Compact Representation of Multidimensional Data Using Tensor Rank-One Decomposition

Hongcheng Wang, Narendra Ahuja
Beckman Institute, University of Illinois at Urbana-Champaign, USA
{*wanghc,ahuja* }@*vision.ai.uiuc.edu*

Abstract

*This paper presents a new approach for representing multidimensional data by a compact number of bases. We consider the multidimensional data as tensors instead of matrices or vectors, and propose a Tensor Rank-One Decomposition (*TROD*) algorithm by decomposing Nth-order data into a collection of rank-1 tensors based on multilinear algebra. By applying this algorithm to image sequence compression, we obtain much higher quality images with the same compression ratio as Principle Component Analysis (*PCA*). Experiments with gray-level and color video sequences are used to illustrate the validity of this approach.*

1. Introduction

In computer vision and graphics, we often encounter multidimensional data, such as images, video, range data and medical data such as CT and MRI. Each dimension is associated with a variation along this direction. For example, a color image can be considered as 3D data, two of the dimensions (rows and columns) being spatial, and the third being spectral (color), while a color video sequence can be considered as 4D data, time being the fourth dimension besides spatial and spectral. How to compactly represent the data using a limited number of bases is an active research problem.

Traditional methods to reduce the dimensionality of the multidimensional data are usually to reshape the data into vectors or matrices in order to apply the classical second order array processing methods. One example of these methods is Principle Component Analysis (PCA), which has been widely used, e.g., for face recognition [13] and representing 3D geometric animation sequences [1]. The spatial dimensions (image matrix in [13] and 3D coordinates in [1]) are regularized into row vectors before PCA is applied. One resulting limitation of this is that the spatial redundancies among the values of image matrix or 3D coordinates are not investigated. For example, when the number of training images is small in face recognition, PCA captures mostly the temporal redundancy in corresponding pixels in the training images. To overcome this problem, Shashua and Levin [12] recently proposed representing a collection of images using a set of rank-1 matrices. They represented an image using a matrix instead of forcing it into a vector in order to capture the spatial redundancy. Their approach is aimed at data of dimensionality 3 or less.

In this paper, we consider the multidimensional data as a tensor (first order data as vector, second order data as matrix, and third or higher order data as tensor), and propose a new algorithm to decompose a tensor into a set of rank-1 tensors based on multilinear algebra. This allows processing data in arbitrary dimensions.

2. Related Work

Our approach belongs to the class of image/video coding methods that use a compact set of basis functions. PCA [13, 1] is used to find a set of mutually orthogonal basis functions which capture the largest variation in the training data. Independent Component Analysis (ICA) is another method used for image coding. ICA [3] model is a generative model, which describes how the images are generated by a process of mixing a set of independent components. ICA is very closely related to Blind Source Separation (BSS) [5]. There are many other methods, such as, minimum entropy coding [2], sparse coding using simple-cell receptive field properties [10] and rank-1 matrix coding [12]. All these image coding methods consider each image as a vector or matrix. It is not trivial to extend these methods to higher dimensional data.

We propose using multilinear algebra for coding Nth order multidimensional data. Recently, multilinear algebra has been studied by many researchers, and applied to psychology, chemometric and signal processing [4]. Higher-Order Singular Value Decomposition (HOSVD) based on multilinear algebra has recently been used in some computer vision problems, such as motion signature analy-

0-7695-2128-2/04 $20.00 (C) 2004 IEEE

sis [14] and face recognition [15] by Vasilescu et.al, and facial expression decomposition [17] by the authors. Moreover, the approaches of the best rank-1 or best rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors have been studied [6, 8]. Lower-rank tensor approximation has been applied to noise reduction in color images by treating it as a third order tensor [9] and performing dimensionality reduction of image ensembles [16]. However, the root mean square error (RMSE) of the reconstructed images using lower-rank tensor approximation is larger than that obtained by PCA [16]. In this paper, we focus on tensor rank-one *decomposition* instead of tensor rank-one *approximation*. The RMSE of reconstructed images using our method is much smaller than that obtained by PCA for the same compression ratio.

3. Tensor Rank-One Decomposition

3.1. Overview of Multilinear Algebra

In this section, we first introduce the relevant preliminary material concerning multilinear algebra. A high-order tensor is represented as: $\mathcal{A} \in \Re^{I_1 \times I_2 \times \ldots \times I_N}$. Like SVD for matrices, HOSVD has been recently developed for tensors [7]. Any tensor $\mathcal{A}$ can be expressed as the product:

$$\mathcal{A} = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \times \cdots \times_N U^{(N)} \tag{1}$$

where, $\mathcal{S}$ is a $I_1 \times I_2 \times \ldots \times I_N$ tensor of which the subtensors $\mathcal{S}_{i_n=\alpha}$ have the properties of all-orthogonality and ordering based on the Frobenius-norms $\|\mathcal{S}_{i_n=\alpha}\|$, and $U^{(n)} = (U_1^{(n)} U_2^{(n)} \ldots U_{I_n}^{(n)})$ is a unitary $(I_n \times I_n)$ matrix. $\mathcal{S}$, known as core tensor, is in general a full matrix (not pseudodiagonal). $U^{(n)}$ provides directions of maximal oriented energy along the *n*th mode. Unfolding a tensor $\mathcal{A}$ along the *n*th mode is denoted as *uf($\mathcal{A}$, n)*. [8, 7, 6] are good sources of details of multilinear algebra.

The *n*-rank of $\mathcal{A}$, denoted by $R_n = rank_n(\mathcal{A})$, is the dimension of the vector space spanned by the *n*-mode vectors. An *N*th-order tensor has rank 1 when it equals the outer product of *N* vectors $U^1, U^2, \ldots, U^N$, i.e., $a_{i_1 i_2 \ldots i_N} = u_{i_1}^{(1)} u_{i_2}^{(2)} \ldots u_{i_N}^{(N)}$, for all values of the indices, written as:

$$\mathcal{A} = U^{(1)} \circ U^{(2)} \circ \cdots \circ U^{(N)}$$

3.2. Tensor Rank-One Decomposition Algorithm

Given a real *N*th-order tensor $\mathcal{A} \in \Re^{I_1 \times I_2 \times \ldots \times I_N}$, we find scalars λ_i and unit-norm vectors $U_i^{(1)}, U_i^{(2)}, \ldots U_i^{(N)}$, such that the new tensor is the summation of a collection of rank-1 tensors, i.e.,

$$\tilde{\mathcal{A}} = \sum_{i=1}^{r} \lambda_i U_i^{(1)} \circ U_i^{(2)} \circ \ldots \circ U_i^{(N)} \tag{2}$$

where *r* is the number of rank-1 tensors. This new tensor is an optimal solution in the sense that it minimizes the least-squares cost function:

$$f(\tilde{\mathcal{A}}) = \left\| \mathcal{A} - \tilde{\mathcal{A}} \right\|^2 \tag{3}$$

To obtain an optimal solution, we propose an iterative tensor rank-one decomposition algorithm given toward the end of this section. The basic idea is first to find a best rank-1 tensor approximation of the original tensor $\mathcal{A}$, i.e.,find a scalar λ_1 and $U_1^{(1)}, U_1^{(2)}, \ldots, U_1^{(N)}$, and then find the residual tensor $\hat{\mathcal{A}} = \mathcal{A} - \lambda_1 U_1^{(1)} \circ U_1^{(2)} \circ \cdots \circ U_1^{(N)}$, which is the difference between the reconstructed and original tensors. Repeating the process using the residual tensor $\hat{\mathcal{A}}$, we can find a second rank-1 tensor, and so on.

To find a rank-1 approximation of the original tensor, we use a formulation similar to that used in [8, 6]. A new least-squares cost function, which minimizes the distance between $\mathcal{A}$ and its approximation $\hat{\mathcal{A}}$ on the rank-1 manifold, is defined as:

$$f(\hat{\mathcal{A}}) = \left\| \mathcal{A} - \hat{\mathcal{A}} \right\|^2, s.t. \sum_{i_n} (u_{i_n}^{(n)})^2 = 1 \tag{4}$$

where $1 \leq n \leq N$.

We apply the Alternative Least Square (ALS) and the Lagrange multipliers techniques to obtain the best rank-1 tensor, i.e., in each iteration we optimize only for one of the scalars λ_i and one of the vectors $U_i^{(1)}, U_i^{(2)}, \cdots, U_i^{(N)}, 1 \leq i \leq r$, while keeping other vectors constant. From the Lagrange equations, we can obtain:

$$\lambda_i = \mathcal{A} \times {}_1 U_i^{(1)^T} \times {}_2 U_i^{(2)^T} \times \cdots \times {}_N U_i^{(N)^T} \tag{5}$$

The vectors are updated as:

$$\begin{aligned} U_{i,j+1}^{(n)} \quad = \quad & \mathcal{A} \times {}_1 U_{i,j+1}^{(1)^T} \times \cdots \times {}_{n-1} U_{i,j+1}^{(n-1)^T} \times \\ & {}_{n+1} U_{i,j}^{(n+1)^T} \times \cdots \times {}_N U_{i,j}^{(N)^T} \end{aligned}$$

with the constraint of $\left\| U_{i,j}^{(n)} \right\| = 1$. This equation can also be expressed in matrix format for the ease of implementation:

$$U_{i,j+1}^{(n)} = A \otimes (U_{i,j+1}^{(1)} \otimes \cdots \otimes U_{i,j+1}^{(n-1)} \otimes U_{i,j}^{(n+1)} \otimes \cdots \otimes U_{i,j}^{(N)}),$$

where $\otimes$ represents Kronecker product, denoted as $kr(U_{i,j+1}^{(1)}, \cdots, U_{i,j+1}^{(n-1)}, U_{i,j}^{(n+1)}, \cdots, U_{i,j}^{(N)})$.

We use HOSVD to initialize the $U_n^{(0)}$ as in [8] even though this approximation is not necessarily the globally optimal one. Regalia and Kofidis [11] recently proposed an alternate initialization strategy and gave a simple convergence proof for the rank-1 tensor approximation problem.

Figure 1. The first component (color images) of the first 4 rank-one tensors obtained by our algorithm for the video sequence used in Figure 4.

Our tensor rank-one decomposition algorithm is as follows:

Algorithm 1: Tensor Rank-One Decomposition

Data: Given an Nth-order tensor, $\mathcal{A}$, and the number of rank-1 tensors, r

Result: find λ_i and $U_i^k, (1 \leq i \leq r, 1 \leq k \leq N)$

Initialize small positive value, ε, and the maximum number of iterations, κ;

for $i = 1 : r$ **do**

  $U_n^{(0)} = \text{HOSVD}(\mathcal{A}), 2 \leq n \leq N$; we use HOSVD to initialize the dominant left singular vector of unfolded matrices;

  while $j < \kappa$ & $(\delta u_1 > \varepsilon | \delta u_2 > \varepsilon | ... | \delta u_N > \varepsilon)$ **do**

    $U_{i,j+1}^{(1)} = uf(\mathcal{A}, 1) \cdot kr(U_{i,j}^{(2)}, U_{i,j}^{(3)}, \cdots, U_{i,j}^{(N)})$;

    $U_{i,j+1}^{(1)} = U_{i,j+1}^{(1)} / norm(U_{i,j+1}^{(1)})$;

    $\delta u_1 = norm(U_{i,j+1}^{(1)} - U_{i,j}^{(1)})$;

    $\cdots$

    $U_{i,j+1}^{(N)} = uf(\mathcal{A}, N) \cdot kr(U_{i,j+1}^{(1)}, \cdots, U_{i,j+1}^{(N-1)})$;

    $U_{i,j+1}^{(N)} = U_{i,j+1}^{(N)} / norm(U_{i,j+1}^{(N)})$;

    $\delta u_N = norm(U_{i,j+1}^{(N)} - U_{i,j}^{(N)})$;

    $\lambda_i = norm(U_{i,j+1}^{(N)})$;

    $\mathcal{A} = \mathcal{A} - \lambda_i U_{i,j+1}^{(1)} \circ U_{i,j+1}^{(2)} \circ \cdots \circ U_{i,j+1}^{(N)}$;

  end

end

The algorithm presented by Shashua and Levin [12] is a special case of our algorithm in that it can be considered as an equivalent formulation of the triadic decomposition problem by considering only the p slices of $\mathcal{A}$ as a collect of matrices $A_1, A_2, \ldots, A_p$.

The compression ratio provided by our algorithm can be computed as the ratio of the sizes of the original data and its final representation. For example, a third-order tensor with dimension (m, n, k) has a compression ratio of $(m \times n \times k)/(r + r(m + n + k))$ using TROD, while the same ratio for PCA is $(m \times n \times k)/(m \times n \times p)$, where p is the number of principle components. Our algorithm can be shown to converge although we will give a proof in a longer version of this paper. Figure 1 illustrates the first component of the first 4 rank-one tensors obtained by our algorithm for the video sequence used in Figure 4.

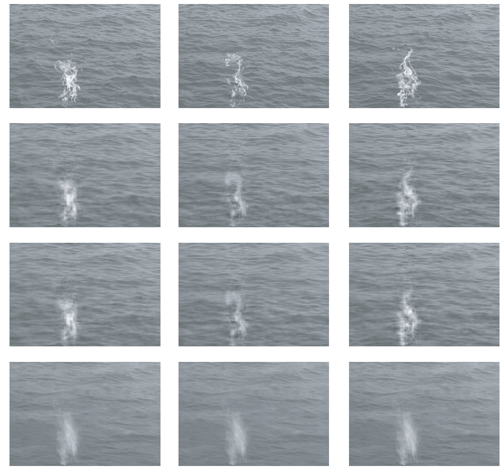

Figure 2. Reconstructed video sequence by rank-1 tensor decomposition and PCA. FIRST ROW: three frames from original sequence ; SECOND ROW: Results using tensor rank-1 decomposition with 126 rank-1 tensors; THIRD ROW: Results using matrix rank-1 decomposition with 126 rank-1 matrices; FOURTH ROW: Results using PCA with 1 principle component (compression ratio fixed at that obtained using 126 rank-1 tensors)

4. Results and Conclusion

We tested our algorithm using two video sequences. The first shows a gray-scale, dynamic, texture video composed of water and fire. The second shows the facial expression of happiness. In all the experiments[1], we choose $\varepsilon = 1e$-12, and the maximum number of iterations, $\kappa = 500$.

For the water-and-fire sequence, we use a third-order tensor with dimensions $220 \times 320 \times 20$ (20-frame sequence). We compare our results using tensor rank-1 decomposition with those obtained by PCA and matrix rank-1 decomposition methods. Figure 2 illustrates the compressed sequences and Figure 3 shows the corresponding residual errors (mean

[1] We thank D. Cremers and Jian Yuan for the video sequences.

0-7695-2128-2/04 $20.00 (C) 2004 IEEE

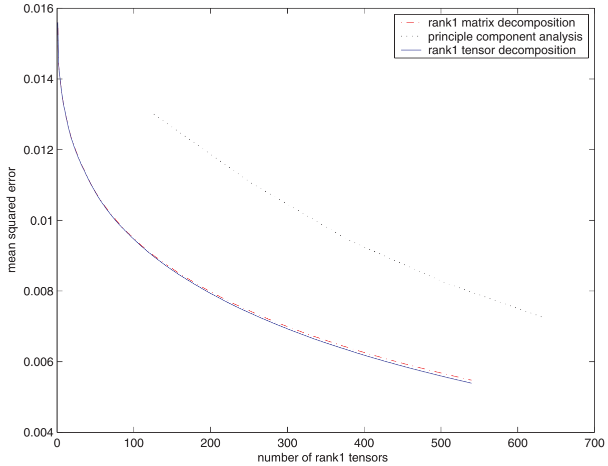

Figure 3. Residual errors obtained by our algorithm vs. rank-1 matrix decomposition and PCA. The horizontal axis represents the number of rank-1 tensors (For PCA, we find the corresponding number of rank-1 tensors of number of principle components having the same compression ratio)

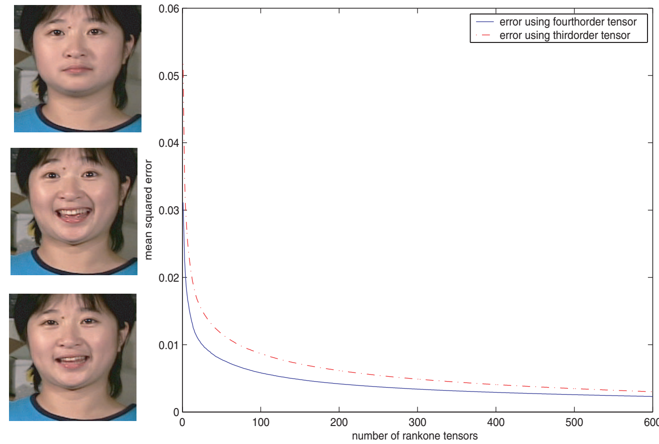

Figure 4. Facial Expression Sequence. LEFT: three frames from the original facial expression sequence. RIGHT: error comparison: Solid blue line is the error by TROD using a fourth-order tensor with 600 rank-1 tensors; Dashed red line is the error by TROD using three third-order tensors.

squared image difference) using the three methods. From residual error plots, we can see that our algorithm performs much better than PCA, and only a little better than matrix rank-1 decomposition. The latter is to be expected since matrix rank-1 decomposition is a special case of ours.

For the facial expression sequence, we construct a fourth-order tensor with dimensions $240 \times 240 \times 3 \times 9$ (3 channels and 9 frames). We compare our result using a fourth-order tensor with that obtained using three third-order tensors with dimensions of $240 \times 240 \times 9$ each. Figure 4 shows that the fourth-order case has a lower reconstruction error.

The support of the Office of Naval Research under grant N00014-03-1-0107 is gratefully acknowledged.

References

[1] M. Alexa and W. Muller. Representing animations by principle components. *Eurographics 2000*, 19(2), 2000.

[2] H. Barlow. Unsupervised learning. *Neural Computation*, pages 295–311, 1989.

[3] P. Comon. Independent component analysis: a new concept? *Signal Processing*, 36(3):11–20, 1994.

[4] P. Comon. Tensor decompositions: State of the art and applications. *IMA Conf. Mathematics in Signal Processing*, pages 18–20, Dec. 2000.

[5] C. Jutten and J. Herault. Blind separation of sources, part ɪ: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, pages 1–10, 1991.

[6] E. Kofidis and P. A. Regalia. On the besrt rank-1 approximation of higher-order supersymmetric tensors. *SIAM J. Matrix Anal. Appl.*, 23(3):863–884, 2002.

[7] L. Lathauwer, B. D. Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000.

[8] L. Lathauwer, B. D. Moor, and J. Vandewalle. On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.*, 21(4):1324–1342, 2000.

[9] D. Muti and S. Bourennane. Multidimensional signal processing using lower-rank tensor approximation. *ICASSP*, 2003.

[10] B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(13), 1996.

[11] P. A. Regalia and E. Kofidis. The higher-order power method revisited: convergence proofs and effective initialization. *ICASSP*, 2000.

[12] A. Shashua and A. Levin. Linear image regression and classification using the tensor-rank principle. *CVPR*, 2001.

[13] M. Turk and A. Pentland. Eigen faces for recognition. *J. of Cognitive Neuroscience*, 1991.

[14] M. A. O. Vasilescu. Human motion signatures: analysis, synthesis, recognition. *ICPR'02*, 2002.

[15] M. A. O. Vasilescu and D. Terzopoulos. Multilinear image analysis for facial recognition. *ICPR'02*, Aug. 2002.

[16] M. A. O. Vasilescu and D. Terzopoulos. Multilinear subspace analysis of image ensembles. *CVPR'03*, June 2003.

[17] H. Wang and N. Ahuja. Facial expression decomposition. *Intern. Conf. on Computer Vision, ICCV03*, pages 958–965, 2003.