Compression of High-Dimensional Multispectral Image Time Series Using Tensor Decomposition Learning

Anastasia Aidini, Grigorios Tsagkatakis, Panagiotis Tsakalides
Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH)
Computer Science Department, University of Crete
Heraklion, Crete, Greece
{aidini, greg, tsakalid}@ics.forth.gr

***Abstract*—Multispectral imaging is widely used in many fields, such as in medicine and earth observation, as it provides valuable spatial, spectral and temporal information about the scene. It is of paramount importance that the large amount of images collected over time, and organized in multidimensional arrays known as tensors, be efficiently compressed in order to be stored or transmitted. In this paper, we present a compression algorithm which involves a training process and employs a symbol encoding dictionary. During training, we derive specially structured tensors from a given image time sequence using the CANDECOMP/PARAFAC (CP) decomposition. During run-time, every new image time sequence is quantized and encoded into a vector of coefficients corresponding to the learned CP decomposition. Experimental results on sequences of real satellite images demonstrate that we can efficiently handle higher-order tensors and obtain the decompressed data by composing the learned tensors by means of the received vector of coefficients, thus achieving a high compression ratio.**

***Index Terms*—Compression, multispectral image time series, high-order tensors, CP decomposition, learning**

I. Introduction

The acquisition of a large number of image sequences characterizes many applications, such as functional brain imaging studies [1], live cell microscopy [2], and satellite-based Earth Observation [3]. Especially in satellite-based Earth observation, a given scene can be observed repeatedly, providing valuable spatial, spectral and temporal information, like growth, maturation, or harvest of crops, which leads to the generation of huge quantities of observations [4]. The copious amounts of images collected from the sensors over time introduce considerable challenges in terms of data storage and data transfer. Especially in remote sensing cases where images are collected on board satellites or unmanned aerial vehicles and need to be transferred to the ground-based stations, an efficient compression algorithm is mandatory in order to reduce bandwidth and increase the system lifetime.

Observations acquired over multiple time instances and spectral bands can be formulated by means of high dimensional data structures. For instance, a time series of grayscale images is a 3D object defined by one index for the temporal dimension and two indices for the spatial variables, while a time series of spectral images is a 4D object with an additional variable for the wavelength. Representing such high dimensional observations in a mathematically consistent manner is accomplished using tensors, higher-order extensions of vectors and matrices [5], [6].

In this paper, we propose a compression algorithm that achieves a considerable compression ratio of high dimensional data by appropriately exploiting their structure through tensor decomposition. The proposed method involves a training process, in which we derive specially structured tensors from a given image time series using CANDECOMP/PARAFAC (CP) decomposition [7], and a symbol encoding dictionary. During run-time, tensors are mapped to learned decompositions and the associated coefficients are quantized and encoded, as it is illustrated in Figure 1.

In short, the key novelties of this work are the following.

- An end-to-end compression algorithm is proposed that includes quantization and encoding of the information data, making it directly applicable in real-world scenarios.
- A machine learning based algorithm is developed for the compression of multidimensional data, by learning from a training sample specially structured tensors that are used during run-time, without requiring the transmission of all the information but only a vector of coefficients.
- The performance of the proposed method is investigated on 3D real data, as well as on 4D data since our method can seamlessly handle high dimensional observations.

II. Related Work

To compress image time series, both spatial and temporal correlations must be simultaneously removed, so typical compression algorithms apply a 3D wavelet transform on the full data cube [8], [9], or a combination of a 1D spectral decorrelator, such as the Karhunen-Loeve Transform (KLT) [10] or Principal Components Analysis (PCA) [11]. The above methods usually are followed by JPEG2000 [12], which is among the best-performing algorithms for 2D still image compression. Although these methods offer significant gains

The research work was supported by the Hellenic Foundation for Research and Innovation (HFRI) and the General Secretariat for Research and Technology (GSRT), under the HFRI PhD Fellowship grant (GA. no. 4824).

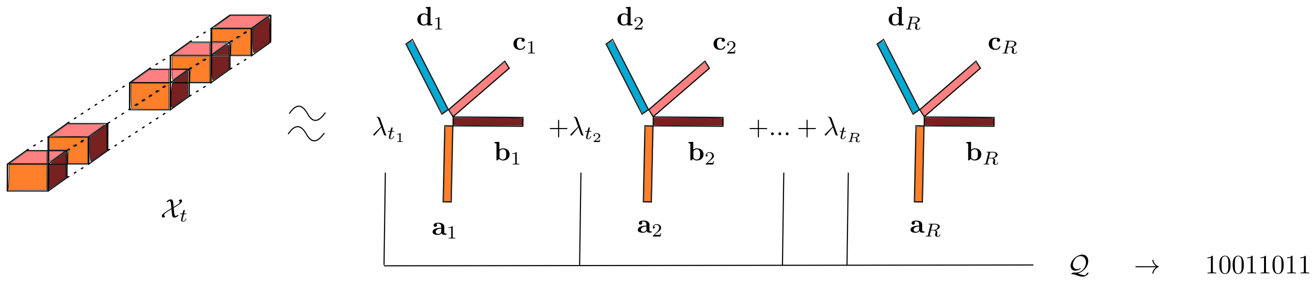

Fig. 1. The compression process of the proposed method, which consists of the expression of a multispectral image time series $\mathcal{X}_t$ as a linear combination of the learned tensors, and the quantization of the coefficients $\boldsymbol{\lambda}_t$ to b bits, using the symbols of the learned encoding dictionary.

in terms of compression ratio, increasing data-rates of the new-generation sensors require even higher compression.

Analysis of high dimensional data using tensor models is an active research topic in signal processing domains including cyber-physical systems [13], wearable Internet-of-Things platforms [14] and remote sensing [15] among others. In [16] a compression-based nonnegative CP decomposition algorithm was proposed for compression of hyperspectral image time series where the input data array was represented by one or a few arrays with reduced dimensions, computing their CP approximation with compressed versions of the original factor matrices. However, in their work, the spatial variables are represented by a single index of the tensor and as such, they cannot sufficiently capture the spatial characteristics of the data and they cannot handle higher-order structures.

Other tensor-based approaches for compression of hyperspectral images are presented in [17], [18]. Specifically, a compression algorithm based on the CP decomposition is proposed in [17], in which a data cube can be compressed into R rank-1 tensors. In this model, the parameter R indicates the compression ratio. Similarly, a compression algorithm based on the Tucker decomposition is proposed in [18], in which the original tensor data is approximately decomposed into a core tensor multiplied by a factor matrix along each mode. Thus, a data cube in this model is compressed to the core tensor and the factor matrices with compressed dimensionality in each mode. However, the above models that have been applied only on 3D hyperspectral images, require either the transmission of all rank-1 tensors with their coefficients, or the core tensor and the factor matrices, which is much more information than the one required by our model. Specifically, in our method, we learn the rank-1 tensors and we only need to transmit the coefficients, achieving an extremely high compression ratio with a good reconstruction quality.

III. Tensor based observation modeling

An N-way or Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as a multidimensional array, whereby the order of a tensor is the number of its dimensions. A fundamental property of tensors is the tensor rank, a generalization of matrix rank. Specifically, considering that the outer product of two vectors is a rank-1 matrix, we can define the matrix rank as the minimum number of rank-1 matrices needed to synthesize a given matrix. In a similar way, a rank-1 N-th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ is the outer product of N vectors $\mathbf{a}^{(n)} \in \mathbb{R}^{I_n}$, $n = 1, ..., N$, with elements the product of the corresponding vector elements, i.e.,

$$\mathcal{X} = \mathbf{a}^{(1)} \circ \cdots \circ \mathbf{a}^{(N)}, \text{ with } x_{i_1 i_2 \cdots i_N} = \mathbf{a}^{(1)}_{i_1} \cdots \mathbf{a}^{(N)}_{i_N}, \quad (1)$$

where $i_n \in \{1, ..., I_n\}$. For instance, a three-way rank-1 tensor is the outer product of three vectors. Therefore, the rank of a tensor $\mathcal{X}$ is defined as the minimum number of rank-1 tensors needed to produce the original tensor.

A common framework for tensor computations is to turn the tensor into a matrix, which is called matricization or unfolding of the tensor [19], where for a given tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_n \times \ldots \times I_N}$, its mode-$n$ matricization is denoted as $\text{unfold}_n(\mathcal{X}) = X_{(n)} \in \mathbb{R}^{I_n \times \prod_{i \neq n} I_i}$ and corresponds to a matrix with columns being the vectors obtained by fixing all indices of $\mathcal{X}$ except the n-th index. This procedure allows the use of matrix factorization algorithms, like the SVD. However, in doing so, the structure of the data is not preserved and therefore the high dimensional relationships, e.g., across neighbouring pixels or time instances, are lost.

A more appropriate approach involves expressing a tensor of arbitrary rank as a linear combination of rank-1 tensors. The CANDECOMP/PARAFAC (CP) decomposition represents an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ as

$$\mathcal{X} = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \cdots \mathbf{a}_r^{(N)} = [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, ..., \mathbf{A}^{(N)}]\!], \quad (2)$$

where R is a positive integer, $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)} \ \mathbf{a}_2^{(n)} \ \cdots \ \mathbf{a}_R^{(n)}]$ are the factor matrices, $\boldsymbol{\lambda} \in \mathbb{R}^R$ and $\mathbf{a}_r^{(n)} \in \mathbb{R}^{I_n}$, for $r = 1, ..., R$ and $n = 1, ..., N$. The matrix form of CP decomposition can be obtained via the Khatri-Rao products as:

$$\mathbf{X}_{(n)} = \mathbf{A}^{(n)} \mathbf{G} (\mathbf{A}^{(N)} \odot \cdots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \cdots \odot \mathbf{A}^{(1)})^T,$$

where $\mathbf{G} = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_R)$ [5], [20]. Compared to matrix decompositions, CP decomposition is unique under more relaxed conditions, that only require the components to be sufficiently different and their number reasonably large [21].

The minimum number R of components in an exact CP decomposition of a given tensor, is the rank of the tensor. Unfortunately, there is no straightforward algorithm to determine the rank of a given tensor; in fact, the problem is NP-hard

[22]. However, in most applications, one is really interested in fitting a model that has the essential or meaningful number of components that we usually call the rank, and it is much less than the actual rank of the tensor that we observe, due to noise and sensor imperfections [5].

In order to compute approximate low-rank models of tensor data, we minimize the Frobenius norm (i.e. the square root of the sum of the squares of all elements) of the difference between the given data tensor and its CP approximation. Since the computation of CP decomposition is intrinsically multilinear, we can arrive at the solution through a sequence of linear sub-problems as in the Alternating Least Squares (ALS) framework [5], whereby the least squares cost function is optimized for one component matrix at a time, while keeping the other component matrices fixed. Specifically, for the case of 4^{rd} order tensors, in order to estimate the factor matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ where $\mathbf{A} = \mathbf{A}^{(1)}, \mathbf{B} = \mathbf{A}^{(2)}, \mathbf{C} = \mathbf{A}^{(3)}, \mathbf{D} = \mathbf{A}^{(4)}$, from possibly noisy data $\mathcal{X} \in \mathbb{R}^{I \times J \times K \times T}$, we can adopt a least squares criterion and then the problem becomes

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}} \|\mathcal{X} - \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \circ \mathbf{d}_r\|_F^2. \tag{3}$$

The above model is non-convex in $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$. By fixing $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$, it becomes (conditionally) linear in $\mathbf{A}$. Similarly, for the other factor matrices, using the matrix representations of the tensor, we can update

$$\begin{aligned} \mathbf{A} &\leftarrow \arg\min_{\mathbf{A}} \|\mathbf{X}_{(1)} - \mathbf{A}((\mathbf{D} \odot \mathbf{C} \odot \mathbf{B})^T\|_F^2, \\ \mathbf{B} &\leftarrow \arg\min_{\mathbf{B}} \|\mathbf{X}_{(2)} - \mathbf{B}((\mathbf{D} \odot \mathbf{C} \odot \mathbf{A})^T\|_F^2, \\ \mathbf{C} &\leftarrow \arg\min_{\mathbf{C}} \|\mathbf{X}_{(3)} - \mathbf{C}((\mathbf{D} \odot \mathbf{B} \odot \mathbf{A})^T\|_F^2, \\ \mathbf{D} &\leftarrow \arg\min_{\mathbf{D}} \|\mathbf{X}_{(4)} - \mathbf{D}((\mathbf{C} \odot \mathbf{B} \odot \mathbf{A})^T\|_F^2, \end{aligned} \tag{4}$$

until little, or no change in the factor matrices is observed. Then, by normalizing the columns of $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$, we obtain the weights $\boldsymbol{\lambda}$.

IV. Proposed Compression Method

A. Training process

The compression method presented in this paper requires a training process with a given image time series, before the compression of new data that need to be transmitted. Specifically, we assume that we have a time series of multispectral images $\mathcal{M} \in \mathbb{R}^{I \times J \times K \times T}$ as training samples, where I and J represent the spatial dimensions, K represents the number of spectral bands and T the number of temporal instances, such as days. We obtain the R rank-1 tensors $\mathcal{M}_r = \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \circ \mathbf{d}_r$, $r = 1, ..., R$, that synthesize the training sample $\mathcal{M}$, by minimizing

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C},\mathbf{D}} \|\mathcal{M} - \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \circ \mathbf{d}_r\|_F^2 \tag{5}$$

using the ALS method that was described above, and obtain the factor matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ which are used in the subsequent compression of the acquired observations.

A crucial step for data transmission or storage is the need for representation of the compressed data in a binary format, which involves a coding algorithm. For a given number of bits b, the coding algorithm maps each binary number to a discrete set of 2^b symbols, $S = \{q_1, q_2, ..., q_{2^b}\}$. To populate the dictionary of symbols, we split the range of values of the coefficients $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_R]$ into $2^b - 1$ equal partitions and we define as the set of symbols S of the dictionary to be the boundaries of those partitions, $q_s, \ s = 1, ..., 2^b$, where q_1 and q_{2^b} are the minimum and the maximum values of $\boldsymbol{\lambda}$, respectively. The correspondence between the symbols and their binary representation is the dictionary of the coding algorithm, which is also estimated during the training process.

B. Run-time Compression and Decompression

Assuming that $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_t, ... \in \mathbb{R}^{I \times J \times K \times T}$ are the test samples, each of them being a multispectral image time series contained by T time intervals, we express each test sample $\mathcal{X}_t$ as a linear combination of the learned rank-1 tensors $\mathcal{M}_r = \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \circ \mathbf{d}_r$, $r = 1, ..., R$, i.e.,

$$\mathcal{X}_t = \sum_{r=1}^{R} \lambda_{t_r} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \circ \mathbf{d}_r, \tag{6}$$

where the coefficients $\boldsymbol{\lambda}_t$ can be obtained through a least squares estimation. Then, we quantize the vector of coefficients $\boldsymbol{\lambda}_t = (\lambda_{t_1}, .., \lambda_{t_R})$ to b bits, as we can see in Figure 1. Specifically, we use a uniform quantizer $\mathcal{Q} : \mathbb{R} \to S$, assuming that the set of symbols $S = \{q_1, q_2, ..., q_{2^b}\}$ are the quantization levels and $c_s = \frac{q_s + q_{s+1}}{2}, \ s = 1, ..., 2^b - 1$ are the quantization boundaries. So, we obtain the quantized vector of coefficients

$$\boldsymbol{\lambda}_{\boldsymbol{q}t} = [\mathcal{Q}(\lambda_{t_1}), \mathcal{Q}(\lambda_{t_2}), ..., \mathcal{Q}(\lambda_{t_R})] = [\lambda_{q_{t_1}}, \lambda_{q_{t_2}}, ..., \lambda_{q_{t_R}}],$$

which needs to be encoded, in order to be transmitted. Therefore, we transmit only R numbers, instead of the whole image time series $\mathcal{X}_t$, achieving extremely high compression ratios. As the final compression stage, we employ Huffman coding on the quantized vector $\boldsymbol{\lambda}_{\boldsymbol{q}t}$, using the dictionary obtained from the training process.

In order to decompress the received data $\boldsymbol{\lambda}_{\boldsymbol{q}t}$, we employ the rank-1 tensors $\mathcal{M}_r = \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \circ \mathbf{d}_r$, $r = 1, ..., R$, obtained from the CP decomposition of the training sample $\mathcal{M}$. Therefore, the reconstructed image time series $\mathcal{X}_t^{'}$ will be

$$\mathcal{X}_t^{'} = \sum_{r=1}^{R} \lambda_{q_{t_r}} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \circ \mathbf{d}_r, \tag{7}$$

which is an efficient approximation of the test sample $\mathcal{X}_t$, as will be demonstrated in the experimental results Section. The same algorithm can be applied for the case of third or any higher-order tensors by utilizing the appropriate number of factor matrices.

V. Experimental results

The efficacy of the proposed compression algorithm is evaluated over a publicly available dataset of satellite image time series acquired by the MODIS satellite, over the region of Cyprus [23], for 160 days, providing information about the vegetation of the area. The dataset consists of 4D sequences of color images of size $136 \times 278 \times 3 \times 160$, and 3D sequences of grayscale images of size $136 \times 278 \times 160$, with the last dimension being the number of days. In our experiments, we train our model with an image time series and we test it by examining sliding temporal windows, such that each test sequence consists of the same number of days as in the training sequence. For instance, if the window size is 30, then the first 30 days of the dataset are used for the training process and the remaining 130 days for testing, by sliding a temporal window of size 30 days.

The recovery performance is measured in terms of the Normalized Mean Square Error (NMSE) which is defined as

$$\text{NMSE} = \frac{\|\mathcal{X}_t - \mathcal{X}_t'\|_2^2}{\|\mathcal{X}_t\|_2^2},$$

where $\mathcal{X}_t$ and $\mathcal{X}_t'$ are the original and the reconstructed signal, respectively.

An important parameter of the proposed method is the tensor rank, i.e., the number of rank-1 components identified by the CP decomposition during training. Figure 2 reports on the reconstruction quality for each time series as a function of values of rank on third and fourth order tensor data. The results indicate that using larger rank leads to better performance. In addition, we observe that the performance of our algorithm is getting worse over time. The same behavior is observed when the order of the tensor is increasing.

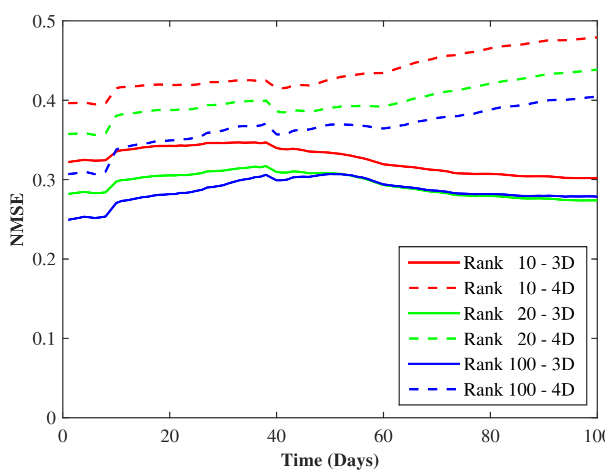

Fig. 2. Reconstruction quality for different values of rank, on third and fourth order tensor data.

Another parameter that was examined is the effect of the temporal window size, i.e., the number of days used to produce each time series. Figure 3 presents the NMSE for different window sizes, on third and fourth order tensor data, using 8 bits and rank 100. Notably, longer window sizes offer higher quality reconstruction and more stable performance compared to smaller ones.

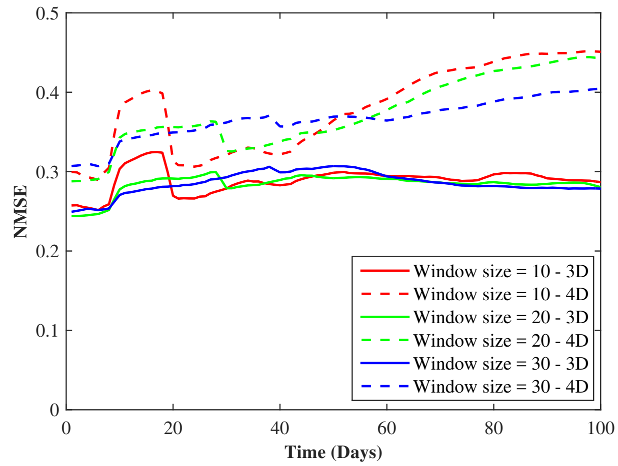

Fig. 3. Reconstruction quality for different temporal window sizes, on third and fourth order tensor data.

Our study also emphasizes on the impact of the number of bits used for the quantization of the coefficients vector. We performed experiments on fourth order tensor data, using a temporal window of size 30 days, and rank 100. The results reported in Table I indicate that as the number of bits increases, the reconstruction error decreases. However, the improvement is negligible, demonstrating a robust behavior.

TABLE I
Reconstruction quality for several days as a function of the number of bits, on fourth order tensor data.

Number of Bits	NMSE			
	1st Day	20th Day	50th Day	78th Day
4	0.2518	0.2937	0.3528	0.3712
6	0.2493	0.2916	0.3498	0.3681
8	0.2490	0.2913	0.3496	0.3680

To examine the efficacy of our method on different compression rates, we performed experiments on third order tensor data, with the results presented in Table II. As it was expected, the more the bits per pixel per band (bpppb), the better the performance of our algorithm. However, there is a minor improvement of the reconstruction since the compression ratio is extremely high in each case.

TABLE II
Reconstruction quality for different number of bpppb, on third order tensor data.

	bpppb			
	0.0003	0.0007	0.0011	0.0014
NMSE	0.3948	0.2751	0.2658	0.2649

Finally, we compared our algorithm with state-of-the-art compression algorithms, namely pure JPEG2000, and Discrete Wavelet Transform (DWT) and PCA spectral decorrelation followed by JPEG2000. Figure 4 presents the NMSE for several days under investigation, on third and fourth order tensor data, with window size 30 days and rank 100, using 7.0532×10^{-4} and 2.3511×10^{-4} bpppb. Note that for the case of fourth order tensor data, the results from competing methods correspond to the application of compression on a single color image at a time, as it has been designed only for 3D data. Compared to the other methods, our technique can handle better higher-order tensor data as it provides a better approximation of the original data.

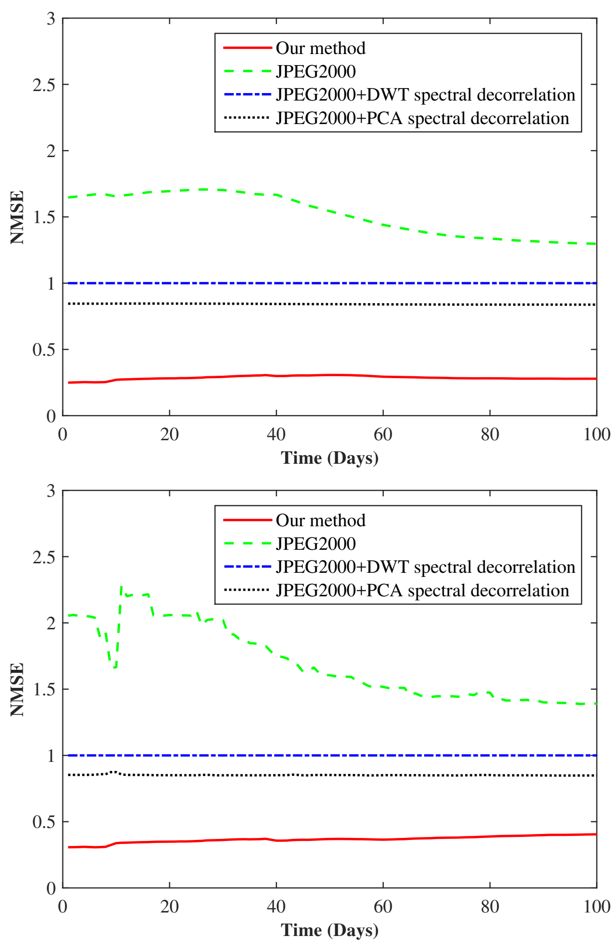

Fig. 4. Reconstruction quality for different compression methods, on third (top) and fourth order (bottom) tensor data.

VI. Conclusion

In this work, we presented a new approach for the compression of high dimensional observations, focusing on the case of color image time series. Our approach achieves extremely high compression ratios by exploiting the structure of multidimensional data through tensor decomposition learning. A major benefit of the proposed scheme is that it can directly be extended to *arbitrary high dimensions*, providing a mathematically concrete solution for simultaneously encoding multiple sources of observations. An extension of the proposed algorithm involves the incremental update of the learned rank-1 tensors that are used in the compression of observations.

References

[1] T Siva Tian, "Functional data analysis in brain imaging studies," *Frontiers in psychology*, vol. 1, pp. 35, 2010.

[2] Simon Gordonov, Mun Kyung Hwang, Alan Wells, Frank B Gertler, Douglas A Lauffenburger, and Mark Bathe, "Time series modeling of live-cell shape dynamics for image-based phenotypic profiling," *Integrative Biology*, vol. 8, no. 1, pp. 73–90, 2016.

[3] Thomas Guyet and Hervé Nicolas, "Long term analysis of time series of satellite images," *Pattern Recognition Letters*, vol. 70, pp. 17–23, 2016.

[4] David M Tralli, Ronald G Blom, Victor Zlotnicki, Andrea Donnellan, and Diane L Evans, "Satellite remote sensing of earthquake, volcano, flood, landslide and coastal inundation hazards," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 59, no. 4, pp. 185–198, 2005.

[5] Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, and Christos Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[6] Tamara G Kolda and Brett W Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[7] J Douglas Carroll and Jih-Jie Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[8] Michael Weeks and Magdy A Bayoumi, "Three-dimensional discrete wavelet transform architectures," *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 2050–2063, 2002.

[9] Amar Aggoun, "Compression of 3d integral images using 3d wavelet transform," *Journal of Display Technology*, vol. 7, no. 11, pp. 586–592, 2011.

[10] Lei Liu, Jingwen Yan, Xianwei Zheng, Hong Peng, Di Guo, and Xiaobo Qu, "Karhunen-loève transform for compressive sampling hyperspectral images," *Optical Engineering*, vol. 54, no. 1, 2015.

[11] Qian Du and James E Fowler, "Hyperspectral image compression using jpeg2000 and principal component analysis," *IEEE Geoscience and Remote sensing letters*, vol. 4, no. 2, pp. 201–205, 2007.

[12] Tzong-Jer Chen, Sheng-Chieh Lin, You-Chen Lin, Ren-Gui Cheng, Li-Hui Lin, and Wei Wu, "Jpeg2000 still image coding quality," *Journal of digital imaging*, vol. 26, no. 5, pp. 866–874, 2013.

[13] Grigorios Tsagkatakis, Konstantina Fotiadou, Michalis Giannopoulos, Anastasia Aidini, Athanasia Panousopoulou, and Panagiotis Tsakalides, "Matrix and tensor signal modelling in cyber physical systems," *Smart Water Grids: A Cyber-Physical Systems Approach*, p. 107, 2018.

[14] Sofia Savvaki, Grigorios Tsagkatakis, Athanasia Panousopoulou, and Panagiotis Tsakalides, "Matrix and tensor completion on a human activity recognition framework," *IEEE journal of biomedical and health informatics*, vol. 21, no. 6, pp. 1554–1561, 2017.

[15] Bo Du, Mengfei Zhang, Lefei Zhang, Ruimin Hu, and Dacheng Tao, "Pltd: Patch-based low-rank tensor decomposition for hyperspectral images," *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 67–79, 2017.

[16] Miguel Angel Veganzones, Jérémy E Cohen, Rodrigo Cabral Farias, Jocelyn Chanussot, and Pierre Comon, "Compression-based nonnegative tensor cp decomposition of hyperspectral big data," .

[17] Leyuan Fang, Nanjun He, and Hui Lin, "Cp tensor-based compression of hyperspectral images," *JOSA A*, vol. 34, no. 2, pp. 252–258, 2017.

[18] Lefei Zhang, Liangpei Zhang, Dacheng Tao, Xin Huang, and Bo Du, "Compression of hyperspectral remote sensing images by tensor approach," *Neurocomputing*, vol. 147, pp. 358–363, 2015.

[19] Tamara Gibson Kolda, *Multilinear operators for higher-order decompositions*, vol. 2, United States. Department of Energy, 2006.

[20] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.

[21] Nicholas D Sidiropoulos and Rasmus Bro, "On the uniqueness of multilinear decomposition of n-way arrays," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 14, no. 3, pp. 229–239, 2000.

[22] Johan Håstad, "Tensor rank is np-complete," *Journal of Algorithms*, vol. 11, no. 4, pp. 644–654, 1990.

[23] NASA (National Aeronautics and Space Administration), "Worldview," .