Self-Tuned Unsupervised Learning of Motion Trajectories

Shehzad Khalid and Andrew Naftel

***Abstract*—This paper presents a novel technique for clustering of video motion clips using coefficient-space representation of object trajectories. Trajectories are treated as time series and modelled using orthogonal basis function representation. Various function approximations have been compared including Chebyshev polynomials, Piecewise Aggregate Approximation, Discrete Fourier Transform (DFT) and Modified DFT (DFT-MOD). A novel framework (HSACT-SOM) is proposed for unsupervised learning of motion patterns without having prior knowledge of number and types of patterns hidden in datasets. Experiments, using simulated and complex real life trajectory datasets, demonstrate the superiority of our proposed HSACT-SOM based motion learning technique compared with other recent approaches. The comparison is performed in terms of the quality of clustering, based on cluster validity indices, and the number of clusters discovered in variety of datasets.**

***Index Terms*—Object trajectory, dimensionality reduction, trajectory clustering, event mining.**

I. Introduction

TECHNIQUES for understanding video object motion activity are becoming increasingly important. Object motion plays the key role in the behaviour analysis. Behaviour, in context of visual surveillance, implies purpose of motion such as running, walking from one location to another or loitering etc. Unsupervised learning of patterns in motion data is considered to have a pivatol position in motion data analysis. These techniques are essential for the development of next generation actionable intelligence surveillance systems.

The literature on trajectory-based motion understanding and pattern discovery is less mature but advances using Learning Vector Quantization (LVQ)[29], Self-Organising Maps (SOMs) [24][34], Hidden Markov Models (HMMs) [9][6], and fuzzy neural networks [25] have all been reported. Most of these techniques attempt to learn high-level motion behaviour patterns from sample trajectories using discrete point sequences as input to a machine learning algorithm. Trajectory representation is crucial in this respect. Using a point vector sequence, it is difficult to efficiently learn motion patterns because of high dimensionality.

Much of the earlier research focus in motion analysis has been on representation schemes for indexing and retrieval [7][16][17][5][22][8][38][28][31]. This work presupposes the existence of some low-level tracking scheme for reliably extracting object-based motion trajectories. A description of relevant tracking algorithms is not within the scope of this paper but recent surveys can be found in [4][21][23][32]. Related work within the data mining community on approximation schemes for modelling time series data is highly relevant to the parameterisation of motion trajectories. Trajectories are defined as set of points representing the ordered observation of location of moving object taken at different points in time. Trajectories can therefore be represented as time series data which makes the indexing techniques for time series applicable to motion data. However, computer vision researchers have been slow to realize the potential of this work. For example, Discrete Fourier Transforms (DFT) [20], Discrete Wavelet Transforms (DWT) [15], Adaptive Piecewise Constant Approximations (APCA) [30], and Chebyshev polynomials [13] have been used to conduct similarity search in time series data.

In this paper, we develop a novel technique for automatic motion learning without any manual labelling of the training dataset. Trajectories are modelled using orthogonal basis function representation which is used as an input feature vector to a novel HSACT-SOM learning algorithm for learning of motion patterns without having any information about the number and type of patterns hidden in training data. Our approach is significantly different from most of the existing work on behaviour learning which mainly took a supervised learning approach where complete information about normal motion patterns is available [9][10][42]. Supervised learning techniques have their place but the applications are appropriate in more constrained problems e.g. vehicle motion analysis, sign language recognition. However, in majority of surveillance applications, complete information about normal motion patterns is not available.

The remainder of the paper is organized as follow. We review some relevant background material in section 2. In section 3 we present some function approximation approaches to trajectory representation. In section 4, a hierarchical semi-agglomerative learning algorithm has been proposed for unsupervised learning of patterns from unclassified datasets. Experiments have been performed to show the effectiveness of proposed system for trajectory-based learning of motion patterns as compared to competitive techniques. These experiments are reported in section 5. The paper concludes with a discussion and proposals for further work.

II. Background and related work

In this section, we provide a survey of recent work on trajectory-based representation and learning. Motion trajectory descriptors are known to be useful candidates for compressed representation of video object motion. Previous work

S. Khalid is with the Department of Computer Science and Engineering, Bahria University, Islamabad, 44000, Pakistan e-mail: (shehzad_khalid@hotmail.com).

Andrew Naftel is with the Department of Computer Science, University of Manchester, Manchester, M60 1QD, UK e-mail: (A.Naftel@manchester.ac.uk).

has sought to represent moving object trajectories through piecewise linear or quadratic interpolation functions [16][27], motion histograms [5], discretised direction-based schemes [17][37][38] or basis functions [3][15][13]. Spatiotemporal representations using piecewise-defined polynomials were proposed by Hsu [22], although consistency in applying a trajectory-splitting scheme across query and searched trajectories can be problematic. Affine and more general spatiotemporally invariant schemes for trajectory retrieval have also been presented [7][8][28]. The importance of selecting the most appropriate trajectory model and similarity search metric has received relatively scant attention [31].

Learning of patterns from trajectory data to extract high level information, for behaviour profiling, has gained interest quite recently. The work on motion learning has either used probabilistic models such as HMMs [26] or discrete point-based trajectory flow vectors (PBF) [29][24][25] as a means of learning patterns of motion activity. An agglomerative clustering algorithm based on the Longest Common Subsequence (LCSS) approach is proposed in PBF space [12][41] for grouping similar motion trajectories. The problem with PBF vector-encoded trajectory representation is the heavy computational burden making prospects for online learning of motion patterns remote.

Earlier work rely upon labelled training data for model training [9][10][42]. There exists some work on learning from unclassified training data such as [2][12][29][34][35][39][40][41]. A number of eigenspace clustering techniques have been proposed recently [18][33]. However, these approaches normally require known number of clusters. Given an unclassified dataset, the number of motion classes are normally unknown. Some approaches, based on spectral clustering, attempts to approximate the number of clusters [11][36]. These approaches, as outlined in section 5, are not scalable to the number of options from which to search the correct number of clusters. They are also unable to give a reliable estimate of the number of groupings hidden in unlabelled training data.

The contribution of this paper is to propose a novel mechanism for learning of motion patterns without having prior knowledge about the number and type of patterns hidden in datasets. Trajectory clustering is carried out in coefficient feature space that results in efficient discovery of patterns of similar object motion behaviour.

III. Feature space representation of trajectory

A moving object registers its location in the three dimensional space corresponding to the *x*- and *y*- axes projection of the object's centroid at each instant of time. Without loss of generality, we consider the projection of a moving object trajectory T in a 2-D image plane, where

$$T = [(x_1, y_1, t_1), (x_2, y_2, t_2), ..., (x_n, y_n, t_n)] \qquad (1)$$

where n is the sequence length. With this representation, trajectories are now treated as motion time series.

We consider four alternative techniques to achieve dimensionality reduction of trajectories including Chebyshev (CS), Piecewise Aggregate Approximation (PAA), Discrete Fourier Transform (DFT) and Modified DFT (DTF-MOD). The performance of the four different trajectory representation schemes is compared experimentally in section 5.

A. Chebyshev polynomials

The projection of spatiotemporal trajectory in (x_i, t_i) space can be approximated by a function $P_{m_x}(t)$ expressed as a weighted sum of Chebyshev polynomials $C_k(t)$ up to degree *m*, defined as:

$$P_{m_x}(t) \approx \sum_{k=0}^{m} a_k C_k \qquad (2)$$

where $C_k(t) = \cos(k \cos^{-1}(t))$ and

$$a_0 = \frac{1}{m}\sum_{k=1}^{m} P_{m_x}(t_k), \;\; a_i = \frac{2}{m}\sum_{k=1}^{m} P_{m_x}(t_k) C_i(t_k) \qquad (3)$$

for $t \in [-1, 1]$ and $i = 1, ..., m$. Similar expressions $(b_0, ..., b_m)$ can be obtained for projection in (t_i, y_i) space. Thus the motion trajectories are represented by a feature vector of Chebyshev polynomial coefficients

$$\tilde{F}_{CS} = [a_0, ..., a_m, b_0, ..., b_m] \qquad (4)$$

B. Piecewise Aggregate Approximation

1-dimensional projection of trajectory in (x_i, t_i) space can be approximated by segmenting the data sequences into equi-length sections and calculating the mean values of this section. Let *n* be the length of the time series and *m* be the dimensionality of the transformed space that is required such that $1 \leq m \leq n$. It is assumed that *m* is a factor of *n*. Mathematically, a time series X of length *n*, $X = x_1, x_2, ..., x_n$, is represented in the *m*-dimensional feature space as $\overline{X} = \overline{x}_1, \overline{x}_2, ..., \overline{x}_m$ where

$$\overline{x}_i = \frac{m}{n} \sum_{j=\frac{n}{m}(i-1)+1}^{\frac{n}{m}i} x_j \qquad (5)$$

C. Discrete Fourier Transform (DFT)

Without loss of generality, a spatiotemporal trajectory $(x_i, t_i), i = 0, ..., n-1$ can be considered as a 1-D time series x_i if $t_i = i$. The *n*-point Discrete Fourier Transform of x_i, defined to be a sequence $\{X_f\}$ of *n* complex numbers $(f = 0, ..., n-1)$, is given in eq. (6). A similar expression can be defined for y_i given in eq. (7).

$$X_f = \frac{1}{\sqrt{n}} \sum_{t=0}^{n-1} x_t exp(-j2\pi ft/n) \quad f = 0, 1, ..., n-1 \quad (6)$$

$$Y_f = \frac{1}{\sqrt{n}} \sum_{t=0}^{n-1} y_t exp(-j2\pi ft/n) \quad f = 0, 1, ..., n-1 \quad (7)$$

where j is the imaginary unit $j = \sqrt{-1}$, X_f and Y_f are complex numbers with the exception of X_0 and Y_0 which are real numbers. Typically, the DFT sequence is truncated after *m* terms, $f = 0, ..., m-1$. In this case, the motion trajectory feature vector consists of $2m + 2$ entries (from real and imaginary parts) for each time series in x_i and y_i.

D. Discrete Fourier Transform-Modified

Discrete Fourier Transform-Modified (referred to as DFT-MOD) is an extension of DFT. DFT-MOD is generated by augmenting the DFT-based feature vector with extra information regarding the start/end points and length of the trajectory. These important information are not modelled correctly by DFT as it selects only top few DFT coefficients. Truncating most of the coefficients results in discarding the length information and distorts the starting position. All these factors may contribute to the fall-off in retrieval accuracies using simple DFT based dimensionality reduction, where starting point and duration of motion are important features for distinguishing different trajectories. Let (x_0, y_0) is the starting point and n is the length of trajectory, DFT-MOD based feature space representation of trajectory is represented as

$$\tilde{F}_{DFT-MOD} = [n, x_0, X_f, y_0, Y_f] \tag{8}$$

where X_f and Y_f are the DFT based feature space representation of x_i and y_i time series as obtained using eq. (6) and (7) respectively.

IV. Learning of Patterns in Coefficient-Space Representation

In this section, a novel unsupervised learning algorithm has been proposed to identify patterns in motion trajectory dataset. The motivation of proposed clustering approach is to effectively identify the right number of clusters without having multiple passes through the learning process for different number of cluster options. The proposed clustering mechanism is a cooperative learning algorithm that combines Self Organizing Maps (SOM) with Hierarchical Semi-Agglomerative ClusTering (HSACT). The architecture chosen for the SOM consists of a single layer of input neurons connected directly to a single 1-dimensional layer of output neurons. The network topology of our SOM architecture is shown in Fig. 1. Here, *F* is the input feature vector and *W* is a weight vector associated to each output neuron. SOM component is responsible for extracting fine groupings in trajectory dataset. HSACT component uses these fine clusters to generate coarse clusters and, in the process, discovering the actual number of groupings in the trajectory dataset. It is an iterative process and at each iteration, the most similar clusters are merged. The closest pair of clusters, indexed by *a* and *b*, is selected as

$$a, b = arg\,min \|W_i - W_j\| \quad \forall i, j \wedge i \neq j \tag{9}$$

where W_i and W_j are the cluster centres associated to clusters *i* and *j* and $\|.\|$ is the Euclidean distance metric. Motion trajectory datsets, from surveillance environments, are extracted by tracking objects from video sequences. These datasets normally suffer from the problem of perspective effects due to the presence of depth in scene. This results in difficulties to differentiate between events that are occurring farther from the scene and will be over sensitive for the events that are occurring closer to the camera. To avoid these problems, we use ground plane homography to map image coordinates to ground plane, given the availability of calibration information. Given a point set X_i in image plane and a corresponding point set X_i' in ground plane, we need to find a homography H such that

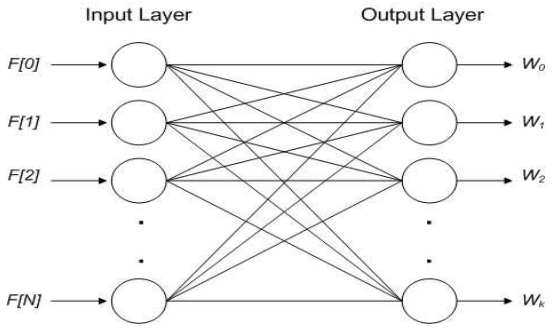


Fig. 1. SOM network architecture used for trajectory clustering

$$X_i' = HX_i \tag{10}$$

The homography matrix H is estimated using the method outlined in [43].

Bearing in mind the nature of motion trajectory data and the drawbacks associated with its high dimensionality, learning of motion patterns is done in coefficient feature space representation of trajectory.

A. Learning Algorithm

The learning algorithm, for unsupervised learning of motion patterns from corrupted training data, comprises the following steps:

1) Initialize the SOM network with greater number of output neurons than the number of clusters we want to identify in the motion trajectory data using:

$$\#_{output} = \begin{cases} \xi & if\, \xi < 100 \\ 100 & otherwise \end{cases} \tag{11}$$

where

$$\xi = size(DB)/4 \tag{12}$$

2) Estimate a single multivariate Gaussian (PDF) form the training data *DB* as:

$$PDF = \frac{1}{\sqrt{2\pi\Sigma}} exp\left[-\frac{(W-\mu)^2}{2\Sigma}\right] \tag{13}$$

where $X \in DB$, μ is the mean and Σ is the covariance estimate associated to DB. Generate $\#_{output}$ samples from the $PDF\ N(\mu, \Sigma)$ and use them to initialize the weight vectors associated with each of the output neurons.

3) Determine the winning output node *k* (indexed by *c*) such that the Euclidean distance between the current input vector *B* and the weight vector W_k is a minimum amongst all output neurons, given by the condition

$$c = arg\ min\ _k \|F - W_k(t)\| \qquad \forall k \tag{14}$$

4) Train SOM network by adjusting the weight vectors so that it starts representing the trend of the data. If *c* is the winning output node, then a subset of the weights

constituting a neighbourhood centred around node *c* are updated using

$$W_k(t+1) = W_k(t) + \alpha(t)\zeta(k,c)(F - W_k(t)) \quad (15)$$

where $\zeta(k,c) = exp(-|r_k - r_c|^2/2\delta^2)$ is a neighbourhood function that has value 1 when $k = c$ and falls off with distance $|r_k - r_c|$ between nodes k and c in the output layer, δ is the neighbourhood radius that is gradually decreased over time, $\alpha(t)$ is the learning rate of SOM and t is the training cycle index.

5) Decrease the learning rate $\alpha(t)$ exponentially over time using:

$$\alpha(t) = 1 - e^{t - t_{max}} \quad (16)$$

where t_{max} is the maximum number of training iterations.

6) Decrease the neighbourhood size $\delta(t)$ exponentially with training iterations as:

$$\delta(t) = \lceil \delta_{init}(1 - e^{t - t_{max}}) \rceil \quad (17)$$

where δ_{init} is the neighbourhood size at the start of learning process.

7) Repeat steps 3-6 for all the training data. Complete training data is repeatedly passed through the network till two consecutive passes produces identical classification for each of the training sample (convergence).

8) Ignore output neurons with no training data associated to them.

9) Calculate Cluster Validity Index (*CVI*) to check the quality of current state of cluster. For the current number of clusters, the mathematical expression of CVI is given as:

$$CVI(k) = \left(\frac{1}{k} \times \frac{E_1}{E_k} \times D_k\right)^2 \quad (18)$$

$$E_k = \sum_{j=1}^{k} \sum_{X \in \Gamma_j} \|X - W_j\| \quad (19)$$

$$D_k = max_{i,j=1}^{k} \|W_i - W_j\| \quad (20)$$

where k represents number of clusters, X represents a sample training data associated to valid clusters and W_j represents weight vector associated to cluster Γ_j. In eq. (18), the factor $\frac{1}{k}$ will increase CVI index as k is increased. On the other hand, $\frac{E_1}{E_k}$ increases CVI index as E_1 is a constant and E_k decreases with increase in k. The third factor D_k will increase with the value of k. These three factors tend to balance each other nicely. Values of k, resulting in higher values of CVI index, indicate better clustering.

10) Identify the closest pair of cluster. This is done by computing the $k \times k$ proximity matrix P as

$$P = [dist(i,j)], \qquad \forall i,j \quad (21)$$

where

$$dist(i,j) = [(W_i - W_j)^T (W_i - W_j)]^{1/2} \quad (22)$$

where W_i and W_j represents the weight vectors for output nodes *i* and *j*. The closest pair of output nodes (i,j) (indexed by (a,b)) is selected such that the distance between the weight vectors W_i and W_j is a minimum amongst all possible pairs of output neurons, given by the condition

$$(a,b) = arg\ min_{(i,j)} dist(i,j), \quad \forall i,j \wedge i \neq j \quad (23)$$

After finding the most similar pair of clusters, the two clusters are merged into one using

$$W_{ab} = \frac{mW_a + nW_b}{m+n} \quad (24)$$

where *m*, *n* are the number of sample trajectories mapped to clusters *a* and *b* respectively.

11) If information about the number of patterns, hidden in the training data, is known, iterate through step 10 till the number of clusters get equivalent to the number of patterns present in the data.

12) If no information about the number of patterns in the training data is available, iterate through steps 9-10 till the number of clusters get equivalent to 1. Algorithm keeps track of cluster memberships for different number of clusters. *CVI* value is computed after each cluster merge and the number of clusters corresponding to highest *CVI* value is selected to be the approximate number of groupings hidden in the training data.

V. Experimental Results

This section analyzes the performance of proposed HSACT-SOM based algorithm for unsupervised learning of motion patterns hidden in motion trajectory dataset.

A. Experimental Datasets

The experiments are conducted using seven different synthetic and real life motion trajectory datasets. These include Australian Sign Language (ASL), CAV-FRNT, CAV-TRK, LAB, Cylinder-Bell-Funnel (CBF), SIM_3 and SIM_5 dataset. The characteristics of these datasets are summarized in Table I.

B. Peformance evaluation of dimensionality reduction techniques

The performance of four different trajectory representation schemes has been compared. The purpose of the experiment is to investigate the robustness of various dimensionality reduction mechanisms to the real life problem of noise and occlusion in motion data. Selection of appropriate trajectory matching mechanism is critical to the performance of the proposed techniques for unsupervised learning of motion patterns. Experiments have been performed using CAV-FRNT dataset.

The dataset is corrupted with noise by adding a uniform random noise, scaled by some factor, to every point in the original trajectory dataset. Let *S* represents an original motion trajectory dataset, a noise corrupted dataset S_C is produced by adding the term $w * U[-0.5, 0.5] * rangeValues$ to each (x,y) coordinate in the original set *S*, where w is a scaling factor, $U[-0.5, 0.5]$ is uniform random noise on the interval

Dataset	Description	# of trajectories	Extraction method	Labelled (Y/N)
SIM_3/SIM_5	Simulated datasets comprising of two dimensional coordinates generated from Gaussian distributions to form 3 or 5 clusters.	arbitrary	Simulation.	Y
CBF	A 3 class labelled dataset normally used for benchmarking time series data mining algorithm.	arbitrary	Simulation.	Y
CAV-FRNT	A manually annotated video sequences of moving people from corridor view in a shopping centre. Object tracking coordinates are generated using interactive program and stored in XML files.	126	Parsing XML files containing motion coordinates.	N
CAV-CORR	A manually annotated video sequences of moving people from corridor view in a shopping centre. Object tracking coordinates are generated using interactive program and stored in XML files.	126	Parsing XML files containing motion coordinates.	N
LAB	Realistic dataset generated in the laboratory controlled environment for testing purposes. Trajectories can be categorized into 4 classes.	152	Tracking moving object and storing motion coordinates.	Y
ASL	Trajectories of right hand of signers as different words are signed. Dataset consists of signs for 30 different word classes with 27 samples per word.	810	Extracting (x,y) coordinates of the mass of right hand from files containing complete sign information.	Y

TABLE I
OVERVIEW OF DATASETS USED FOR EXPERIMENTAL EVALUATION.

$[-0.5, 0.5]$, and *rangeValues* is the range on *x* and *y* coordinates. Different levels of noise are simulated using difference values for noise scaling factor *w*. Feature space representation of trajectories in *S* and S_C are generated separately using CS, DFT, DFT-MOD and PAA. Each corrupted trajectory in S_C is then selected as an example query Q_C and we search for its closest match *Q* in the original dataset *S*. This is defined by $argmin_{Q \in S}\, ED(Q_C, Q)$. A set of rankings $\forall Q_C \in S_C$ is produced. In the absence of noise and occlusion, the closest match to Q_C should be its corresponding uncorrupted version in *S* which produces a rank value of unity. For ease of comparison we record the proportion of times (as a percentage) the query trajectory is ranked correctly as unity when taken over all S_C. This is repeated for different number of coefficients in CS, PAA, DFT and DFT-MOD and for various values of scale factor *w*. The results are summarised in Fig. 2. It is apparent that DFT-MOD and PAA performs better than other dimensionality reduction mechanisms followed by DFT. CS does not give good retrieval accuracies and its performance degrades significantly with increasing noise levels as compared to PAA, DFT-MOD and DFT. The retrieval experiments are now repeated but this time the trajectory dataset is corrupted with occlusion. The occlusion is simulated by replacing the occluded sub-sequence with the predicted location of object during occlusion, based on the motion trend of the object before the occlusion. Let $(\Delta x, \Delta y)$ be the difference between two points immediately before the start of the occluded subsequence. Assume (x_p, y_p) is the first point in the occluding sequence and *k* is the subsequence length. Then

$$(\hat{x_i}, \hat{y_i}) = (x_{i-1} + \Delta x,\ y_{i-1} + \Delta y)\ , \quad i = p, ..., p + k \qquad (25)$$

where

$$\Delta x = x_{p-1} - x_{p-2} \qquad (26)$$

and

$$\Delta y = y_{p-1} - y_{p-2} \qquad (27)$$

The occluded subsequence is then replaced with the estimated vector in eq. (25). Average retrieval accuracies, obtained using different types of representation schemes with varying numbers of function coefficients and different levels of occlusion are shown in Fig. 3. The retrieval accuracies were averaged over 10 random subsequence removals to reduce bias due to choice of starting position of occluding subsequence. The results from Fig. 3 show that CS and DFT-MOD gives good retrieval accuracies. PAA, in contrast to its performance in the presence of noise, does not give good retrieval accuracies in the presence of occlusion.

Overall, the experiments demonstrate that DFT-MOD is the best choice of feature space representation. It is fairly robust to varying amounts of noise and occlusion, compared to PAA whose performance degrades with occlusion. Therefore, it is recommended to use DFT-MOD as a feature space representation in unsupervised learning of motion trajectory patterns.

C. Experiment 2: Learning motion patterns using HSACT-SOM

The purpose of experiment is to evaluate the suitability of HSACT-SOM algorithm for clustering in the specific domain of motion trajectory datasets. The experiment also aims to investigate the ability of HSACT-SOM algorithm to identify the correct number of clusters in unclassified dataset when

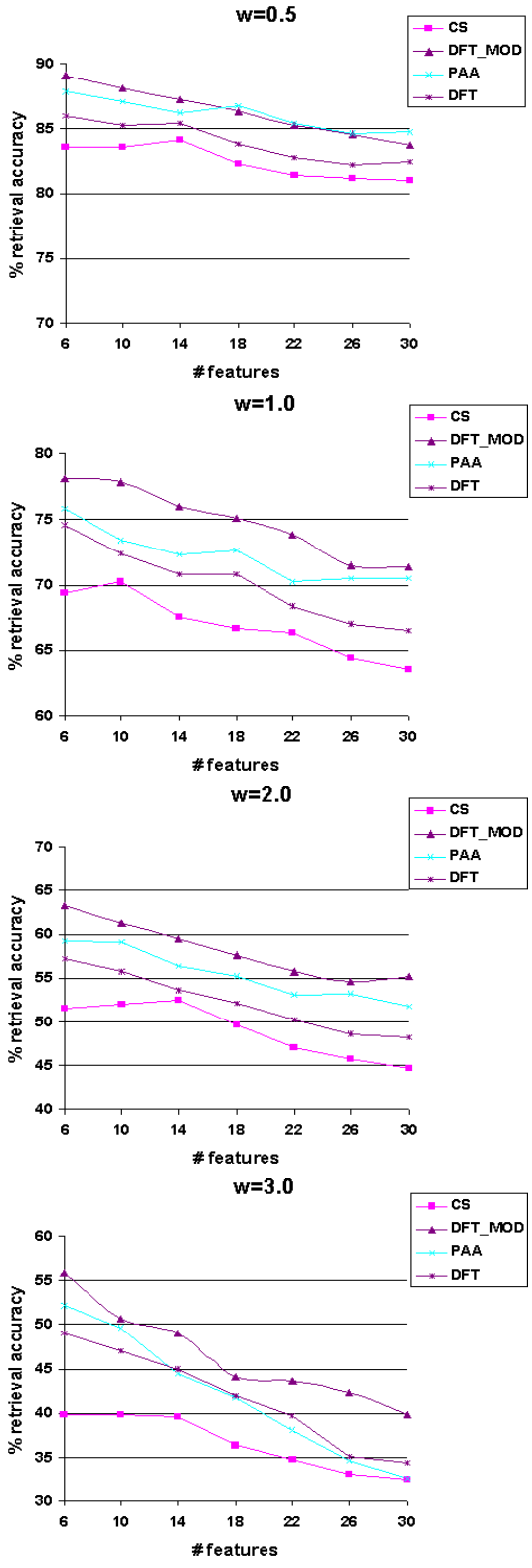


Fig. 2. Effect of scaled uniform noise on trajectory retrieval accuracy using CAV-FRNT dataset

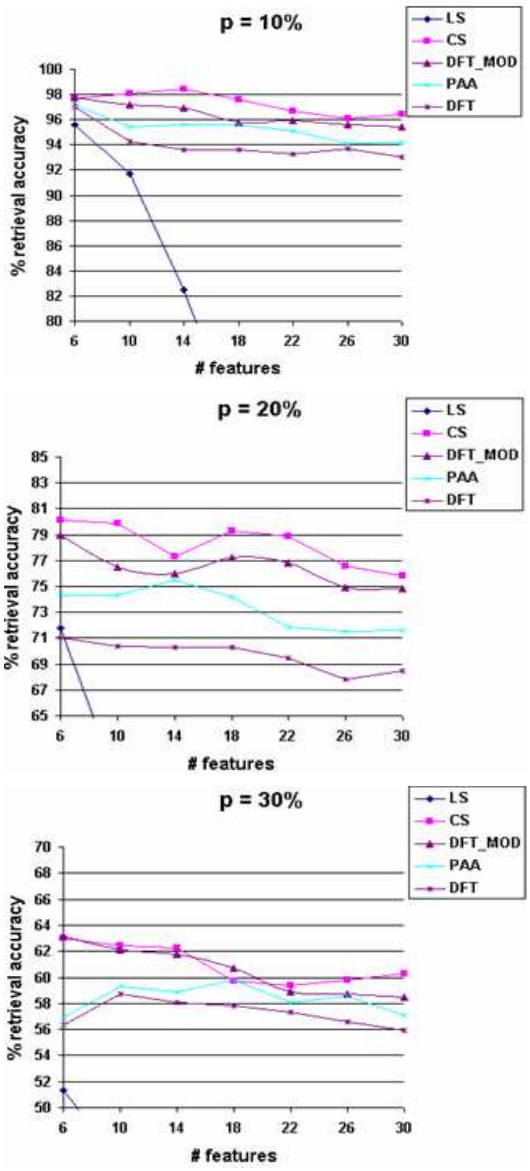


Fig. 3. Effect of different lengths of occluding sub-sequences on trajectory retrieval accuracy using CAV-FRNT dataset

no such information is available. The experiment has been conducted on real life CAV-CORR and CAV-FRNT motion datasets, as shown in Fig. 4. CAV-CORR dataset is also suffering from the problem of perspective effects due to the presence of depth in CAV-CORR scene. To avoid this problem, we use ground plane homography to map image coordinates to ground plane. The calibration is available from [1]. Trajectories are modelled using DFT-MOD based coefficient feature vectors. We assume $m = 4$ in eq. (6) and eq. (7) based on retrieval experiments as outlined in previous section. Patterns are learned using HSACT-SOM based learning algorithm. Training samples from the dataset are passed through HSACT-SOM network one by one and the network is trained for $t_{max} = 10000$ number of iterations. This results in identification of fine clusters in unclassified dataset. Correct number of clusters is then identified by merging the most similar patterns and calculating the cluster validity index

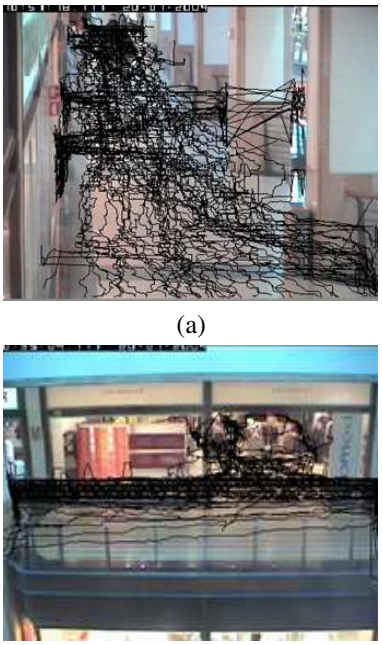


Fig. 4. Background scene with overlaid trajectories from (a) CAV-CORR dataset (b) CAV-TRK dataset.

CVI, as specified in eq. (18), after each merge. The number of clusters corresponding to highest *CVI* value is selected. The clustering results obtained, by applying the HSACT-

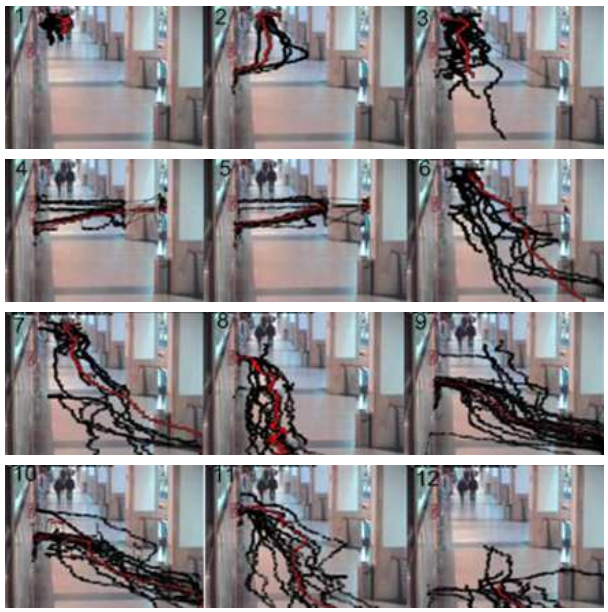


Fig. 5. Motion trajectory clustering of CAV-CORR dataset after applying homography.

SOM methodology, on CAV-CORR dataset is shown in Fig. 5. The red trajectory in each class represents the trajectory that is closest to the class mean. Qualitatively, similar motion trajectory patterns appear to have been grouped together quite successfully. The experiment is also repeated using CAV-FRNT dataset. The learning of patterns obtained by applying the proposed approach on CAV-FRNT dataset is shown in Fig. 6. Clustering results on CAV-CORR and CAV-FRNT dataset provide support to the claim that HSACT-SOM is an effective approach for learning patterns in complex real-life motion datasets.

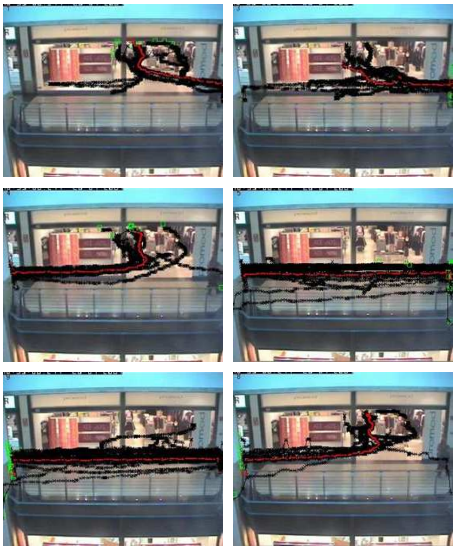


Fig. 6. Learning of motion patterns from CAV-FRNT dataset.

D. Comparison of HSACT-SOM with Spectral Clustering

The purpose of this experiment is to compare the performance of proposed HSACT-SOM algorithm with an adaptation of spectral clustering as proposed in Porikli *et al.* [36], for clustering vehicle trajectories and recently used in [10][11] for motion sub-trajectories clustering. Comparative evaluation is provided in terms of the quality of clustering, detection of correct number of clusters and response time.

The experiment has been conducted on simulated SIM_3, SIM_5, CBF and real life trajectory based CAV-FRNT, LAB and ASL datasets. Samples from the datasets are modelled using DFT-MOD based coefficient feature vectors except for SIM_3 and SIM_5 where the 2D-points are used as it is for learning purposes. Learning of patterns is done and three different cluster validity indices are employed to evaluate the quality of clustering results obtained by HSACT-SOM and spectral clustering. These include Davies-Bouldin (*DB*) index [18], *Dunn*'s index [19] and Calinski-Harabasz (*CH*) index [14].

The number of clusters identified by HSACT-SOM and spectral clustering along with the quality of clustering, as indicated by three different cluster validity indices, are presented in Table II. For *DB* index, lower values indicates better quality of clustering whereas for *Dunn* and *CH* index, higher values indicate better clustering. The number of clusters identified, using each of the clustering algorithm, is also provided in Table II. For CBF and SIM_5 datasets, clustering algorithms are unable to identify a unique number of clusters hidden in datasets. This phenomenon is explained by the fact that data samples for simulated datasets are generated each time they are used for experimental evaluation. This may result in slightly different orientation in the data thus resulting in

Datasets		HSACT-SOM				Spectral			
	Ground Truth	# of clusters	DB	CH	Dunn	# of clusters	DB	CH	Dunn
CAV-FRNT	6	6	0.46	307.07	2.70	7	0.67	282.3	2.4
LAB	4	4	0.67	347.7	2.91	8	1.07	370.5	1.87
ASL_4 (alive, all, crazy, drink)	4	4	1.19	46.3	1.59	7	1.69	21.3	0.89
ASL_6 (alive, all , crazy, drink, god, go)	6	6	1.35	48.4	1.52	8	1.42	32.7	0.91
CBF	3	2/3/4	1.19	94.9	1.64	10/11/12	1.89	5.59	0.68
SIM_3	3	3	0.41	1643.4	3.04	2	0.46	1173.7	2.98
SIM_5	5	5	0.38	2088.7	2.57	4/5	0.51	1463.4	1.85

TABLE II
COMPARISON OF HSACT-SOM AND SPECTRAL CLUSTERING BASED ON THE NUMBER AND QUALITY OF CLUSTERS (USING VARIOUS CLUSTER INDICES) ON DIFFERENT DATASETS

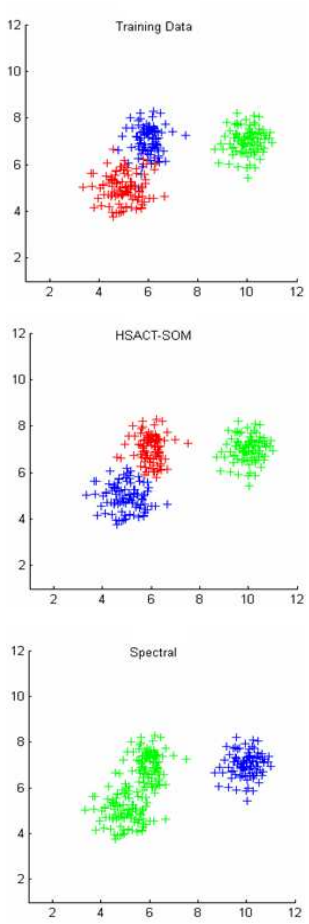
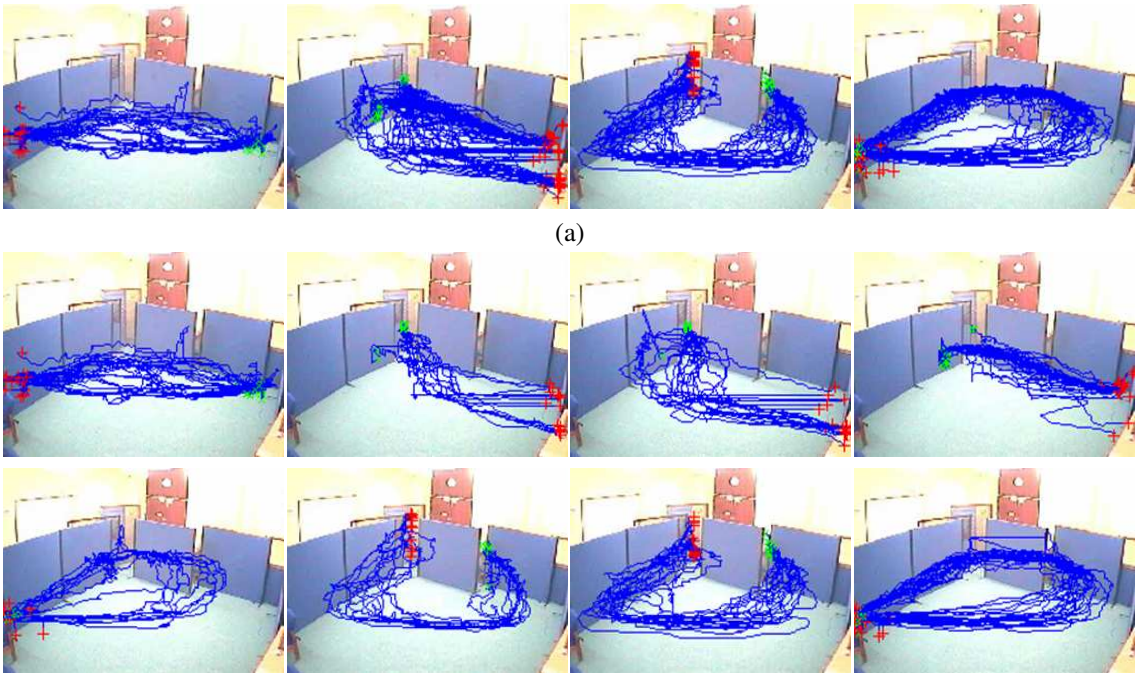


Fig. 7. Clustering of SIM_3 dataset using HSACT-SOM and spectral clustering.

different number of clusters. Based on these results, we can see that proposed HSACT-SOM algorithm performs consistently better then spectral clustering, for all the datasets that have been used in the experiment. Moreover, the number of clusters identified using our proposed approach is consistent with the number of groupings hidden in the dataset. On the other hand, spectral clustering is not able to identify the correct number of clusters. This is verified by matching the numbers of cluster, identified using HSACT-SOM and spectral clustering, with the actual number of groupings hidden in classified datasets such as SIM_3, SIM_5, ASL and LAB datasets.

Effectiveness of proposed HSACT-SOM algorithm, as compared to spectral clustering, is demonstrated graphically for SIM_3, and LAB datasets. Fig. 7 shows clustering results for SIM_3 dataset. The blue and green clusters, in trainind data, have covariance matrix $\begin{pmatrix} 0.2 & 0 \\ 0 & 0.3 \end{pmatrix}$ and mean for left cluster at (6,7) and mean for right cluster at (10,7). The red cluster has covariance matrix $\begin{pmatrix} 0.7 & 0 \\ 0 & 0.4 \end{pmatrix}$ and mean at (7,4). Comparing clustering results with the ground truth for SIM_3 dataset shows that HSACT-SOM identifies the right number of clusters. On the other hand, spectral clustering has merged two slightly overlapping clusters and is not able to identify the right number of groupings hidden in SIM_3 dataset. Clustering results for LAB dataset is presented in Fig. 8. Fig. 8(a) shows the clustering results obtained using proposed HSACT-SOM and Fig. 8(b) shows the results obtained using spectral clustering. Green marks in the figure identify the starting point of the trajectories whereas red marks identify the ending points. As LAB dataset is a classified dataset which contains trajectories belonging to one of the four planned motion behaviour patterns, HSACT-SOM generates the right number of clusters. On the other hand, spectral clustering manages to identify much finer clusters. Comparison of HSACT-SOM and spectral clustering is now provided by investigating the scalability of these algorithms to the number of options from which to identify the correct number of clusters present in the dataset. We have implemented these algorithms using MATLAB 7 and running times are noted on an Intel Pentium IV 1.73 GHz machine with 504 MB of RAM. Experiment has been conducted on SIM_3 dataset and the response time of clustering algorithms, for different number of candidate clusters, are presented in Table III. It is evident from the results in Table III that our proposed approach is scalable to number of options from which to select the right number of patterns hidden in the dataset. This is one of the important advantages of incorporating HSACT component with SOM based learning that the proposed clustering algorithm takes same amount of time for any number of cluster options. On the other hand, spectral clustering exhibits increasing time complexity with increasing number of cluster options.



(a)

(b)

Fig. 8. Clustering results for LAB dataset using (a) HSACT-SOM (b) spectral clustering.

# of cluster options	Response Time of SOM (sec.)	Response Time of Spectral (sec.)
2	5.61	4.16
4	5.56	9.94
6	5.61	14.70
8	5.67	20.01
10	5.71	28.85
12	5.62	36.51
14	5.60	45.98
16	5.57	52.43
18	5.78	59.64
20	5.77	74.36

TABLE III
COMPARISON OF HSACT-SOM AND SPECTRAL CLUSTERING BASED ON THE RESPONSE TIME FOR DIFFERENT NUMBER OF CLUSTER OPTIONS

VI. DISCUSSION AND CONCLUSIONS

In this paper, we have provided a detailed discussion on the topic of unsupervised learning of motion patterns without having prior knowledge of number and types of patterns hidden in datasets. A novel technique, based on HSACT-SOM, has been proposed for learning of trajectory-based object motion sequences. Trajectory clustering is carried out in coefficient feature space to discover patterns of similar object motion behaviour. We have compared the performance of different dimensionality reduction techniques. DFT-MOD has been selected as dimensionality reduction mechanism as it gives overall best results in real time situation where motion data is susceptible to high level of noise and occlusion. Mapping trajectories from point sequence vectors to DFT-MOD coefficient feature space improves learning efficiencies.

Experimental results are presented to demonstrate the effectiveness of learning of motion patterns using HSACT-SOM in DFT-MOD coefficient feature subspace. SOM, along with hierarchical semi-agglomerative clustering technique (HSACT), better preserves the topology of original point sequence space. It automatically determines the number of hidden patterns by evaluating the quality of clustering with different number of patterns and selecting the one which gives best result. HSACT-SOM yields better clustering results as compared to spectral clustering and consistently determines the right number of patterns hidden in various datasets. It is also scalable to number of options from which to select the right number of patterns hidden in the dataset and exhibit a consistent time complexity with the increasing number of cluster options. Our techniques have been validated using variety of simulated and real-life video tracking data.

A possible drawback of this approach is for catering the presence of anomalies in training data itself. Sometimes, in visual surveillance, there exist some movements which are not normal and represent an abnormal activity. These abnormal activities can affect the quality of clusters formed. In our future work, we aim to address the problem of learning activity patterns without having any affect of the presence of anomalies in training data.

REFERENCES

[1] Caviar test sequences [online]. available:. http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/.

[2] Christophe Abraham, Pierre-Andre Cornillon, Eric Matzner-Lober, and Nicolas Molinari. Unsupervised curve clustering using b-splines. In *Scandinavian Journal of Statistics*, volume 30 of *3*, pages 581–595, September 2003.

[3] Rakesh Agarwal, Christos Faloutsos, and Arun Swami. Efficient similarity search in sequence databases. In *4th International Conference of Foundations of Data Organization and Algorithms*, pages 69–84, Evanston, Illinois, USA, October 1993.

[4] J. Aggarwal and Q. Cai. Human motion analysis: A review. In *Computer Vision and Image Understanding*, volume 73, pages 428–440, 1999.

[5] Z. Aghbari, K. Kaneko, and A. Makinouchi. Content-trajectory approach for searching video databases. In *IEEE Transanction on Multimedia*, volume 5 of *4*, pages 516–531, December 2003.

[6] Faisal I. Bashir, Ashfaq A. Khokhar, and Dan Schonfeld. View-invariant motion trajectory based activity classification and recognition. In *ACM Multimedia Systems, special issue on Machine Learning Approaches to Multimedia Information retrieval*, pages 45–54, 2006.

[7] Faisal I. Bashir, Ashfaq A. Khokhar, and Dan Schonfield. Segmented trajectory based indexing and retrieval of video data. In *IEEE International Conference on Image Processing*, volume 3, pages II– 623–6, Barcelona, Spain, September 2003.

[8] Faisal I. Bashir, Ashfaq A. Khokhar, and Dan Schonfield. A hybrid system for affine-invariant trajectory retrieval. In *Proc. MIR'04*, pages 235–242, 2004.

[9] Faisal I. Bashir, Ashfaq A. Khokhar, and Dan Schonfield. Hmm based motion recognition system using segmented pca. In *IEEE International Conference on Image Processing*, pages 1288–1291, Genova, Italy, Sept. 11-14 2005.

[10] Faisal I. Bashir, Ashfaq A. Khokhar, and Dan Schonfield. Object trajectory-based activity classification and recognition using hidden markov models. In *IEEE Transactions on Image Processing*, volume 16 of *7*, pages 1912–1919, July 2007.

[11] Faisal I. Bashir, Ashfaq A. Khokhar, and Dan Schonfield. Real-time motion trajectory based indexing and retrieval of video sequences. In *IEEE Transactions on Multimedia*, volume 9 of *1*, pages 58–65, January 2007.

[12] D. Buzan, S. Sclaroff, and G. Kollios. Extraction and clustering of motion trajectories in video. In *International Conference on Pattern Recognition*, pages 521–524, Cambridge, UK, 2004.

[13] Yuhan Cai and Raymond Ng. Indexing spatio-temporal trajectories with chebyshev polynomials. In *ACM SIGMOD/PODS Conference*, pages 599–610, France, June 13-18 2004.

[14] R.B. Calinski and J. Harabasz. A dendrite method for cluster analysis. In *Comm. in Statistics*, volume 3, pages 1–27, 1974.

[15] K. Chan and A. Fu. Efficient time series matching by wavelets. In *Proc. of International Conference on Data Engineering*, pages 126–133, Sydney, March 1999.

[16] S. F. Chang, W. Chen, J. M. Horace, H. Sundaram, and D. Zhong. A fully automated content based video search engine supporting spatiotemporal queries. In *IEEE Transactions on Circuits and System for Video Technology*, volume 8 of *5*, pages 602–615, September 1998.

[17] S. Dagtas, W. Ali-Khatib, A. Ghafor, and R.L. Kashyap. Models for motion-based video indexing and retrieval. In *IEEE Transactions on Image Processing*, volume 9 of *1*, pages 88–101, January 2000.

[18] D. L. Davie and D. W. Bouldin. A cluster separation index. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume 1, pages 224–227, 1979.

[19] J.C. Dunn. A fuzzy relative of isodata process and its use in detecting compact well-separated clusters. In *J. Cybernetics*, volume 3, pages 32–57, 1973.

[20] Christos Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, pages 419–429, 1994.

[21] D. M. Gavrila. The visual analysis of human movement: A survey", in: Computer vision and image understanding. In *Proc. IEEE International Joint Conference on Neural Networks*, volume 73 of *1*, pages 82–98, January 1999.

[22] C-T. Hsu and S-J. Teng. Motion trajectory based video indexing and retrieval. In *IEEE International Conference on Image Processing*, volume 1, pages 605–608, 2002.

[23] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. In *IEEE Transactions on Systems, Man & Cybernetic*, volume 34 of *3*, pages 334–352, August 2004.

[24] W. Hu, X. Xiao, D. Xie, T. Tan, and S. Maybank. Traffic accident prediction using 3-d model based vehicle tracking. In *IEEE Transactions on Vehicular Tech*, volume 53 of *3*, pages 677–694, May 2004.

[25] W. Hu, D. Xie, T. Tan, and S. Maybank. Learning activity patterns using fuzzy self-organizing neural networks. In *IEEE Transactions on Systems, Man & Cybernetic*, volume 34 of *3*, pages 1618–1626, June 2004.

[26] G. Kollios V. Pavlovic J. Alon, S. Sclaroff. Discovering clusters in motion time-series data. In *Proc. IEEE CVPR*, volume 1, pages I–375–I–381, June 2003.

[27] S. Jeanin and A. Divakaran. Mpeg-7 visual motion descriptors. In *IEEE Trans. Circuits Syst. Video Technol.*, volume 11 of *6*, pages 720–724, June 2001.

[28] Y. Jin and F. Mokhtarian. Efficient video retrieval by motion trajectory. In *Proceedings of British Machine Vision Conference*, pages 667–676, Kingston, September 2004.

[29] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. In *Proceedings of British Conference on Machine Vision*, pages 582–592, 1995.

[30] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrota. Locally adaptive dimensionality reduction for indexing large time series databases. In *Proc. ACM SIGMOD Conference*, pages 151–162, 2001.

[31] S. Khalid and A. Naftel. Evaluation of matching metrics for trajectory based indexing and retrieval of video clips. In *Proceedings of IEEE WACV*, pages 242–249, Colorado, USA, January 2005.

[32] T. B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision based human motion capture and analysis. In *Computer Vision and Image Understanding*, volume 104.

[33] Andrew Y. Ng, Micahel I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information and Processing Systems*, volume 14, 2001.

[34] J. Owens and A. Hunter. Application of the self-organising map for trajectory classification. In *Proceedings of Third IEEE International Workshop on Visual Surveillance*, page 77, Dublin, Ireland, July 1 2000.

[35] J. Owens and A. Hunter. Novelty detection in video surveillance using hierarchical neural networks. In *Proceedings of ICANN*, pages 1249–1254, Madrid, Spain, August 28-30 2002.

[36] F. Porikli and T. Haga. Event detection by eigenvector decomposition using object and frame features". In *International Conference on Computer Vision and Pattern Recognition*, 2004.

[37] C. Shim and J. Chang. Content based retrieval using trajectories of moving objects in video databases. In *Proceedings of IEEE 7th International Conference on Database Systems for Advanced Applications*, pages 169–170, 2001.

[38] C. Shim and J. Chang. Trajectory based video retrieval for multimedia information systems. In *Proceedings of ADVIS*, pages 372–382, 2004.

[39] C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. In *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*, volume 22 of *8*, pages 747–757, 2000.

[40] N. Sumpter and A. J. Bulpitt. Learning spatio-temporal patterns for predicting object behaviour. In *Image and Vision Computing*, volume 18, pages 697–704, 2000.

[41] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *Proceedings of the International Conference on Data Engineering*, pages 673–684, San Jose, CA, 2002.

[42] D. Zhang, Gatica-Perez, S. Bengio, and I. McCowan. Semi-supervised adapted hmms for unusual event detection. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 611–618, 2005.

[43] A. Zisserman and R. Harley. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.