

# Tensor-based Multiple Object Trajectory Indexing and Retrieval

Xiang Ma, Faisal I. Bashir, Ashfaq A. Khokhar, Dan Schonfeld

University of Illinois at Chicago  
Department of Electrical and Computer Engineering  
815 S.Morgan St., Chicago, IL, 60607  
{mxiang,fbashir,ashfaq,ds}@ece.uic.edu

## ABSTRACT

This paper presents novel tensor-based object trajectory modelling techniques for simultaneous representation of multiple objects motion trajectories in a content based indexing and retrieval framework. Three different tensor decomposition techniques-PARAFAC, HOSVD and Multiple-SVD-are explored to achieve this goal with the aim of using a minimum set of coefficients and data-dependant bases. These tensor decompositions have been applied to represent full as well as segmented trajectories. Our simulation results show that the PARAFAC-based representation provides higher compression ratio, superior precision-recall metrics, and smaller query processing time compared to the other tensor-based approaches.

## 1. INTRODUCTION

In most of the existing content-based video indexing and retrieval systems, object motion (trajectory) stands out as the best cue for describing the rich dynamic content of video clip [1][2]. Among the systems that use object motion, majority of them represent each object trajectory individually, thus limiting the system to only single object motion based queries. At times observing the trajectories of multiple interacting objects provides better clue to the underlying activity which otherwise may not be apparent from observing the motions of the same objects individually. However, the problem of simultaneously representing multiple object trajectories in a single framework is hard.

In this paper we present a novel and pioneering approach to deal with multiple object trajectories. In the proposed approach multiple object trajectories are mathematically represented as tensors. Different tensor decomposition techniques, namely PARAFAC, HOSVD and Multiple-SVD, are investigated to analyze, index and query the multiple objects trajectory data. In this presentation, we introduce a new mathematical framework based on tensor analysis for indexing and retrieval of simultaneous multiple object trajectories in video sequences.

## 2. TENSOR-BASED INDEXING AND RETRIEVAL

A tensor, also known as n-way array or multidimensional matrix or n-mode matrix, is a higher order generalization of a vector ( first order tensor) and a matrix (second order tensor)[3]. A tensor  $\mathcal{A}$  can be represents as

$$\mathcal{A} \in \mathbf{R}^{I_1 \times I_2 \times \dots \times I_N}. \tag{1}$$

There are 3 major tensor decomposition tools: HOSVD (also called TUCKER), PARAFAC (also called CANDECOMP), and Multiple-SVD. Details of the 3 decomposition tools will be discussed in the following subsections.

### 2.1. High Order SVD

In SVD, a matrix or order-2 tensor  $\mathcal{A}$  can be decomposed as matrix product:  $\mathcal{A} = U_1 \Sigma U_2^T$ . This matrix product can be rewritten as[4]

$$\mathcal{A} = \sigma \times_1 U_1 \times_2 U_2. \tag{2}$$

By extension, a tensor  $\mathcal{A}$  of order  $N > 2$  is an N-dimensional matrix comprising N spaces. "High Order SVD (HOSVD)" is an extension of SVD that orthogonalizes these N spaces and expresses the tensor as the mode-n product of N-orthogonal spaces[4]:

$$\mathcal{A} = \mathcal{Z} \times_1 U_1 \times_2 U_2 \dots \times_n U_n \dots \times_N U_N. \tag{3}$$

Tensor  $\mathcal{Z}$ , known as the *core tensor*, is analogous to the diagonal singular value matrix in conventional matrix SVD.

Our HOSVD-based algorithm is as follows:

1. Decompose the tensor using HOSVD:

$$\mathcal{A} = \mathcal{Z} \times_1 U_1 \times_2 U_2 \times_3 U_3. \tag{4}$$

2. Project each trajectory pair on the 2 bases  $U_1$  and  $U_2$ :

$$M_{Coeff} = U_1^T \times M_{TrajPair} \times U_2. \tag{5}$$

3. Project query trajectory pair on the 2 bases, derived in step 2 above, using the following:

$$M_{QueryCoeff} = U_1^T \times M_{TrajQuery} \times U_2. \tag{6}$$

4. Calculate the Euclidean distance norm D between HOSVD coefficients of query trajectory and those of each trajectory in the tensor, and return the ones that have the minimum distance.

$$D = \|(M_{Coeff} - M_{QueryCoeff})\|^2. \quad (7)$$

where in equ.(2)-(7),  $U_i$  is the  $i$ th HOSVD base,  $U_i^T$  is transpose of  $U_i$ ,  $M_{Coeff}$  and  $M_{QueryCoeff}$  are the coefficient matrices of trajectory pair in tensor and query trajectory,  $M_{TrajPair}$  and  $M_{TrajQuery}$  are trajectory matrices of trajectory in tensor and query trajectory, respectively.

## 2.2. PARAFAC

The Parallel Factors (PARAFAC) [5] Decomposition or Canonical Decomposition (CANDECOMP) of a tensor  $\mathcal{A}$  is a decomposition of  $\mathcal{A}$  as a linear combination of a minimal number of rank-1 tensors as following

$$\mathcal{A} = \sum_{r=1}^R \rho_r \times_1 W_1^{(r)} \times_2 W_2^{(r)} \times_3 \dots \times_N W_N^{(r)}. \quad (8)$$

where R is the number of components.

Our PARAFAC-based algorithm is as follows:

1. Decompose the tensor into 3 loading vectors:

$$\mathcal{A} = \sum_r \rho_r \times_1 W_1^{(r)} \times_2 W_2^{(r)} \times_3 W_3^{(r)}. \quad (9)$$

2. Project each trajectory pair on the first 2 bases using the formula below:

$$M_{Coeff} = W_1^T \times M_{TrajPair} \times W_2. \quad (10)$$

3. Project query trajectory pair on the 2 bases, derived in step 2 above, using the formula:

$$M_{QueryCoeff} = W_1^T \times M_{TrajQuery} \times W_2. \quad (11)$$

4. Calculate the Euclidean distance norm D between PARAFAC coefficients of query trajectory and those of each trajectory in the tensor, and return the ones that have the minimum distance.

$$D = \|(M_{Coeff} - M_{QueryCoeff})\|^2. \quad (12)$$

where in equ.(8)-(12),  $W_i^{(r)}$  is the  $i$ th loading(base) of PARAFAC,  $W_i^T$  is transpose of  $W_i$ ,  $M_{Coeff}$  and  $M_{QueryCoeff}$  are the coefficient matrices of trajectory pair in tensor and query trajectory,  $M_{TrajPair}$  and  $M_{TrajQuery}$  are trajectory matrices of trajectory in tensor and query trajectory.

## 2.3. Multiple-SVD

The Multiple-SVD decomposition of a tensor can be viewed as recursive use of SVD. By viewing slices of tensor, which are matrices, as vectors, the tensor can be viewed as a matrix, then first use SVD on the matrix, and then use SVD on the slice-matrices, as shown below:

$$\mathcal{A} = \sigma_1 \times_1 A_{1\dots(N-1)} \times_2 V_N. \quad (13)$$

$$A_{1\dots(N-1)} = \sigma_2 \times_1 A_{1\dots(N-2)} \times_2 V_{N-1}. \quad (14)$$

...

$$A_{12} = \sigma_{N-1} \times_1 V_1 \times_2 V_2. \quad (15)$$

Our Multiple-SVD algorithm is summarized as:

1. Decompose the tensor as follows:

$$\mathcal{A} = \sigma_1 \times_1 A_{12} \times_2 V_3. \quad (16)$$

$$U_{12} = \sigma_2 \times_1 V_1 \times_2 V_2. \quad (17)$$

2. Project each trajectory pair in the tensor onto the 2 bases of interests (For example, V1 and V3), following these formulas:

$$M_{Coeff1} = V_{11}^T \times M_{TrajPair} \times V_{31}. \quad (18)$$

$$M_{Coeff2} = V_{12}^T \times M_{TrajPair} \times V_{32}. \quad (19)$$

3. Project query trajectory pair on the 2 bases, derived in step 2 above, using the formulas:

$$M_{QueryCoeff1} = V_{11}^T \times M_{TrajQuery} \times V_{31}. \quad (20)$$

$$M_{QueryCoeff2} = V_{12}^T \times M_{TrajQuery} \times V_{32}. \quad (21)$$

4. Calculate the Euclidean distance norm D between Multiple-SVD coefficients of query trajectory and those of each trajectory in the tensor, and return the ones that have the minimum distances.

D=

$$\sqrt{(M_{Coeff1} - M_{QueryCoeff1})^2 + (M_{Coeff2} - M_{QueryCoeff2})^2}. \quad (22)$$

where in equ.(13)-(21),  $A_{1\dots N}$  is matrix indexed by 1 to N,  $V_i$  are bases of conventional matrix SVD.  $V_i^T$  is transpose of  $V_i$ .  $M_{Coeff1}$ ,  $M_{Coeff2}$ ,  $M_{QueryCoeff1}$ ,  $M_{QueryCoeff2}$  are coefficient matrices;  $M_{TrajPair}$ ,  $M_{TrajQuery}$  are trajectory in the tensor and query trajectory matrix.

## 2.4. Comparison of HOSVD, PARAFAC and Multiple-SVD

As discussed above, both HOSVD and PARAFAC are effective tensor analysis tools and are widely used. One major difference is that in PARAFAC, a tensor is represented as a linear combination of rank-1 tensors which is the outer products of loading vectors; while in HOSVD, a tensor is represented as a product of the core tensor and loading vectors.

Specifically, suppose we have a third-order tensor  $\mathcal{A}$ . In PARAFAC, the tensor  $\mathcal{A}$  will be represented as a linear combination of rank-1 tensors. The element  $a_{ijk}$  of the tensor  $\mathcal{A}$  is expressed as following:

$$a_{ijk} = \sum_{l=1}^W a_{il}b_{jl}c_{kl}. \quad (23)$$

where  $W$  is the number of components (factors) of PARAFAC. While in HOSVD, the core tensor  $\mathcal{Z}$  is of size  $W_1 \times W_2 \times W_3$ , then the element  $a_{ijk}$  of the tensor  $\mathcal{A}$  is expressed as following:

$$a_{ijk} = \sum_{l=1}^{W_1} \sum_{m=1}^{W_2} \sum_{n=1}^{W_3} a_{il}b_{jm}c_{kn}z_{lmn}. \quad (24)$$

Multiple-SVD is a recursive use of SVD.

### 3. GLOBAL AND SEGMENTED APPROACH

When full trajectory data is available, we can use it to construct the trajectory tensor, then apply the three tensor decomposition methods to construct our indexing and query systems. This approach is called Global approach, which indicates that we use global (full) trajectory data.

When only partial trajectory data is available, or when we wish to evaluate similarity based on partial trajectory information only, we use the segmented trajectory approach described next.

We segment the full trajectory pairs into atomic meaningful "units" which are called subtrajectory pairs. These atomic units of actions are defined as motion events due to significant changes such as the points of change in velocity (1st order derivative) and acceleration (2nd order derivative). The spatial curvature of a 2-D curve is given by :

$$\mathcal{K}[k] = \frac{x'[k]y''[k] - y'[k]x''[k]}{[(x'[k])^2 + (y'[k])^2]^{3/2}}. \quad (25)$$

The value of curvature at any point is a measure of inflection point, an indication of concavity or convexity in the curve. A hypothesis testing based approach is used to locate these points of change. From the curvature pair, two non-overlapping windows of equal dimension are extracted. Let  $X$  and  $Y$  be two such windows where  $X$  contains the first  $n$  samples of the curvature pair, and  $Y$  contains the next  $n$  samples. Let  $Z$  be the  $2 \times 2n$  dimension window formed by concatenating  $X$  and  $Y$ . Then we perform the likelihood ratio test to determine if the two windows  $X$  and  $Y$  have data drawn from the same distribution. Specifically, we verify the following two hypothesis:

$$\begin{aligned} H_0 : f_x(X; \theta_x) &= f_y(Y; \theta_y) = f_z(Z; \theta_z). \\ H_1 : f_x(X; \theta_x) &\neq f_y(Y; \theta_y). \end{aligned} \quad (26)$$

Assume that curvature pair data in each window form an i.i.d random variable and they are 2-D jointly Gaussian. We first

compute the maximum likelihood estimator of mean and variance in each window. Then calculate the distance  $d$  between the distributions of  $X$  and  $Y$ . We define the distance  $d$  between  $X$  and  $Y$  as

$$\begin{aligned} d(X, Y) &= -\log \sqrt{(2\pi)} \frac{\sigma_{11}\sigma_{12}\sqrt{1-\rho_1^2}\sigma_{21}\sigma_{22}\sqrt{1-\rho_2^2}}{\sigma_{31}\sigma_{32}\sqrt{1-\rho_3^2}} \\ &+ \frac{1}{2} \left\{ \frac{1}{1-\rho_3^2} \left[ \sum_{i=1}^2 \frac{(x_{3i} - \mu_{3i})^2}{\sigma_{3i}^2} - \frac{2\rho_3(x_{31} - \mu_{31})(x_{32} - \mu_{32})}{\sigma_{31}\sigma_{32}} \right] \right. \\ &- \frac{1}{1-\rho_1^2} \left[ \sum_{i=1}^2 \frac{(x_{1i} - \mu_{1i})^2}{\sigma_{1i}^2} - \frac{2\rho_1(x_{11} - \mu_{11})(x_{12} - \mu_{12})}{\sigma_{11}\sigma_{12}} \right] \\ &\left. - \frac{1}{1-\rho_2^2} \left[ \sum_{i=1}^2 \frac{(x_{2i} - \mu_{2i})^2}{\sigma_{2i}^2} - \frac{2\rho_2(x_{21} - \mu_{21})(x_{22} - \mu_{22})}{\sigma_{21}\sigma_{22}} \right] \right\}. \end{aligned} \quad (27)$$

where  $\mu_{i1}, \mu_{i2}, \sigma_{i1}, \sigma_{i2}, \rho_i$  are means, variances and correlation coefficient of 2-D Gaussian distributions,  $i=1,2,3$  represents distribution of  $X, Y$  and  $Z$ , respectively.  $d$  is larger if  $X$  and  $Y$  have different distributions. We define decision rule which decides in favor of  $H_1$  if  $d$  is above a threshold  $\alpha$ .

### 4. SIMULATION RESULTS

In our simulation, we use the ASL (Australian Sign Language) dataset, which consists of 95 different sign trajectory classes, each class consists of 20 trajectory pairs.

We first compare the three tensor decomposition in terms of compactness. PARAFAC decomposition significantly reduces the size of data required to represent the underlying trajectory data. For example, if the original data is 1024 trajectory pairs, then the HOSVD uses a matrix of  $1024 \times 2$  coefficients to represent the data, Multiple-SVD uses a matrix of  $1026 \times 2$  coefficients, and PARAFAC with 12 components represents the entire data using only a matrix of  $12 \times 12$  coefficients.

For the quantitative assessment of the three tensor decompositions, we compute the conventional Precision-Recall curves. The conventional definition of precision ( $P_p$ ) and recall ( $P_r$ ) metrics are the following:

$$P_p = |X_i \in T| \setminus |N|. \quad (28)$$

$$P_r = |X_i \in T| \setminus |T|. \quad (29)$$

where the  $||$  operator returns the size of the set;  $|X_i \in T|$  means retrieved and relevant,  $N$  is the size of returned list and  $T$  is the size of target set in database. Figures 1 and 2 show the Precision and Recall curve for Tensor-based indexing and retrieval of full trajectory pairs and segmented trajectory pairs, respectively. PARAFAC shows superior performance than the other two methods for both full trajectory and segmented trajectory based systems, particularly it outperforms the other two in the case of segmented trajectory based system.

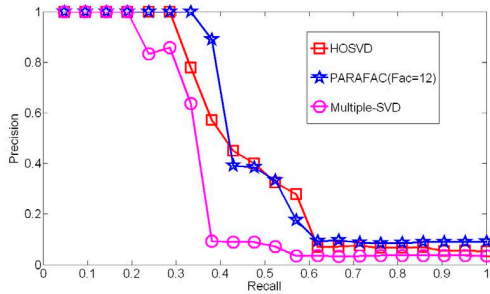


Fig. 1. Precision and Recall Curve for Global Approach

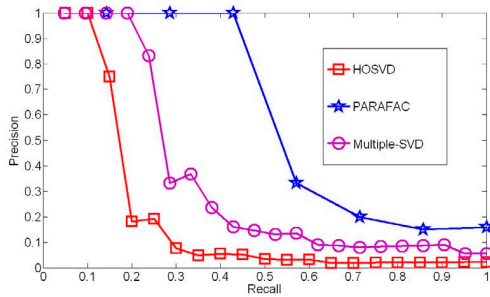


Fig. 2. Precision and Recall Curve for Segmented Approach

Since indexing and query times are also very important criteria in evaluating the performance of the indexing and retrieval system, we next compare the three decompositions in term of these parameters. As we can see in Figure 3, PARAFAC has a relatively insignificant query time (130 msec vs 19 seconds) compared to the other decompositions. However, it is at the expense of large indexing time. Note that generally indexing is an offline process and query time dictates whether a system can be used for realtime applications. We assert that for practical application, the reduction of retrieval time is much more important than the reduction of the indexing time.

Figure 4 shows the indexing and retrieval performance of the three decompositions for segmented trajectory based systems. PARAFAC yields superior performance again with much smaller retrieval time.

### 5. CONCLUSIONS

In this paper, three original tensor-based approaches solving the problem of indexing and retrieval of multiple-object trajectories have been proposed. Simulation results on the ASL dataset show higher precision-recall values, much smaller query processing times, and higher compression ratios for PARAFAC-based systems compared with other tensor-based approaches.

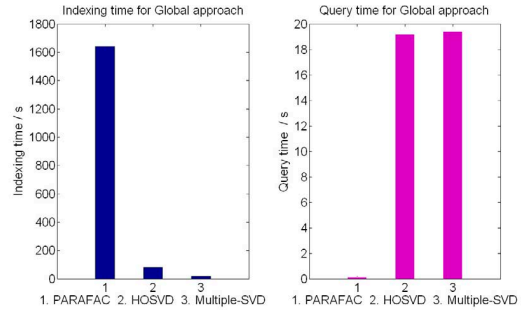


Fig. 3. Comparison of Indexing and Query Time of Global Approach

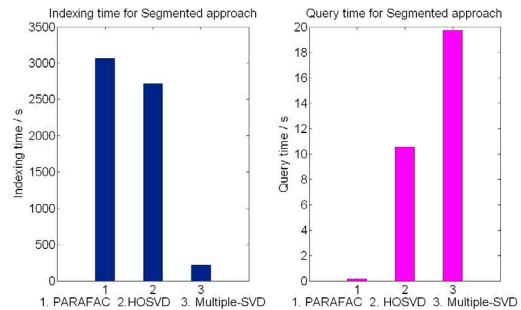


Fig. 4. Comparison of Indexing and Query Time of Segmented Approach

### 6. REFERENCES

- [1] Nicholas D. Sidiropoulos, Georgios B. Giannakis, and Rasmus Bro, "Blind parafac receivers for ds-cdma systems," vol.48, No. 3, pp.813-815, 2000.
- [2] F.Bashir, A.Khokhar, and D.Schonfeld, "Segmented trajectory based indexing and retrieval of video data," in *Proceedings of the IEEE International Conference on Image Processing*, 2003.
- [3] Lieven De Lathauwer and Bart De Moor, "From matrix to tensor : Multilinear algebra and signal processing," in *Proceedings of 4th IMA Int. Conf. on Mathematics in Signal Processing*, 1996.
- [4] M.Alex O.Vasilescu and Demetri Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *Proceedings of the 7th European Conference on Computer Vision*, 2002.
- [5] Richard A.Harshman and Margaret E.Lundy, "Parafac: Parallel factor analysis," *Computational Statistics and Data Analysis*, vol.18, pp.39-45, 1994.