. Figure 10-12 show the visual comparisons of the eleven methods for recovering the House, Lena, and Baboon images respectively. Table 6 shows the PSNR (dB)/SSIM results of the nine methods for recovering different color images under different missing patterns. Figure 13 depicts the

PSNR curves of the inpainting results of the different methods, the missing ratio ranges from 10% to 70% under a random missing pattern.

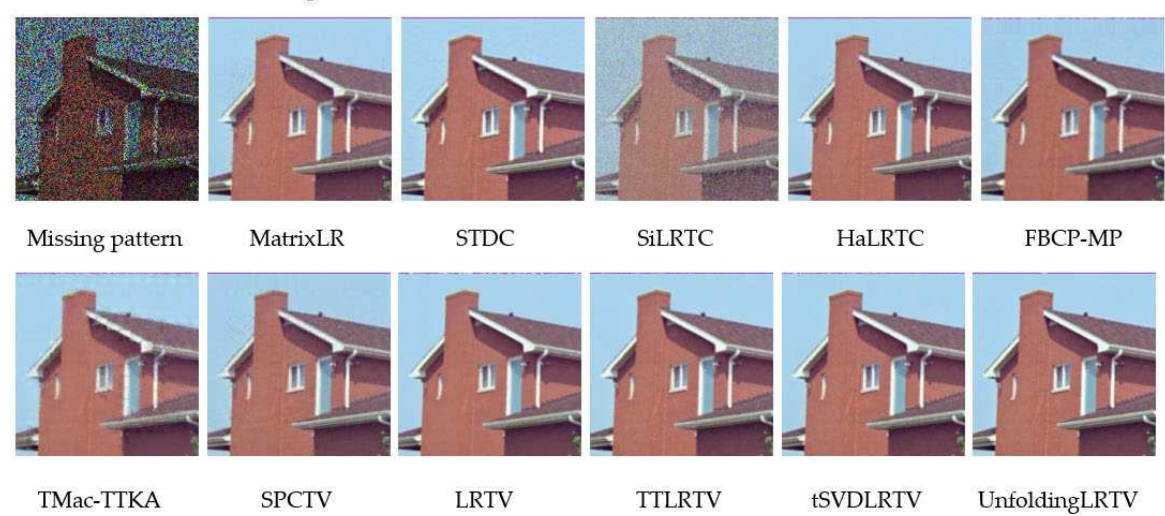


Figure 10. The missing patterns and inpainting results of House image solved by different methods.



Figure 11. The missing patterns and inpainting results of Lena image solved by different methods.

As shown in Figure 10-13 and Table 6, compared to the numerous close STDC, HaLRTC, FBCP, TMac-TTKA, SPCTV, and LRTV methods, the UnfoldingLRTV, tSVDLRTV, and TTLRTV methods have the super performance on both visual and quantity results. The SPCTV and LRTV methods also enhance the low rankness and sparsity simultaneously, but the results are worse than our methods.

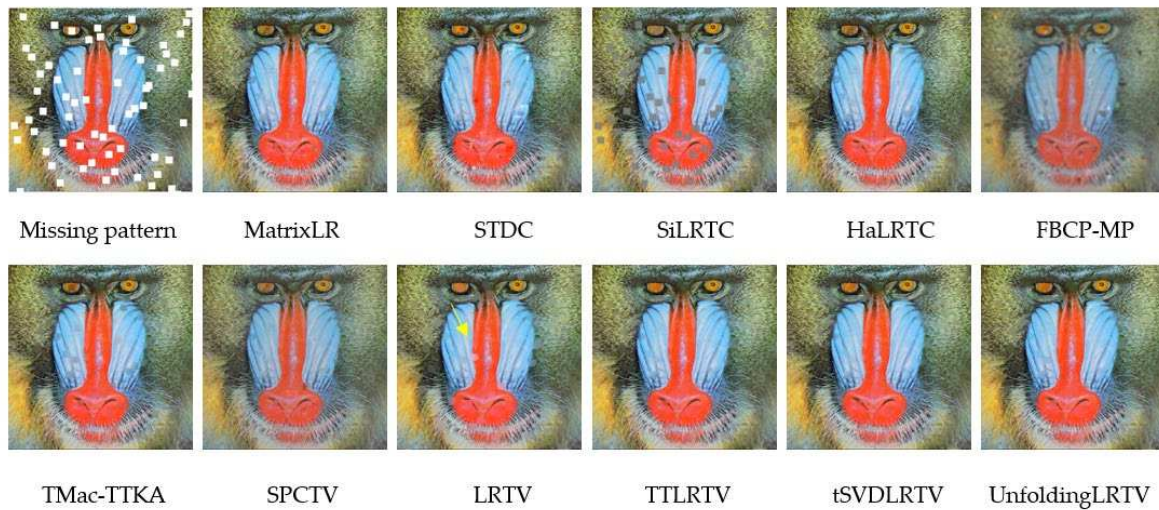


Figure 12. The missing patterns and inpainting results of Baboon image solved by different methods.

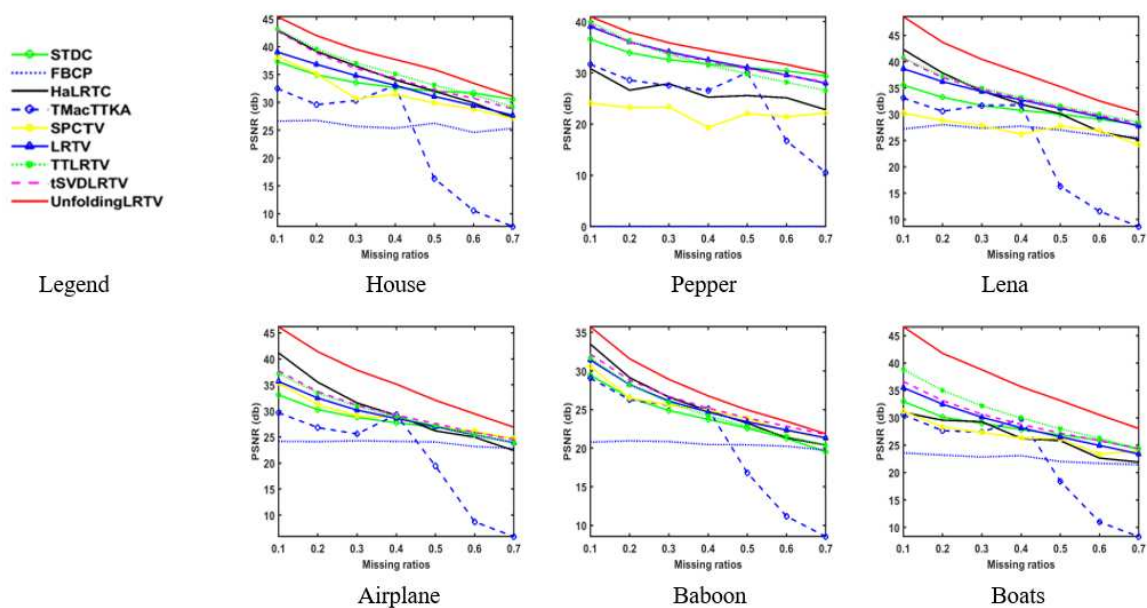


Figure 13. The PSNR curves of the inpainting results of the six images solved by different methods. The missing ratio ranges from 10% to 70% under random missing pattern.

Table 7 shows the PSNR (dB)/SSIM results of the eight methods: MatrixLR method only constrains the low matrix rank; TV method only exploits the TV prior; The UnfoldingLR, tSVDLR, and TTLR methods only constrain the low unfolding matrix rank, low tubal rank and low TT rank respectively; The UnfoldingLRTV, tSVDLRTV, and TTLRTV methods combine both sparsity and low tensor rankness. As shown in Table 7, the combination of sparsity and low tensor rankness constraints can yield better inpainting results than enforcing sparsity or low rankness alone. TTLR method is more efficient than the MatrixLR, and TV methods. The results of the tSVDLR method and TTLR method are comparable. The UnfoldingLR method provides the best results among the TTLR, tSVDLR, TuckerLR, MatrixLR, and TV methods. UnfoldingLRTV, tSVDLRTV, and TTLRTV methods have improved numerical results than the corresponding UnfoldingLR, tSVDLR, and TTLR methods, which demonstrates that TV prior is efficient in improving the accuracy of low-rank based inpainting methods.

The visual and numerical PSNR (dB)/SSIM comparisons of our methods for recovering the pepper image under 80% random missing patterns are shown in Figure 14. In the first row of Figure 14, the methods only exploit low-rank constraints. As shown in the color box, there are small blocky errors in the recovered image, these are caused by the QA scheme. This phenomenon can be solved by combining the constraints of low rank and sparsity (TV), as shown in the second row of Figure 14.

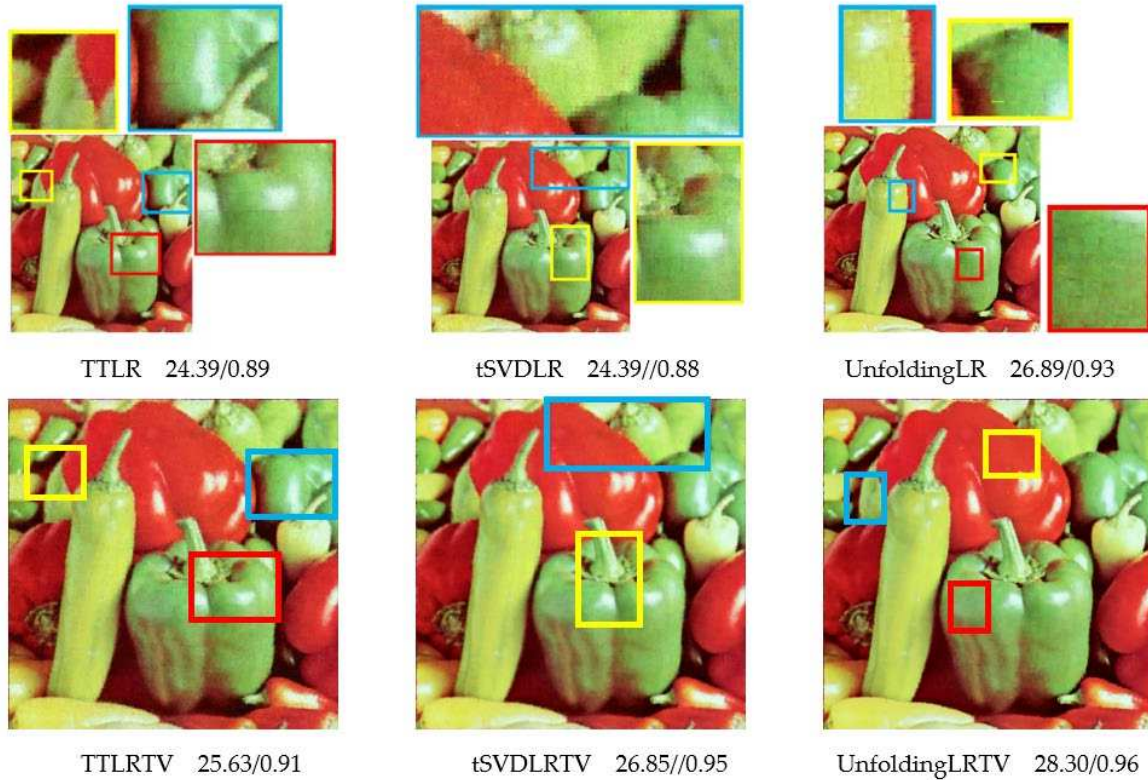


Figure 14. The visual and numerical PSNR (dB)/SSIM comparisons of our methods for recovering the pepper image under 80% random missing patterns. In the first row, the methods only exploit low-rank constraints. As shown in the color box, there are small blocky errors in the repaired image, this is caused by the QA scheme. This phenomenon can be solved by combining the constraints of low rank and sparsity (TV) as shown in the second row.

All in all, due to the support of the QA scheme and the efficient TV prior, the low tensor-rank based methods (UnfoldingLRTV, tSVDLRTV, and TTLRTV) are superior to other close low tensor-rank based methods. The UnfoldingLRTV method provides the best results among all the methods conducted in this paper.

4.3. Analyses of TTLR and TTLRTV methods

In this section, we mainly focus on the analyses of the TT based methods (i.e. TTLR and TTLRTV) in detail. Since TT rank is multi-rank, how does every TT matrix rank affect the final result? We answer this question with the below experimental results.

We conducted the experiments on recovering House, Lena, and Airplane images with a size of $256 \times 256 \times 3$. The random missing patterns have four missing ratios: 10%, 30%, 50%, and 70% respectively. We label the 8 TT matrices as $k=1, 2, \dots, 8$. Then the PSNR (dB) results of X_n^* (the optimal solution of the n^{th} subproblem which exploits the n^{th} TT matrix rank) in TTLR and TTLRTV methods are shown in Figure 15.

From Figure 15, we can see that, the PSNR (dB) results of each subproblem is steeply different in the TTLR method, which demonstrates that each TT matrix rank contributes different PSNR result. Since there are no rules to find which TT matrix rank meets the best PSNR result, we should combine

each solution of X_n^* to obtain the final X^* , i.e. $X^* = \sum_{n=1}^{N-1} \alpha_n X_n^*$. Comparing the PSNR curves of TTLR and TTLRTV method in Fig. 15, the PSNR (dB) results of each subproblem is slightly different in the TTLRTV method which demonstrates that the combination of TT and TV can make the PSNR more balanced among all k .

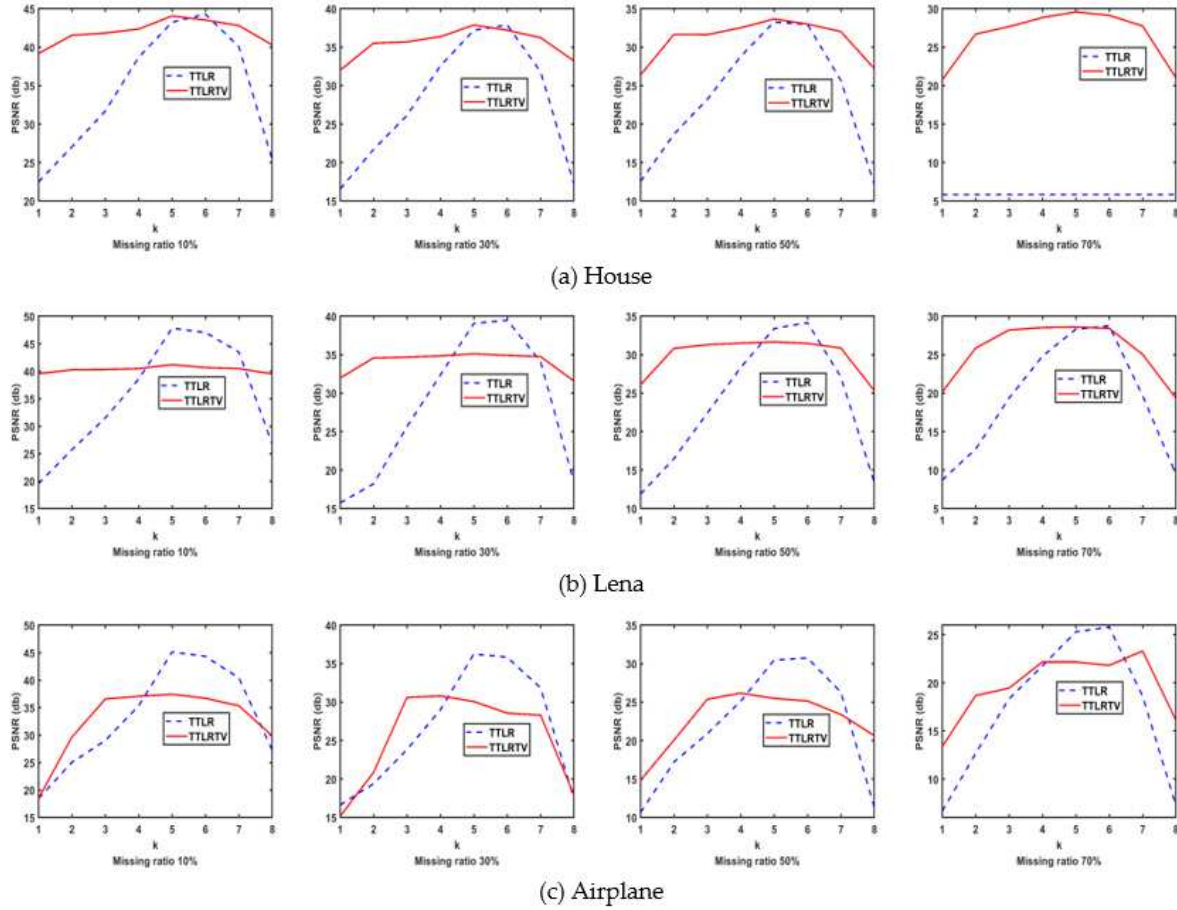


Figure 15. PSNR (dB) results contributed by each TT matrix in TTLR method and TTLRTV method. We permute the image size of $256 \times 256 \times 3$ to an order-9 tensor by the QA scheme. Then we labeled the TT matrices of this order-9 tensor as $k=1, 2, \dots, 8$. We use the random missing patterns with four missing ratios: 10%, 30%, 50%, and 70% respectively. The tested color images for the PSNR curves in (a)-(c) are House, Lena and Airplane images respectively.

4.4. Runtime and complexity analysis

From a high-dimensional curse perspective, converting an image to a higher-order tensor can result in increased complexity, which inevitably leads to a longer runtime. We compare our methods (UnfoldingLRTV, tSVDLRTV, and TTLRTV) with the traditional MatrixLR method and the close STDC, HaLRTC, FBCP, SPCQV, and LRTV methods in running time, as shown in Table 8.

Table 5. PSNR (dB)/SSIM and SSIM of the seven methods without rearrangement and with rearrangement.

Methods		PSNR (dB)/SSIM of different color images under different missing patterns			
		House	Lena	Airplane	Boats
		Random 50%	Lines	Random line	Random 80%
Without Rearrangement	MatrixLR	9.38/0.8970	13.34/0.5850	7.118/0.1308	19.18/0.5680
	TTLR	28.61/0.871	13.34/0.585	7.11/0.130	19.25/0.519
	tSVDLR	32.30/0.932	13.34/0.585	7.11/0.130	21.60/0.707
	UnfoldingLR	7.83/0.093	13.34/0.585	7.11/0.130	6.32/0.102
	TTLR	30.21/0.9251	31.79/0.9559	25.77/0.8796	21.44/0.7144
With Rearrangement	tSVDLR	29.79/0.8989	31.20/0.9561	18.91/0.8386	21.34/0.6879
	UnfoldingLR	32.58/0.9416	33.45/0.9771	28.75/0.9464	23.46/0.8139

Table 6. PSNR (dB)/SSIM and SSIM of the nine methods.

No. Methods			PSNR (dB)/SSIM of different color images under different missing patterns					
			House	Peppers	Lena	Airplane	Baboon	Boats
			Random 50%	Text	Lines	Random line	Blocks	Random 80%
Other methods	1	STDC	32.04/0.9300	33.61/0.9813	28.56/0.8995	23.49/0.7756	27.01/0.9293	21.88/0.7340
	2	HaLRTC	32.07/0.9423	25.84/0.9496	13.34/0.5850	19.94/0.6334	28.04/0.9397	20.56/0.6858
	3	FBCP	26.41/0.8701	NAN	14.56/0.5242	10.25/0.1954	18.71/0.5546	20.91/0.6947
	4	TMac-TTKA	23.18/0.8113	29.47/0.9681	29.93/0.9462	20.82/0.7521	28.04/0.9429	8.83/0.1229
	5	SPCTV	29.56/0.9133	23.38/0.9154	16.02/0.6107	18.58/0.6894	24.21/0.9144	20.98/0.7254
	6	LRTV	30.93/0.9382	36.98/0.9945	34.07/0.9724	26.82/0.9228	27.10/0.9319	21.62/0.7541
Our methods	1	TTLRTV	33.02/0.9579	37.27/0.9945	34.94/0.9823	28.82/0.9561	29.46/0.9559	22.37/0.7487
	2	tSVDLRTV	32.20/0.9550	37.49/0.9950	34.70/0.9818	28.03/0.9507	29.56/ 0.9574	22.86/0.8021
	3	UnfoldingLRTV	35.61/0.9689	37.72/0.9952	34.87/0.9821	29.55/0.9639	29.59/0.9556	25.43/0.8863

Table 7. PSNR (dB)/SSIM and SSIM of the eight methods.

No. Methods		PSNR (dB)/SSIM of different color images under different missing patterns					
		House	Peppers	Lena	Airplane	Baboon	Boats
		Random 50%	Text	Lines	Random line	Blocks	Random 80%
1	MatrixLR	9.38/0.8970	33.23/0.9814	13.34/0.5850	7.118/0.1308	27.62/0.9343	19.18/0.5680
2	TV	29.70/0.8816	34.14/0.9913	29.21/0.9107	22.85/0.8463	23.18/0.9066	20.32/0.6103
3	TTLR	30.21/0.9251	34.86/0.9892	31.79/0.9559	25.77/0.8796	25.42/0.9239	21.44/0.7144
4	tSVDLR	29.79/0.8989	33.86/0.9840	31.20/0.9561	18.91/0.8386	28.03/0.9373	21.34/0.6879
5	UnfoldingLR	32.58/0.9416	36.86/0.9938	33.45/0.9771	28.75/0.9464	22.22/0.9238	23.46/0.8139
6	TTLRTV	33.02/0.9579	37.27/0.9945	34.94/0.9823	28.82/0.9561	29.46/0.9559	22.37/0.7487
7	tSVDLRTV	32.20/0.9550	37.49/0.9950	34.70/0.9818	28.03/0.9507	29.56/ 0.9574	22.86/0.8021
8	UnfoldingLRTV	35.61/0.9689	37.72/0.9952	34.87/0.9821	29.55/0.9639	29.59/0.9556	25.43/0.8863

UnfoldingLRTV methods: In the first step, the QA scheme is used to decompose a single image into several small graphs. Because of the similarity of these small graphs, the QA tensor can be reduced to a matrix with a low-rank structure in an unfolding way. Ignoring TV constraints, the unfoldingLRTV method only needs to solve the low-rank matrix completion problem of an unfolding matrix, so the running time is similar to the traditional MatrixLR method, and the accuracy is higher than the traditional MatrixLR method.

Table 8. Runtimes(s) of the different methods.

Methods	Runtime (s)			
	<i>House</i>	<i>Lena</i>	<i>Airplane</i>	<i>Boats</i>
	<i>Random 50%</i>	<i>Lines</i>	<i>Random lines</i>	<i>Random 80%</i>
MratrixLR	4.95	0.17	0.16	5.01
STDC	5.43	5.13	5.17	5.16
HaLRTC	8.00	0.88	0.84	6.84
FBCP	188.32	86.45	132.09	219.33
SPCTV	19.25	16.37	16.03	17.69
LRTV	19.08	20.17	21.04	21.05
TTLRTV	145.5	143.2	142.6	142.3
tSVDLRTV	15.23	15.07	15.17	15.14
UnfoldingLRTV	9.49	8.53	8.69	8.72

The tSVDLRTV methods: Since the color image is highly unbalanced in the size of three dimensions, which is not suitable for the low tubal rank constraint, we use the QA scheme to rearrange an image into a third-order tensor with a more balanced size of every dimension. Then we use TNN to constrain the low tubal rank of the rearranged tensor, due to the fast Fourier scheme, it is necessary to perform a low-rank matrix constraint on each frontal slice after the third-dimensional Fourier transform. At this time, the SVD decomposition process will increase the time consumption.

TTLRTV methods: TT multi-rank is the combination of the rank of each TT matrix. The TTLRTV method essentially completes the same data amount $N-1$ times, where N is the order of the QA tensor. So, although the TITRTV method is effective, it is necessarily more computationally expensive than the low matrix-rank completion method.

In summary, among the three methods, the UnfoldingLRTV method achieves the best performance both in accuracy and runtime; The TTLRTV method reaches better accuracy, but it is time-consuming; The tSVDLRTV method has moderate performance both in runtime.

All in all, the above three methods can deeply exploit the potential low-rank prior of an image and have been successfully used for image inpainting problems, which demonstrates that the three flexible QA schemes are perfect ways to explore the low-rank prior of an image.

5. Conclusions

To effectively explore the potential of low tensor rank prior to an image, we first exploited a rearrangement scheme (QA) for permuting the color image (3-order) into three flexible rearrangement forms (with more efficient low tensor rank structure). Based on the scheme, three optimization models by exploiting the low unfolding matrix rank, low tensor tubal rank, and low TT multi-rank were proposed to improve the accuracy in image inpainting. Combined with TV constraints, we developed efficient ADMM algorithms for solving those three optimization models. The experimental results demonstrate that our low tensor-rank-based methods are effective for image inpainting, and are superior to the low matrix-rank completion method and numerous close methods. The low tensor rank constraint is effective for image inpainting, which is mainly due to the support of the QA scheme.

Author Contributions: Conceptualization, S.M.; methodology, S.M.; investigation, S.M. and Y.F.; resources, S.M. and S.F.; writing—original draft preparation, S.M. and S.F.; writing—review and editing, S.F., W.Y. and Y.F.; supervision, L.L. and W.Y.; funding acquisition, S.M. and L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Key Laboratory of Science and Technology on Space Microwave, No. HTKJ2021KL504012; Supported by the Science and Technology Innovation Cultivation Fund of Space Engineering University, No. KJCX-2021-17; Supported by the Information Security Laboratory of National Defense Research and Experiment, No.2020XXAQ02.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pendu M.; Jiang X.; Guillemot C. Light field inpainting propagation via low rank matrix completion. *IEEE Trans. Image Process.* **2018**, *27*, 1989-1993.
2. Yu Y.; Peng J.; Yue S. A new nonconvex approach to low-rank matrix completion with application to image inpainting, *Multidim. Syst. Signal Process.* **2018**, *30*, 145-174.
3. Gong X.; Chen W.; Chen J. A low-rank tensor dictionary learning method for hyperspectral image denoising. *IEEE Trans. Signal Process.* **2020**, *68*, 1168-1180.
4. Su X.; Ge H.; Liu Z.; Shen Y. Low-Rank tensor completion based on nonconvex regularization. *Signal Process.* **2023**, *212*, 109157.
5. Ma S.; Ai J.; Du H.; Fang L.; Mei W. Recovering low-rank tensor from limited coefficients in any orthonormal basis using tensor-singular value decomposition. *IET Signal Process.* **2021**, *19*, 162-181.
6. Liu Y.; Long Z.; Zhu C. Image completion using low tensor tree rank and total variation minimization. *IEEE Trans. Multimedia.* **2018**, *21*, 338-350.
7. Gong W.; Huang Z.; Yang L. Accurate regularized Tucker decomposition for image restoration. *Appl. Math. Model.* **2023**, *123*, 75-86.
8. Long Z.; Liu Y.; Chen L. Low rank tensor completion for multiway visual data. *Signal Process.* **2019**, *155*, 301-316.
9. Kolda T.; Bader B. Tensor decompositions and applications. *SIAM Rev.* **2009**, *51*, 455-500.
10. Kilmer M.; Braman K.; Hao N. Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging. *SIAM J. Matrix Anal. Appl.* **2013**, *34*, 148-172.
11. Semerci O.; Hao N.; Kilmer M. Tensor-based formulation and nuclear norm regularization for multienergy computed tomography. *IEEE Trans. Image Process.* **2014**, *23*, 1678-1693.
12. Zhou P.; Lu C.; Lin Z.; Zhang C. Tensor factorization for low-rank tensor completion. *IEEE Trans. Image Process.* **2018**, *3*, 1152-1163.
13. Oseledets I.; Tyrtysnikov E. TT-cross approximation for multidimensional arrays. *Linear Algebra Appl.* **2010**, *432*, 70-88.
14. Oseledets I. Tensor-train decomposition. *Siam J. Sci. Comput.* **2011**, *33*, 2295-2317.
15. Hackbusch W.; Kuhn S. A new scheme for the tensor representation. *J. Fourier Anal. Appl.* **2009**, *15*, 706-722.
16. Zhang Z.; Aeron S. Exact tensor completion using T-SVD. *IEEE Trans. Signal Process.* **2017**, *65*, 1511-1526.
17. Lu C.; Feng J.; Chen Y.; Liu W.; Lin Z.; Yan S. Tensor robust principal component analysis with a new tensor nuclear norm. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 925-938.
18. Du S.; Xiao Q.; Shi Y.; Cucchiara R.; Ma Y. Unifying tensor factorization and tensor nuclear norm approaches for low-rank tensor completion. *Neurocomput.* **2021**, *458*, 204-218.
19. Chen Y.; Hsu C.; Liao H. Simultaneous tensor decomposition and completion using factor priors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *36*, 577-591.
20. Xue J.; Zhao Y.; Liao W.; Enhanced sparsity prior model for low-rank tensor completion. *IEEE Trans. Neural. Netw. Learn. Syst.* **2019**, *31*, 4567-4581.
21. Liu J.; Musialski P.; Wonka P. Tensor completion for estimating missing values in visual data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 208-220.
22. Qin M.; Li Z.; Chen S.; Guan Q.; Zheng J. Low-Rank Tensor Completion and Total Variation Minimization for Color Image Inpainting. *IEEE Access* **2020**, *8*, 53049-53061.
23. Yokota T.; Zhao Q.; Cichocki A. Smooth PARAFAC decomposition for tensor completion. *IEEE Trans. Signal Process.* **2016**, *64*, 5423-5436.
24. Yokota T.; Hontani H. Simultaneous visual data completion and denoising based on tensor rank and total variation minimization and its primal-dual splitting algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, 3732-3740.
25. Li L.; Jiang F.; Shen R. Total Variation Regularized Reweighted Low-rank Tensor Completion for Color Image Inpainting. In *25th IEEE International Conference on Image Processing (ICIP)*, Athens, Greece, 2018, pp. 2152-2156.
26. Zhao Q.; Zhang L.; Cichocki A. Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1751-1763.
27. Wang X.; Philip L.; Yang W.; Su J. Bayesian robust tensor completion via CP decomposition. *Pattern Recognition. Letters* **2022**, *163*, 121-128.
28. Zhu Y.; Wang W.; Yu G. A Bayesian robust CP decomposition approach for missing traffic data imputation. *Multimed Tools Appl.* **2022**, *81*, 33171-33184.

29. Cui G.; Zhu L.; Gui L. Multidimensional clinical data denoising via Bayesian CP factorization. *Sci. China Technol.* **2020**, *63*, 249-254.
30. Bengua J.; Phien H.; Tuan H. Efficient tensor completion for color image and video recovery: Low-Rank Tensor Train. *IEEE Trans. Image Process.* **2017**, *26*, 1057-7149.
31. Ma S.; Du H.; Hu J.; Wen X.; Mei W. Image inpainting exploiting tensor train and total variation. In 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2019, 1-5.
32. Ma S. Video inpainting exploiting tensor train and sparsity in frequency domain. In IEEE 6th International Conference on Signal and Image Processing (ICSIP), 2021, 1-5.
33. Ma S.; Du H.; Mei W. Dynamic MR image reconstruction from highly undersampled (k, t)-space data exploiting low tensor train rank and sparse prior. *IEEE Access* **2020**, *8*, 28690-28703.
34. Latorre J. Image compression and entanglement. Available online: <https://arxiv.org/abs/quant-ph/0510031>
35. Kilmer M.; Martin C. Factorization strategies for third-order tensors. *Linear Algebra Appl.* **2011**, *435*, 641-658.
36. Martin C.; Shafer R.; Larue B. An order-p tensor factorization with applications in imaging. *Siam J. Sci. Comput.* **2013**, *35*, A474-A490.
37. Oseledets I. Compact matrix form of the d-dimensional tensor decomposition. Preprint 2009-01, INM RAS, March 2009.
38. Lingala S.; Hu Y.; Dibella E. Accelerated dynamic MRI exploiting sparsity and low-rank structure: k-t SLR. *IEEE Trans. Med. Imaging.* **2011**, *30*, 1042-1054.
39. Recht B.; Fazel M.; Parrilo P. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.* **2010**, *52*, 471-501.
40. Signoretto M.; Cevher V.; Suykens J. An SVD-free approach to a class of structured low rank matrix optimization problems with application to system identification. In IEEE Conference on Decision and Control (CDC), 2013. no. EPFL-CONF-184990.
41. Liu H.; Xiong R.; Zhang X.; Zhang Y. Nonlocal gradient sparsity regularization for image restoration. *IEEE Trans. Circ. Syst. Vid.* **2017**, *27*, 1909-1921.
42. Feng X.; Li H.; Li J.; Du Q. Hyperspectral Unmixing Using Sparsity-Constrained Deep Nonnegative Matrix Factorization with Total Variation. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 1-13.
43. Z Wen.; Yin W.; Zhang Y. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Math. Program. Comput.* **2012**, *4*, 333-61.
44. Ai J.; Ma S.; Du H.; Fang L. Dynamic MRI Reconstruction Using Tensor-SVD. In 14th IEEE International Conference on Signal Processing, Beijing, 2018, 1114-1118.
45. Su X.; Ge H.; Liu Z.; Shen Y. Low-rank tensor completion based on nonconvex regularization. *Signal Process.* **2023**, *212*, 109157.
46. Wang Z.; Bovik A.; Sheikh H. Simoncelli E. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600-612.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.