Thai Journal of Mathematics
Volume 21 Number 2 (2023)
Pages 385–398

http://thaijmath.in.cmu.ac.th

ISSN 1686-0209

International Conference on Digital Image Processing and Machine Learning 2022

Randomized Algorithm on Tensor Singular Value Decomposition for Image and Video Reconstructions

Siriwan Intawichai and Saifon Chaturantabut*

Department of Mathematics and Statistics, Faculty of Science and Technology, Thammasat University, Pathumthani, 12120, Thailand
e-mail : siriwan.in@up.ac.th (S. Intawichai); saifon@mathstat.sci.tu.ac.th (S. Chaturantabut)

Abstract In this work, we introduce a randomized algorithm for analyzing and capturing multi-linear data structure. Tensor singular value decomposition (T-SVD) is a useful method to extract the most dominant features of a given tensor data and to compute a low multi-linear rank basis. However, computing T-SVD can be time-consuming for large-scale data. The randomized algorithm is therefore employed on T-SVD, so called randomized T-SVD, to reduce the computational complexity on large-scale problems. We proposed a method that employs the randomized T-SVD to construct an efficient tensor basis used in the least-squares approximation for estimating the missing values on data recovery applications. Numerical experiments on data recovery are performed for image and video reconstructions. These results show that the proposed method is considerably faster than the traditional tensor approach while achieving a comparable peak signal-to-noise ratio.

MSC: 65D18; 65F55; 49M27
Keywords: tensor SVD; randomized tensor SVD; tensor basis; least-squares approximation

Submission date: 31.12.2022 / Acceptance date: 01.05.2023

1. Introduction

A tensor is a higher generalization of vector and matrix which can be a representation of high-dimensional and multi-way real world data such as color images, video, etc. Factorization strategies for tensor have been studied in many researches [1–10] with many applications such as data processing [3, 10], images and video processing [4–7].

The order of tensor is the number of ways or modes of tensor, thus the first order and second order tensors are vectors and matrices, respectively. Additional, a grey-scale image can be considered as a second order tensor. A third order tensor can not only represent a color image but also a grey-scale video. For a color image, its third index component represents the intensities on red, green, and blue scales. For a grey-scale video, the third

*Corresponding author.

Published by The Mathematical Association of Thailand.

index component represents the number of frames. Moreover, images on each channel mode or gray-scale mode can be represented by matrix and the elements in matrix are represented as pixels which can be solved by matrix completion methods.

Singular value decomposition (SVD) [11, 14, 17] and randomized singular value decomposition (rSVD) [12, 13, 15] are low-rank matrix completion techniques and widely used for matrix decomposition. Furthermore, SVD and rSVD method is applied on image processing [14, 16–18] for solving image compression and image reconstruction.

S. Intawichai et al. [19–22] applied the rSVD method in image reconstruction problems. The missing pixels in gray-scaled images are recovered by the notion of least-squares approximation with the proper orthogonal decomposition (POD) basis. The POD basis is an optimal low-rank basis in Euclidean norm which can compute by SVD method [11] and also rSVD method. The rSVD method can substantially decrease the computation work and preserve the accuracy while a full SVD is expensive and memory intensive.

Based on tensor completion, tensor SVD (t-SVD) is one of the tensor factorization which is applied in image processing [8, 9]. A randomized algorithm is applied on the tensor SVD [6], called randomized tensor SVD (randomized t-SVD) which is an improved method of the t-SVD but that is more computationally efficient on extremely large datasets.

In this paper, we focus on the third order tensor which represents color images and gray-scaled videos. Based on tensor completion, the t-SVD is used for constructing a POD tensor which is an optimal basis for tensor. With the similar properties to the t-SVD, the randomness can be applied in the t-SVD, called t-rSVD and it can be also used to compute the POD tensor. With the notion of rSVD method, we obtain randomized t-SVD to recover the missing components in high dimensional data. The reconstruction method in [19, 20] is modified for the color image and gray-scaled video which are represented by third order tensor. The proposed method need the POD tensor which can be constructed by t-SVD, t-rSVD or randomized t-SVD. The performance is compared between the three methods. In addition, we compare this proposed reconstruction method to the previous method in [19].

This paper is organized as follows. After establishing basic notations and some preliminaries in Section 2, the formulation of tensor completion is introduced in section 3. The reconstruction algorithm is detailed in section 4. Experimental results are presented in section 5, and some concluding remarks are given in section 6.

2. Preliminaries on Tensors and Notations

In this section, we provide explanations of the notations and introduce some tensor operations that appear in this paper.

For convenience, we let scalars, vectors, matrices and tensors be denoted as lowercase letters $\{a, b, ...\}$, bold - case letters $\{\mathbf{a}, \mathbf{b}, ...\}$, capital letters $\{A, B, ...\}$ and Euler script letters $\{\mathcal{A}, \mathcal{B}, ...\}$, respectively. For a third-order tensor $\mathcal{A} \in \mathbb{R}^{m\times n\times l}$, its mode-1, mode-2, and mode-3 fibers are denoted by $\mathcal{A}(:, i_2, i_3), \mathcal{A}(i_1, :, i_3)$, and $\mathcal{A}(i_1, i_2, :)$, where $i_1 = 1, ..., m, i_2 = 1, ..., n$ and $i_3 = 1, ..., l$. Its horizontal slices $\mathcal{A}(i_1, :, :), i_1 = 1, ..., m$ lateral slices $\mathcal{A}(:, i_2, :), i_2 = 1, ..., n$ and frontal slices $\mathcal{A}(:, :, i_3), i_3 = 1, ..., l$ are shown in Fig. 1.

In this paper, the frontal slices represent the channel color modes, and we refer to the frontal slice of $\mathcal{A}$ as a matrix $\mathcal{A}(\cdot)$. First, we define an operator that converts the third-order tensor $\mathcal{A} \in \mathbb{R}^{m\times n\times l}$ into a block circulant matrix.

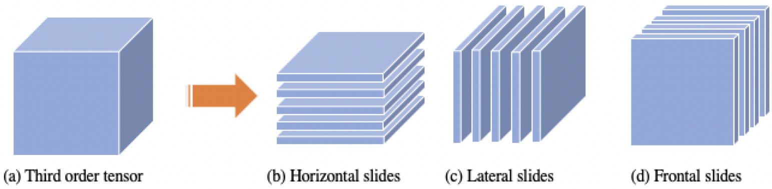


FIGURE 1. The illustration of (a) A third-order tensor $\mathcal{A} \in \mathbb{R}^{m\times n\times l}$, (b) Horizontal slides $\mathcal{A}(i_1, :, :), i_1 = 1, ..., m$, (c) lateral slices $\mathcal{A}(:, i_2, :), i_2 = 1, ..., n$ and (d) frontal slices $\mathcal{A}(:, :, i_3), i_3 = 1, ..., l$.

Definition 2.1. ([10] p.9) Let $\mathcal{A} \in \mathbb{R}^{m\times n\times l}$ be a third-order tensor with $m \times n$ frontal slices denoted $A(\cdot)$, then a block circulant matrix of $\mathcal{A}$ with size $ml \times nl$ is written as

$$\texttt{bcirc}(\mathcal{A}) = \begin{bmatrix} A(1) & A(l) & A(l-1) & \dots & A(2) \\ A(2) & A(1) & A(l) & \dots & A(3) \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A(l) & A(l-1) & \ddots & A(2) & A(1) \end{bmatrix} \tag{2.1}$$

It is well-known that block circulant matrices can be block diagonalized by using the Fourier transform as

$$(F_l \otimes I_m)\texttt{bcirc}(\mathcal{A})(F_l^H \otimes I_n) = \begin{bmatrix} \hat{A}(1) & 0 & \dots & 0 \\ 0 & \hat{A}(2) & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{A}(l) \end{bmatrix},$$

where $\otimes$ denotes the Kronecker product, H denotes the conjugate transpose, F_l is a $l \times l$ normalized discrete Fourier transform (DFT) matrix and I_m, I_l are identity matrices with size $m \times m$ and $l \times l$ respectively. $\hat{A}(i)$ is the ith frontal slice of $\hat{A}$. Using Matlab notation, define $\hat{A} =: \texttt{fft}(A, [\,], l)$ as the tensor obtained by applying the fast Fourier transform (FFT) along each tubal-element of $\mathcal{A}$.

The `unfold` command is setting the third-order tensor $\mathcal{A}$ to a block $ml \times n$ matrix, whereas the `fold` command undoes this operation:

$$\texttt{unfold}(\mathcal{A}) = \begin{bmatrix} A(1) \\ A(2) \\ \vdots \\ A(l) \end{bmatrix}, \ \texttt{fold}(\texttt{unfold}(\mathcal{A})) = \mathcal{A}. \tag{2.2}$$

Definition 2.2. (**T-product** [10] p.9). Let $\mathcal{A} \in \mathbb{R}^{m\times n\times l}$ and $\mathcal{B} \in \mathbb{R}^{n\times J_1\times l}$. Then the T-product $\mathcal{A} * \mathcal{B}$ is the $m \times J_1 \times l$ tensor

$$\mathcal{A} * \mathcal{B} = \texttt{fold}(\texttt{bcirc}(\mathcal{A}) \cdot \texttt{unfold}(\mathcal{B})). \tag{2.3}$$

Definition 2.3. (**Transpose** [10] p.6). Let $\mathcal{A} \in \mathbb{R}^{m\times n\times l}$, then $\mathcal{A}^T$ is $n \times m \times l$ tensor obtained by transposing each of the frontal slices and then reversing the order of transposed frontal slices 2 through l.

Definition 2.4. (**Identity** [10] p.11). The $m \times m \times l$ identity tensor $\mathcal{I}$ is the tensor whose first frontal slice is the $m \times m$ identity matrix, and whose other frontal slices are all zeros.

Definition 2.5. (**Orthogonal** [10] p.11). The $m \times m \times l$ real-valued tensor $\mathcal{Q}$ is orthogonal if $\mathcal{Q}^T * \mathcal{Q} = \mathcal{Q} * \mathcal{Q}^T = \mathcal{I}$

Tensor Decompositions

We now introduce the tensor QR (t-QR), tensor SVD (t-SVD), and truncated t-SVD, which builds on the above operations of tensors.

Definition 2.6. (**t-QR factorization** [2] p.652). Given an $m \times n \times l$ tensor $\mathcal{A}$, the t-QR factorization of $\mathcal{A}$ is $\mathcal{A} = \mathcal{Q} * \mathcal{R}$, where tensor $\mathcal{Q}$ is orthogonal and $\mathcal{R}$ is f-upper triangular (i.e. each frontal slide of $\mathcal{R}$ is upper triangular).

Definition 2.7. (**t-SVD** [2] p.651). Given an $m \times n \times l$ tensor $\mathcal{A}$, the t-SVD of $\mathcal{A}$ is

$$\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T, \tag{2.4}$$

where tensor $\mathcal{U} \in \mathbb{R}^{m \times m \times l}$, $\mathcal{V} \in \mathbb{R}^{n \times n \times l}$ are orthogonal and $\mathcal{S} \in \mathbb{R}^{m \times n \times l}$ is f-diagonal tensor (i.e. each frontal slide of $\mathcal{S}$ is diagonal).

Definition 2.8. (**truncated t-SVD** [2] p.652). Given a tensor $\mathcal{A} \in \mathbb{R}^{m \times n \times l}$, the truncated t-SVD of $\mathcal{A}$ is

$$\mathcal{A}_k = \mathcal{U}_k * \mathcal{S}_k * \mathcal{V}_k^T, \tag{2.5}$$

where k is a target truncation term, tensors $\mathcal{U}_k \in \mathbb{R}^{m \times k \times l}$, $\mathcal{V}_k \in \mathbb{R}^{n \times k \times l}$ are orthogonal and $\mathcal{S}_k \in \mathbb{R}^{m \times n \times l}$ is f-diagonal tensor.

Algorithm 1: The truncated t-SVD algorithm

INPUT	: A tensor $\mathcal{A} \in \mathbb{R}^{m \times n \times l}$ and target truncation term k.
OUTPUT	: $\mathcal{U}_k \in \mathbb{R}^{m \times k \times l}, \mathcal{S}_k \in \mathbb{R}^{k \times k \times l}, \mathcal{V}_k \in \mathbb{R}^{n \times k \times l}$
Step 1.	$\widehat{\mathcal{A}} \leftarrow \texttt{fft}(\mathcal{A}, [], 3)$
Step 2.	**for** $i = 1$ **to** I_3 **do**
Step 3.	$[U, S, V] = \texttt{SVD}(\mathcal{A}(i))$
Step 4.	Truncating U, S, V with target truncation term k
Step 5.	Form $\widehat{\mathcal{U}}_k(i) = U_k, \widehat{\mathcal{S}}_k(i) = S_k, \widehat{\mathcal{V}}_k(i) = V_k$
Step 6.	**end for**
Step 7.	$\mathcal{U}_k \leftarrow \texttt{ifft}(\widehat{\mathcal{U}}_k, [], 3)$
Step 8.	$\mathcal{S}_k \leftarrow \texttt{ifft}(\widehat{\mathcal{S}}_k, [], 3)$
Step 9.	$\mathcal{V}_k \leftarrow \texttt{ifft}(\widehat{\mathcal{V}}_k, [], 3)$

Next, we will introduce the randomized tensor SVD (randomized t-SVD). Since it follows from the randomized SVD in matrix completion, random samping is used to generate a Gaussian random tensor and form a random projection of tensor. The random projection is used to construct a reduced tensor whose range approximates the range of original tensor.

Definition 2.9 (**Gaussian random tensor** [6]**).** A tensor $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is called a Gaussian random tensor, if the elements of $\mathcal{G}(1)$ satisfy the standard normal distribution, and the other frontal slices are all zeros.

Randomized t-SVD [6]: Given a tensor $\mathcal{A} \in \mathbb{R}^{m\times n\times l}$, the randomized t-SVD of $\mathcal{A}$ is

$$\mathcal{A}_k = \mathcal{U}_k * \mathcal{S}_k * \mathcal{V}_k^T, \tag{2.6}$$

where k is a target truncation term, tensor $\mathcal{U}_k \in \mathbb{R}^{m\times k\times l}$, $\mathcal{V}_k \in \mathbb{R}^{n\times k\times l}$ are orthogonal and $\mathcal{S}_k \in \mathbb{R}^{m\times n\times l}$ is f-diagonal tensor.

Algorithm 2: The randomized t-SVD algorithm

INPUT	: A tensor $\mathcal{A} \in \mathbb{R}^{m\times n\times l}$, target truncation term k and oversampling parameter p.
OUTPUT	: $\mathcal{U}_k \in \mathbb{R}^{m\times k\times l}, \mathcal{S}_k \in \mathbb{R}^{k\times k\times l}, \mathcal{V}_k \in \mathbb{R}^{n\times k\times l}$
Step 1.	Generate a Gaussian random tensor $\mathcal{G} \in \mathbb{R}^{n\times (k+p)\times l}$
Step 2.	Form a random projection of the tensor $\mathcal{A}$ as $\mathcal{Y} = \mathcal{A} * \mathcal{G}$
Step 3.	Construct the tensor $\mathcal{Q}$ by using t-QR factorization of tensor $\mathcal{Y}$
Step 4.	Form a reduced tensor $\mathcal{B} = \mathcal{Q}^T * \mathcal{A}$, whose size is $(k+p)\times n\times l$
Step 5.	Compute truncated t-SVD of $\mathcal{B}$, $\mathcal{B}_k = \widetilde{\mathcal{U}}_k * \mathcal{S}_k * \mathcal{V}_k^T$
Step 6.	Set $\mathcal{U}_k = \mathcal{Q} * \widetilde{\mathcal{U}}_k$

3. Proposed Method

This section describes the proposed method for reconstructing the missing data components in a color image which is represented as a third-order tensor.

Let $\mathcal{A} \in \mathbb{R}^{m\times n\times l}$ be a target tensor whose some pixels are missing. This tensor is separated to two sets by the lateral slides, which are a set of complete lateral slides and a set of incomplete data tensor. Let $\{\mathcal{A}(:, i_2, :)\}_{i_2=1}^{n_s}$, be a set of complete lateral slides and form a complete data tensor as $\mathcal{X} \in \mathbb{R}^{m\times n_s\times l}$. Then $\{\bar{\mathcal{A}}(:, i_2, :)\}_{i_2=1}^{n_t}$, is a set of incomplete lateral slides, where $n = n_s + n_t$.

3.1. POD Tensor

We will investigate an optimal basis of third-order tensor which is called **Proper orthogonal decomposition tensor** or **POD tensor**. It follows from the matrix completion, that proper orthogonal decomposition (POD) basis is the optimal low-rank basis in Euclidean norm. POD basis can be computed by using SVD, i.e., it can be obtained from the left singular vector of SVD[19]. For tensor Completion, the POD tensor can be obtained by the left tensor of the t-SVD.

Algorithm 3 : POD Tensor

INPUT	: A set of lateral slides $\{Y(i_2)\}_{i_2=1}^{n_s} \subset \mathbb{R}^{m\times l}$ and target truncation term k.
OUTPUT	: POD tensor $\mathcal{U}_k$.
Step 1.	Create a tensor : $\mathcal{Y} = [Y_1, Y_2, ..., Y_{i_2}] \in \mathbb{R}^{m\times n_s\times l}$
Step 2.	Compute the truncated t-SVD of $\mathcal{Y}$ with truncation term k, $\mathcal{Y} = \mathcal{U}_k * \mathcal{S}_k * \mathcal{V}_k^T$.
Step 3.	The POD tensor $\mathcal{U}_k$.

Note that, we can approximate the POD tensor by the left tensor of the randomized t-SVD for reducing computation time. Moreover, the POD tenser is more accurate than the POD basis which is computed from each frontal slides by using SVD method [6].

Since, the randomized SVD is used to decrease the computation time when compare to SVD. Thus, we modify an above algorithm by using randomized SVD for reducing time consuming as Algorithm 4.

Algorithm 4: The t-rSVD algorithm

INPUT	: A tensor $\mathcal{A} \in \mathbb{R}^{m \times n \times l}$ and target k.
OUTPUT	: $\mathcal{U}_k \in \mathbb{R}^{m \times k \times l}, \mathcal{S}_k \in \mathbb{R}^{k \times k \times l}, \mathcal{V}_k \in \mathbb{R}^{n \times k \times l}$
Step 1.	$\widehat{\mathcal{A}} \leftarrow \texttt{fft}(\mathcal{A}, [], 3)$
Step 2.	**for** $i = 1$ **to** I_3 **do**
Step 3.	$[U_k, S_k, V_k] = \texttt{rSVD}(\mathcal{A}(i), k)$
Step 4.	Form $\widehat{\mathcal{U}}_k(i) = U_k, \widehat{\mathcal{S}}_k(i) = S_k, \widehat{\mathcal{V}}_k(i) = V_k$
Step 5.	**end for**
Step 6.	$\mathcal{U}_k \leftarrow \texttt{ifft}(\widehat{\mathcal{U}}_k, [], 3)$
Step 7.	$\mathcal{S}_k \leftarrow \texttt{ifft}(\widehat{\mathcal{S}}_k, [], 3)$
Step 8.	$\mathcal{V}_k \leftarrow \texttt{ifft}(\widehat{\mathcal{V}}_k, [], 3)$

Next, the tensor reconstruction is considered by extending the previous reconstruction approaches [19–21].

3.2. Tensor Reconstruction

Suppose that $\overline{A} = [\overline{\mathbf{y}}_1, \overline{\mathbf{y}}_2, ..., \overline{\mathbf{y}}_l] := \overline{\mathcal{A}}(:, i_2, :) \in \mathbb{R}^{m \times l}$ is an incomplete lateral slide sample which is represented an incomplete data matrix. Let $\widehat{\mathbf{x}} := \overline{\mathbf{y}}_i \in \mathbb{R}^m, i = 1, ..., l$ be an incomplete sample with $m = m_c + m_g$, where m_c, m_g are the numbers of known and unknown components respectively.

Define the matrices $C = [\mathbf{e}_{c_1}, \ldots, \mathbf{e}_{c_{m_c}}] \in \mathbb{R}^{m \times m_c}$ and $G = [\mathbf{e}_{g_1}, \ldots, \mathbf{e}_{g_{m_g}}] \in \mathbb{R}^{m \times m_g}$, where $\{c_1, c_2, \ldots, c_{m_c}\}$, $\{g_1, g_2, \ldots, g_{m_g}\} \subset \{1, 2, \ldots, m\}$ are the index sets of the known and unknown components, respectively. The column vector $\mathbf{e}_j = [0, ..., 0, 1, 0, ..., 0]^T$ is the j-th, column of the identity matrix I_m.

Note that, C^T (G^T) is equivalent to extracting the m_c (m_g) rows corresponding to the indices $c_1, \ldots, c_{m_c}$ $(g_1, \ldots, g_{m_g})$. Let $\widehat{\mathbf{x}}_c := C^T \widehat{\mathbf{x}} \in \mathbb{R}^{m_c}$ and $\widehat{\mathbf{x}}_g := G^T \widehat{x} \in \mathbb{R}^{m_g}$. Then, the known components and the unknown components are given in the vectors $\widehat{\mathbf{x}}_c$ and $\widehat{\mathbf{x}}_g$, respectively.

Given a basis tensor $\mathcal{U}$ with truncation term k which can considered by the left tensor of truncated t-SVD or randomized t-SVD. Suppose that $U := \mathcal{U}(i), i = 1, ..., l$ is a frontal slide of $\mathcal{U}$. Since the basis tensor is constructed by the t-SVD, then U is also a basis matrix with orthonormal columns. The missing components contained in $\widehat{\mathbf{x}}_g$ will be approximated by first projecting $\widehat{\mathbf{x}}$ onto the column span of the basis matrix U with rank k.

$$\widehat{\mathbf{x}} \approx U\mathbf{a}, \quad \text{or} \quad \widehat{\mathbf{x}}_c \approx U_c\mathbf{a} \;\; \text{and} \;\; \widehat{\mathbf{x}}_g \approx U_g\mathbf{a}, \tag{3.1}$$

for some coefficient vector $\mathbf{a} \in \mathbb{R}^k$, and $U_c := C^T U \in \mathbb{R}^{m_c \times k}$, $U_g := G^T U \in \mathbb{R}^{m_g \times k}$.

Using $\widehat{\mathbf{x}}_c$ to determine $\mathbf{a}$ through $\widehat{\mathbf{x}}_c \approx U_c\mathbf{a}$ from least-squares problem:

$$\min_{\mathbf{a} \in \mathbb{R}^k} \|\widehat{\mathbf{x}}_c - U_c\mathbf{a}\|_2^2. \tag{3.2}$$

Then the solution of (3.2) is $\mathbf{a} = U_c^{\dagger}\widehat{\mathbf{x}}_c$, which $U_c^{\dagger} = (U_c^T U_c)^{-1} U_c^T$, where $U_c^{\dagger}$ is called the Moore-Penrose inverse. Therefore

$$\widehat{\mathbf{x}}_g \approx U_g\mathbf{a} = U_g U_c^{\dagger} \widehat{\mathbf{x}}_c. \tag{3.3}$$

The steps described above, which will called Tensor Reconstruction, are summarized in Algorithm 4.

Algorithm 5: Tensor Reconstruction

INPUT : A set of complete lateral slides $\{\mathcal{A}(:,i_2,:)\}_{i_2=1}^{n_s}$, and target rank k.
Incomplete lateral slide $\overline{\mathcal{A}} \in \mathbb{R}^{m\times 1\times l}$

OUTPUT : Approximation of $\overline{\mathcal{A}}_g$

Step 1. Create a complete tensor : $\mathcal{X} \in \mathbb{R}^{m\times n_s\times l}$ which the lateral slides are $\mathcal{A}(:,i_2,:), i_2 = 1, ..., n_s$.
Step 2. Construct a basis tensor $\mathcal{U}$ of $\mathcal{X}$.
Step 3. **for** $i = 1$ **to** l **do**
Step 4. Let $\widehat{\mathbf{x}} = \overline{\mathcal{A}}(i) \in \mathbb{R}^m$ and separate to known and unknown components, $\widehat{\mathbf{x}}_c$ and $\widehat{\mathbf{x}}_g$ respectively.
Step 5. Using $\widehat{\mathbf{x}}_c$ to find coefficient vector $\mathbf{a}$ through Least Squares problem: $\min_{\mathbf{a}\in\mathbb{R}^k} \|\widehat{\mathbf{x}}_c - U_c\mathbf{a}\|_2^2$, where $U_c = C^T U$ and U is the corresponding frontal slide.
Step 6. Compute the approximation $\widehat{\mathbf{x}}_g \approx U_g\mathbf{a}$ then $\overline{\mathcal{A}}_g(i) = \widehat{\mathbf{x}}_g$
Step 7. **end for**

3.3. Approximate Tensor Decomposition

Definition 3.1 (**Frobenius norm of tensor** [2]). Suppose $\mathcal{A} = (a_{ijk}) \in \mathbb{R}^{m\times n\times l}$. Then

$$\|\mathcal{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{l} a_{ijk}^2}$$

Based on the tensor SVD, the low-rank approximation is given by

$$\mathcal{A}_k = \texttt{argmin}_{\mathcal{A}\in\mathcal{M}} \|\mathcal{A} - \widetilde{\mathcal{A}}\|_F,$$

where $\mathcal{M}$ describes a special class of tensors that can be written as a product of tensors of appropriate dimension as seen in Theorem 3.2.

Theorem 3.2. *([2] P.653) Let the t-SVD of $\mathcal{A} \in \mathbb{R}^{m\times n\times l}$ and $\mathcal{A} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$ and for $k < \min(m,n)$. Define $\mathcal{A}_k = \mathcal{U}_k * \mathcal{S}_k * \mathcal{V}_k^T$, then $\mathcal{A}_k = \texttt{argmin}_{\mathcal{A}\in\mathcal{M}} \|\mathcal{A} - \widetilde{\mathcal{A}}\|_F$, where $\mathcal{M} = \{\mathcal{X} * \mathcal{Y} | \mathcal{X} \in \mathbb{R}^{m\times k\times l}, \mathcal{Y} \in \mathbb{R}^{k\times n\times l}.\}$*

4. Experimental Results

In this section, we compare the performance of the t-SVD, the t-rSVD and the randomized t-SVD methods to compute POD tensor for recovering missing data in color images through the least-squares approach.

We consider two numerical experiments: color image and gray-scale video. We compare the performance of the reconstructed examples by the computation times and the accuracy.

4.1. Image Quality

The accuracy of the reconstruction is measured by the relative error of the reconstructed missing data which defined as

$$\textbf{error} = \frac{\|\mathcal{A}^* - \mathcal{A}\|_F^2}{\|\mathcal{A}\|_F^2}. \tag{4.1}$$

In order to evaluate the images recovery quality of the missing images, we employ the peak signal-to-noise ratio (PSNR) [7], defined as

$$\mathbf{PSNR} = 10\log_{10}\left(\frac{mnl\|\mathcal{A}\|_\infty^2}{\|\mathcal{A}^* - \mathcal{A}\|_F^2}\right), \tag{4.2}$$

where $\mathcal{A}$ and $\mathcal{A}^* \in \mathbb{R}^{m\times n\times l}$ are a sample image and a reconstruction image, respectively. $\|\mathcal{A}\|_\infty$ is the absolute value maximum of $\mathcal{A}$ and $\|\mathcal{A}\|_F$ is the Frobenius norm of $\mathcal{A}$.

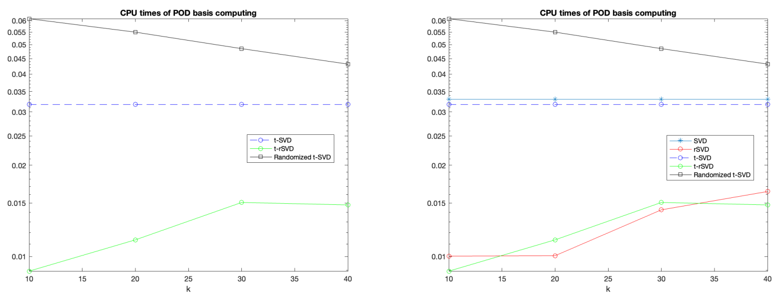
4.2. Test 1: Color Image

We investigate the color image, **Girl** (200x200x3) with 20% and 40% missing, as shown in Figure 2. From the color test image with different dimensions, the 20% and 40% missing pixels are reconstructed in different $[k, k, 3]$, where $k = 10, 20, 30, 40$. The performance of them are shown in Figure 3.

Figure 2. The test image: **Girl**, with the examples 20% and 40% missing pixels.

Figure 3. The reconstructed images for images with 20% missing pixels by using t-SVD, t-rSVD and randomized t-SVD when truncation term $k = 10, 20, 30$, and 40 respectively.

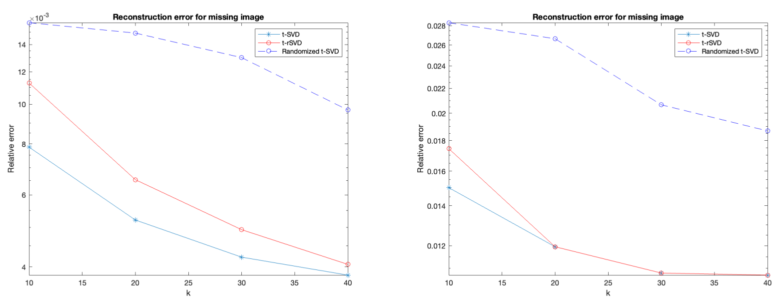
The computational times of the reconstruction process are compared between using t-SVD, t-rSVD and randomized t-SVD for constructing the POD tensor in the reconstruction method, are shown in Figure 4. Since the rSVD method can reduce the computation time when computing the POD basis of the matrix [19–22], the t-rSVD uses less CPU time than the others while randomized t-SVD is the most time consuming. For (B) in Figure 4, we compare the result from using SVD and rSVD methods to compute POD basis for each frontal slices of image tensor. The t-rSVD and rSVD give similar results which are better than the others.



(A) (B)

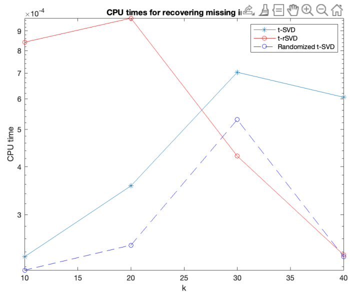
FIGURE 4. Computational time for constructing the POD tensor (A) by using t-SVD, t-rSVD and randomized t-SVD when truncation term $k = 10, 20, 30,$ and 40 respectively. Additional, (B) compared to the previous methods from [19–22] by using SVD and rSVD.

The relative errors in Figure 5 show that the t-SVD and t-rSVD methods are more accurate than the randomized t-SVD while the comparison between using t-SVD and t-rSVD in the proposed method give the same order of accuracy. Furthermore, the proposed method using the t-SVD and the randomized t-SVD uses less CPU times than t-rSVD as shown in Figure 6.

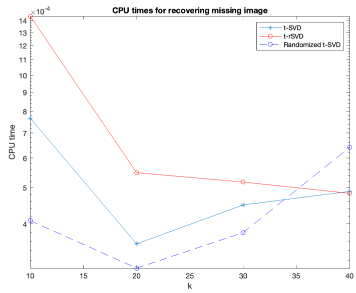


(A) 20% missing (B) 40% missing

FIGURE 5. Relative errors of the reconstruction results of 20% and 40% missing images when truncation term $k = 10, 20, 30,$ and 40 respectively.



(A) 20% missing



(B) 40% missing

FIGURE 6. Computational time for reconstructing missing images by using t-SVD t-rSVD and randomized t-SVD when truncation term $k = 10, 20, 30,$ and 40 respectively.

4.3. TEST 2: VIDEO

We investigate a gray-scale video **Girl** (200x200x10), which has 10 frames. The illustration is shown in Figure 7.

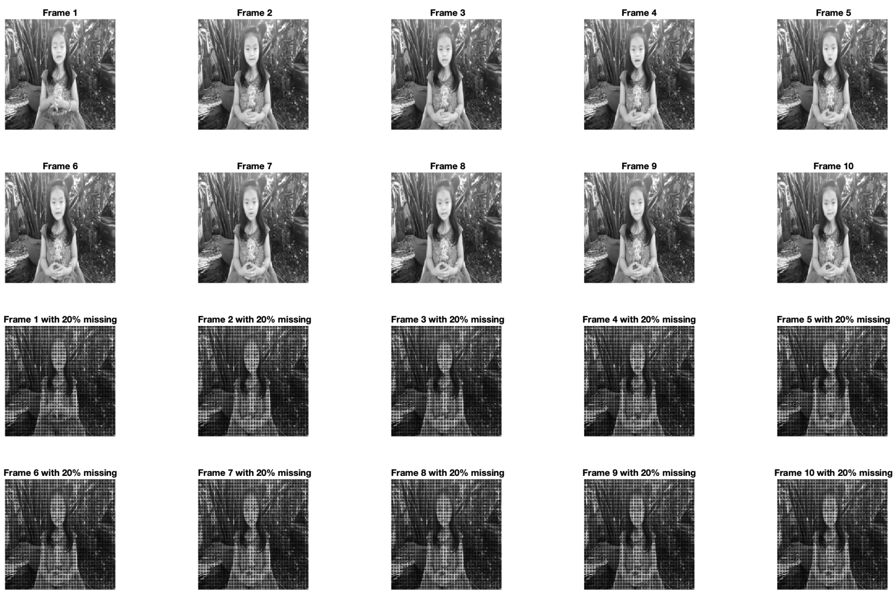
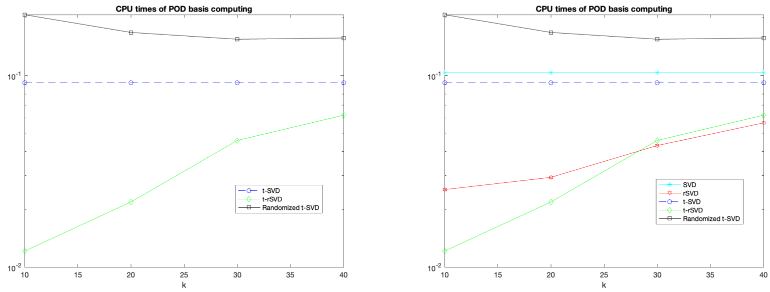


FIGURE 7. The test video: 10 frames gray-scale video with 25% missing pixels.

The computational times for constructing the POD tensor of the 10 frames video are shown in Figure 8. Comparing between using t-SVD, t-rSVD and randomized t-rSVD is the fastest while the randomized t-SVD is the most time consuming. For (B) in Figure 8, the results from the t-rSVD and rSVD are quite similar and using t-SVD is faster than computing each slices by SVD.

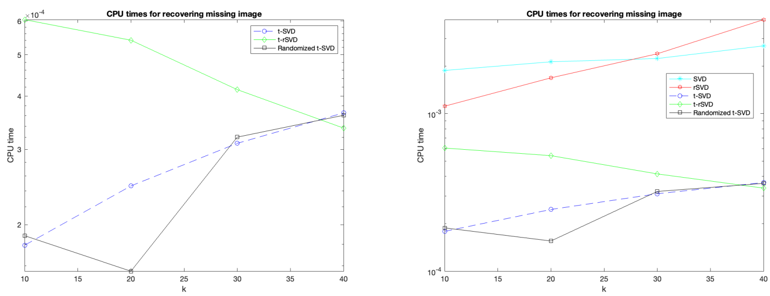


(A) (B)

FIGURE 8. Computational time for constructing the POD tensor (A) by using t-SVD, t-rSVD and randomized t-SVD (B) adding compare to the previous methods from [19–21] by using SVD and rSVD when truncation term $k = 10, 20, 30$, and 40 respectively.

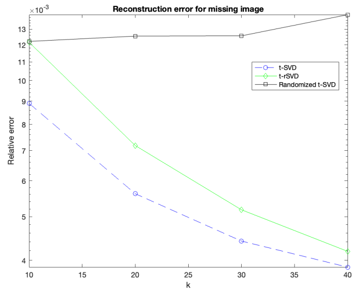
The computational times of the video reconstruction process are shown in Figure 9. Here, the randomized t-SVD uses less CPU time than the others. When comparing to the previous method in [19, 22], the three methods for video tensor are better than the previous methods using SVD as well as rSVD to reconstruct each video frames.

The relative errors of video reconstruction are shown in Figure 10, which illustrates that the t-SVD and t-rSVD methods are more accurate than the randomized t-SVD based on PSNR, while the comparison between using t-SVD and t-rSVD in the proposed method show that they give the same order of accuracy. Furthermore, the proposed method is compared to the previous method in [19, 22], using t-SVD and SVD and the corresponding results are shown to be similar to the ones from using the t-rSVD and rSVD.

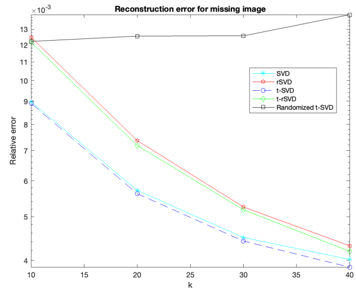


(A) (B)

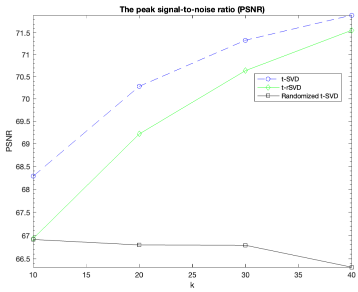
FIGURE 9. Computational time for reconstruction missing images by using t-SVD t-rSVD and randomized t-SVD when truncation term $k = 10, 20, 30$, and 40 respectively.



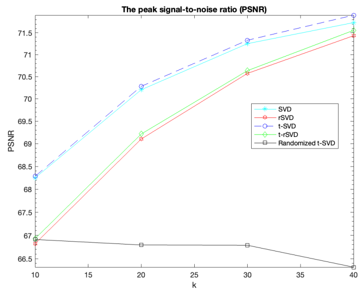
(A)



(B)



(C)



(D)

FIGURE 10. Relative errors and PSNR of the reconstruction results of 25% missing images when truncation term $k = 10, 20, 30$, and 40 respectively.

5. Conclusions

In this paper, we employ the tensor reconstruction approach to recover the missing data pixels in color images and gray-scale video. We have investigated and performed a comparative study of the tensor-based approach with existing techniques, t-SVD, t-rSVD and randomized t-SVD, which can be used to speed up the reconstruction process with some trade off on accuracy. For the color images and gray-scale video represented as the third-order tensors of dimension $m \times n \times l$ in our experiments, the t-rSVD and the t-SVD approaches are suitable for small l, while the randomized t-SVD approach is particularly favorable for large l. The randomness in the reconstruction approaches makes it more efficient in terms of both computation time and memory. Moreover, the effectiveness of proposed method has been compared to the previous method in [19, 22].

Acknowledgements

The authors gratefully acknowledge the financial support provided by Faculty of Science and Technology, Contract No. SciGR 12/2565 and Royal Thai Government Scholarship in the Area of Science and Technology (Ministry of Science and Technology).

References

[1] B. Karen, Third-order tensors as linear operators on a space of matrices, Linear Algebra and Its Applications 433 (2010) 1241–1253.

[2] M.E. Kilmer, C.D. Martin, Factorization strategies for third-order tensors, Linear Algebra and its Applications 435 (3) (2011) 641–658.

[3] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1) (2013) 208–220.

[4] S. Sara, K. Misha, H.P. Christian, A tensor-based dictionary learning approach to tomographic image reconstruction, BIT Numerical Mathematics 56 (2015) 1425–1454.

[5] P. Zhou, C. Lu, Z. Lin, C. Zhang, Tensor factorization for low-rank tensor completion, IEEE Transactions on Image Processing 27 (3) (2018) 1152–1163.

[6] J. Zhang, Design and Application of Tensor Decompositions to Problems in Model and Image Compression and Analysis, Ph.D. Thesis, Mathematics, Tufts university, 2017.

[7] F. Yali, G. Zhou, Y. Qiu, W. Sun, Orthogonal random projection based tensor completion for image recovery, Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (2018) 1350–1354.

[8] Q. Zhenzhi, M. Zhenyu, Z. Liping, Singular value decomposition of third order quaternion tensors, Applied Mathematics Letters (2021).

[9] Y. Itin, S. Reches, Decomposition of third-order constitutive tensors, Mathematics and Mechanics of Solids 27 (2) (2022) 222–249.

[10] Y. Liu, Tensors for Data Processing: Theory, Methods, and Applications, Academic Press is an imprint of Elsevier, 2022.

[11] S. Volkwein, Proper Orthogonal Decomposition: Theory and Reduced-Order Modelling, Lecture Notes, University of Konstanz, 2013.

[12] T. Sarlos, Improved approximation algorithms for large matrices via random projections, In Foundations of Computer Science, 47th Annual IEEE Symposium (2006) 143–152.

[13] E. Liberty, F. Woolfe, P.G. Martinsson, V. Rokhlin, M. Tygert, Randomized algorithms for the low-rank approximation of matrices, in Proc. Natl. Acad. Sci. 104 (51) (2007) 20167–20172.

[14] J. Cai, E. Cands, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM Journal on Optimization 20 (4) (2010) 1956–1982.

[15] N. Halko, P.G. Martinsson, J.A. Tropp, Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions, Siam Review 53 (2) (2011) 217–288.

[16] X. Hu, Q. Zhang, J. Zhang, J. Erway, R. Plemmons, Randomized svd methods in hyperspectral imaging, Journal of Electrical and Computer Engineering 2012 (2012) 2737–2764.

[17] S. Kahu, R. Rahate, Image compression using singular value decomposition, International Journal of Advancements in Research and Technology 2 (8) (2013) 244–248.

[18] H. Ji, Y. Li, Gpu accelerated randomized singular value decomposition and its application in image compression (2014).

[19] S. Intawichai, S. Chaturantabut, An application of randomized singular value decomposition on image reconstruction using least-squares approach, Preprint in Thai Journal of Mathematics (2018).

[20] S. Intawichai, S. Chaturantabut, A numerical study of eicient sampling strategies for randomized singular value decomposition, Thai Journal of Mathematics, Special Issue: Annual Meeting in Mathematics 2019 (2020) 371–385.

[21] S. Intawichai, S. Chaturantabut, Missing image data reconstruction based on least-squares approach with randomized SVD, In International Conference on Intelligent Computing & Optimization, Springer, Cham (2020) 1059–1070.

[22] S. Intawichai, S. Chaturantabut, A missing data reconstruction method using an accelerated least-squares approximation with randomized SVD, Algorithms 15 (190) (2022) 1–16.