

Multiscale Tensor Decomposition

Alp Ozdemir¹, Mark A. Iwen^{1,2} and Selin Aviyente¹

¹Department of Electrical and Computer Engineering, Michigan State University

²Department of the Mathematics, Michigan State University

Abstract—Large datasets usually contain redundant information and summarizing these datasets is important for better data interpretation. Higher-order data reduction is usually achieved through low-rank tensor approximation which assumes that the data lies near a linear subspace across each mode. However, nonlinearities in the data cannot be captured well by linear methods. In this paper, we propose a multiscale tensor decomposition to better approximate local nonlinearities in tensors. The proposed multiscale approach constructs hierarchical low-rank structure by dividing the tensor into subtensors sequentially and fitting a low-rank model to each subtensor. The proposed method is evaluated on compressing 3 and 4-mode tensors containing video and functional connectivity brain networks, respectively.

I. INTRODUCTION

There is a growing interest to collect and store a variety of large originating from multiple sensors such as hyperspectral images, high resolution videos and medical images, for a variety of applications [1]–[3]. These large datasets, when vectorized into elements of \mathbb{R}^D , are often assumed to be lying near a lower d -dimensional manifold, or subspace, with $d \ll D$. Many techniques including Principal Component Analysis (PCA), Independent Component Analysis (ICA), Morphological Component Analysis (MCA) have been proposed to discover these intrinsic lower dimensional subspaces from high dimensional data [4]. However, these lower dimensional bases obtained by linear methods may not efficiently encode the data points belonging to nonlinear manifolds [5], [6]. To address this issue, various approaches have been developed to discover such embedded low dimensional manifolds [5]–[8].

In order to better deal with the curse of dimensionality, both linear and nonlinear subspace learning methods have been extended to higher order data reduction. In early work in the area, Vasilescu and Terzopoulos [9] extended the eigenface concept to the tensorface by using higher order SVD and taking different modes such as expression, illumination and pose into account. Similarly, 2D-PCA for matrices has been used for feature extraction from face images without converting the images into vectors [10]. More recently, Higher Order SVD (HoSVD) and Parallel Factor Analysis (PARAFAC) have been proposed as direct extensions of SVD and PCA to higher order tensors [11]–[14]. De Lathauwer [15] proposed block-term decomposition which unifies PARAFAC and HoSVD where PARAFAC decomposes the tensor into scalar blocks and HoSVD fits a low n -rank representation to each block. Alternatively, Hierarchical Tucker Decomposition (HT) [16] and Tensor-Train Decomposition (TT) [17] have been

developed to compress large tensor data into smaller core tensors through matrix product forms. However, most of these subspace learning algorithms are interested in fitting a low-rank model to data which lies near a *linear* subspace and are similar to PCA and SVD with the goal of finding the best linear low-rank approximation. The major shortcoming of these methods is that they are limited to discovering linear structure and cannot capture nonlinearities.

More recently, many of the manifold learning approaches have been extended from the vector case to tensors. He et al. [18] extended locality preserving projections [19] to second order tensors for face recognition. Moreover, Dai and Yeung [20] presented generalized tensor embedding methods such as the extensions of local discriminant embedding [21], neighborhood preserving embedding [22], and locality preserving projection [19] to tensors. Li et al. [23] proposed a supervised manifold learning method for vector type data which preserves local structures in each class of samples, and then extended the algorithm to tensors to provide improved performance for face and gait recognition. Similar to vector-type manifold learning algorithms, the aim of these methods is finding an optimal *linear* transformation for the tensor-type data samples without vectorizing them and mapping these samples to a low dimensional subspace while preserving the neighbourhood information. In contrast, herein, we propose novel unsupervised higher-order manifold learning approaches for summarizing higher-order data by taking advantage of multiscale structure to better deal with intrinsic nonlinearities.

In this paper, we propose a novel multiresolution analysis technique to efficiently encode nonlinearities in tensor type data. The proposed method constructs data-dependent multiscale dictionaries to better represent the data. The proposed algorithm consists of two main steps: 1) Constructing a tree structure by decomposing the tensor into a collection of permuted subtensors, and 2) Constructing multiscale dictionaries by applying HoSVD to each subtensor. Finally, we apply the proposed algorithm to real datasets to illustrate the improvement in the compression performance compared to HoSVD.

II. BACKGROUND

A. Tensor Algebra

An order N tensor is denoted as $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ where x_{i_1, i_2, \dots, i_N} corresponds to the $(i_1, i_2, \dots, i_N)^{th}$ element of the tensor \mathcal{X} . Vectors obtained by fixing all indices of the tensor except the one that corresponds to n^{th} mode are called mode- n fibers.

Mode- n product The mode- n product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_n \times \dots \times I_N}$ and a matrix $\mathbf{U} \in \mathbb{R}^{J \times I_n}$ is denoted as $\mathcal{Y} = \mathcal{X} \times_n \mathbf{U} = (\mathcal{Y})_{i_1, i_2, \dots, i_{n-1}, j, i_{n+1}, \dots, i_N} = \sum_{i_n=1}^{I_n} x_{i_1, \dots, i_n, \dots, i_N} u_{j, i_n}$ and is of size $I_1 \times \dots \times I_{n-1} \times J \times I_{n+1} \times \dots \times I_N$.

Tensor matricization Process of reordering the elements of the tensor into a matrix is known as matricization or unfolding. The mode- n matricization of tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times \dots \times I_n \times \dots \times I_N}$ is denoted as $\mathbf{Y}_{(n)} \in \mathbb{R}^{I_n \times \prod_{i \in \{1, \dots, N\} / \{n\}} I_i}$ and is obtained by arranging mode- n fibers to be the columns of the resulting matrix.

Rank-One Tensors An N -way rank-one tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ can be written as the outer product of N vectors.

$$\mathcal{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \dots \circ \mathbf{a}^{(N)} \quad (1)$$

where ' \circ ' is the vector outer product.

Tensor Rank Unlike matrices which have a unique definition of rank, there are multiple rank definitions for tensors including *tensor rank* and *tensor n -rank*. The *rank* of a tensor \mathcal{X} is the smallest number of rank-one tensors that form \mathcal{X} as their sum.

The n -Rank Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ be an N -way tensor, the n -rank of \mathcal{X} is the collection of rank of mode matrices $\mathbf{X}_{(n)}$ and is denoted as:

$$\text{rank}_n(\mathcal{X}) = \{\text{rank}(\mathbf{X}_{(1)}), \text{rank}(\mathbf{X}_{(2)}), \dots, \text{rank}(\mathbf{X}_{(n)})\} \quad (2)$$

where $n = 1, 2, \dots, N$.

III. MULTISCALE ANALYSIS OF HIGHER-ORDER DATASETS

In this section, we present a multiscale analysis procedure named as Multiscale HoSVD (MS-HoSVD) for an N th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$. The proposed method recursively applies the following two-step approach: (i) Low-rank tensor approximation, (ii) Decomposing the residual (original minus low-rank) tensor into subtensors.

A tensor \mathcal{X} is decomposed using HoSVD as follows:

$$\mathcal{X} = \mathcal{C} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \dots \times_N \mathbf{U}^{(N)}, \quad (3)$$

where $\mathbf{U}^{(n)}$ s are the left singular vectors of $\mathbf{X}_{(n)}$ s. The low-rank approximation of \mathcal{X} is obtained by

$$\hat{\mathcal{X}}_0 = \mathcal{C}_0 \times_1 \hat{\mathbf{U}}^{(1)} \times_2 \hat{\mathbf{U}}^{(2)} \dots \times_N \hat{\mathbf{U}}^{(N)} \quad (4)$$

where $\hat{\mathbf{U}}^{(n)} \in \mathbb{R}^{I_n \times r_n}$ s are the truncated projection matrices obtained by keeping the first r_n columns of $\mathbf{U}^{(n)}$ and $\mathcal{C}_0 = \mathcal{X} \times_1 \hat{\mathbf{U}}^{(1), \top} \times_2 \hat{\mathbf{U}}^{(2), \top} \dots \times_N \hat{\mathbf{U}}^{(N), \top}$. The tensor \mathcal{X} can now be written as

$$\mathcal{X} = \hat{\mathcal{X}}_0 + \mathcal{W}_0, \quad (5)$$

where \mathcal{W}_0 is the residual tensor.

For the first scale analysis, to better encode the details of \mathcal{X} , we adapted an idea similar to the one presented in [24]. The residual tensor of 0th scale \mathcal{W}_0 is first decomposed into subtensors as follows.

Tensor $\mathcal{W}_0 \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is unfolded across each mode yielding $\mathbf{W}_{0,(n)} \in \mathbb{R}^{I_n \times \prod_{j \neq n} I_j}$ whose columns are the mode- n fibers of \mathcal{W}_0 . For each mode, rows of $\mathbf{W}_{0,(n)}$ are partitioned into c_n non-overlapping clusters by a clustering algorithm and the Cartesian product of the partitioning labels coming from

different modes yields index sets of $K = \prod_{i=1}^N c_i$ subtensors $\mathcal{X}_{1,k}$ where $k \in \{1, 2, \dots, K\}$.

Let J_0^n be the index set corresponding to the n th mode of \mathcal{W}_0 where $J_0^n = \{1, 2, \dots, I_n\}$, $J_{1,k}^n$ be the index set of the subtensor $\mathcal{X}_{1,k}$ for the n th mode, where $J_{1,k}^n \subset J_0^n$ with $n \in \{1, 2, \dots, N\}$. Index sets of subtensors satisfy $\cup_{k=1}^K J_{1,k}^n = J_0^n$ and $J_{1,k}^n \cap J_{1,l}^n = \emptyset$ when $k \neq l$ for all $k, l \in \{1, 2, \dots, K\}$. For example, the index set of the first subtensor $\mathcal{X}_{1,1}$ can be written as $J_{1,1}^1 \times J_{1,1}^2 \times \dots \times J_{1,1}^N$ and the k th subtensor $\mathcal{X}_{1,k}$ is obtained by

$$\begin{aligned} \mathcal{X}_{1,k}(i_1, i_2, \dots, i_N) &= \mathcal{W}_0(J_{1,k}^1(i_1), J_{1,k}^2(i_2), \dots, J_{1,k}^N(i_N)), \\ \mathcal{X}_{1,k} &= \mathcal{W}_0(J_{1,k}^1 \times J_{1,k}^2 \times \dots \times J_{1,k}^N), \end{aligned} \quad (6)$$

where $i_n \in \{1, 2, \dots, J_{1,k}^n\}$. Low-rank approximation for each subtensor is obtained by applying HoSVD as:

$$\hat{\mathcal{X}}_{1,k} = \mathcal{C}_{1,k} \times_1 \hat{\mathbf{U}}_{1,k}^{(1)} \times_2 \hat{\mathbf{U}}_{1,k}^{(2)} \dots \times_N \hat{\mathbf{U}}_{1,k}^{(N)}, \quad (7)$$

where $\mathcal{C}_{1,k}$ and $\hat{\mathbf{U}}_{1,k}^{(n)}$ s correspond to the core tensor and low-rank projection matrices of $\mathcal{X}_{1,k}$ with rank $r_{1,k}^{(n)} < |J_{1,k}^{(n)}|$, respectively. $\hat{\mathcal{X}}_1$ is the 1st scale approximation of \mathcal{X} formed by mapping all of the subtensors onto $\hat{\mathcal{X}}_{1,k}$ as follows:

$$\hat{\mathcal{X}}_1(J_{1,k}^1 \times J_{1,k}^2 \times \dots \times J_{1,k}^N) = \hat{\mathcal{X}}_{1,k}. \quad (8)$$

Similarly, 1st scale residual tensor is obtained by

$$\mathcal{W}_1(J_{1,k}^1 \times J_{1,k}^2 \times \dots \times J_{1,k}^N) = \mathcal{W}_{1,k}, \quad (9)$$

where $\mathcal{W}_{1,k} = \mathcal{X}_{1,k} - \hat{\mathcal{X}}_{1,k}$. Therefore \mathcal{X} can be rewritten as:

$$\mathcal{X} = \hat{\mathcal{X}}_0 + \mathcal{W}_0 = \hat{\mathcal{X}}_0 + \hat{\mathcal{X}}_1 + \mathcal{W}_1. \quad (10)$$

The j th scale approximation of \mathcal{X} is obtained by decomposing $\mathcal{W}_{j-1,k}$ s into subtensors $\mathcal{X}_{j,k}$ s and fitting low-rank model to each one of them. Finally, the j^{th} scale decomposition of \mathcal{X} can be written as:

$$\mathcal{X} = \sum_{i=0}^j \hat{\mathcal{X}}_i + \mathcal{W}_j. \quad (11)$$

A pseudo code of the algorithm for a single scale decomposition is given in Algorithm 1. This procedure can be easily extended for multiple scales.

Algorithm 1 Multiscale HoSVD with 1-Scale Analysis

- 1: Input: \mathcal{X} : tensor, $\mathbf{C} = \{c_1, c_2, \dots, c_N\}$: the desired number of clusters for each mode.
 - 2: Output: $\hat{\mathcal{X}}$: 1-scale low-rank approximation of \mathcal{X} .
 - 3: $\mathcal{C}_0, \{\hat{\mathbf{U}}^{(1)}, \dots, \hat{\mathbf{U}}^{(N)}\} \leftarrow$ truncatedHOSVD(\mathcal{X}).
 - 4: $\hat{\mathcal{X}}_0 = \mathcal{C}_0 \times_1 \hat{\mathbf{U}}^{(1)} \times_2 \hat{\mathbf{U}}^{(2)} \dots \times_N \hat{\mathbf{U}}^{(N)}$.
 - 5: $\mathcal{W}_0 \leftarrow \mathcal{X} - \hat{\mathcal{X}}_0$.
 - 6: Create subtensors $\mathcal{X}_{1,k}$ and index sets $J_{1,k}^n$ from \mathcal{W}_0 where $k \in \{1, 2, \dots, K\}$, $n \in \{1, 2, \dots, N\}$ and $K = \prod_{n=1}^N c_n$.
 - 7: **for** $k = 1$ to K **do**
 - 8: $\mathcal{C}_{1,k}, \{\hat{\mathbf{U}}_{1,k}^{(1)}, \dots, \hat{\mathbf{U}}_{1,k}^{(N)}\} \leftarrow$ truncatedHOSVD($\mathcal{X}_{1,k}$).
 - 9: $\hat{\mathcal{X}}_1(J_{1,k}^1 \times J_{1,k}^2 \times \dots \times J_{1,k}^N) = \hat{\mathcal{X}}_{1,k}$.
 - 10: **end for**
 - 11: $\hat{\mathcal{X}} = \hat{\mathcal{X}}_0 + \hat{\mathcal{X}}_1$.
-

A. Computational Complexity

Computational complexity of HoSVD of an N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ where $I_1 = I_2 = \dots = I_N = I$ is $O(I^{(N+1)})$ [25]. By assuming that the clustering is performed using K-means with $c_i = c$ along each mode, the complexity of first scale MS-HoSVD analysis also includes the sum of the complexity of clustering along each mode, $N \cdot O(I^N \cdot c \cdot i)$, where i is the number of iterations and the complexity of HoSVD for each subtensor $c^N \cdot O(I/c)^{(N+1)}$. Therefore, first scale MS-HoSVD requires $O(I^{(N+1)}) + N \cdot O(I^N \cdot c \cdot i) + c^N \cdot O(I/c)^{(N+1)}$ computations and order of complexity is similar to HoSVD when $N \cdot c \cdot i$ is small compared to I .

IV. MEMORY COST OF THE FIRST SCALE DECOMPOSITION

Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ be the N th order tensor. To simplify the notation, assume that the dimension of each mode is the same, i.e. $I_1 = I_2 = \dots = I_N$ and denoted by I . Assume \mathcal{X} is approximated by HoSVD as:

$$\hat{\mathcal{X}} = \mathcal{C}_H \times_1 \mathbf{U}_H^{(1)} \times_2 \mathbf{U}_H^{(2)} \dots \times_N \mathbf{U}_H^{(N)}, \quad (12)$$

by keeping the rank of each mode matrix fixed as $\text{rank}(\mathbf{U}_H^{(i)}) = R_H$ for $i \in \{1, 2, \dots, N\}$. Let $\mathbb{F}(\cdot)$ be a function that measures the memory cost for the data, then the storage cost of \mathcal{X} decomposed by HoSVD is $\mathbb{F}(\mathcal{C}_H) + \sum_{i=1}^N (\mathbb{F}(\mathbf{U}_H^{(i)})) = R_H^N + N \cdot I \cdot R_H$.

For multiscale analysis at scale 1, $\mathcal{X} = \hat{\mathcal{X}}_0 + \hat{\mathcal{X}}_1$. The cost of storing $\hat{\mathcal{X}}_0$ is $\mathbb{F}(\mathcal{C}_0) + \sum_{i=1}^N (\mathbb{F}(\hat{\mathbf{U}}^{(i)})) = R_0^N + N \cdot I \cdot R_0$ where the rank of each mode matrix is $\text{rank}(\mathbf{U}^{(i)}) = R_0$ for $i \in \{1, 2, \dots, N\}$. The cost of storing $\hat{\mathcal{X}}_1$ is the sum of the storing cost of each $K = \prod_{i=1}^N c(i)$ subtensor $\hat{\mathcal{X}}_{1,k}$ s. Assume $c(i) = c$ for all $i \in \{1, 2, \dots, N\}$ yielding c^N subtensors and each $\hat{\mathcal{X}}_{1,k}$ is decomposed using HoSVD as $\hat{\mathcal{X}}_{1,k} = \mathcal{C}_{1,k} \times_1 \hat{\mathbf{U}}_{1,k}^{(1)} \times_2 \hat{\mathbf{U}}_{1,k}^{(2)} \dots \times_N \hat{\mathbf{U}}_{1,k}^{(N)}$. Let the rank of each mode matrix be fixed as $\text{rank}(\hat{\mathbf{U}}_{1,k}^{(i)}) = R_1$ for all $i \in \{1, 2, \dots, N\}$ and $k \in \{1, 2, \dots, K\}$. Then, the cost of the first scale is $\sum_{k=1}^K (\mathbb{F}(\mathcal{C}_{1,k}) + \sum_{i=1}^N \mathbb{F}(\hat{\mathbf{U}}_{1,k}^{(i)})) = c^N (R_1^N + N \cdot \frac{I}{c} \cdot R_1)$. Choosing $R_1 \leq \frac{R_0}{c^{(N-1)}}$ assures that the storage cost does not grow exponentially since $\mathbb{F}(\hat{\mathcal{X}}_1) < \mathbb{F}(\hat{\mathcal{X}}_0)$ and the total cost becomes $R_0^N (1 + \frac{1}{c^{N^2-2N}}) + 2N \cdot I \cdot R_0$. Similarly, picking $R_0 = R_H/2$ provides lower storage cost for the first scale analysis than HoSVD.

V. RESULTS

A. Datasets

The proposed multiscale approach is applied to 3 and 4-mode tensors containing a video from PIE dataset [26] and functional connectivity brain networks obtained from EEG measurements, respectively. The proposed approach is compared with HoSVD in terms of reconstruction error and compression ratio. In the tables below, the error refers to the normalized tensor approximation error $\frac{\|\mathcal{X} - \hat{\mathcal{X}}\|_F}{\|\mathcal{X}\|_F}$ and the compression ratio is computed as $\frac{\# \text{ total bits to store } \hat{\mathcal{X}}}{\# \text{ total bits to store } \mathcal{X}}$, where lower compression ratio indicates more compressed data.

1) *ERN data*: The proposed approach is applied to a set of EEG data containing the error-related negativity (ERN). The ERN is a brain potential response that occurs following performance errors in a speeded reaction time task usually 25-75 ms after the response [27]. EEG data from 63-channels was collected in accordance with the 10/20 system on a Neuroscan Synamps2 system (Neuroscan, Inc.) sampled at 128 Hz from 91 subjects. For each subject and response type, the pairwise average phase locking value within the ERN time window and theta frequency band was computed as described in [28] yielding a 63×63 connectivity matrix indicating the average synchrony between brain regions. Then, a 4-mode tensor $\mathcal{X}_i \in \mathbb{R}^{63 \times 63 \times 91 \times 256}$ is created for ERN data where the first and second modes represent the functional connectivity brain networks while the third and fourth mode correspond to the number of subjects and time points, respectively.

2) *PIE dataset*: A 3-mode tensor $\mathcal{X} \in \mathbb{R}^{122 \times 160 \times 69}$ is created from PIE dataset [26]. The tensor contains 138 images from 6 different yaw angles and varying illumination conditions collected from a subject where each image is converted to gray scale.

B. Fixed Rank Experiments

In the following experiments, clustering is performed by local subspace analysis (LSA) [29] and the cluster number along each mode is chosen as $c_i = 2$. The rank used in HoSVD is selected based on the size of the datasets and gradually increased to illustrate the relationship between reconstruction error and compression rate. In MS-HoSVD with 1-scale analysis, rank of each scale is selected according to the criterion $R_1 \leq \frac{R_0}{c^{(N-1)}}$ derived in Section IV.

As seen in Figure 1, MS-HoSVD provides better compression performance for both datasets with the compression performances of both approaches being very close to each other in the PIE experiment. Moreover, compressing the tensors by selecting smaller rank, i.e. lower compression ratio, increases the normalized error for both MS-HoSVD and HoSVD as expected. Therefore, as the compression rate goes down the performance of HoSVD and MS-HoSVD become comparable.

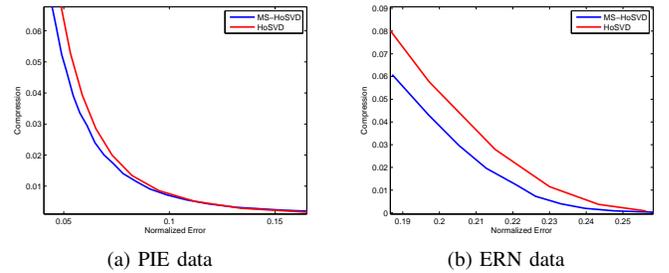


Fig. 1: Compression rate versus Normalized Error for MS-HoSVD (blue line) and HoSVD (red line) for a) PIE dataset and b) ERN data with fixed rank criterion. MS-HoSVD provides better compression performance with lower compression ratio for both datasets.

C. Adaptive Rank Experiments

In this section, we evaluated the performance of MS-HoSVD on 1-scale and 2-scale analysis compared to the HoSVD. In the following experiments, clustering is performed using local subspace analysis (LSA) [29] and the cluster number along each mode is chosen as $c_i = 2$. The rank used in HoSVD is selected adaptively depending on the energy criterion. Energy criterion determines the number of singular values kept during the SVD of the unfolded tensors along each mode such that the preserved energy is above a certain threshold as:

$$r^{(n)} = \arg \min_i \sum_{a=1}^i \sigma_a^{(n)} \quad s.t. \quad \frac{\sum_{a=1}^i \sigma_a^{(n)}}{\sum_{a=1}^{J^{(n)}} \sigma_a^{(n)}} > \tau, \quad (13)$$

where $\sigma_a^{(n)}$ is the a th singular value of the matrix obtained by unfolding the tensor \mathcal{X} along the n th mode and τ is the threshold.

Tables I and II explore the interplay between compression and approximation error for MS-HoSVD approach in comparison to the HoSVD for both ERN and PIE datasets. For ERN dataset, comparing 1-scale MS-HoSVD with energy threshold $\tau = 0.75$ to HoSVD with $\tau = 0.9$, we can see that the proposed approach outperforms HoSVD with respect to compression (Table I). Similarly, for PIE dataset, 2-scale MS-HoSVD with $\tau = 0.75$ outperforms HoSVD with $\tau = 0.92$ with respect to both error and compression ratio (Table II). Fig. 2 illustrates the influence of scale on visual quality where it can be seen that the visual quality improves with higher order scale.

TABLE I: Reconstruction error computed for compression of a 4-mode ERN tensor.

ERN	data size: $63 \times 63 \times 91 \times 256$		
	τ	Compression	Error
MS-HoSVD (1 scale)	0.7	0.1892	0.1609
MS-HoSVD (1 scale)	0.75	0.2996	0.1425
HoSVD	0.85	0.2637	0.1704
HoSVD	0.90	0.4180	0.1428

TABLE II: MSE computed for compression of a video from PIE data.

PIE	data size: $122 \times 160 \times 69$		
	τ	Compression	Error
MS-HoSVD (1 scale)	0.7	0.0614	0.0710
MS-HoSVD (2 scale)	0.7	0.0869	0.0655
MS-HoSVD (1 scale)	0.75	0.1058	0.0566
MS-HoSVD (2 scale)	0.75	0.1310	0.0529
HoSVD	0.90	0.0914	0.0637
HoSVD	0.92	0.1413	0.0545

VI. CONCLUSIONS

In this paper, we proposed a new tensor decomposition technique for better approximating the local nonlinearities in generic tensor data. The proposed approach constructs a tree structure by considering local similarities in the tensor and decomposes the tensor into lower dimensional subtensors hierarchically. A low-rank approximation of each subtensor is then obtained by HoSVD. The proposed approach is applied

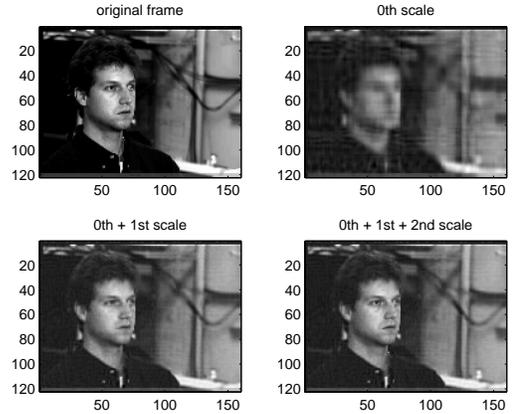


Fig. 2: A single frame of the PIE dataset showing increasing accuracy with scale ($\tau = 0.75$).

to a set of 3-way and 4-way tensors containing real datasets to evaluate its performance.

Future work will consider automatic selection of parameters such as the number of clusters and the appropriate rank along each mode. Parallelization of the algorithm will be considered to make the algorithm faster such as parallel construction of subtensors and parallel implementation of HoSVD. Faster implementation will also enable us to implement finer scale decompositions. Proposed algorithm currently constructs the tree structure based on decomposing the tensor using HoSVD. Integrating the proposed multiscale approach into other tensor decomposition methods such as PARAFAC will also be considered.

REFERENCES

- [1] D. Letexier, S. Bourennane, and J. Blanc-Talon, "Nonorthogonal tensor matricization for hyperspectral image filtering," *Geoscience and Remote Sensing Letters, IEEE*, vol. 5, no. 1, pp. 3–7, 2008.
- [2] T.-K. Kim and R. Cipolla, "Canonical correlation analysis of video volume tensors for action categorization and detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 8, pp. 1415–1428, 2009.
- [3] F. Miwakeichi, E. Martinez-Montes, P. A. Valdés-Sosa, N. Nishiyama, H. Mizuhara, and Y. Yamaguchi, "Decomposing eeg data into space-time-frequency components using parallel factor analysis," *NeuroImage*, vol. 22, no. 3, pp. 1035–1045, 2004.
- [4] A. Cichocki, "Era of big data processing: a new approach via tensor networks and tensor decompositions," *arXiv preprint arXiv:1403.2048*, 2014.
- [5] W. K. Allard, G. Chen, and M. Maggioni, "Multi-scale geometric methods for data sets ii: Geometric multi-resolution analysis," *Applied and Computational Harmonic Analysis*, vol. 32, no. 3, pp. 435–462, 2012.
- [6] Z. Zhang, J. Wang, and H. Zha, "Adaptive manifold learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 2, pp. 253–265, 2012.
- [7] J. B. Tenenbaum, V. De Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.
- [8] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.
- [9] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear image analysis for facial recognition," in *null*. IEEE, 2002, p. 20511.

- [10] J. Yang, D. Zhang, A. F. Frangi, and J.-y. Yang, "Two-dimensional pca: a new approach to appearance-based face representation and recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 1, pp. 131–137, 2004.
- [11] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [12] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
- [13] E. Acar and B. Yener, "Unsupervised multiway data analysis: A literature survey," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 1, pp. 6–20, 2009.
- [14] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *arXiv preprint arXiv:1607.01668*, 2016.
- [15] L. De Lathauwer, "Decompositions of a higher-order tensor in block terms-part ii: definitions and uniqueness," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1033–1066, 2008.
- [16] L. Grasedyck, "Hierarchical singular value decomposition of tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 4, pp. 2029–2054, 2010.
- [17] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.
- [18] X. He, D. Cai, and P. Niyogi, "Tensor subspace analysis," in *Advances in neural information processing systems*, 2005, pp. 499–506.
- [19] X. Niyogi, "Locality preserving projections," in *Neural information processing systems*, vol. 16. MIT, 2004, p. 153.
- [20] G. Dai and D.-Y. Yeung, "Tensor embedding methods," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 330.
- [21] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local discriminant embedding and its variants," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 846–853.
- [22] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1208–1213.
- [23] X. Li, S. Lin, S. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 38, no. 2, pp. 342–352, 2008.
- [24] A. Ozdemir, M. A. Iwen, and S. Aviyente, "Locally linear low-rank tensor approximation," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2015, pp. 839–843.
- [25] A. Karami, M. Yazdi, and G. Mercier, "Compression of hyperspectral images using discrete wavelet transform and tucker decomposition," *IEEE journal of selected topics in applied earth observations and remote sensing*, vol. 5, no. 2, pp. 444–450, 2012.
- [26] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression database," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 12, pp. 1615–1618, 2003.
- [27] J. R. Hall, E. M. Bernat, and C. J. Patrick, "Externalizing psychopathology and the error-related negativity," *Psychological Science*, vol. 18, no. 4, pp. 326–333, 2007.
- [28] S. Aviyente, E. M. Bernat, W. S. Evans, and S. R. Sponheim, "A phase synchrony measure for quantifying dynamic functional integration in the brain," *Human brain mapping*, vol. 32, no. 1, pp. 80–93, 2011.
- [29] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," in *Computer Vision—ECCV 2006*. Springer, 2006, pp. 94–106.