International Journal of Latest Trends in Engineering and Technology
Vol.(13)Issue(2), pp.102-107
DOI: http://dx.doi.org/10.21172/1.132.19
e-ISSN:2278-621X

TENSOR RING DECOMPOSITION FOR VISUAL DATA DENOISING VIA TENSOR RANDOM PROJECTION

Longhao Yuan[1], Jianting Cao[2], Qibin Zhao[3]

Abstract-Large-scale data have posed a big computational challenge to traditional data processing methods. The recently proposed tensor ring decomposition (TRD) has shown to be a promising tool to process large-scale data. However, the existing TRD algorithms are of high computational cost which makes it less efficient to apply to large-scale datasets. In this paper, by employing the random projection method to TRD, we propose a non-iterative TRD algorithm for fast large-scale data decomposition which can be used for image denoising tasks. In the experiments of large visual data denoising, our method shows satisfying performance and huge speed-up without loss of accuracy, in comparison with traditional algorithms.
Keywords – Large-scale data, tensor ring decomposition, randomized algorithm, dimensionality reduction, denoising

1. INTRODUCTION

Large-scale data are ubiquitous in various research fields such as machine learning [1], signal processing [2], pattern/image recognition [3], and remote sensing [4]. Due to the properties of high-order and high complexity of large-scale data, it is hard for traditional methods like singular value decomposition (SVD) and principal component analysis (PCA) to process the data. On the one hand, high-order data have to be transformed into matrices to fit the matrix-based method, which will cause the loss of structure information of the high-order data. Secondly, traditional methods require a vast computational resource to process large-scale data, which hinders the practical applicability and efficiency of large-scale data processing.

Tensors (i.e., multi-dimensional arrays) are the natural and compact representation of large-scale and high-order data. Tensor decomposition is a data processing method which is to decompose a tensor into a sequence of factors. By this way, it transforms the tensor into a latent space of low dimensionality, thus reducing the data dimensionality. Moreover, the decomposition factors usually can be considered as the extracted features of the tensor. CANDECOMP/PARAFAC decomposition (CPD) [4] and Tucker decomposition (TKD) [5] are the most popular tensor decomposition models, and the tensor methodologies based on the two models have been widely studied. Recently, tensor train decomposition (TTD) [6] and tensor ring decomposition (TRD) [7] have attracted people's attention, due to the high compression ability and multi-linear representation ability. Though TTD and TRD are named matrix product state (MPS) together, compared with the TTD, the TRD release the rank constraint on the first and the last core tensors to $R_0 = R_N$, while the original constraint on TT is rather stringent, i.e., $R_0 = R_N = 1$, thus providing more model flexibility. In this way, TRD can be considered as the generalization of TTD. Despite the success of TR, there are still limitations and problems of TRD that remain unsolved. Although TRD is a promising tool to process large-scale and high-order data, the existing TRD-based algorithms like alternative least squares (ALS) and gradient descent (GD) are of low-efficiency due to their high computational cost and low convergence rate, which require large computational resource. In consideration of this problem, we employ a randomized algorithm to the TRD, to develop fast and efficient TRD algorithm for large-scale data.

The randomized methods are powerful tools for the acceleration of computation, and they have been widely studied for many years [13]. Recently, tensor decomposition algorithms based on randomized methods have become popular. Paper [11] proposes a randomized Tucker decomposition (RTucker) for large-scale TKD. The algorithm can achieve parallel computation so it is suitable for distributed computation. It can obtain nearly optimal approximation even for very noisy and large-scale data. The algorithm is able to process arbitrarily large-scale tensor which has low multi-linear rank, and the algorithm is robust to many different datasets. Two algorithms which are based on ALS and block coordinate descent (CBD)respectively are proposed in [10] to achieve fast tensor CPD. The algorithms are much faster than the traditional CPD algorithms without loss of accuracy. They first generate a relatively small tensor from the large-scale tensor by random projection, and then calculate the CPD of the small tensor. Finally, the CPD of the small tensor is projected to obtain the CPD of the large-scale tensor. Although randomized tensor decomposition methods show high efficiency, to the best of our

---

[1] Graduate School of Engineering, Saitama Institute of Technology, Tensor Learning Unit, RIKEN Center for Advanced Intelligence Project (AIP), Japan
[2] Graduate School of Engineering, Saitama Institute of Technology, Tensor Learning Unit, RIKEN Center for Advanced Intelligence Project (AIP), Japan, School of Computer Science and Technology, Hangzhou Dianzi University, China
[3] Tensor Learning Unit, RIKEN Center for Advanced Intelligence Project (AIP), Japan, School of Automation, Guangdong University of Technology, China

knowledge, there are only a few studies about it and the experiments are only conducted on synthetic data without noise. Moreover, the randomized methods have not yet been applied to TRD, and the performance of TRD algorithms based on randomization technology in real-world data have not yet been explored. Based on the fact that TRD is the most recent and popular tensor decomposition model but lacks efficient algorithms for large-scale tensor, in our study, we aim to develop a randomized TRD algorithm which applies tensor random projection technique on large-scale TRD tasks.

This paper first proposes the randomized tensor ring decomposition (RTRD) based on traditional tensor ring singular value decomposition (TRSVD)and the tensor random projection (TRP). The RTRD algorithm is non-iterative and can automatically choose the TR-rank according to the prescribed approximation error, so it is of high efficiency and almost parameter-tuning-free. The RTRD algorithm is then applied to a large-scale RGB image denoising experiment and shows higher performance and much lower running time than the traditional TRSVD algorithm. A hyperspectral image (HSI) with noise is used in the next denoising experiment and many related large-scale data processing algorithms are compared. The proposed RTRD outperforms all the compared algorithms in the HIS denoising task in both performance and running time.

The rest of the paper is organized as follows. The preliminaries for the demonstration of our algorithm is shown in section II. Our RTRD algorithm is illustrated in section III. The denoising experiments are presented in section IV. The conclusion is given in section V.

2. PRELIMINARIES

2.1. Notations

In this paper, we mainly employ the notations in [9]. Tensors are denoted by calligraphic letters, e.g., $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. Matrices and vectors are denoted by normal capital letters and normal lowercase letters, respectively, e.g., $\mathrm{X} \in \mathbb{R}^{I_1 \times I_2}$ and $\mathrm{x} \in \mathbb{R}^{I}$. Scalars are denoted by italic lowercase letters or italic uppercase letters, e.g., x, X. A tensor sequence $\{\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \cdots, \mathcal{X}^{(\mathrm{N})}\}$ is denoted as $\{\mathcal{X}^{(n)}\}_{n=1}^{N}$, or $[\mathcal{X}]$ for simplicity. Moreover, we employ two types of tensor unfolding (a.k.a., matricization) operations. For a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the mode-n unfolding is denoted by $X_{(n)} \in \mathbb{R}^{I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N}$, and the other similar unfolding named n-unfolding is denoted by $X_{<n>} \in \mathbb{R}^{I_n \times I_{n+1} \cdots I_N I_1 \cdots I_{n-1}}$. Furthermore, the mode-n product of tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and matrix $\mathrm{B} \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{Z} = \mathcal{X} \times_n B \in R^{I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$.

2.2. Tensor ring decomposition

In recent years, the concept of tensor networks has been proposed and has become a powerful and promising aspect of tensor methodology [2]. Tensor ring (TR) decomposition is one of the most recent proposed and successful tensor networks which is based on the matrix product state (MPS). It is a generalization of tensor train (TT) decomposition with super compression and computational efficiency properties. It offers an enhanced representation ability, latent factors permutation flexibility (i.e. tensor permutation is directly related to the permutation of tensor factors) and structure information interpretability (i.e. each tensor factor can represent a specific feature of the original tensor) [8]. The TRD has become a powerful and promising tensor decomposition model to solve large-scale problems because of its multi-linear representation and super compressibility. The most significant advantage of TRD is that the space complexity grows linearly in the tensor order. In this way, TRD provides a natural solution for the "curse of dimensionality" which means the data volume grows exponentially with the increase of the tensor order. TRD decomposes a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ by circular multilinear products to a sequence of 3rd-order core tensors (i.e., TR factors), $\{\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \cdots, \mathcal{G}^{(N)}\}$ where $\mathcal{G}^{(n)} \in \mathbb{R}^{R_{n-1} \times I_n \times R_n}$. $\{R_0, R_1, \dots, R_N\}$ denotes the TR-rank which is the most important parameter to determine the approximation performance and model complexity of TRD. Compared with tensor train decomposition, TRD applies trace computation to make all the TR factors be 3rd-order tensors equivalently. Every element of the tensor is given by the trace operation of the result of multiple matrix product, which can be demonstrated by the below equation:

$$\mathcal{X}(i_1, i_2, \dots, i_N) = \mathrm{Trace}\{\mathcal{G}^{(1)}(i_1)\mathcal{G}^{(2)}(i_2) \cdots \mathcal{G}^{(N)}(i_N)\} \tag{1}$$

where Trace is the matrix trace operator, $\mathcal{G}^{(n)}(i_n)$ is the ith mode-2 slice of $\mathcal{G}^{(n)} \in \mathbb{R}^{R_{n-1} \times I_n \times R_n}$. Moreover, TR is actually a linear combination of TT and thus it provides a more generalized and powerful representation than TTD. The overall relation between the decomposed tensor and its TRD is given by the below equations:

$$X_{<n>} = G_{(2)}^{(n)} (G_{<2>}^{(\neq n)})^T \tag{2}$$

where $G_{<2>}^{(\neq n)}$ is a sub-chain tensor which is calculated by merging all the TR factors but the nth core tensor [8].

3. PROPOSED ALGORITHM

3.1. Model formulation

The problem of finding the TRD of a tensor can be simply formulated as the following model:

$$\min_{[\mathcal{G}]} \| \mathcal{T} - \Psi([\mathcal{G}]) \|_F^2 \tag{3}$$

where $\mathcal{T}$ is the target tensor to be decomposed, $[\mathcal{G}]$ are the TR factors to be optimized, and $\Psi(\cdot)$ is the function which transforms the TR factors into the approximated tensor. In consideration of noise, the obtained tensor $\mathcal{T}$ can be formulated by:

$$\mathcal{T} = \mathcal{X} + \mathcal{N} \tag{4}$$

where $\mathcal{X}$ is the real tensor data, and $\mathcal{N}$ is assumed to be the white noise which is independent from the real tensor and follows the Gaussian distribution. If we assume the real tensor has the TR structure, we can formulate the denoising problem as:

$$\min_{\mathcal{X},[\mathcal{G}]} \parallel \mathcal{T} - \mathcal{X} \parallel_F^2 \text{ , s.t. } \mathcal{X} = \Psi([\mathcal{G}]) \tag{5}$$

Actually, solving the denoising model (5) is equal to first solving the decomposition model (3), and then, transforming the TR factors to the tensor space to get the denoised real tensor $\mathcal{X}$. In [8], the model (3) is solved by various algorithms like TRSVD, TRALS, TRSGD, etc. However, the SVD-based and ALS-based algorithms are calculated on the original size of the tensor, so they are of high computational cost. When facing large-scale tensors, tremendous computing resource is needed. In addition, though TRSGD has low computational complexity in every iteration and is able to process large-scale computation in low storage cost, the convergence speed of TRSGD is rather slow because of the non-convex optimization model. Next, we introduce the random tensor projection method which can calculate the tensor decomposition on projected latent space and greatly reduce the computational complexity.

3.2. Tensor random projection

Tensor random projection (TRP) is a natural extension of matrix random projection (MRP). It has been applied to tensor decomposition in the very recent years, and some studies have proposed TRP-based algorithms to calculate large-scale CPD and TKD [10,11]. Different from the MRP, the TRP method is to conduct random projection at every mode of the target tensor. In this way, a much smaller latent space tensor is obtained, and it can retain most of the features of the original large-scale tensor. The TRP can be demonstrated by:

$$\mathcal{X} \approx \mathcal{X} \times_1 \mathrm{Q}^{(1)}\mathrm{Q}^{(1),\mathrm{T}} \times_2 \cdots \times_1 \mathrm{Q}^{(\mathrm{N})}\mathrm{Q}^{(\mathrm{N}),\mathrm{T}} \approx \mathcal{P} \times_1 \mathrm{Q}^{(1)} \times_2 \cdots \times_\mathrm{N} \mathrm{Q}^{(\mathrm{N})} \tag{7}$$

where $[\mathrm{Q}]$ are the orthogonal matrices, and $\mathcal{P}$ is the projected small tensor of latent space. After projection, the obtained tensor $\mathcal{P}$ is used to calculate the low-rank approximation of the original large-scale tensor. The obtained tensor decomposition factors of the small latent space tensor are finally projected back to the decomposition factors of the large-scale tensor. In this way, we can avoid the direct computation on the large-scale tensor, so as to greatly reduce the computational complexity.

3.3. Randomized tensor ring decomposition

Finding tensor ring decomposition for large-scale data directly needs high computational cost. We first apply tensor random projection method which transforms the large-scale tensor into a relatively small tensor. Then we use TRSVD which is an efficient tensor ring decomposition algorithm to find the TRD of the small tensor. Finally, the TRD of the small tensor is projected back to the TRD of the large-scale tensor. For the step of tensor random projection, the most important step is to calculate the orthogonal matrices $\mathcal{P}$ which can determine how well the small projected tensor can retain the features of the large-scale tensor. Several methods for the projection step are proposed by researchers to improve the numerical stability and performance of the tensor random projection. For instance, [12] uses structured projection matrices to replace the Gaussian distributed matrices to project the tensor. Moreover, in order to achieve fast decay of the spectrum of the mode-n unfolding of the tensor, power iterations of LU decomposition are applied to calculate the projection matrices [13]. In our paper, we adopt both of the tricks to obtain the best TRP performance. By the combination of TRP and TRSVD, our randomized tensor ring decomposition (RTRD) is illustrated in the following algorithm.

Algorithm: Randomized tensor ring decomposition (RTRD)

Input: tensor $\mathcal{X}$, projection size for every mode $\{k_1, k_2, \dots, k_N\}$, TR-rank $\{R_0, R_1, \dots, R_N\}$, power iteration p.

Output: TR factors $[\mathcal{G}]$ which approximates $\mathcal{X}$.

$\mathcal{P} \leftarrow \mathcal{X}$

For n=1,…,N

Generate matrix $\mathrm{M} \in R^{numel(\mathcal{X})/I_n \times k_n}$ of which every element follows the below distribution:

$$m_{ij} \begin{cases} +\sqrt{3}, \textit{with probability } 1/6 \\ 0, \textit{with probability } 2/3 \\ -\sqrt{3}, \textit{with probability } 1/6 \end{cases}$$

$Y^{(n)} = P_{(n)}M$

For i=1,…,q

$[Q^{(n)}, \sim]$=lu($Y^{(n)}$)

[M,~]=lu($P_{(n)}^T Q^{(n)}$)

$Y^{(n)} = X_{(n)}M$

End
$[Q^{(n)}, \sim]$=qr($Y^{(n)}$)
$\mathcal{P} \leftarrow \mathcal{P} \times_n Q^{(n),T}$
End
Decompose the projected tensor $\mathcal{P}$ by TRSVD [8] to obtain its TR factors $[\mathcal{Z}]$.
For n=1,…,N
$\mathcal{G}^{(n)} = \mathcal{Z}^{(n)} \times_2 Q^{(n)}$
End

4. EXPERIMENT AND RESULT

In the experiment section, we firstly test the performance of our method in different size of TRP and compare our RTRD algorithm with the traditional counterpart (i.e., RTRD vs. TRSVD). Then, we conduct experiments on a hyperspectral image (HSI) which is corrupted by white noise by the related large-scale data processing methods. For the noising performance evaluation, we apply relative square error (RSE) and peak signal-to-noise ratio (PSNR) in our experiments. The RSE is calculated by

$$\text{RSE} = \| \mathcal{T}_{real} - \mathcal{Y} \|_F / \| \mathcal{T}_{real} \|_F,$$

where $\mathcal{T}_{real}$ is the real tensor without noise, $\mathcal{Y}$ is the denoised tensor which is approximated by the obtained decomposition factors, $numel(\cdot)$ calculates the number of elements of a tensor. The PSNR is calculated by

$$\text{PSNR} = 10\log\left(255^2/\text{MSE}\right),$$

where $\text{MSE} = \| \mathcal{T}_{real} - \mathcal{Y} \|_F^2 / numel(\mathcal{Y})$. In addition, we manually add noise on the clean tensor data to see the denoising performance of the algorithms. The noise data is calculated by

$$\mathcal{T}_{noise} = \mathcal{T}_{real} + \text{randn}(\text{size}(\mathcal{T}_{real}))\sqrt{10^{-dB/10} \| \mathcal{T}_{real} \|_F^2 / numel(\mathcal{T}_{real})},$$

where $\text{randn}(\text{size}(\mathcal{X}))$ generates a tensor, which is the same size as the target tensor and all the elements follows the normal distribution, and dB is the noise strength. All the computations in the experiments are conducted on a Mac PC computer with Intel Core i7 and 16GB DDR3 memory.

4.1. RGB image denoising

The projection size $\{k_1, k_2, \dots, k_N\}$ is the most important hyper-parameter of the TRP step of our algorithm because it determines the amounts of residual features of the original tensor to be retained and it balances the computational speed and the accuracy. This experiment is to explore the influence of the different size of the projected tensor to the performance and running time of our algorithm, and compare the performance with the TRSVD algorithm. We compare the traditional TRSVD and our RTRD to see the performance and speed differences w.r.t. different projection size and noise strength. We choose the RGB image of size 5690×4234×3 as the target tensor and add noise of 0 dB and 10 dB respectively. The RGB image is a typical 3rd-order tensor of large-scale and the image modes are considered to have strong low-rankness, so the projection of mode-1 and mode-2 can largely reduce the computational cost. The projection size of mode-1 and mode-2 of the tensor data are chosen from {10, 50, 100}. The mode-3 of the tensor is small so it remains as 3. As for parameter settings, because only one iteration is needed and the TR-rank is automatically chosen in our algorithm, we set the tolerance of TRSVD as 0.01. Figure 1 shows the approximation error (i.e., RSE and PSNR) and computational time of the compared algorithms. From the experiment we can see, our method runs much faster than TRSVD and the performance is always higher than TRSVD. When the noise is 0 dB, be best denoising performance is obtained when the projection size is {50, 50, 3}, and the performance of the randomized algorithms remain steady and when the projection size is {10, 10, 3}. Moreover, when the noise is 10 dB, the best performance is obtained when the projection size is {100, 100, 3}, and the denoising performance falls when the projection size decreases. The results indicate that the running time increases when the projection size increases, and the image with less noise requires higher projection size to reach a better performance.

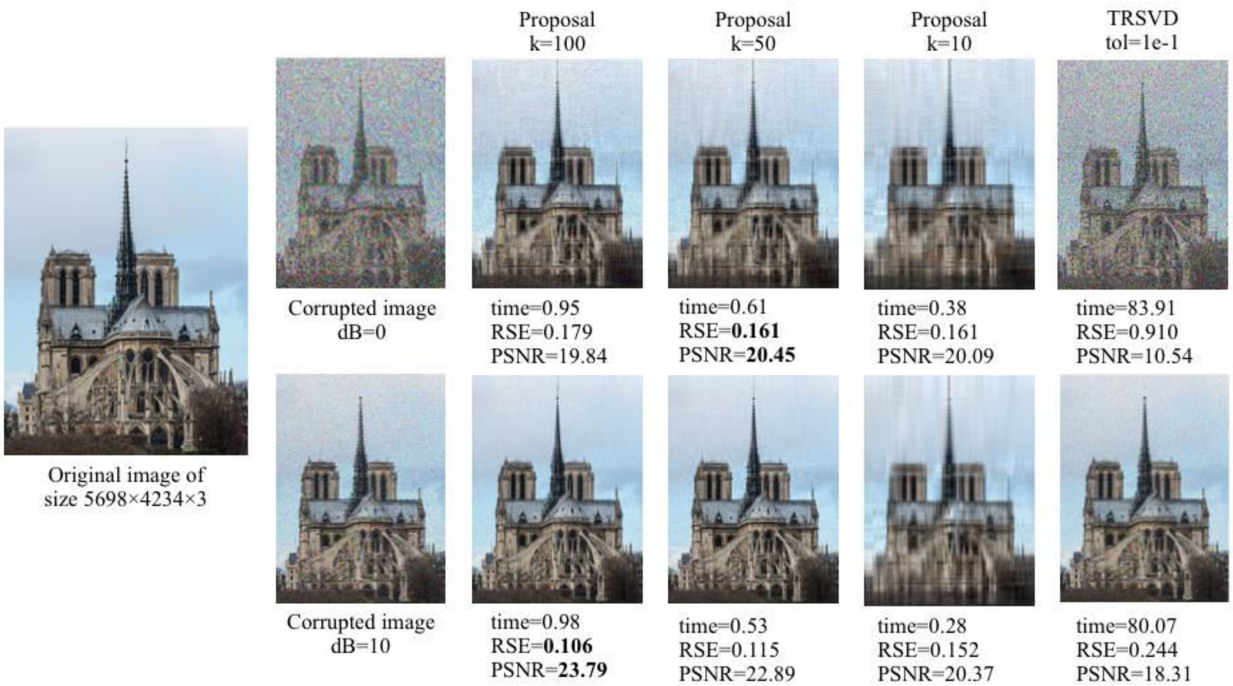

Figure 1 Denoising results of a large-scale image with different projection size and noise strength.

4.2. Hyperspectral image denoising

Hyperspectral image (HSI) is a kind of large-scale 3rd-order tensor of height×width×bands. The band-mode of HSI is usually considered to own strong property low-rankness, so the projection of the band-mode will largely reduce the computational cost and running time. In this experiment, we employ the most related large-scale tensor decomposition and matrix completion algorithms to compare with our RTRD. We apply TRSGD [8], RCPALS [10], RTucker [11] in this experiment. The RSVD [13] which is based on matrix random projection and often used in HSI reconstruction and denoising is also employed. The RSVD is implemented by mode-3 unfolding of the HSI data. The projection size of all the algorithms are set as {100, 100, 3} for the HIS image of size 200×200×80, and the other parameters for each algorithm are tuned to get the best performance. Figure Table 1 and Figure 2 show the numerical results and visual results of all the compared algorithms respectively. RTRD outperforms the compared algorithms in both performance and running time.

Table 1. Numerical results of the HSI denoising experiment.

Noise		RTRD	TRSGD	RCPALS	RTucker	RSVD
-	RSE	**0.00934**	0.249	0.100	0.0110	0.0303
	time	**0.45**	9.45	5.38	0.50	1.84
20 dB	RSE	**0.0331**	0.253	0.101	0.0388	0.0594
	time	**0.20**	206.82	3.97	0.54	2.33
10 dB	RSE	**0.0895**	0.293	0.107	0.114	0.156
	time	**0.29**	210.89	3.91	0.46	2.08
0 dB	RSE	**0.138**	0.437	0.166	0.367	0.431
	time	**0.38**	206.62	3.95	0.44	1.87

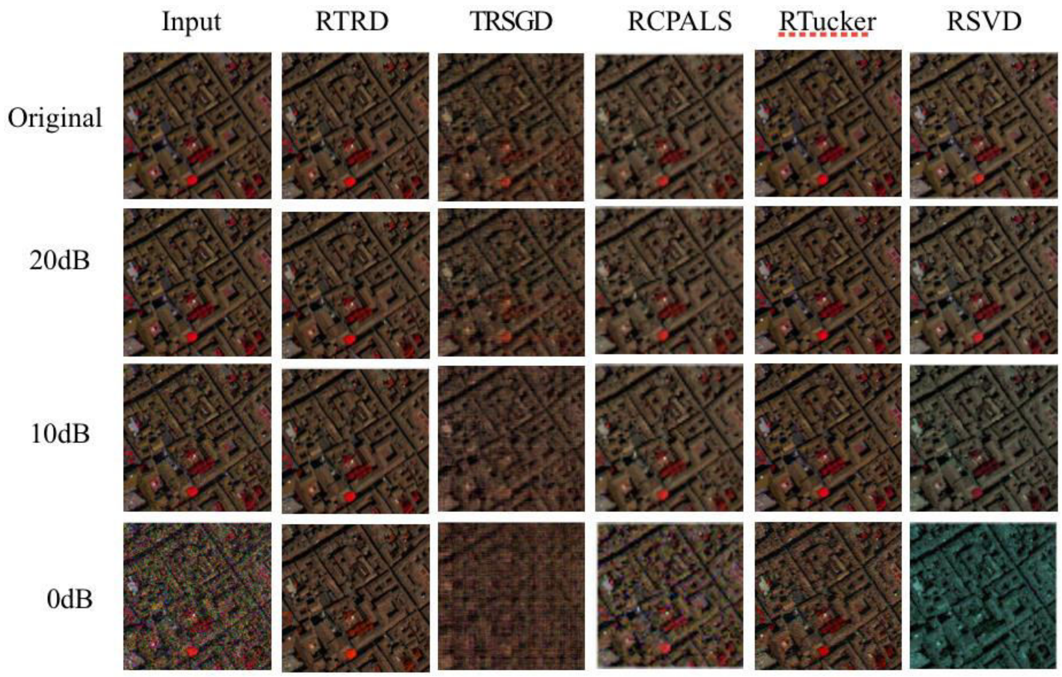

Figure 2. Visual results of the HSI denoising experiment.

5. CONCLUSION

In this paper, based on tensor random projection method, we proposed RTRD algorithm for fast and reliable TR decomposition. Without losing accuracy, our algorithm performs much faster than their traditional counterparts and outperforms the compared randomized algorithms in visual data denoising experiments. Randomized method is a promising aspect for large-scale data processing. In our future work, we will focus on further improving the performance of decomposition and applying randomized algorithms to the sparse and incomplete tensors.

6. ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI (Grant No. 17K00326, 15H04002, 18K04178), JST CREST (Grant No. JPMJCR1784), and the National Natural Science Foundation of China (Grant No. 61773129).

7. REFERENCES

[1] A. Novikov, D. Podoprikhin, A. Osokin, and D. P. Vetrov, "Tensorizing neural networks," Advances in neural information processing systems, pp. 442-450, 2015.
[2] A. Cichocki, "Era of big data processing: A new approach via tensor networks and tensor decompositions," arXiv preprint arXiv:1403.2048,2014.
[3] L. Yuan, C. Li, D. Mandic, J. Cao, and Q. Zhao, "Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion," arXiv preprint arXiv:1809.02288, 2018.
[4] W. He, H. Zhang, L. Zhang, and H. Shen, "Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration,"IEEE transactions on geoscience and remote sensing, vol. 54(1), pp. 178-188, 2016.
[5] N. K. M Faber, R. Bro, and P. K. Hopke, "Recent developments in CANDECOMP/PARAFAC algorithms: A critical review,"Chemometrics and Intelligent Laboratory Systems, vol. 65(1), pp. 119-137, 2003.
[6] L. R.Tucker, "Some mathematical notes on three-mode factor analysis," Psychometrika, vol. 31(3), pp. 279-311, 1966.
[7] I. V. Oseledets, "Tensor-train decomposition," SIAM Journal on Scientific Computing, vol. 33(5), pp. 2295-2317, 2011.
[8] Q. Zhao, M. Sugiyama, L. Yuan and A. Cichocki, "Learning efficient tensor representations with ring-structured networks,"2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8608-8612,2019.
[9] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," SIAM review, vol. 51(3), pp. 455-500, 2009.
[10] N. B. Erichson, K. Manohar, S. L. Brunton, and J. N. Kutz, "Randomized CP tensor decomposition," arXiv preprint arXiv:1703.09074, 2017.
[11] G.Zhou, A. Cichocki and S. Xie, "Decomposition of big tensors with low multilinear rank," arXiv preprint arXiv:1412.1885, 2014.
[12] D. Achlioptas, "Database-friendly random projections," In Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 274-281, 2001.
[13] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," SIAM review, vol. 53(2), pp. 217-288, 2011.