Int. J. Nonlinear Anal. Appl. 14 (2023) 3, 223–230
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.24206.2688

The application of tensor ring and tensor train completion to image recovery

Hamid Reza Yazdani[a,], Alireza Shojaeifard[a], Mohsen Shahrezaei[b]

[a] *Department of Mathematics, Faculty of Basic Sciences, Imam Hossein Comprehensive University, Tehran, Iran*

[b] *Faculty of Defense and Engineering, Imam Hossein University, Tehran, Iran*

(Communicated by Mugur Alexandru Acu)

Abstract

Tensor completion has numerous applications in digital image processing such as image recovery and video overlay. In this paper, we consider two new approaches to tensor completion. Efficient low-rank tensor with tensor train and tensor ring for image recovery, some basic concepts about tensor algebra and completion problems are presented, after that Tensor completion based on the tensor train and tensor ring are offered and implemented on some examples for image recovery with different observed ratios. The results of these implementations are compared and final results are proposed.

Keywords: Image Recovery, Tensor Completion (TC), Tensor Ring (TR), Tensor Train (TT)
2020 MSC: 15A69, 58J60, 68T07

1 Introduction

Here, we consider efficient low-rank tensor completion (TC) based on the tensor ring (TR) and tensor train (TT) methods for image recovery and video completion [10]. Tensor decompositions for expressing and saving data have lately attracted significant concentration due to their effectiveness in compressing data for statistical and mathematical signal processing [3]. In this paper, we concentrate on TR decomposition [10] and in particular, its connection to Matrix Product States (MPS) [6] representation for tensor and utilize it for completing data from missing entries. The tensor ring completion uses a different initialization. This method is based on the TR analysis. The type of decomposition removes unit rank limitations for boundary tensor factors and applied the sign operator in decomposition. Multi-linear multiplications between factors are also used by generalizing matrix states. To complete the data by using tensorial decompositions, a primary issue is rank [7]. Although the rank in TR is a vector, we can assume that all ranks are the same as in other cases, so a single parameter is provided that can be adjusted based on the data and the number of available samples in each application [1]. The use of a sign operator in the TR structure has some challenges to completing the tensor analysis in TT mode. It can be said that the structure of TR is equivalent to the distance structure in tensor networks (TN). Understanding this structure can help to know the completion for TN in general cases [10].

Email addresses: hamidreza.yazdani@gmail.com (Hamid Reza Yazdani), Ashojaeifard@ihu.ac.ir (Alireza Shojaeifard), mshahrezaee@mail.bmn.ir (Mohsen Shahrezaei)

Received: August 2021 *Accepted:* September 2022

The remainder of the paper is constructed as follows. In section 2, we present the basic notation and preliminaries on the TC, TR, and TT decomposition. In section 3, we describe the problem statement and propose the principal algorithm. We also explain the computational complexity of the proposed algorithm based on the TT and TR. In section 4, we perform and examine the algorithm extensively against competing methods on a number of real and artificial data experiments in section 5, we provide conclusions and future research directions.

2 Preliminaries

In this section, we introduce some preliminaries and notations about tensor theory and related topics.

Definition 2.1. If $\mathbb{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ ia a tensor, and $M \in \mathbb{R}^{m_k \times n_k}$ is a matrix, then mode-k matrix product of M and $\mathbb{X}$ defined as follows [2]:

$$\sum_{j=1}^{n_k} M(i,j).\mathbb{X}(\alpha_1 \dots \alpha_{k-1}, j, \alpha_{k+1}, \dots, \alpha_d) \tag{2.1}$$

In practice, The mode-k matrix product is a contraction between a tensor and a matrix that produces another tensor.

Definition 2.2. A tensor can be defined in various ways at different levels of abstraction. We follow the most general way and define it as a multidimensional array [8]. The dimensionality of it is described as its order. An Nth-order tensor is an N-way array, also known as N-dimensional or N-mode tensor, denoted by $\mathbb{X}$. We use the term order to refer to the dimensionality of a tensor (e.g., Nth-order tensor), and the term mode to express operations on a specific dimension (e.g., mode-n product), for more examples, see figure 1. The tensor dimension with mode i could be an expression, where the expression inside () is evaluated as a scalar, e.g. $\mathbb{X} \in \mathbb{R}^{(I_1 I_2)\times(I_3 I_4)\times(I_5 I_6)}$ represents a 3-mode tensor where dimensions along each mode is $I_1 I_2$, $I_3 I_4$, and $I_5 I_6$ respectively. An entry inside a tensor $\mathbb{X}$ is represented as $\mathbb{X}(i_1, i_2, \dots, i_n)$, where $i_k : k = 1, 2, \dots, n$ is the location index along the k^{th}-mode [1].

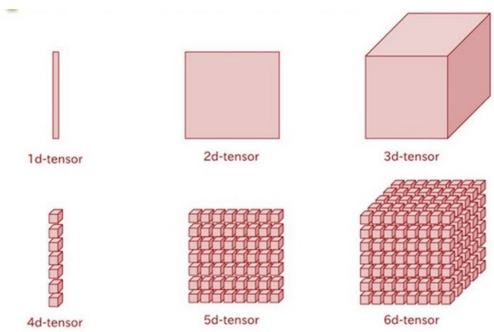

Figure 1: The representation of tensors based on the dimension.

In linear algebra, the singular value decomposition (SVD) is a factorization of a real or complex matrix that generalizes the eigendecomposition of a square normal matrix into an orthonormal eigenbasis to any $m \times n$ matrix via an extension of the polar decomposition. Here, this idea generalizes to tensors. The SVD is a robust tool for displaying the structure of a matrix and for achieving its essence through optimal, data-sparse representations [2].

Definition 2.3. High order Singular Value Decomposition (HoSVD) of a tensor $\mathbb{X} \in \mathbb{R}^{n_1 \times \dots \times n_d}$ involves computing the matrix SVDs of modal unfoldings $\mathbb{U}_{(1)}, \dots, \mathbb{U}_{(d)}$ [4]. This result in a reperestation of $\mathbb{X}$ as a sum of rank-1 tensors as follows:

$$\mathbb{X} = \mathbb{S} \times_1 U_1 \times_2 U_2 \dots \times_d U_d \tag{2.2}$$

where $\mathbb{S}$ is a low rank tensor core of $\mathbb{X}$, U_is for $i = 1, \dots, d$ is inverse tensor factors, and $\times_d$ is a d-mode production, see figure 2 [1].

Tensor completion is defined as the problem of filling the missing elements of partially observed tensors. As its special matrix case [9], to avoid being an underdetermined and intractable problem, low rank is a certain hypothesis to limit the degree of freedoms of the missing entries [5]. Since a tensor has different types of rank definitions, to give a nearly general mathematical formulation of the low-rank tensor completion (LRTC) problem [5].

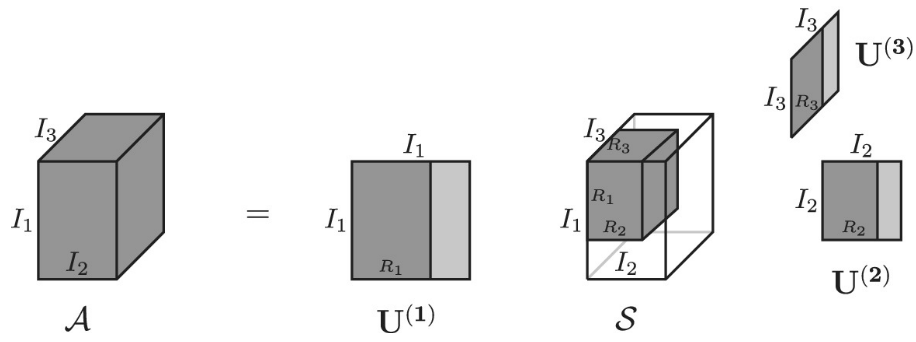

Figure 2: The representation of HoSVD.

Definition 2.4. Given a low-rank tensor $\mathbb{T}$ with missing entries, the goal of completing it can be formulated as the following optimization problem:

$$\begin{array}{ll} \text{minimiz } e_{\mathbb{X}} & \text{rank}_*(\mathbb{X}), \\ \text{Subject } to & \mathbb{X}_\omega = \mathbb{T}_\Omega. \end{array}$$

where $rank_*$ denote a specific type of tensor rank based on the rank assumption of the given tensor $T, \mathbb{X}$ represents the completed low-rank tensor of τ, and Ω is an index set denoting the indices of observations. The intuitive explanation of the above equation is that: we expect to find a tensor $\mathbb{X}$ with minimum rank, which is subjects to the equality constraints given by the observations, please see figure 3 [9].

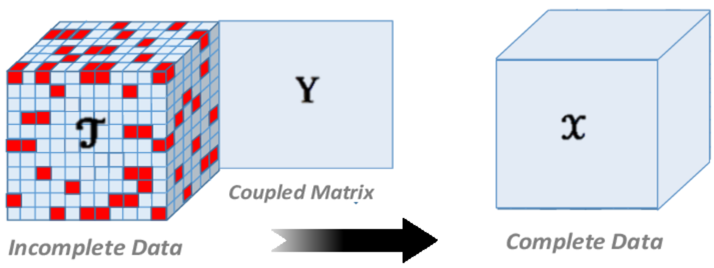

Figure 3: Tensor Completion Problem (TC).

For $\mathbb{X} \in \mathbb{R}^{I_1 \times \ldots \times I_n}$ as a n-mode tensor, mode-i unfolding of $\mathbb{X}$ denoted as $\mathbb{X}_i$, matricized the tensor $\mathbb{X}$ by putting the i-th mode in the matrix rows and remaining modes with the original order in the columns such that:

$$\mathbb{X}_i \in \mathbb{R}^{I_i \times (I_1 \ldots I_{i-1} I_{i+1} \ldots I_n)}$$

Definition 2.5. Let $\mathbb{X} \in \mathbb{R}^{I_1 \times \ldots \times I_n}$ be an n-order tensor with I_i-dimension along the i^{th} mode, then any entry inside the tensor, denoted as $\mathbb{X}(i_1, \ldots, i_n)$, is represented by:

$$\mathbb{X}(i_1, \ldots, i_n) = \sum_{r_i=1}^{R_1} \ldots \sum_{r_n=1}^{R_n} U_1(r_n, i_1, r_1) \ldots U_n(r_{n-1}, i_n, r_n),$$

where $U_i \in \mathbb{R}^{R_i - 1 \times I_i \times R_i}$ is a set of 3-order tensors, also named matrix product states (MPS), which consist the bases of the tensor ring structures.

Remark 2.6. Note that, In the formulation of the tensor ring, tensor ring rank is the vector $[R_1, \ldots, R_n]$. In general R_is are not necessary to be the same. We set $R_i = R$ for $i = 1, \ldots, n$ and the scalar R is referred as the tensor ring rank [10].

Definition 2.7. Tensor train is a special case of tensor ring when $R_n = 1$ [11].

3 Algorithms

Given a tensor $\mathbb{X} \in \mathbb{R}^{I_1\times\dots\times I_n}$ that is partially observed at locations Ω, let $\mathbb{P}_\Omega \in \mathbb{R}^{I_1\times\dots\times I_n}$ be the corresponding binary tensor in which 1 denotes an observed entry and 0 represents a missing entry. The problem is to obtain a low tensor ring rank (TR-Rank) approximation of the tensor $\mathbb{X}$, denoted by $f(U_1 \dots U_n)$, such that the recovered tensor matches $\mathbb{X}$ at $\mathbb{P}_\Omega$. This problem is referred to as the tensor completion problem under the ring model, which is equivalent to the following problem:

$$\min \quad ||\mathbb{P}_\Omega \circ (f(U_1 \dots U_n) - \mathbb{X}||_F^2. \tag{3.1}$$

Note that the rank of the tensor ring R is predefined and the dimension of $U_{i:i=1,\cdots,n}$ is $\mathbb{R}^{R\times I_i\times R}$ [10].

To solve this problem, we introduce an algorithm, referred to as Tensor Ring completion by Alternating Least Square (TR-ALS) in two steps:

1) First, Choose an initial starting point by using Tensor Ring Approximation (TRA). This initialization algorithm is detailed in subsection 3.1.

2) Update the solution by utilizing Alternating Least Square (ALS) that alternatively (in cyclic order) determines a factor say U_i keeping the other factors fixed. This algorithm is detailed in subsection 3.2.

3.1 Tensor Ring Approximation (TRA)

Here, a heuristic initialization algorithm, namely TRA, for solving (3.1) is proposed. This algorithm is a revised version of TR decomposition as proposed in [7]. We first conduct a TT decomposition on the zero-filled data, where the rank is constrained by SVD. Next, an estimate for the TR is formed by extending the acquired factors to the desired dimensions by filling the remaining entries with small random numbers. We remark that the small entries show faster convergence as compared to zero entries based on our studied small examples, and consequently induces the choice in the algorithm. Further, non-zero random entries help the algorithm initialize with larger ranks since the TT decomposition has the corner ranks as 1. Having non-zero entries can help the algorithm not getting stuck in local optima of low corner rank [10]. Figure 4 shows the main algorithm steps.

Input: Zero-filled Tensor $\mathcal{X}_\Omega \in \mathbb{R}^{I_1\times I_2\times\dots\times I_n}$, binary observation index tensor $\mathcal{P}_\Omega \in \mathbb{R}^{I_1\times I_2\times\dots\times I_n}$, tensor ring rank R, thresholding parameter tot, maximum iteration $maxiter$

Output: Recovered tensor $\mathcal{X}_R$

Apply tensor ring approximation in Algorithm 1 on $\mathcal{X}_\Omega$ to initialize the MPSs $\mathcal{U}_{i:i=1,\cdots,n} \in \mathbb{R}^{R\times I_i\times R}$. Set iteration parameter $\ell = 0$.
while $\ell \leq maxiter$ **do**
    $\ell = \ell + 1$
    for $i = 1$ to n **do**
        Solve by Least Square Method $\mathcal{U}_i{}^{(\ell)} = \operatorname{argmin}_{\mathcal{U}} \|\mathcal{P}_\Omega \circ (\mathcal{U}\mathcal{U}_{i+1}^{(\ell-1)} \dots \mathcal{U}_n^{(\ell-1)} \mathcal{U}_1^{(\ell)} \dots \mathcal{U}_{i-1}^{(\ell)} - \mathcal{X})\|_F^2$
    end for
    if $\frac{\|\mathcal{U}_n^{(\ell+1)} - \mathcal{U}_n^{(\ell)}\|_F}{\|\mathcal{U}_n^{(\ell)}\|_F} \leq tot$ **then**
        Break
    end if
end while
Return $\mathcal{X}_R = \text{reshape}(\mathcal{U}_1^{(\ell)}\mathcal{U}_2^{(\ell)} \dots \mathcal{U}_{n-1}^{(\ell)}\mathcal{U}_n^{(\ell)})$

Figure 4: Tensor Ring Approximation (TRA)

3.2 TR-ALS Algorithm

The proposed tensor ring completion by alternating the least square method (TR-ALS) solves (3.1) by solving the following problem for each i iteratively. The factors are initialized from the TRA algorithm presented in the previous subsection. Figure 5 shows the main algorithms. The stopping step in TR-ALS is measured via the changes of the

Input: Missing entry zero filled tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$, TR-Rank R, small random variable depicting the standard deviation of the added normal random variable σ

Output: Tensor train decomposition $\mathcal{U}_{i:i=1,\cdots,n} \in \mathbb{R}^{R \times I_i \times R}$

1: Apply mode-1 canonical matricization for $\mathcal{X}$ and get matrix $\mathbf{X}_1 = \mathcal{X}_{<1>} \in \mathbb{R}^{I_1 \times (I_2 I_3 \cdots I_n)}$

2: Apply SVD and threshold the number of singular values to be $T_1 = \min(R, I_1, I_2 \cdots I_n)$, such that $\mathbf{X}_1 = \mathbf{U}_1 \mathbf{S}_1 \mathbf{V}_1^\top, \mathbf{U}_1 \in \mathbb{R}^{I_1 \times T_1}, \mathbf{S}_1 \in \mathbb{R}^{T_1 \times T_1}, \mathbf{V}_1 \in \mathbb{R}^{T_1 \times (I_2 I_3 \cdots I_n)}$. Reshape $\mathbf{U}_1$ to $\mathbb{R}^{1 \times I_1 \times T_1}$ and extend it to $\mathcal{U}_1 \in \mathbb{R}^{R \times I_1 \times R}$ by filling the extended entries by random normal distributed values sampled from $\mathcal{N}(0, \sigma^2)$.

3: Let $\mathbf{M}_1 = \mathbf{S}_1 \mathbf{V}_1^\top \in \mathbb{R}^{T_1 \times (I_2 I_3 \cdots I_n)}$.

4: **for** $i = 2$ to $n - 1$ **do**

5: Reshape $\mathbf{M}_{i-1}$ to $\mathbf{X}_i \in \mathbb{R}^{(T_{i-1} I_i) \times (I_{i+1} I_{i+2} \cdots I_n)}$.

6: Compute SVD and threshold the number of singular values to be $T_i = min(R, T_{i-1} I_i, I_{i+1} \cdots I_n)$, such that $\mathbf{X}_i = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^\top, \mathbf{U}_i \in \mathbb{R}^{(T_{i-1} I_i) \times T_i}, \mathbf{S}_i \in \mathbb{R}^{T_i \times T_i}, \mathbf{V} \in \mathbb{R}^{T_i \times (I_{i+1} I_{i+2} \cdots I_n)}$. Reshape $\mathbf{U}_i$ to $\mathbb{R}^{T_{i-1} \times I_i \times T_i}$ and extend it to $\mathcal{U}_i \in \mathbb{R}^{R \times I_i \times R}$ by filling the extended entries by random normal distributed values sampled from $\mathcal{N}(0, \sigma^2)$.

7: Set $\mathbf{M}_i = \mathbf{S}_i \mathbf{V}_i^\top \in \mathbb{R}^{T_i \times (I_{i+1} I_{i+2} \cdots I_n)}$

8: **end for**

9: Reshape $\mathbf{M}_{n-1} \in \mathbb{R}^{T_{n-1} \times I_n}$ to $\mathbb{R}^{T_{n-1} \times I_n \times 1}$, and extend it to $\mathcal{U}_n \in \mathbb{R}^{R \times I_n \times R}$ by filling the extended entries by random normal distributed values sampled from $\mathcal{N}(0, \sigma^2)$ to get $\mathcal{U}_n$

10: Return $\mathcal{U}_1, \cdots, \mathcal{U}_n$

Figure 5: TR-ALS Algorithm

last tensor factors U_n since if the last factor does not change, the other factors are less possible to change.

We note that tensor train completion provides a similar complexity as tensor ring completion. Nevertheless, tensor train rank is a vector and it is hard for tuning to reach the optimal completion. The middle ranks in the tensor train are large in general, leading to the significantly higher computational complexity of tensor train [11]. This is alleviated in part by the tensor ring structure which can be parametrized by the tensor ring rank which can be smaller than the intermediate ranks of the tensor train in general. In addition, the single parameter in the tensor ring structure leads to ease in characterizing the performance for different ranks and can be easily tuned for practical applications. The lower ranks lead to the lower computational complexity of data completion under the tensor ring structure as compared to the tensor train structure [10].

4 Implementation of algorithms

In this section, we performed our codes, algorithms, and experiments on the Asus Laptop with configuration as given in table 1.

Table 1: The configuration of expriment system.

CPU	Ci7-2670QM (8 Cores)
Frequency	2.2-3.1 GHz
RAM	16 GB (DDR3)
GPU	Nvidia Geforce GT 540M (2GB)
O.S.	Windows-10 Pro (64bit)
Software	MATLAB R2021a (64bit)

It must be noted that in this set of codes, the software package of *Tensorlab* is also used. We note that the recovery error is defined as:

$$RE = \frac{||\mathbb{T} - \mathbb{X}||_F}{||\mathbb{X}||_F} \tag{4.1}$$

Tensor ring completion by alternating least square (TR-ALS) algorithm is an iterative algorithm and the maximum iteration, *maxiter*, is set to be 300. The convergence is obtained by the change of the last factorization term U_n, where the error threshold is set to be 10^{-10}.

4.1 Image Completion

In this section, we examine the completion of RGB Einstein Image, that employed ad a 3-order tensor $\mathbb{X} \in \mathbb{R}^{600\times600\times3}$. In the initial step, a reshaping operation is utilized to transform the image into a 7-order tensor of size $\mathbb{R}^{6\times10\times10\times6\times10\times10\times3}$. Reshaping low order tensors into high order tensors is a general practice in research papers and has shown enhanced achievement in classification and completion [11].

In the field of image recovery, with only a 10 percent observation ratio (i.e. 90 percent distortion ratio), this method has a minimum error of 10.83 percent and accordingly can recover and restore up to 90 percent distortion with 89.17 percent confidence and certainty. You can see the achievement results in Figure 6.

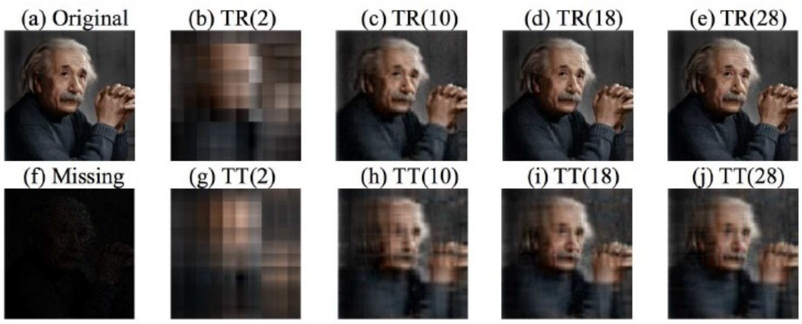

Figure 6: Comparison of the implementation of the TR against TT method in different cases of tensor rank on Einstein image.

In the second step, we consider Yale Face dataset that includes 38 people with 9 poses under 64 illumination conditions. Each images has the size of 192×168, wherein we down-sample the size of each image to 48×42 for ease of computation. We investigate the image subsets of 38 people under 64 illumination with 1 pose by formatting the data into a 4-order tensor in $\mathbb{R}^{48\times42\times64\times38}$, which is further reshape into a 8-order tensor $\mathbb{X} \in \mathbb{R}^{6\times8\times6\times7\times8\times8\times19\times2}$.

In the area of recovery face images in various exposure modes by the aforementioned Yale dataset, with a 10 percent observation ratio (90 percent distortion) of the original images, the minimum error under the rank of 30 reaches 21.57 percent. So this method can recover up to 90 percent of distorted images with 78.43 percent accuracy.

Figure 7 shows the original image, missing images, and recovered images using TR-ALS and TT-ALS algorithms for ranks of 10, 20, and 30, wherever the completion results given by TR-ALS better capture the detailed information given from the image and recovers the image with a better resolution.

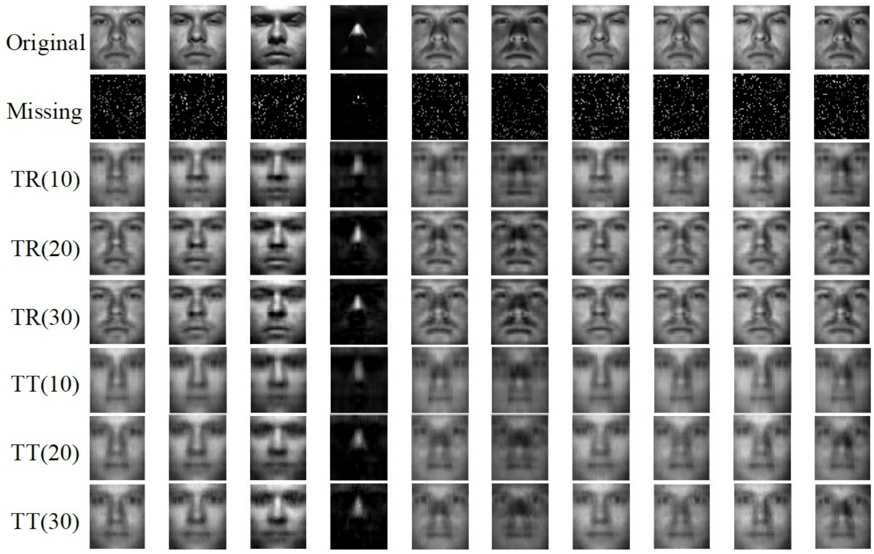

Figure 7: Comparison of the results of the implementation of the tensor completion method based on TR and TT in different ranks in the Yale image data set of different facial exposure modes.

4.2 Video Completion

The video data, we employed is high-speed camera video for gun shooting, that is downloaded from You Tube with 85 frames in total and each frame consists of a $100 \times 260 \times 3$ image. Hence the video is a 4-order tensor of size $100 \times 260 \times 3 \times 85$, which is further reshaped into an 11-order tensor of size $5 \times 2 \times 5 \times 2 \times 13 \times 2 \times 5 \times 2 \times 3 \times 5 \times 17$ for completion. Video is multi-dimensional data with different color channels a time dimension in addition to the $2D$-image structure [10].

In the area of video completion, our studies show that by observation ratio of 10 percent (90 percent distortion), TC based on the TR in the rank 30, has an error of 6.25 percent, while the TC based on the TT has an error of 16.99 percent. Therefore, the method of TT can recover video images by a distortion ratio of 90 percent with an accuracy of 93.75 percent. For more details, please see figure 8.

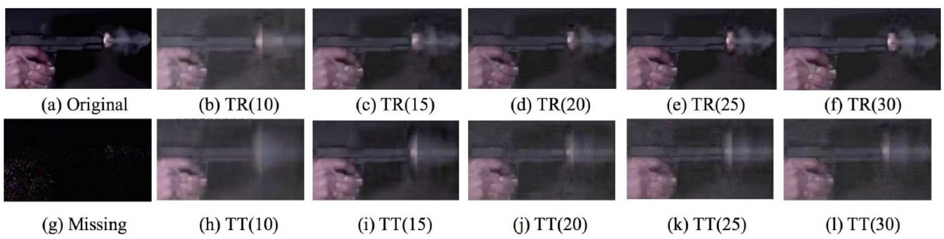

Figure 8: Comparison of the implementation of TC in video completion based on TR and TT in different ranks for the bullet shooting video.

5 Conclusions

In this paper, we consider the low-rank tensor completion method based on the TT and TR. The TC on the TT and TR are presented and implemented on some standard data sets and compared with other famous and related methods. Our main goal is to recover and reconstruct distorted and noisy images and video completion by this method. This algorithm employs the MPS representation and utilizes alternating minimization over the low-rank factors for completion. In general, it can be said that this method has a significant efficiency in recovery images, video completions under various exposure modes with a maximum distortion percentage of up to 90 percent. This method can be used for image completion issues, completing various exposure modes of the face, and completing the video successfully up to 90 percent distortion and aberration.

References

[1] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa and H. A. Phan, *Tensor decompositions for signal processing applications: From two-way to multi-way component analysis*, IEEE Signal Process. Mag. **32** (2015), no. 2, 145—163.

[2] E.J. Candes and B. Recht, *Exact matrix completion via convex optimization*, Commun. ACM **55** (2012), no. 6, 111–119.

[3] T. Kolda and B. Bader. *Tensor decompositions and applications*, SIAM Rev. **51** (2009), no. 3, 455-–500.

[4] M. Kurucz, A. Benczur and K. Csalogany, *Methods for large scale svd with missing values*, Proc. KDD Cup Workshop, San José, USA: Citeseer, 2007, vol. 12.

[5] J. Liu, P. Musialski , P. Wonka, and Ye J., *Tensor completion for estimating missing values in visual data*, IEEE Trans. Pattern Anal. Machine Intel. **35** (2012), no. 1, 208–220.

[6] R. Orus, *A practical introduction to tensor networks: Matrix product states and projected entangled pair states*, Ann. Phys. **349** (2014), 117—158.

[7] I. V. Oseledets. *Tensor train decomposition*, SIAM J. Sci. Comput. **33** (2011), no. 5, 2295-–23171.

[8] L. Qi, *The spectral theory of tensors*, In textitarXiv: 1201.3424v1 (2012).

[9] Q. Song, H. Ge, J. Caverlee and X. Hu, *Tensor completion algorithms in big data analytics*, ACM Trans. Knowledge Disc. Data, **6** (2019), no. 13, 1—48.

[10] W. Wang, V. Aggrawal and S. Aeron, *Efficient low rank tensor ring completion*, IEEE Int. Conf. Comput. Vision (ICCV), 2017, pp. 5697–5705.

[11] Q. Zhao, G. Zhou, S. Xie, L. Zhang and A. Cichocki, *Tensor ring decomposition*, *arXiv preprint arXiv:1606.05535* (2016).