Classifying Spatiotemporal Object Trajectories using Unsupervised Learning of Basis Function Coefficients

Shehzad Khalid Andrew Naftel
School of Informatics
University of Manchester
Manchester M60 1QD, United Kingdom
+44 161 306 5837
s.khalid-2@postgrad.manchester.ac.uk a.naftel@manchester.ac.uk

ABSTRACT

This paper proposes a novel technique for clustering and classification of object trajectory-based video motion clips using spatiotemporal functional approximations. A Mahalanobis classifier is then used for the detection of anomalous trajectories. Motion trajectories are considered as time series and modeled using the leading Fourier coefficients obtained by a Discrete Fourier Transform. Trajectory clustering is then carried out in the Fourier coefficient feature space to discover patterns of similar object motions. The coefficients of the basis functions are used as input feature vectors to a Self-Organising Map which can learn similarities between object trajectories in an unsupervised manner. Encoding trajectories in this way leads to efficiency gains over existing approaches that use discrete point-based flow vectors to represent the whole trajectory. Experiments are performed on two different datasets – synthetic and pedestrian object tracking - to demonstrate the effectiveness of our approach. Applications to motion data mining in video surveillance databases are envisaged.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering

General Terms

Algorithms

Keywords

Object trajectory, Event mining, Motion classification, Anomaly detection.

1. INTRODUCTION

The current ubiquity of video surveillance systems has prompted a flurry of research activity aimed at the development of sophisticated content-based video data management techniques. General purpose tools are now urgently required for visual event mining including discovery and grouping of similar motion patterns, detection of anomalous behaviour and object motion prediction. Such tools are needed for the development of next generation 'actionable intelligence' surveillance systems.

Much of the earlier research focus has been on high-level object trajectory representation schemes that are able to produce compressed forms of motion data [1-11]. This work presupposes the existence of some low-level tracking scheme for reliably extracting object-based trajectories[12, 13]. The literature on trajectory-based motion understanding and pattern discovery is less mature but advances using Learning Vector Quantization (LVQ) [14], Self-Organising Maps (SOMs) [15, 16], hidden Markov Models (HMMs) [17], and fuzzy neural networks [18] have all been reported. Most of these techniques attempt to learn high-level motion behaviour patterns from sample trajectories using discrete point-based flow vectors as input to a machine learning algorithm. For realistic motion sequences, convergence of these techniques is slow and the learning phase is usually carried out offline due to the high dimensionality of the input data space.

Related work within the data mining community on approximation schemes for indexing time series data is highly relevant to the parameterisation of object trajectories. However, computer vision researchers have been slow to adopt this work. For example, Discrete Fourier Transforms (DFT) [19], Discrete Wavelet Transforms (DWT) [20], Adaptive Piecewise Constant Approximations (APCA) [21], and Chebyshev polynomials [22] have successfully been used to conduct similarity search in time series databases.

In this paper, we apply time series modeling of spatiotemporal data to the problem of object trajectory classification and show how to learn motion patterns by projecting the high-dimensional trajectory data into a low-dimensional manifold represented by a suitably chosen coefficient feature space. The coefficients are derived using functional approximation. The derived coefficient vector is used as an input feature vector to a neural network learning algorithm – in this instance a SOM - which can learn similarities between object trajectories in an unsupervised manner. It is shown that significant improvements in the accuracy of trajectory classification and recognition result when learning

VSSN'05, November 11, 2005, Singapore.

takes place in the coefficient feature space rather than the original trajectory space.

The remainder of the paper is organized as follow. We review some relevant background material in section 2. The system architecture and trajectory learning algorithm is presented in section 3 within the framework of a self-organising map. In section 4, the trajectory classification and anomaly detection procedure is discussed and experimental results for synthetic and object tracking data are reported in section 5. The paper concludes with a discussion and proposals for further work.

2. BACKGROUND AND RELATED WORK

Trajectory descriptors are known to be useful candidates for compressed representation of video object motion. Previous work has sought to represent moving object trajectories through a wide variety of direction schemes, polynomial models and other function approximations [1-10, 19-22]. The importance of selecting the most appropriate trajectory model has received relatively scant attention [11]. It is also surprising to find that many of these candidate indexing schemes have not yet been applied to the problem of motion data mining and trajectory classification. Recent work has either used probabilistic models such as HMMs [17] or discrete point-based trajectory flow vectors [14, 16, 18] as a means of learning patterns of motion activity.

In [25, 26] an agglomerative clustering algorithm is presented, based on the Longest Common Subsequence (LCSS) approach, for grouping similar motion trajectories. Yacoob [27] and Bashir *et al*. [28, 29] have presented a framework for modeling and recognition of human motion based on a trajectory segmentation scheme. Classification is performed using Gaussian Mixture Model (GMMs) and HMMs with trajectory modeling that relies on a PCA-based representation of segmented object trajectories. In [30], a semantic event detection technique based on discrete HMMs is applied to snooker videos. Various machine learning algorithms used for classifying biological motion trajectories are compared in [31].

The contribution of this paper is to show that a trajectory-encoding scheme based on an input coefficient feature space can be used to learn motion patterns more efficiently than previous approaches relying on discrete flow vectors. Clustering, classification and the detection of anomalous trajectories can then be carried out in the basis function coefficient space with reduced computational burden.

3. LEARNING TRAJECTORY PATTERNS USING SELF-ORGANIZING MAPS

Self-organised maps (SOMs) have been previously used for motion activity classification [15, 16] with trajectories encoded as point-based flow vectors. However, the use of point-based feature vectors results in high dimensionality and reduced system efficiency. In this case, unsupervised learning of motion patterns takes place offline. To achieve dimensionality reduction, we consider object trajectories as motion time series and use a coefficient indexing scheme. A performance comparison of different motion indexing schemes can be found in [11].

3.1 Trajectory Representation and Similarity Search Metric

The output of a motion tracking algorithm is usually a set of noisy 2-D tracker points (x_i, y_i) representing the object's motion path over a sequence of n frames, where $i = 1,\ldots,n$. Often the representative point is taken to be the centroid or edge mid-point of the object's minimum bounding rectangle. The motion trajectory can be considered as two independent 1-dimensional time series, $<t_i, x_i>$ and $<t_i, y_i>$, the horizontal and vertical components of object displacement against time where $t_1 < \ldots < t_n$.

In this paper we use the DFT to map object trajectory time series to the frequency domain. The DFT coefficients can be calculated using the well known Fast Fourier Transform (FFT). The formulae for evaluating the Fourier coefficients can be found in [19].

The Euclidean distance is used as the basis for comparing the similarity of two motion trajectories. Each function approximation produces a vector of leading Fourier coefficients which can be used to index a 2-dimensional spatiotemporal trajectory. Given two trajectories Q and S, we can index these by a vector of $2m$ coefficients $Q = \left\{\overrightarrow{q_1},\ldots.,\overrightarrow{q_m}\right\}$ and $S = \left\{\overrightarrow{s_1},\ldots.,\overrightarrow{s_m}\right\}$, where $\vec{q}_i, \vec{s}_i$ are $\vec{q}_i = [q_{xi}, q_{yi}]^T$ and $\overrightarrow{s_i} = [s_{xi}, s_{yi}]^T$ ($i = 1,\ldots, m$). The constant coefficient can be removed by shift normalising the data.

A Euclidean distance function (ED) on the coefficient space can be expressed as

$$ED(Q,S) = \sqrt{\sum_{i=1}^{m} (\vec{q}_i - \vec{s}_i)^2} \qquad (1)$$

An overview of the system architecture used for trajectory learning is shown in Fig. 1.

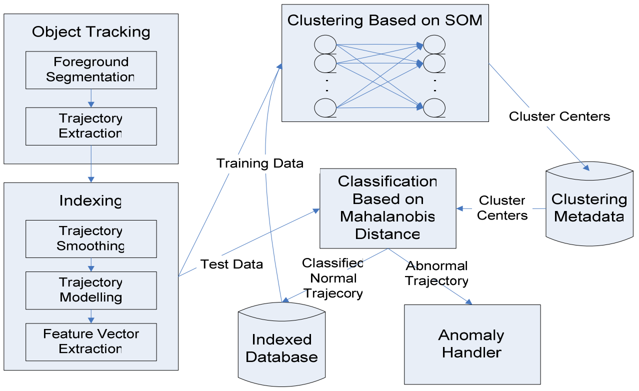


Figure 1. Overview of system architecture for learning object trajectories.

3.2 Network Model

The network topology chosen for the SOM is a layer of input neurons connected directly to a single 1-dimensional output layer.

Each input neuron is connected to every output neuron with the connection represented by a weight vector. The network topology is shown in Fig. 2. A similar network model was proposed in [16] for learning vehicle trajectories as a means for accident prediction.

In a SOM network, physically adjacent output nodes encode patterns in the trajectory data that are similar and hence, it is known as a topology-preserving map. Consequently, similar object trajectories are mapped to the same output neuron. The number of input neurons is determined by the size of the feature vector which in this case relates to the selected number of DFT coefficients in the model. The number of output neurons represents the number of distinct patterns in the trajectory data and is selected manually at present.

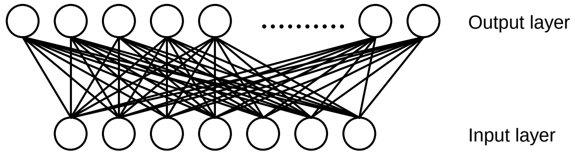


Figure 2. SOM network architecture used for trajectory learning.

3.3 Learning Algorithm

The algorithm used to cluster the trajectories differs slightly from the original SOM proposed by Kohonen [23]. The number of output neurons is initially set to a value greater than the desired number of cluster patterns that we wish to produce. After training the network, clusters representing the most similar patterns are merged in an agglomerative manner until the cluster count is reduced to the target number. The final number of trajectory cluster patterns is empirically tuned at present.

Let B be the input feature vector representing the set of trajectory basis function coefficients, and W the weight vector associated to each output neuron. The learning algorithm comprises the following steps:

1. 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

   $$c = \arg\min_k \left\| B - W_k(t) \right\| \quad \forall k \tag{2}$$

2. Train the network by updating the weights. A subset of the weights constituting a neighbourhood centred around node c

   are updated using

   $$W_k(t+1) = W_k(t) + \alpha(t)\eta(k,c)(B - W_k(t)) \tag{3}$$

   where $\eta(k, c) = \exp(-|r_k - r_c|^2 / 2\sigma_t^2)$ is a neighbourhood function that has a value of 1 when $k = c$ and falls off with distance $|r_k - r_c|$ between output nodes k and c, σ_t is a width parameter that is gradually decreased and t is the training cycle index.

3. Decrease the learning rate $\alpha(t)$ linearly over time.

4. After a pre-determined number of training cycles, decrease the neighbourhood size.

5. At the end of the training phase, merge the most similar cluster pairs until the desired number of groupings is achieved. Assuming W_a and W_b are the weight vectors associated with output neurons representing the most similar clusters, and m, n are the number of sample trajectories mapped to these neurons respectively, a new weight value W_{ab} for the merged cluster can be calculated as

   $$W_{ab} = \frac{mW_a + nW_b}{m+n} \tag{4}$$

4. TRAJECTORY CLASSIFICATION AND ANOMALY DETECTION

The SOM algorithm can be used to learn a set of motion patterns for the trajectory training dataset. The resulting labelled classes can then be used to classify new unseen trajectory data as normal (i.e. belonging to one of the existing labelled classes) or abnormal (distant from one of the existing classes). We use a simple k-NN classifier with the optimum value of k chosen by leave-one-out analysis. Classification results are presented in the following section using hand-labelled trajectories as ground truth. Visualisation of the clusters in the coefficient feature space shows that it is a reasonable assumption to represent class conditional probability density functions as multivariate normal. Anomalous trajectories can be detected through analysis of the covariance structure of a pattern at each output node. Hotelling's T^2 test is used to determine if the Mahalanobis distance of a sample trajectory to its nearest class centre makes it an outlier and thus abnormal. The test is now described in more detail.

Assume that sample x belongs to pattern class Γ_i , where $\#\{\Gamma_i\}$ denotes the number of sample vectors x assigned to class Γ_i and $i = 1,..,K$. The class mean is denoted by μ_i and the covariance estimate $\sum_i$ is thus

$$\sum_i = \sum_{x\in\Gamma_i} (x-\mu_i)(x-\mu_i)^T \,/(\#\{\Gamma_i\}-1) \tag{5}$$

where μ_i and $\sum_i$ are calculated during training. The T^2 statistic based on the Mahalanobis distance can be calculated as

$$T^2 = \frac{n}{n+1}(x-\mu_i)^T \sum^{-1}(x-\mu_i) \tag{6}$$

where $n = \#\{\Gamma_i\}$ and μ_i is the class mean to which the sample vector is closest. A hypothesis test can be conducted to determine whether x is 'too far' from μ_i and hence anomalous [32]. Given that p is the size of the input feature vector in the coefficient space, we have that

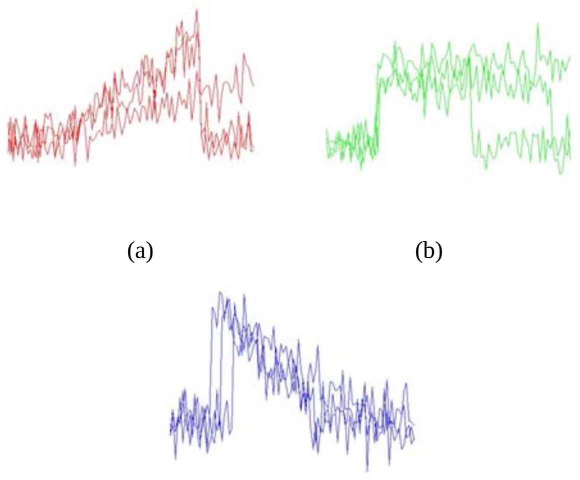
$$\alpha = P\left(T^2 > \frac{(n-1)p}{n-p} F_{p,n-p} \right) \tag{7}$$

where $F_{p,n-p}$ is a random variable with an F-distribution and p, n-p degrees of freedom. $F_{p,n-p}(\alpha)$ is the upper $(100\alpha)^{\text{th}}$ percentile of the $F_{p,n-p}$ distribution.

5. EXPERIMENTAL RESULTS

We now present some results to demonstrate the effectiveness of the proposed clustering, classification and anomaly detection techniques that are performed in the basis function coefficient feature space. The experiments have been performed on two different datasets – synthetic and pedestrian object tracking data.

The synthetic dataset is known as the Cylinder-Bell-Funnel data and has been widely used to benchmark time series data mining algorithms [33]. Examples from each of the 3 classes are shown in Fig. 3. We performed two tests on this dataset. In the first experiment, we compared the classification error using the Euclidean distance similarity measure on the original data points and then on the Fourier coefficient feature vector obtained using 8 DFT coefficients. Each time series was of fixed length consisting of 100 points. The classification accuracy was determined by 1-NN leave-one-out cross-validation. For each class, we generated n_c = 50 and n_c = 120 instances and averaged the classification error over 50 runs. The results are presented in Table 1. These illustrate that improved classification accuracy is obtained using DFT-coefficient indexing of time series instead of discrete point vectors. The classification errors resulting from unsupervised learning using *K*-means clustering are also shown for comparison. In the case of unsupervised learning, we assume no a priori knowledge of class labels.



(a) (b)

(c)

Figure 3. Examples of the synthetic dataset of which there are 3 classes: (a) bell, (b) cylinder, and (c) funnel.

In the second experiment, we tested the anomaly detection procedure using the Fourier coefficient representation. In this case it was not possible to use the point sequence as the covariance matrices for each class were near-singular and hence non-invertible. We generated a set of 100 instances (i.e. coefficient vectors) from each class and tested each instance in the class against all the other classes seperately to determine the average number of detected anomalies. The mean and covariances of each class were generated from a sample size of 100 instances of each class and maintained constant for each run. The reported results were averaged over 50 repetitions. We would expect that few instances drawn from class *X* would be recorded as anomalous when tested against the same class, whereas nearly all instances would be detected as anomalous when tested against a different class *Y*. The results demonstrating this fact are shown in Table 2. The diagonal entries in the table show instances of class *X* tested against *X* and the off-diagonal entries instances of class X tested against *Y*. It should be noted that instances of Funnel class are less likely to be detected as anomalous when tested against Cylinder class (but not *vice versa*). It can be seem from Fig. 3 that there is some overlap between instances from these classes. The results of both experiments for the synthetic dataset were promising and encouraged us to try out real motion trajectory data.

We now evaluate the performance of the SOM machine learning algorithm using the CAVIAR visual tracking database [24]. The database consisted of hand annotated video sequences of moving and stationary people and are intended to provide a test-bed for benchmarking vision understanding algorithms. The dataset consisting of 222 independent object trajectories is shown in Fig. 4.

Table 1. Classification error based on Euclidean distance similarity measure using 1-NN leave-one-out cross-validation.

Model		% Error
Point sequence	(n_c = 50)	3.2
DFT coefficients	(n_c = 50)	2.0
Point sequence	(n_c = 120)	2.5
DFT coefficients	(n_c = 120)	0.8
Unsupervis. DFT	(n_c = 100)	21.2
Unsupervis. Point	(n_c = 100)	35.4

Table 2. Percentage of instance vectors detected as anomalous. If the classes were completely separable the diagonal table entries would be zero and the off-diagonal entries would be 100.

Sample\Test class	Cylinder	Bell	Funnel
Cylinder	0.8	90.1	7.3
Bell	99.0	1.5	98.8
Funnel	98.9	100.0	0.3

The trajectories are represented using Fourier coefficient feature vectors with 8 coefficients for each spatial coordinate. We initially train a SOM network with 50 output neurons and then reduce these to 9 using the agglomerative clustering method described in section 3.3. In the SOM learning algorithm the neighbourhood size σ_t is decreased linearly after every Q training cycles, where Q is fixed at the start of training. The learning rate $\alpha(t)$ is reduced linearly over time until it reaches a set minimum value and then remains constant over the fine tuning stage until maximum number of iterations is achieved. The weight vectors are randomly initialized to lie within the expected range of the input feature vectors. This type of initialization improves the stability of the training network during the learning phase.

Sample trajectories from the test set are then classified using the classification technique described in section 4. The resulting trajectory cluster patterns are shown in Fig. 5. Visual inspection confirms that qualitatively similar motion trajectories have been clustered together quite successfully. Motions across the shopping mall corridor from left-to-right and right-to-left are grouped into separate clusters as expected. Although the proposed time series representation is velocity dependent, spatial similarities in object trajectories can still be identified in the cluster patterns. In this case we have chosen a motion representation that is view dependent and this would necessitate training the system on each camera separately. A method that deals with small PTZ motions of the camera can be developed based on techniques described in [7].

Sample trajectories are identified as abnormal when eq. (7) is satisfied ($P < 0.01$). These are shown highlighted in colour in the bottom right hand plot in Fig. 5.

In order to visualise the effects of trajectory clustering in the transformed feature space, we perform Principal Component Analysis (PCA) on the DFT coefficient vectors. The first 3 PCs account for 94% of the total variability. Fig. 6 shows the trajectories plotted in the PCA subspace of DFT coefficients. Each point represents an instance trajectory and these are colour/marker coded to highlight the separate cluster groups each trajectory is allocated to. These plots show a good degree of cluster separation in the low-dimensional PCA subspace.

To investigate the effectiveness of clustering in the basis function coefficient feature space compared to clustering in the original trajectory point space, we compared the classification accuracy achieved with each approach. The class labels of the motion trajectory patterns were learnt using the SOM and *K*-means unsupervised techniques on the CAVIAR dataset. The dataset *S* was then randomly partitioned into training and test sets of equal sizes for cross-validation. We used a 1-NN classifier to classify all instance trajectories from the test set and generated the overall classification accuracy. To avoid bias, we repeated the random partitioning 500 times and aggregated the classification errors by averaging over all the test sets. The results summarised in Table 3 demonstrate the superiority of learning trajectory patterns in the coefficient space. The classification accuracy obtained using coefficient feature space learning is higher than that of discrete point vector encoding for both SOM and *K*-means algorithms.

In the next experiment we compare the performance of all 4 methods in trajectory classification and prediction. From the original set *S*, we defined a set of partial trajectories S_P by sub-sampling a set of points from the original trajectories from 10% of the original length up to 100% in steps of 10. The classification was performed based on the input vectors consisting of DFT coefficients or discrete point-based flow (PBF) vectors. This is repeated for the SOM and *K*-means defined set of codebook vectors. The Mahalanobis distance is calculated between the input vector x representing the partial trajectory and cluster mean μ_i associated with ith output neuron. The input sample is classified to class i if it satisfies

$$\arg\min_i \left[\log(\det(\textstyle\sum_i)) + (x-\mu_i)^T \textstyle\sum^{-1}(x-\mu_i) \right] \qquad (8)$$

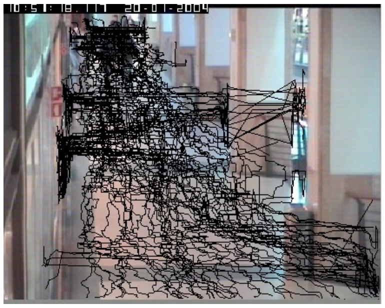


Figure 4. Background scene containing database of 222 hand-labelled object trajectories.

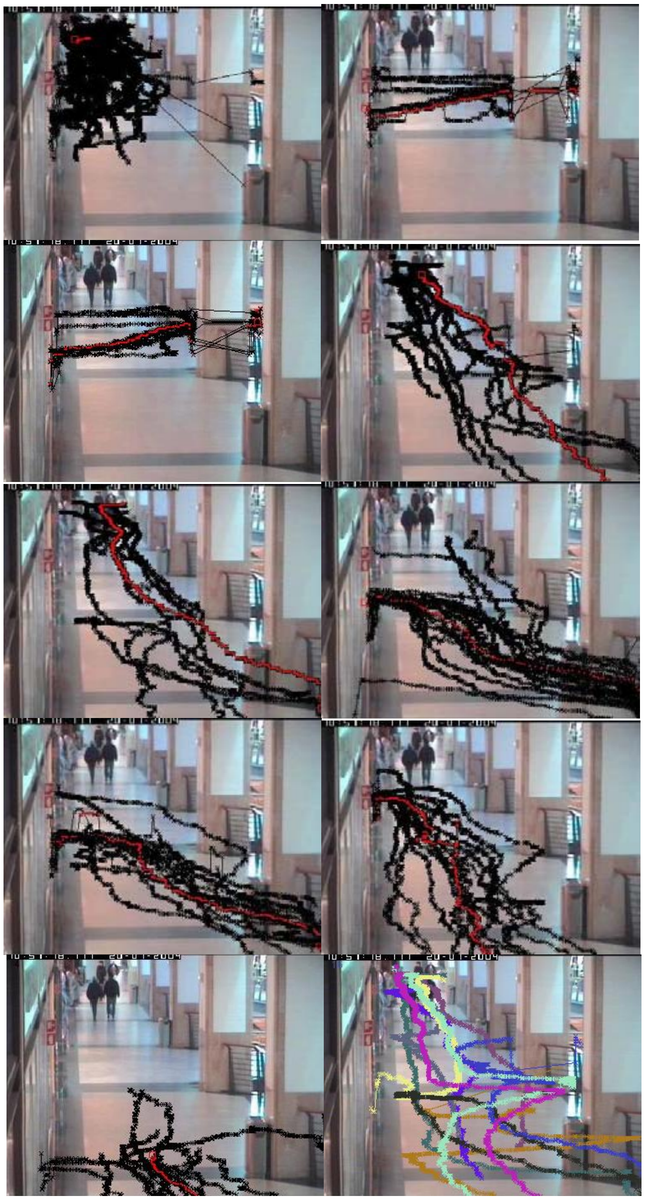


Figure 5. Clustering of motion trajectories in CAVIAR dataset using SOM with DFT-based coefficient input feature vectors. The figure at bottom right shows the anomalous traectories detected using Hotelling's test with *P* < 0.01.

where Σ_i is the class covariance estimate and is calculated using eq.(5). A partial trajectory is said to be misclassified if it is not assigned to the original class based on the full trajectory set S. As evidenced from Fig. 7, the classifier derived from SOM in the coefficient feature space once again outperforms K-means.

It is clear that DFT-parameterized models prove more effective than point-based flow vectors in the trajectory prediction and classification task. These results give further impetus to the development of alternative dimensionality reduction techniques for learning and prediction of motion trajectory patterns.

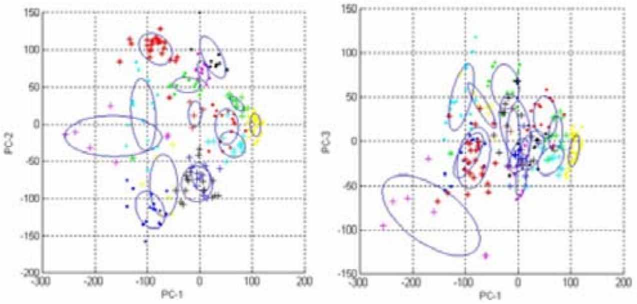


Figure 6. Clustering visualised in the PCA subspace of DFT coefficient vectors. The plots represent PC$_1$ vs PC$_2$ (left), PC$_1$ vs PC$_3$ (right). Error ellipses (1-sigma contours) for the covariance matrix are shown for each cluster pattern. This data corresponds to that of Fig. 5.

Table 3. Comparison of mean overall classification accuracy for 2 different clustering techniques (SOM and K-means) and 2 different trajectory encodings (coefficient subspace and discrete PBF vectors). #classes : #trajectories = 9 : 111.

Method type	% Accuracy
SOM: coefficient subspace	91.9
SOM: PBF vectors	79.9
K-Means: coefficient subspace	88.8
K-Means: PBF vectors	85.6

6. DISCUSSION AND CONCLUSION

This paper presents a neural network learning algorithm for classifying spatiotemporal object trajectories. Global features of motion trajectories are represented by Fourier approximations and this is apparent in the cluster visualizations. Using the coefficients of basis functions as input feature vectors to a neural network learning algorithm offers an efficient alternative to the use of discrete point-based flow vectors for trajectory classification and anomaly detection.

A drawback of this approach is for representation of highly complex trajectories which are inherently unsuited to a global Fourier approximation. One possibility is to use a trajectory segmentation or multiscale approach and augment the feature vector with additional entries relating to object shape or colour. A more comprehensive performance evaluation is now required using real video sequences where occlusions and target misdetections will result in fragmented noisy trajectories. The robustness of the classification technique requires thorough investigation under these circumstances. We would also like to compare other dimensionality reduction and machine learning techniques for trajectory classification, e.g. ICA, HMMs and semi-supervised learning.

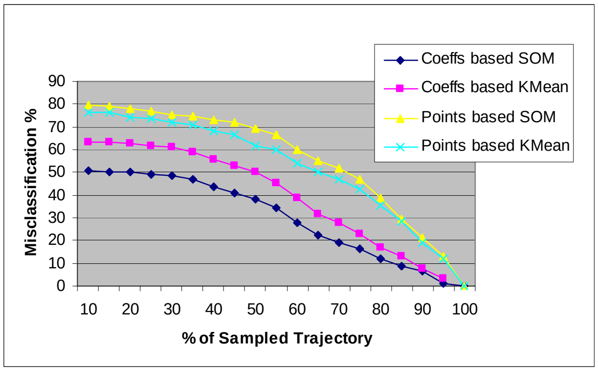


Figure 7. Comparison of mean overall classification accuracy in trajectory prediction using 2 different clustering techniques (SOM and K-means) and 2 different trajectory encodings (coefficient subspace and discrete PBF vectors). #classes : #trajectories = 9 : 111

7. REFERENCES

[1] Chang, S.-F., Chen, W. Meng, H.J. Sundaram, H. and Zhong, D. A fully automated content-based video search engine supporting spatiotemporal queries. *IEEE Trans. Circuits Syst. Video Technol.,* vol. 8, no. 5, pp. 602-615, Sept. 1998.

[2] Jeannin, S. and Divakaran, A. MPEG-7 visual motion descriptors. *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 6, pp. 720-724, June 2001.

[3] Dagtas, S., Ali-Khatib, W., Ghafor, A. and Kashyap, R.L. Models for motion-based video indexing and retrieval. *IEEE Trans. Image Proc.*, vol. 9, no. 1, pp. 88-101, Jan 2000.

[4] Aghbari, Z., Kaneko, K. and Makinouchi, A. Content-trajectory approach for searching video databases. *IEEE Trans. Multimedia*, vol. 5, no. 4, pp. 516-531, Dec. 2003.

[5] Bashir, F., Khokhar, A. and Schonfeld, D. Segmented trajectory-based indexing and retrieval of video data. In *Proc. IEEE Int. Conf. Image Processing*, Spain, 2003, pp. 623-626.

[6] Hsu, C.-T. and Teng, S.-J. Motion trajectory based video indexing and retrieval. In *Proc. IEEE Int. Conf. Image Processing*, pt 1, 2002, pp. 605-608.

[7] Bashir, F., Khokhar, A. and Schonfeld, D. A hybrid system for affine-invariant trajectory retrieval. In *Proc. MIR'04*, 2004, pp. 235-242.

[8] Shim, C. and Chang, J. Content-based retrieval using trajectories of moving objects in video databases. In *Proc.*

IEEE. 7th Int. Conf. Database Systems for Advanced Applications, 2001, pp. 169-170.

[9] Shim, C. and Chang, J. Trajectory-based video retrieval for multimedia information systems. In *Proc. ADVIS*, LNCS 3261, 2004, pp. 372-382.

[10] Jin, Y. and Mokhtarian, F. Efficient video retrieval by motion trajectory. In *Proc. British Machine Visiom Conference'04*, 2004.

[11] Khalid, S. and Naftel, A. Evaluation of matching metrics for trajectory-based indexing and retrieval of video clips. In *Proc.7th IEEE Workshop Applications Comp. Vision (WACV/MOTION'05)*, Colorado, USA, Jan. 2005, 242-249.

[12] Wang, L., Hu, W. and Tan, T. Recent developments in human motion analysis. *Pattern Recognition*, vol. 36, no. 3, pp. 585-601, 2003.

[13] Hu, W., Tan, T., Wang, L. and Maybank, S. A survey on visual surveillance of object motion and behaviors. *IEEE Trans. Systems, Man & Cybernetic*, Part C, vol.34, no.3, pp. 334-352, August 2004.

[14] Johnson, N. and Hogg, D. Learning the distribution of object trajectories for event recognition. *Image Vis. Comput.*, vol. 14, no. 8, pp. 609-615, 1996.

[15] Owens, J. and Hunter, A. Application of the self-organising map to trajectory classification. In *Proc. IEEE Int. Workshop Visual Surveillance*, pp. 77-83, 2000.

[16] Hu, W., Xiao, X., Xie, D., Tan, T. and Maybank, S. Traffic accident prediction using 3-D model-based vehicle tracking. *IEEE Trans. Vehicular Tech*., vol. 53, no. 3, pp. 677-694, May 2004.

[17] Alon, J., Sclaroff, S., Kollios, G. and Pavlovic, V. Discovering clusters in motion time-series data. In *Proc. IEEE CVPR*, June 2004.

[18] Hu, W., Xie, D., Tan, T. and Maybank, S. Learning activity patterns using fuzzy self-organizing neural networks. *IEEE Trans. Systems, Man & Cybernetic*, Pt. B, vol. 34, no. 3, pp. 1618-1626, June 2004.

[19] Faloutsas, C. Ranganathan, M. and Manolopoulos, Y. Fast subsequence matching in time-series databases. In *Proc. ACM SIGMOD Conf.*, 1994, pp. 419-429.

[20] Chan, K. and Fu, A. Efficient time series matching by wavelets. In *Proc. Int. Conf. Data Engineering*, Sydney, March 1999, pp. 126-133.

[21] Keogh, E., Chakrabarti, K., Pazzani, M. and Mehrota, S. Locally adaptive dimensionality reduction for indexing large time series databases. In *Proc. ACM SIGMOD Conf.*, 2001, pp. 151-162.

[22] Cui, Y. and Ng, R. Indexing spatio-temporal trajectories with Chebyshev polynomials. In *Proc. ACM SIGMOD Conf.*, Paris, June 2004, pp. 599-610.

[23] Kohonen, T. *Self-Organizing Maps,* 2nd ed. New York: Springer-Verlag, 1997, vol. 30.

[24] CAVIAR: Context aware vision using image-based active recognition. (2004, Jan. 10). [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIAR

[25] Buzan D., Sclaroff S., Kollios G. Extraction and clustering of motion trajectories in video. In *Proc. International Conference on Pattern Recognition,* 2004.

[26] Vlachos, M.., Kollios, G. and Gunopulos, D. Discovering similar multidimensional trajectories. In *Proc. International Conference on Data Engineering,* p. 673, 2002.

[27] Yacoob, Y. and Black, M. J. Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding,* Vol. 73 (2), pp. 232-247, Feb. 1999.

[28] Bashir, F., Khokhar, A. and Schonfeld, D. HMM-based motion recognition system using segmented PCA, To appear in *Proc. IEEE International Conference on Image Processing (ICIP 2005)*, Sept. 11 - Sept. 14, 2005. Genova, Italy.

[29] Bashir, F., Ashfaq, A., Khokhar, A. and Schonfeld, D. Object trajectory-based motion modeling and classification using Hidden Markov Models, *submitted to IEEE Transactions on Image Processing.*

[30] Rea, N., Dahyot, R. and Kokaram, A. Semantic event detection in sports through motion understanding. In *Proc. of Conference on Image and Video Retrieval*, Dublin, Ireland, July 21-23, 2004.

[31] Ivo, F. and Sbalzarinii, J.T. Machine learning for biological classification applications. Technical Report *Center for Turbulence Research, Proceedings of the Summer Program* 2002.

[32] Johnson, R.A. and Wichern, D.W. *Applied Multivariate Statistical Analysis*, 4th Ed., Prentice-Hall, NJ, 1998.

[33] Keogh, E. and Kasetty, S. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. *Knowledge and Data Discovery*, 2002, pp. 102-111.