

Real-Time Motion Trajectory-Based Indexing and Retrieval of Video Sequences

Faisal I. Bashir, *Student Member, IEEE*, Ashfaq A. Khokhar, *Senior Member, IEEE*, and Dan Schonfeld, *Senior Member, IEEE*

Abstract—This paper presents a novel motion trajectory-based compact indexing and efficient retrieval mechanism for video sequences. Assuming trajectory information is already available, we represent trajectories as temporal ordering of subtrajectories. This approach solves the problem of trajectory representation when only partial trajectory information is available due to occlusion. It is achieved by a hypothesis testing-based method applied to curvature data computed from trajectories. The subtrajectories are then represented by their Principal Component Analysis (PCA) coefficients for optimally compact representation. Different techniques are integrated to index and retrieve subtrajectories, including PCA, spectral clustering, and string matching. We assume a query by example mechanism where an example trajectory is presented to the system and the search system returns a ranked list of most similar items in the dataset. Experiments based on datasets obtained from UCI's KDD archives and Columbia University's multimedia group demonstrate the superiority of our proposed PCA-based approaches in terms of indexing and retrieval times and precision recall ratios, when compared to other techniques in the literature.

Index Terms—Trajectory Retrieval, Principal Component Analysis, Spectral Clustering, String Matching.

I. INTRODUCTION

Object motion-based analysis and recognition has gained significant interest in scientific circles lately. This is primarily due to unprecedented advances in hardware and software technologies that allow spatio-temporal data of objects to be easily derived from video and non-video sequences. These recent developments have led to a vast amount of spatio-temporal data resulting from spatial localization of objects of interest as a function of time. The object trajectory is typically modeled as a sequence of consecutive locations of the object on a coordinate system resulting in a vector in 2-D or 3-D Euclidean space. Examples of the object trajectory in this setting include tracking results from video trackers, sign language data measurements gathered from wired glove interfaces fitted with sensors, GPS coordinates of satellite phones and cars using CNS, animal mobility experiments, etc. One of the new areas of trajectory

analysis and understanding is automated analysis of sports videos to assist the players, coaches and sports analysts with strategies used on the field based on the motion patterns of players and their mutual interaction. Gesture- and sign language- recognition are other areas where the trajectory information plays more important role than other cues. Lastly, video surveillance which is based on techniques from object detection and tracking, human motion analysis and activity recognition relies heavily on robust methods of trajectory indexing. We emphasize that object motion plays the key role in the domain of activity analysis in general and in video surveillance in particular. Psychological studies have shown that human beings can routinely discriminate and recognize the kind of object motion using motion pattern, even in large viewing distances or poor visibility conditions where other familiarity cues such as clothes, appearance or hair style tend to disappear [17].

In all of these applications, a robust representation of the trajectories is needed to capture the spatio-temporal actions performed, particularly knowing the fact that dimensions of the feature spaces representing diverse motion trajectories are relatively large. Object trajectory indexing is thus the cornerstone of successful motion analysis systems. This paper focuses on the design of robust representations, efficient indexing mechanisms, and fast (real-time) retrieval techniques for motion trajectory-based indexing and retrieval of video sequences. The contributions of our work include:

- A fully translation-invariant method of indexing and retrieval of trajectories and subtrajectories.
- A statistically robust mechanism of motion representation based on trajectory segmentation.
- Principal Components based representation of subtrajectories in a reduced-dimension space.
- Spectral clustering to automatically decide on the optimal number of clusters and to map subtrajectories into alphabets.
- An efficient retrieval mechanism to minimize the system response time to spatio-temporal queries.

The remaining sections of this paper are organized as follows: Section II surveys related work on motion trajectory-based indexing and retrieval systems. Section III presents our two subtrajectory-based approaches using PCA coefficients.

Manuscript received January 7, 2005.

Faisal I. Bashir, Ashfaq A. Khokhar and Dan Schonfeld are with University of Illinois at Chicago, Chicago, IL, 60607 USA (phone: 312-996-5847; fax: 312-996-6465; e-mail: dans@uic.edu).

Section IV provides a comparison and analysis of the motion trajectory-based indexing and retrieval methods developed in this paper. Experiments are performed on two datasets: the Australian Sign Language dataset obtained from University of California Irvine's Knowledge Discovery in Databases archive [13], and the sports video dataset provided by the Columbia University's Digital Video and Multimedia Group (DVMM) [6]. Details of these datasets and experiments conducted are provided in Section IV. Finally, in Section V, we present a brief summary and conclusion and outline future research in this area.

II. RELATED WORK

This section provides a survey of the related work from recent literature in the areas of motion feature computation for trajectory representation, principal component analysis and applications of trajectory-based indexing and retrieval. Studies into human psychology have shown the extra-ordinary ability of human beings to recognize object motion even from minimal information system such as Moving Light Displays (MLDs) [17]. Given its importance, MPEG-7 adopted the notion of motion activity and motion trajectory in a collection of motion descriptors that capture the different aspects of motion in videos with a broad range of precision. The standard defines concise descriptors including motion activity and motion trajectory that are easy to extract and match without providing the details of the indexing and retrieval process [16][29][10]. Object motion has been an important feature for object representation and activity modeling in video applications [1]. An object trajectory-based system for video indexing is proposed in [27], in which the normalized x- and y- projections of trajectory are separately processed by wavelet transform using Haar wavelets. Chen et al [6] segment each trajectory into subtrajectories using fine-scale wavelet coefficients at high levels of decomposition. A feature vector is then extracted from each subtrajectory and Euclidean distances between each subtrajectory in the query trajectory and all the indexed subtrajectories are computed to generate a list of similar trajectories in the database. In [4], the Longest Common Subsequence (LCSS) approach is used for grouping similar motion trajectories in an agglomerative clustering algorithm. The LCSS is defined recursively as increasing distance between two sequences based on their x- and y- projections. The similarity is then computed from the two sequences as the least distance under a set of translations. Shim et al [28] propose a modification of the DTW algorithm using a k-warping distance algorithm by permitting up to k replications for an arbitrary motion of a query trajectory to measure the similarity between two trajectories. Their approach is tested on Content Based Soccer Video Retrieval (CSVR) system in which they extract the trajectory of the soccer ball by manually tracing the ball in a ground field using linear segments.

Object trajectory data can be viewed as a time series when x- and y- projections are combined for representation. There

has been tremendous amount of activity in time series representation and retrieval in recent years. Lin et al [20],[21] have presented a symbolic representation of time series approach (SAX) using Piecewise Aggregate Approximation (PAA). Although quite close to our string matching-based system, there are two major problems with it for trajectory data. The PAA uses fixed box-bases to represent continuous noisy data which might not be very suitable for most time series. Also, the notion of fixed breakpoints used in their approach to map time series segments to symbols is questionable as it is perceptually not quite appealing for trajectory data. Fink et al [12] represent time series by maxima, minima and major inclines. They identify all major inclines of all series in the dataset and index them using a range tree. Their piecewise linear approximation is similar to polynomial approximation which was shown to perform worse than our PCA-based representation in [2]. A lot of recent research effort has gone into devising new space- and time- efficient index structures. Since space-efficient indexing of generic time series is not the main topic of this presentation, we refer the reader to [11][19][23].

An important application area of trajectory-based indexing is human activity modeling. Yacoub et al [30] have presented a framework for modeling and recognition of human motions based on principal components. Each activity is represented by eight motion parameters recovered from five body parts of the human walking scenario. The high-dimensional trajectory using all the eight parameters of object motion is reduced using PCA. In [15], the issue of recognizing a set of plays from American football videos is considered. Using a set of classes each representing a particular game plan and computation of perceptual features from trajectories, the propagation of uncertainty paradigm is implemented using automatically generated Bayesian network. On similar lines, Nevatia et al [14] have addressed the issue of activity recognition in single or multiple actor situations which exhibit some specific patterns of whole body motion.

III. EIGENSPACE DECOMPOSITION FOR TRAJECTORY REPRESENTATION

An object trajectory is represented by a two-dimensional N-tuple corresponding to the x- and y- axes projections of the object centroid location at each instant of time:

$$r[k] = \{x[k], y[k]\}, \quad k = 1, \dots, N. \quad (1)$$

Our approach models the sequence of object trajectory points as a stochastic process with variability in both x- and y- directions. We compute the data-dependant orthogonal bases that transform the trajectory data into reduced dimension subspace keeping most of the original variation in the data intact. This step provides the necessary dimensionality-reduction so the trajectory matching process can be performed in the lower-dimensional subspace spanned by the data-dependant orthogonal bases. In our previous work [2], we presented a PCA-based system that treats x- and y- projections

separately and performs PCA on them individually in a two-pass approach. This section presents our approach to segmented trajectory-based PCA systems for motion trajectory representation.

A. Segmented Trajectory PCA with Euclidean Distance Retrieval

The segmented trajectory-based system indexes trajectories by first segmenting them into subtrajectories using hypothesis testing-based approach. The set of subtrajectories are then processed for principal component analysis to represent the subtrajectories in the reduced dimension subspace spanned by the eigenvectors of the dataset covariance matrix. The retrieval process computes a PCA-based subspace distance between subtrajectories which is merged to form a distance between individual trajectories.

1) Trajectory Segmentation:

As pointed out earlier, the perception of object motion is very sophisticated in humans even under poor visibility conditions as pointed out by Johansson's MLD experiments [17]. Contemporary psychological studies have provided an instructive analysis of the atomic units of actions that are of substantial value to perception. These atomic units of actions are defined as motion events due to significant changes in motion trajectories [31]. Another advantage, apart from its perceptual appeal, is that using this approach, partial queries can also be evaluated when only a part of the object motion is available due to object being temporarily out of the field of view, or tracker losing track of the object for a while.

Given above considerations, we segment the trajectories at perceptually significant points of change. The change points in trajectory data are detected as the points of change in velocity (1st order derivative) and acceleration (2nd order derivative of the data). For this purpose, we use the spatial curvature of a 2-D curve given by:

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

This feature measures the sharpness of a bend in spatial 2-D curve¹. Our hypothesis testing based approach locates these points of change. From the curvature data, two non-overlapping windows of equal dimension n are extracted. Let X and Y be two such windows where X contains the first n samples of curvature and Y contains the next n samples. Let Z be the $2 \times n$ dimension window formed by concatenating X and Y . We perform the likelihood ratio test to determine if the two windows X and Y have data drawn from the same distribution. Specifically, we have two hypotheses:

$$\begin{aligned} H_0 : f_x(X; \theta_x) &= f_y(Y; \theta_y) = f_z(Z; \theta_z) \\ H_1 : f_x(X; \theta_x) &\neq f_y(Y; \theta_y) \end{aligned} \quad (3)$$

Under the assumption of Gaussian i.i.d. random variables,

we first compute the maximum likelihood estimators of the mean and variance in each window with their likelihoods being $L(X; \theta_1)$, $L(Y; \theta_2)$ and $L(Z; \theta_3)$, where θ_i is the corresponding parameter vector of the underlying distribution of X . Then, the formulation in Eq. (4) computes distance d between the distributions of X and Y :

$$\begin{aligned} L_0 &= \frac{1}{\sqrt{2\pi}\sigma_3} \exp\left(-\frac{(x-\mu_3)^2}{2\sigma_3^2}\right) \\ L_1 &= \frac{1}{\sqrt{2\pi}\sigma_1\sigma_2} \exp\left(-\frac{1}{2}\left[\sum_{i=1}^2 \frac{(x-\mu_i)^2}{\sigma_i^2}\right]\right) \\ \lambda_L &= \frac{L_0}{L_1} \\ d(X, Y) &= -\log(\lambda_L) \\ &= -\log\sqrt{2\pi} \frac{\sigma_1\sigma_2}{\sigma_3} + \\ &\quad \frac{1}{2}\left[\frac{(x-\mu_3)^2}{\sigma_3^2} - \frac{(x-\mu_1)^2}{\sigma_1^2} - \frac{(x-\mu_2)^2}{\sigma_2^2}\right] \end{aligned} \quad (4)$$

Distance d is large when X and Y have different distributions. The windows are then slid by $m < n$ samples and the process is repeated for this configuration. This process is repeated for all the curvature data samples. A 1-D vector of distance function is then formed, and distinct local maxima separated by valleys in this distance vector are located. Figure 1 displays the segmentation results on four trajectories, two each from two different classes. Additional details can be found in [3].

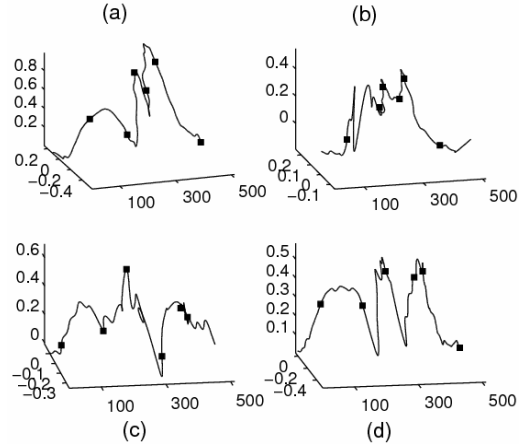


Figure 1: Trajectory Segmentation. (a) & (b): Segmentation of trajectories for 'Norway' signed by two different signers. (c) & (d): 'Alive' signed by same two signers.

2) Translation-invariant Subtrajectory Representation:

The subtrajectories are normalized before PCA-based representation, to achieve temporal- and spatial- translation invariance. The starting point of each subtrajectory is shifted to the origin which will ensure that matching process at retrieval stage does not depend on what point of time the subtrajectory in a given trajectory starts. Spatial invariance is achieved by normalization:

¹ We have also experimented with the so-called 'spatio-temporal curvature' [25]. Based on better segmentation, we stick to this definition of curvature which is more popular in image processing community.

$$\begin{aligned} X'_k &= \frac{X_k - X_{\min}}{X_{\max} - X_{\min}} \\ Y'_k &= \frac{Y_k - Y_{\min}}{Y_{\max} - Y_{\min}} \end{aligned} \quad (5)$$

The median segment size of set of subtrajectories is computed from the segmentation results. The X- and Y- data of each subtrajectory are concatenated into one vector which is then resampled to twice the median segment size determined before. Finally, we form a data matrix from subtrajectory data and perform the PCA on this matrix. We compute the optimal dimensionality m by observing that principal components are successively chosen to have the largest possible variance, and the variance of the k^{th} PC α_k is its eigenvalue λ_k . We compute a cumulative sum threshold as:

$$t_m = 100 \times \frac{\sum_{j=1}^m \lambda_j}{\sum_{j=1}^r \lambda_j}, \quad (6)$$

where $r \gg m$ is the rank of the data matrix. Choosing a cut-off of say $k = 95\%$, and retaining m PCs, where m is the smallest integer for which $t_m > k$, provides a rule which can be used, in practice, to preserve, in the first m PCs, most of the information about the trajectories in the dataset. Based on this computation, the first m principal components resulting from PCA are stored as the PC transformation matrix Φ_m which is used to compute the PCA coefficients of trajectories:

$$B = \Phi_m^T \cdot [A - Avg] , \quad (7)$$

where A is the matrix of original subtrajectories, B is the matrix of corresponding PCA coefficients and Avg is the average subtrajectory. The subtrajectories are then represented by their low-dimensional PCA coefficients.

3) Retrieval using Euclidean Distance:

The retrieval process is tailored to produce results quickly with the aid of pre-computed indexing structures. The query trajectory is segmented and transformed to its PCA coefficients using the PCA transformation matrix computed in the indexing phase:

$$D = \Phi_m^T \cdot [C - Avg] , \quad (8)$$

where C is the set of p -dimensional query trajectories, and D represents the corresponding m -dimensional PCA coefficients where $m \ll p$. The subtrajectory distances are finally merged using the following metric:

$$E_k = \sqrt{\sum_{l=1}^M \sum_{i=1}^{N_k} \min^2(E_{k,l}, \mu)} , \quad (9)$$

where M is the total number of subtrajectories in query trajectory, N_k is the total number of subtrajectories in k^{th} indexed trajectory, μ is an arbitrary penalizing constant for non-matching subtrajectories in a full trajectory and $E_{k,l}$ represents Euclidean distance between the PCA coefficients of l^{th} query subtrajectory and i^{th} subtrajectory of

k^{th} full trajectory in the database.

We have tested the subtrajectory PCA-based algorithm on the Australian Sign Language (ASL) dataset. The details of this dataset are provided in Section IV. The first experiment presents full trajectories from a known class in the dataset and computes the Precision-Recall metrics. The second experiment simulates the effect of partial trajectories where one portion of the trajectory is unavailable due to occlusion, loss of tracking, etc.. Also the system is tested for its spatio-temporal sensitivity in trajectory representation. For each of the 35 full-length queries, we add 5 random x - and y -translations to it and pose the resulting 175 trajectories as queries. The corresponding precision-recall metrics for full-length query and for its shifted version are exactly the same demonstrating that the normalized subtrajectory-based representation is translation invariant. The results are displayed in Figure 2.

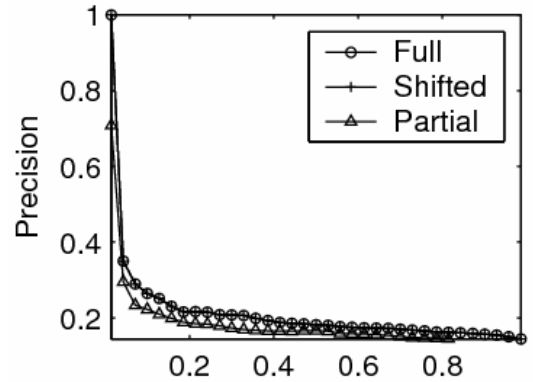


Figure 2: Precision-Recall metrics for PCA-based system using Euclidean distance retrieval. Average precision-recall for 35 full-, 35 partial- and 175 shifted- trajectory queries are shown.

We shall now present a major improvement to the indexing and retrieval system which uses string matching methods on clusters of segmented subtrajectories.

B. Subtrajectory PCA using String Matching for Retrieval

This variation of the PCA-based trajectory retrieval process is motivated by the fact that if the segmentation process is stable, it results in quite similar sets of subtrajectories for the trajectories belonging to one class. Our indexing system proceeds the same way as in the previous approach until the PCA computation of subtrajectories. Once the PCA coefficients of all the subtrajectories have been computed, we use spectral clustering to group the subtrajectory data. Once this grouping is found, each cluster is treated as an alphabet and each trajectory is represented as a string in the dataset. The query trajectory is transformed into a string using the same process and Edit Distance is used to measure the distance between the query trajectory and all trajectories in the dataset. The indexing and retrieval process based on this approach is sketched in Figure 3. The following subsections describe the components of this system in more detail.

1) Spectral Clustering using K-Means on PCA Coefficients:

Spectral clustering is a relatively new technique which

employs eigenspace decomposition of the symmetric similarity matrix between items to be clustered. Ding et al. [8] prove that when optimizing the K-means objective function for a specific value of k , the continuous solutions for the discrete cluster indicator vectors are given by the first $k-1$ principal components of the similarity matrix. In [9], they discuss an eigenspace method for grouping and re-ordering the objects for cluster assignment once the continuous cluster indicator vectors are estimated. Other interesting applications can also be found in [22][24]. Given the set of n subtrajectories in R^p represented in a matrix form S , we look for the number of clusters k and put the n subtrajectories into k clusters. We then compute the cluster validity score proposed in [24]:

$$\alpha_k = \sum_{c=1}^k \frac{1}{N_c} \sum_{i,j \in Z_c} W_{ij}, \quad (10)$$

where Z_c denotes the cluster c , N_c the number of items in cluster c , and W is the matrix formed out of Y , the normalized eigenvector matrix defined below.

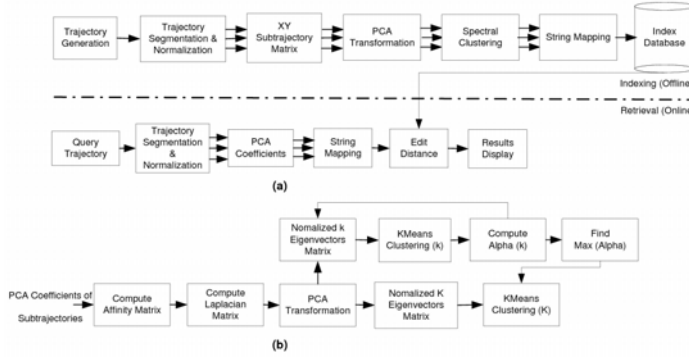


Figure 3: (a) Block diagram of segmented-trajectory-based system using string matching. (b) Spectral clustering using K-Means on PCA coefficients.

We use the following algorithm to find the number of clusters k and to perform the clustering:

1. Form the subtrajectory affinity matrix $A \in R^{n \times n}$ defined by $A_{ij} = \exp(-\|s_i - s_j\|^2 / 2\sigma^2)$ if $i \neq j$, and $A_{ii} = 0$. Here s_i refers to the PCA coefficients of i^{th} subtrajectory.
2. Define D to be the diagonal matrix whose (i,i) -element is the sum of A 's i^{th} row, and construct the Laplacian matrix $L = D^{-1/2} A D^{1/2}$.
3. Find the n principal components x_1, x_2, \dots, x_n of L .
4. Using the matrix formed by stacking k largest PCs $X = [x_1, x_2, \dots, x_k] \in R^{n \times k}$, form the normalized eigenvector matrix Y by renormalizing each of X 's rows to have unit length, $Y_{ij} = X_{ij} / (\sum_j X_{ij}^2)^{1/2}$. Also compute $n \times n$ matrix $W = Y Y'$.
5. Use K-Means clustering on the PCA representation of subtrajectories S to form k clusters.
6. Calculate α_k .
7. Iterate the steps 4 through 6 for $k = 1, 2, \dots, K$, and find the maxima.

Once the clustering is done, each cluster is assigned a distinct alphabet mapping each trajectory to a string. At this time during indexing when the clusters are ready, we also compute their centroidal PCA coefficients representation. These centroidal PCA coefficients will be used in retrieval stage to map the query trajectory into its string representation.

2) String Matching-based Retrieval:

The indexing process represents the trajectories as strings with a sequence of alphabets. At query time, the query trajectory is segmented and transformed to its PCA coefficients. Each query subtrajectory is then mapped to an alphabet by computing minimal PCA-based distance between the query subtrajectory and the set of cluster centroids:

$$\psi(Y_q) = \{i : d(Y_q, Y_i) < d(Y_q, Y_j), \forall i \neq j, 1 \leq i, j \leq |C|\}, \quad (11)$$

where $|C|$ represents the total number of all clusters, Y_q the PCA coefficients representation of the query trajectory, Y_j represents the PCA coefficients of the j^{th} cluster centroid and $d(\dots)$ represents the Euclidean distance operation. The

trajectory matching process then boils down to that of string matching. Our string matching approach is based on the Edit Distance (ED), implemented using dynamic programming, which is widely used in bio-informatics and speech recognition to measure the similarity between two strings, as given by the *Needleman-Wunsch Edit Distance Theorem* [7]. The ED between two sequences is defined as the number of insert, delete or replace operations needed to change one sequence into another. The query trajectory distance with all the trajectories in the database is thus computed and the distance list is then sorted to obtain the ranked list of matching trajectories in the database.

We report the results on the same dataset using the same set of full-, partial- and shifted- query trajectories as in previous section. Figure 4 shows the precision-recall metrics for these experiments.

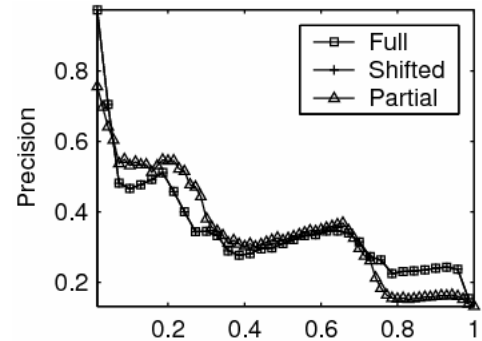


Figure 4: Precision-recall metrics for PCA-based system using string matching for retrieval. Average precision-recall metrics for 35 full-, 35 partial- and 175 shifted trajectory queries are shown.

IV. COMPARISON AND ANALYSIS

This section compares the performance of the two approaches proposed in the previous section with a global approach from recent literature [5]. This method has been selected for comparison since it has superb retrieval

performance and it is a global method for trajectory retrieval. Their approach transforms the x- and y- projections of the trajectory into quantized movement direction and movement distance ratio pairs and mapped to an alphabet character. This results in a string representation of the trajectory. At the time of query, the Normalized Edit Distance (NED) is computed between the query string and all strings in the database. We also compare our results with a global PCA-based approach which represents trajectories using PCA coefficients without segmentation [2].

A. Data Sets

We use two datasets in our simulations: The Australian Sign Language (ASL) dataset is obtained from the UCI's KDD archives² [13]. We extract the x- and y- locations corresponding to 8 words signed by five signers resulting in 552 trajectories. The second dataset in our experiments has been provided to us by Columbia University and contains object trajectories tracked from video clips of sports activities, like high jump, slalom skiing, etc. This dataset, HJSL (108), contains around 40 trajectories of high jump, and 68 trajectories of slalom skiing objects. A superset of this dataset, HJSL(408), is formed by extracting some 300 trajectories from random object motions to be treated as noise.

B. Simulation Results and Discussion

This section summarizes the results of our computer simulation experiments. We measure two performance metrics for all the trajectory indexing and retrieval systems: the accuracy of the content-based information retrieval system is measured in terms of precision and recall metrics, while the efficiency is measured in terms of indexing and retrieval times as well as asymptotic complexity of retrieval process. The precision-recall metrics are computed for multiple trajectories being posed to the system under various degradation conditions such as random translations and partial queries.

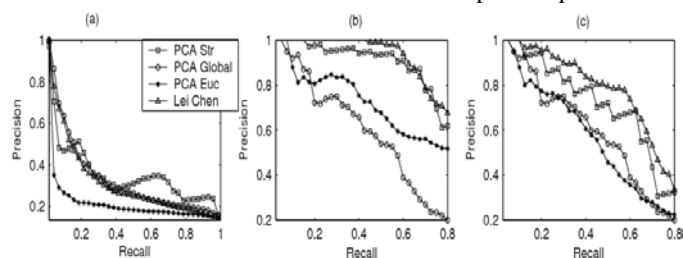


Figure 5: Precision-Recall metrics for proposed systems based on the three datasets. (a) ASL (552). (b) HJSL (108). (c) HJSL (408).

The first experiment poses a set of 35 full-length queries to the underlying systems and retrieves the ranked list of responses. Precision-recall metrics are computed for each of these 35 queries and average value of precision is computed for each value of recall. The results are shown in Figure 5. We also report the F-value curves³ for this experiment for all datasets and retrieval methods in Figure 6. The F-values are

plotted against the total number of true positives in the ranked list of responses for a query. The F-value curves are used since it is sometimes difficult to interpret the results based on the pairs of precision and recall and it is easier to assess the performance based on their harmonic mean which is provided by their F-value. To avoid repetition, other experiments are reported in terms of precision-recall only. Based on these results, we see that our segmented PCA-based system using the string matching approach presents the best trade-off between accuracy and efficiency. It performs at least as well or better as compared to the exact matching-based approach by Lei-Chen [5]. The next two experiments are designed to test the sensitivity of our systems to spatio-temporal shift and partial trajectory information. The 35 full-length query trajectories from the previous experiment are randomly shifted in both the x- and y- directions to produce shifted query trajectories. For each query trajectory, 5 randomly shifted versions are generated which results in 175 query trajectories. These queries are then posed to the underlying system and average precision-recall results are computed. The third experiment highlights the robustness of the indexing and retrieval systems to object occlusion. Here, 35 partial queries synthetically generated from the 35 full-length queries in the previous experiments are used. The partial query trajectories are formed by cutting away the last $\frac{1}{4}$ of the full-length trajectories. The results of these two experiments are shown in Figures 2 and 4. These experiments demonstrate the invariance of our approach to such transformations. Our extensive experiments on the two standard datasets thus reveal that our subtrajectory-based PCA systems provide a superior representation of the trajectory information, yield high precision and perform within real-time requirements. Further investigation of the scalability of trajectory indexing and retrieval methods to extremely large datasets using high-dimensional indexing data structures is required.

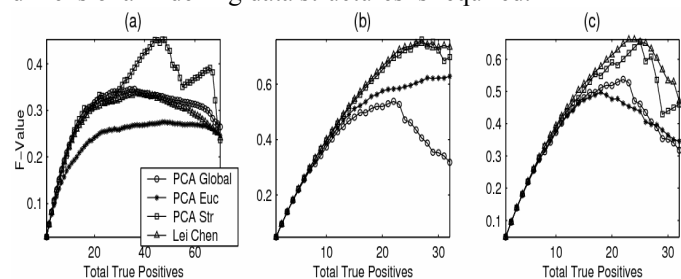


Figure 6: F-Values for proposed systems based on the three datasets. (a) ASL (552). (b) HJSL (108). (c) HJSL (408).

TABLE 1: INDEXING AND RETRIEVAL TIMES OF ALL SYSTEMS FOR THREE DIFFERENT DATASETS. HJSL(108) IS A SUBSET OF HJSL (408).
*- TOTAL TIME FOR 35 QUERIES = 9.3311 HOURS.

² KDD archive: <http://kdd.ics.uci.edu/databases/auslan/auslan.html>
Original donor: <http://www.cse.unsw.edu.au/~waleed/tml/data/>.

³ The F-value is given by $F\text{-value} = 2/(1/\text{precision} + 1/\text{recall})$.

TABLE 2: ASYMPTOTIC COMPLEXITY ANALYSIS OF THE RETRIEVAL TIMES OF PROPOSED AND COMPARED ALGORITHMS.

N: SIZE OF EACH TRAJECTORY (FIXED AFTER NORMALIZATION).

M: TOTAL NUMBER OF TRAJECTORIES IN THE DATABASE.

P: DIMENSIONALITY OF THE PCA SUBSPACE.

K: AVERAGE NUMBER OF SUBTRAJECTORIES IN QUERY AS WELL AS DATABASE TRAJECTORIES.

L: TOTAL NUMBER OF CLUSTERS. SAME AS THE SIZE OF ALPHABETS.

PCA-GI	$T_{retrieval} = T_{projection} + T_{distance} = p.O(N) + M.O(p)$
PCA-Euc	$T_{retrieval} = T_{segmentation} + T_{projection} + T_{distance}$ $= O(N) + k.p.O(N/k) + k.M.O(p)$
PCA-Str	$T_{retrieval} = T_{segmentation} + T_{projection} + T_{mapping} + T_{distance}$ $= O(N) + k.p.O(N/k) + k.l.O(p) + k.M.O(k^2)$
L-C [5]	$T_{retrieval} = M.O(N^2)$

We have implemented the above methods using Matlab 6.5 and running times are noted on an Intel Pentium IV 3 GHz machine with 1 GB RAM. The indexing and retrieval times are presented in Table 1. In this table, ‘PCA-GI’ refers to the global PCA-based system used as baseline [2], ‘PCA-Euc’ refers to the segmented PCA-based system using the Euclidean measure in section III.A, ‘PCA-Str’ refers to the segmented PCA-based system using string matching on segment clusters in section III.B, and ‘L-C[5]’ refers to the string matching-based system proposed by Lei Chen et al [5]. It is evident from the results in Table 1 that while the exact match philosophy-based approach [5] takes impractical amount of time, our PCA-based systems can easily perform in the real-time requirement settings. Table 2 compares the retrieval complexity analysis of the four trajectory-based indexing and retrieval systems considered in this work. Here, only the retrieval times are considered. Table 2 shows an increasing complexity of retrieval processes among the three PCA-based systems while attaining better precision-recall metrics and tolerance for partial trajectory information. The retrieval complexity of PCA-based systems is far less than that of [5].

V. SUMMARY AND CONCLUSIONS

In this paper, we have laid out a detailed discussion on the topic of motion trajectory-based indexing and retrieval of data captured from any form of hardware/software setting capable of object tracking. We alleviate the drawbacks of global processing approaches – inability to handle partial trajectory information – by segmenting the object trajectories and then representing the original segments by their PCA coefficients. The results are further improved by clustering the subtrajectories by using a spectral clustering approach to estimate the optimal number of clusters in an unsupervised

	Indexing Time (sec.)			Average Retrieval Time (sec.)		
	ASL(552)	HJSL(108)	HJSL(408)	ASL(552)	HJSL(108)	HJSL(408)
PCA-GI	22.36	31.124	32.5	0.262	0.226	0.216
PCA-Euc	185.391	43.3330	194.4	0.877	0.549	0.737
PCA-Str	2875.0	76.8800	1199.1	0.426	0.425	0.613
L-C [5]	10.109	3.455	26.2	959.8*	399.4	1356.9

way. We have based our experiments on two parameters to be measured: accuracy and efficiency. The accuracy is measured in terms of retrieval effectiveness using precision-recall metrics. The efficiency is measured in terms of the time taken by indexing and retrieval processes as well as their asymptotic complexity. Based on this analysis, our segmented PCA-based systems have shown a marked improvement in precision at wide range of recall values. As compared to a similar string matching-based approach which operates without trajectory segmentation, our systems show an exceptional amount of improvement in online retrieval time.

Future research must focus on motion trajectory-based indexing and retrieval of video sequences that are robust to camera orientation and movement. An important extension of our approach would be required to perform multiple motion trajectory-based indexing for ‘semantic’ retrieval from video sequences. It is also possible that the basis of our approach could be used for video sequence mining by detection and identification of motion trajectories in the video query.

REFERENCES

- [1] AbouGhazaleh N., Gamal Y.E., “Compressed Video Indexing Based on Object’s Motion”, International Conference on Visual Communication and Image Processing, VCIP’00, June 2000, Australia.
- [2] Bashir F., Khokhar A., Schonfeld D., “Segmented Trajectory-based Indexing and Retrieval of Video Data”, International Conference on Image Processing, ICIP 2003, Barcelona, Spain.
- [3] Bashir F., Khokhar A., Schonfeld D., “A Hybrid System for Affine-Invariant Trajectory Retrieval”, 6th ACM SIGMM Multimedia Information Retrieval workshop, MIR 2004.
- [4] Buzan D., Sclaroff S., Kollios G., “Extraction and Clustering of Motion Trajectories in Video”, International Conference on Pattern Recognition (ICPR), 2004.
- [5] Chen L., Ozsu M. T., Oria V., “Symbolic Representation and Retrieval of Moving Object Trajectories”, 6th ACM SIGMM Multimedia Information Retrieval workshop, MIR 2004, New York.
- [6] Chen W., Chang S.F., “Motion Trajectory Matching of Video Objects”, IS&T/SPIE, San Jose, CA, January 2000.
- [7] Clote P., Backofen R., “Computational Molecular Biology – An Introduction”, John Wiley & Sons, 2000.
- [8] Ding C, He X, “K-means Clustering via Principal Component Analysis”, Proceedings of the 21st International Conference on Machine Learning, ICML 2004.
- [9] Ding C, He X, “Linearized cluster assignment via Spectral Ordering”, Proceedings of the 21st International Conference on Machine Learning, ICML 2004.
- [10] Divakaran A., Pekar K.A., Radharkishnan R., Xiong Z., Cabasson R., “Video Summarization Using MPEG-7 Motion Activity and Audio Descriptors”, Video Mining, Rosenfeld A., Doermann D., DeMenthon D., October 2003, Kluwer Academic Publishers.
- [11] Faloutsos C., Ranganathan M., Manolopoulos Y., “Fast Subsequence Matching in Time-Series Databases”, ACM SIGMOD, Vol. 23 (2), pp. 419-429. June 1994.

- [12] Fink E., Pratt K.B., Gandhi H.S., "Indexing of Time Series by Major Minima and Maxima", IEEE International Conference on Systems, Man, and Cybernetics. pp. 2332-2335, 2003.
- [13] Hettich, S. and Bay, S. D. (1999). The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science.
- [14] Hongeng S, Nevatia R, Bremond F, "Video-based event recognition: activity representation and probabilistic recognition methods", Computer Vision and Image Understanding, Vol. 96 (2004), pp. 129-162.
- [15] Intille S.S., Bobick A.F., "Recognizing planned, multiperson action", Computer Vision and Image Understanding, Vol. 81 (2001), pp. 414-445.
- [16] Jeannin S., Divakaran A., "MPEG-7 Visual Motion Descriptors", IEEE Transactions on Circuits and Systems for Video Technology. Vol.11(6), June 2001, pp. 720-724.
- [17] Johansson G., "Visual Perception of Biological Motion and a Model for its Analysis", Perception and Psychophysics. Vol. 14(2), 1973, pp. 201-211.
- [18] Jolliffe, I.T., "Principal Component Analysis", Springer-Verlag, New York, 1986.
- [19] Kahveci T., Singh, "Variable Length Queries for Time Series Data", Proc. of 17th International Conference on Data Engineering 2001, Washington, D.C., USA. pp. 273-282.
- [20] Keogh E., Lin J., Fu A., "HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence", 5th IEEE International Conference on Data Mining (ICDM 2005). pp 226-233, Houston, TX, Nov. 27-30, 2005.
- [21] Lin J., Keogh E., Lonardi S., Chiu B., "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms", 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. San Diego, CA. June 13, 2003.
- [22] Ng A.Y., Jordan M.I., Weiss Y., "On Spectral Clustering: Analysis and an Algorithm", Advances in Neural Information Processing Systems, Vol. 14, 2001.
- [23] Pfoser D., Jensen C., Theodoridis Y., "Novel Approaches to the Indexing of Moving Object Trajectories", Proc. 26th Int'l Conference on Very Large Databases, VLDB'00, Cairo, Egypt, September 2000.
- [24] Porikli F., Haga T., "Event Detection by Eigenvector Decomposition using Object and Frame Features", International Conference on Computer Vision and Pattern Recognition, CVPR 2004.
- [25] Rao C., Yilmaz A., Shah M., "View-Invariant Representation and Recognition of Actions", International Journal of Computer Vision. Vol. 50 (2), 2002. pp. 203 - 226.
- [26] Rea N., Dahyot R., Kokaram A., "Semantic Event Detection in Sports through motion understanding", Proceedings of Conference on Image and Video Retrieval (CIVR) 2004, Dublin, Ireland, July 21-23, 2004.
- [27] Sahouria E., Zakhori A., "A Trajectory Based Video Indexing System For Street Surveillance", IEEE Int. Conf. on Image Processing, ICIP 1999.
- [28] Shim C.B., Chang J.W., "Efficient Similar Trajectory-based retrieval for moving objects in video databases", Conference on Image and Video Retrieval CIVR 2003, Lecture Notes in Computer Science LNCS 2728, pp. 163-173, 2003.
- [29] Sun X., Divakaran A., Manjunath B.S., "A Motion Activity Descriptor and Its Extraction in Compressed Domain", Second IEEE Pacific Rim Conference on Multimedia, LNCS 2195, pp. 450-457.
- [30] Yacoob Y., Black M. J., "Parameterized Modelling and Recognition of Activities", Computer Vision and Image Understanding, Vol. 73 (2), Feb. 1999. pp. 232 - 247.
- [31] Zacks J., Tversky B., "Event Structure in Perception and Cognition", Psychological Bulletin, Vol. 127(1), 2001, pp. 3 - 21.



Faisal Bashir received his B.S. Electrical Engineering degree from University of Engineering and Technology Lahore, Pakistan, and Ph.D. in Electrical and Computer Engineering from University of Illinois at Chicago, Chicago, in 2000 and 2006 respectively.

He worked at Delta Indus Systems, a machine vision software development company specializing in height measurement through structured light patterns from 2000 to 2001 before joining the Ph.D. program at UIC in Fall 2001. From 2001 to 2005 he was Research Assistant in the Multimedia Systems Lab, UIC. He worked as intern at Mitsubishi Electric Research Labs (MERL) in

Cambridge, MA during Summer 2005. He is currently working at MERL as Computer Vision Consultant.

Dr. Bashir was a recipient of the National Talent Scholarship from 1995 to 2000, and Provost Award for Graduate Research in Spring 2005. His research interests include: content-based multimedia indexing and retrieval, computer vision, machine learning, and statistical signal processing.



Ashfaq A. Khokhar received his M.S. in computer engineering from Syracuse University, in 1989 and Ph.D. in computer engineering from University of Southern California, in 1993. After his Ph.D., he spent two years as a Visiting Assistant Professor in the Department of Computer Sciences and School of Electrical and Computer Engineering at Purdue University. In 1995, he joined the Department of Electrical and Computer Engineering at the University of Delaware, where he first served as Assistant Professor and then as Associate Professor. In Fall 2000, Dr. Khokhar joined UIC in the Department of Computer Science and Department of Electrical and Computer Engineering where he currently serves as a full Professor.

Dr. Khokhar has published over 100 technical papers in refereed conferences and journals in the area of parallel computing, image processing, computer vision, and multimedia systems. He is a recipient of the NSF CAREER award in 1998. His paper entitled "Scalable S-to-P Broadcasting in Message Passing MPPs" has won the Outstanding Paper award in the International Conference on Parallel Processing in 1996. He is currently serving as the Program Chair of the 17th Parallel and Distributed Computing Conference (PDCS), 2004 and Vice Program Chair for the 33rd International Conference on Parallel Processing (ICPP), 2004. He has been nominated for IEEE fellow in 2006.

His research interests include: wireless and sensor networks, multimedia systems, datamining, and high performance computing.



Dan Schonfeld was born in Westchester, Pennsylvania, on June 11, 1964. He received the B.S. degree in Electrical Engineering and Computer Science from the University of California, Berkeley, California, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the Johns Hopkins University, Baltimore, Maryland, in 1986, 1988, and 1990, respectively. In August 1990, he joined the Department

of Electrical Engineering and Computer Science at the University of Illinois, Chicago, Illinois, where he is currently an Associate Professor in the Departments of Electrical and Computer Engineering, Computer Science, and Bioengineering, and Co-Director of the Multimedia Communications Laboratory (MCL) and member of the Signal and Image Research Laboratory (SIRL).

He has authored over 80 technical papers in various journals and conferences. He has served as consultant and technical standards committee member in the areas of multimedia compression, storage, retrieval, communications, and networks. He has also served as an associate editor of the IEEE Transactions on Image Processing on Nonlinear Filtering as well as an associate editor of the IEEE Transactions on Signal Processing on Multidimensional Signal Processing and Multimedia Signal Processing. He was a member of the organizing committees of the IEEE International Conference on Image Processing and IEEE Workshop on Nonlinear Signal and Image Processing. He was the plenary speaker at the INPT/ASME International Conference on Communications, Signals, and Systems. He has been elected as a Senior Member of the IEEE.

He has previously served as President of Multimedia Systems Corp. and provided consulting and technical services to various corporations including AOL Time Warner, Chicago Merchantile Exchange, Dell Computer Corp., Getco Corp., EarthLink, Fish & Richardson, Fitch, Even, Tabin & Flannery, IBM, Jones Day, Latham & Watkins, Mirror Image Internet, Motorola, Multimedia Systems Corp., nCUBE, NeoMagic, Nixon & Vanderhuy, PrairieComm, SmartSignal, Teledyne Systems, Touchtunes Music, Xcelera, and 24/7 Media. His current research interests are in signal, image, and video processing; video communications; video retrieval; video networks; image analysis and computer vision; pattern recognition; and genomic signal processing.