Motion Trajectory Learning in the DFT-Coefficient Feature Space

Andrew Naftel and Shehzad Khalid
School of Informatics, University of Manchester, Sackville Street, Manchester M60 1QD, UK
a.naftel@manchester.ac.uk, s.khalid-2@postgrad.manchester.ac.uk

Abstract

Techniques for understanding video object motion activity are becoming increasingly important with the widespread adoption of CCTV surveillance systems. In this paper we propose a novel vision system for clustering and classification of object-based video motion clips using spatiotemporal models. Object trajectories are modeled as motion time series using the lowest order Fourier coefficients obtained by Discrete Fourier Transform. Trajectory clustering is then carried out in the DFT-coefficient feature space to discover patterns of similar object motion activity. The DFT coefficients 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. Assuming the clusters of trajectory points are distributed normally in the coefficient feature space, we propose a simple Mahalanobis classifier for the detection of anomalous trajectories. Our proposed techniques are validated on three different datasets – Australian sign language, hand-labelled object trajectories from video surveillance footage and real-time tracking data obtained in the laboratory. Applications to event detection and motion data mining for visual surveillance systems are envisaged.

1. Introduction

The current prevalence of visual surveillance systems has prompted much research activity aimed at the development of sophisticated content-based visual data management techniques. General purpose tools are now urgently required for automated surveillance tasks such as video event mining, discovery and understanding of motion activity patterns, detection of anomalous behaviour and motion trajectory prediction.

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 motion 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 proposed. Most of these techniques attempt to learn high-level motion behaviour patterns from sample trajectories using discrete point-based flow vectors as input to the learning phase. For realistic motion sequences, convergence of these techniques is slow and the learning phase is usually performed offline. This is due to the high dimensionality of the input feature space for a point-wise defined trajectory representation.

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], wavelet transforms (DWT) [20], adaptive piecewise constant approximations (APCA) [21], and Chebyshev polynomials [22] have been used to index time series data.

In this paper, we aim to apply time series modeling of spatiotemporal data to the problem of trajectory classification and show that it is possible to learn motion activity patterns by projecting the high-dimensional trajectory data into a low-dimensional manifold given a suitably chosen parameter/coefficient feature space. The vector of parameters is used as an input feature vector to a neural network learning algorithm – in this instance a Self-Organising Map - which can learn similarities between object trajectories

Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (ICVS 2006)
0-7695-2506-7/06 $20.00 © 2006 IEEE

in an unsupervised manner. It is shown that significant improvements in the accuracy of trajectory classification and recognition result when learning takes place in the parameter feature space rather than the original point-wise defined 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 3 different examples of object tracking data are reported in section 5. The paper concludes with a brief discussion and proposals for further work.

2. Review of Background Work

Trajectory descriptors are known to be useful candidates for compressed representation of object motion in videos. 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 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 a combination of Gaussian Mixture Models (GMMs) and HMMs for trajectory representation that relies on PCA-based segmented object trajectories. In [30], a semantic event detection technique based on discrete HMMs is applied to snooker/pool video clips. Various machine learning techniques 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 activity patterns more effectively than previous approaches relying on point-wise defined flow vectors. Clustering, classification and the detection of anomalous trajectories can then be carried out in the parameter/coefficient space with reduced computational burden.

3. Learning Trajectory Patterns using Self-Organizing Maps

Self-organising maps (SOMs) have been previously used for motion activity classification [15,16] with object trajectories encoded as point-based flow vectors. However, the use of point-based feature vectors results in high dimensionality and reduced system efficiency. This necessitates an offline unsupervised learning process. To achieve dimensionality reduction, we consider object trajectories as motion time series and use a coefficient indexing scheme. We have addressed the question of selecting an appropriate trajectory model representation scheme in [11].

An overview of the system architecture used for trajectory modeling and recognition is shown in Figure 1.

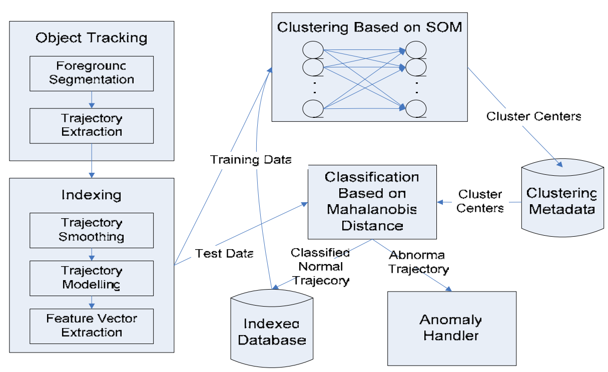

Figure 1. **Overview of system architecture for motion trajectory data mining.**

3.1. Trajectory Representation and Similarity Search Metric

The output of a motion tracking algorithm is usually a set of noisy 2-D candidate points (x_i, y_i) representing the object's motion path over a sequence of n frames, where $i = 0,\ldots,n-1$. 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 independent one-dimensional time series, $<t_i, x_i>$ and $<t_i, y_i>$, representing the x and y projections of object displacement against time where $t_0 < \ldots < t_{n\text{-}1}$.

As mentioned, an experimental comparison of several trajectory representation schemes has been previously carried out [11]. In the presence of additive noise and occlusions in tracking data, it is shown that a DFT-based coefficients outperforms two polynomial indexing schemes – least squares polynomials and Chebyshev approximations – in terms of retrieval accuracy. Hence, we use the DFT to map an object trajectory time series to the frequency domain. We use the first four coefficients of DFT since the energy of the time series in usually concentrated in the low order coefficients. It is found that similarity retrieval accuracy is not sensitive to the exact number of DFT coefficients selected [11] at least for the tracking datasets we examined.

The n-point DFT of a time series $\{x_i\}$, i=0,...,n-1 is defined to be a sequence $\{X_f\}$ of n complex numbers, f = 0,...,n-1 given in eq.(1). A similar expression can be defined for $\{y_i\}$ given in eq.(2)

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

$$Y_f = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} y_i \exp(-j2\pi ft/n), \quad f = 0,1,...,n-1 \tag{2}$$

where $j = \sqrt{-1}$. Here n represents the number of trajectory points. As mentioned, we truncate the DFT sequence after m terms, i.e. f = 0,..,m. Typically m = 4. X_0 and Y_0 are real numbers for real time series data. The feature vector thus consists of 2m+1 real entries (from real and imaginary parts) for each time series.

The Euclidean distance is then used as the basis for comparing the similarity of two motion trajectories. Each function approximation produces a vector of the leading m Fourier coefficients which can be used to index a 2-dimensional spatiotemporal trajectory. Given two trajectories Q and S, we can index each of these by a fecture vector of 2*(2m+1) coefficients $Q = \left\{\overrightarrow{q_0},....,\overrightarrow{q_{m-1}}\right\}$ and $S = \left\{\overrightarrow{s_0},....,\overrightarrow{s_{m-1}}\right\}$, where $\vec{q}_i$, $\vec{s}_i$ are given by $\vec{q}_i = [q_{xi}, q_{yi}]^T$, $\vec{s}_i = [s_{xi}, s_{yi}]^T$ (i= 0,...,m-1).

The constant coefficient can be removed by shift normalising the data. However, in the case of fixed camera surveillance this is undesirable as the trajectory spatial dependence would then be removed.

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

$$ED(Q,S) = \sqrt{\sum_{i=0}^{m-1} (\vec{q}_i - \vec{s}_i)^2} \tag{3}$$

An important observation here is that Parseval's theorem ensures that the Euclidean distance between the time series is preserved in the frequency domain. Truncating the DFT by ignoring all coefficients $f > 4$ results in underestimating the distance between trajectories and introduces no false dismissals [19].

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. 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, common 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 coefficients in the model. The number of DFT coefficients m can be determined empirically although, in practice, the learning algorithm is not sensitive to the exact choice of m when $m > 4$ [11]. The number of output neurons represents the number of distinct patterns in the trajectory data and is also empirically chosen at present.

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 using an agglomerative hierarchical clustering method [32] until the cluster count is reduced to the target number.

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 Wk is a

minimum amongst all output neurons, given by the condition

$$\| B - W_c(t) \| \leq \| B - W_k(t) \| \qquad \forall k \tag{4}$$

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

$$W_k(t+1) = W_k(t) + \alpha(t)\eta(k,c)(B - W_k(t)) \tag{5}$$

where $\eta(k,c) = \exp(-|r_k - r_c|^2/2\sigma^2)$ is a neighbourhood function that has value unity when $k = c$ and falls off with distance $|r_k - r_c|$ between output nodes k and c, σ 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 to shift from coarse to fine tuning.
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

$$W_{ab} = \frac{mW_a + nW_b}{m+n} \tag{6}$$

4. Trajectory Classification and Anomaly Detection

The SOM algorithm can be used to learn a set of motion patterns for the object trajectory training data. The resulting labelled classes can then be used to classify unseen trajectories as normal or abnormal. 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 DFT coefficient feature space (see section 5) 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 abnormal. The test is now described in more detail.

Assuming that a sample trajectory x belongs to the pattern class Γ_i, where $\#\{\Gamma_i\}$ denotes the number of sample trajectories x assigned to class Γ_i and $i = 1,..,K$. The class mean of the sample trajectories x is denoted by μ_i and the covariance estimate is given by Σ_i where

$$\Sigma_i = \Sigma_{x \in \Gamma_i}(x - \mu_i)(x - \mu_i)^T / (\#\{\Gamma_i\} - 1) \tag{7}$$

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

$$T^2 = \frac{n}{n+1}(x - \mu_i)^T \Sigma^{-1}(x - \mu_i) \tag{8}$$

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

$$\alpha = P\left(T^2 > \frac{(n-1)p}{(n-p)} F_{p,n-p}\right) \tag{9}$$

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 indicate the effectiveness of the proposed clustering, classification and anomaly detection. The experiments have been performed on 3 different tracking datasets – high quality recordings of hand signs from the Australian Sign Language (ASL) UCI KDD archive [33], hand-labelled object trajectories taken from CAVIAR dataset [24] and real-time object tracking data recorded in our laboratory.

Firstly, we evaluate the performance of the SOM neural network learning algorithm using the CAVIAR visual tracking data [24]. The data consists of hand-annotated video sequences of moving and stationary people and are intended to provide a test-bed for benchmarking vision understanding algorithms. The individual hand-labelled object trajectories have varying lengths but typically range from 20-1600 points with a mean length of 342. The trajectory points are rectified to the ground plane using the homography mapping data provided [24]. The complete set of object trajectories for the Portuguese shopping centre frontal corridor view are shown in Fig. 2.

These object trajectories are indexed using DFT coefficient feature vectors with 9 coefficients for each spatial coordinate (m = 4). We initially train a SOM network with 50 output neurons and then reduce these to 6 using the agglomerative clustering method described in section 3.2. The final choice for the number of clusters in the dataset is empirically determined. However, this process can be automated by applying a threshold on the distance between the closest clusters when merging the most similar clusters together. The merging process is stopped when the distance between the closest clusters lies below a certain threshold value.

Figure 2. Background scene of shopping centre corridor overlaid with complete set of hand-annotated object trajectories (CAVIAR dataset).

The resulting trajectory cluster patterns are shown in Fig. 3. The trajectory highlighted in red denotes the sample trajectory that lies closest to the weight vector associated to a particular cluster pattern. Visual inspection confirms that qualitatively similar motion trajectories have been clustered together quite successfully. Motions across the shopping mall scene from left-to-right and right-to-left are grouped into separate clusters as expected. This is highlighted by the points marked in green which represent the start point for each trajectory in the cluster.

To investigate the effectiveness of clustering in the coefficient/parameter feature space compared to clustering in the original point-wise defined trajectory space, we performed some classification tests. The class labels of the motion patterns shown in Fig. 2 were assigned using the SOM unsupervised learning algorithm described in section 3. For comparison purposes this was repeated using K-means clustering [32]. These labels were assumed to represent ground truth. The dataset S_T was then randomly partitioned into equal-sized training and test sets for cross validation purposes. We used a k-NN classifier (k = 1) 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 averaged the classification errors over the test set. The results summarised in Table 1 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 point-wise trajectory flow vector encoding for both SOM and K-means algorithms.

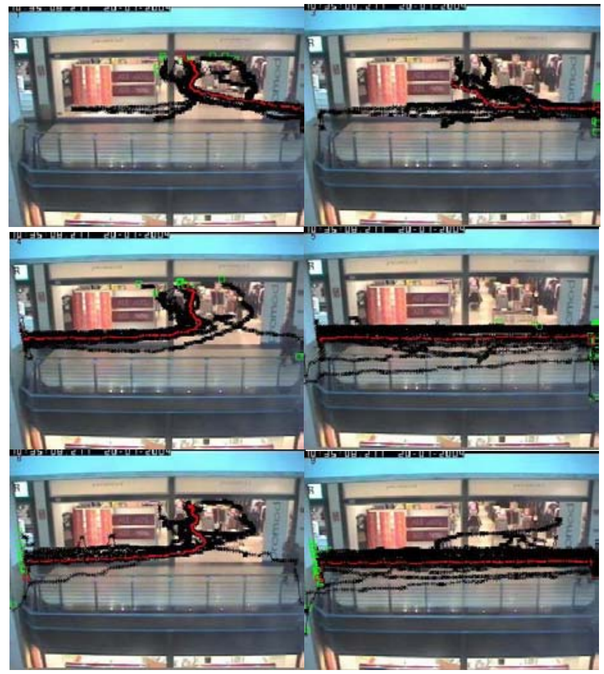

Figure 3. Clustering of motion trajectories in CAVIAR dataset using SOM with DFT coefficient feature vectors. Trajectories highlighted in red denote those closest to the weight vector for each cluster. Points marked in green denote the start point of each trajectory.

Table 1. Comparison of mean overall classification accuracy for 2 different clustering techniques (SOM and *K*-means) and 2 different trajectory encodings (DFT coefficient feature space and discrete point-wise flow vectors). #classes : #trajectories = 6 : 62.

Method type	% Accuracy
SOM: DFT coefficients	93.7
SOM: point-based flow vectors	81.2
K-Means: DFT coefficients	89.7
K-Means: point-based flow vectors	83.0

For the next experiment we compare the performance of all 4 methods in trajectory classification and prediction. From the original set S_T,

Proceedings of the Fourth IEEE International Conference on Computer Vision Systems (ICVS 2006)
0-7695-2506-7/06 $20.00 © 2006 IEEE

we define a set of partial trajectories S_P by removing 50% of the data points from the end of each trajectory. This is decreased down to the last 10% in steps of 10. The partial trajectories are then passed to the learning algorithm for classification. The class assigned to the complete trajectory is treated as the ground truth when assessing classification accuracy for the partial trajectory. We compare the point-based flow vector and DFT coefficient feature vector representation for both SOM and K-means trajectory learning techniques. The classification is based on the Mahalanobis distance between the input vector x_P representing the partial trajectory and cluster mean μ_i associated with ith output neuron/cluster centre. The sample x_P is assigned to class k if

$$k = \arg\min_{i \in K} \{(x_P - \mu_i)^T \Sigma_i^{-1} (x_P - \mu_i)\} \qquad (10)$$

where Σ_i is covariance estimate and is calculated using eq.(7). x_P is said to be misclassified if it is not assigned to the same class when trained on S_T. The mean classification errors based on motion prediction for the partial trajectories using each of the four approaches can be seen in Fig. 4. Once again, the classifier derived from an SOM-based learning technique combined with trajectory representation in the DFT-coefficient feature space outperforms K-means and point-based flow vector encoding. Hence, parameterized models prove more effective than point-based flow vectors in the trajectory prediction and classification task.

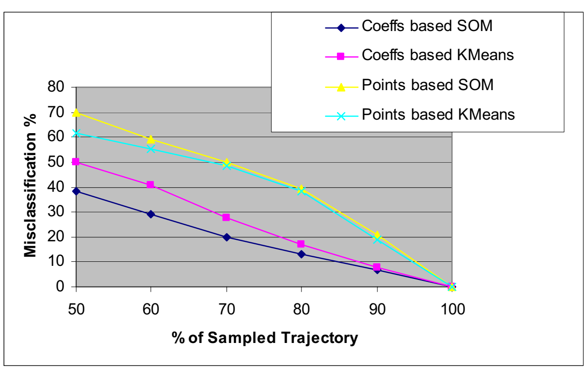

Figure 4. Comparison of mean overall classification accuracy in motion activity prediction using 2 different clustering techniques (SOM and *K*-means) and 2 different trajectory representations (DFT-coefficient feature space and point-based flow vectors). #classes : #trajectories = 6:125.

Classification experiments have also been performed on the Australian Sign Language (ASL) dataset [33]. The trajectories derive from the x, y-displacements of the signer's hand over a sequence of frames as different word classes are signed. The results may be compared with those reported in [28] for HMM-based motion recognition. Hand sign trajectories for the word class *forget* are shown in Fig. 5.

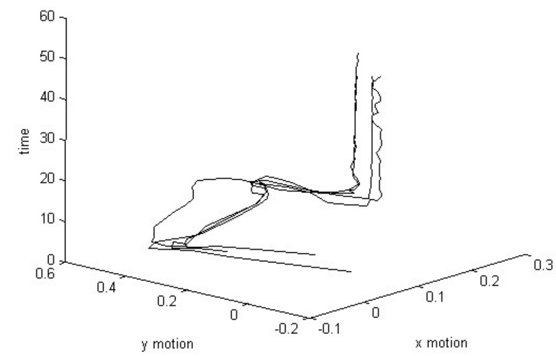

Figure 5. Three examples of hand sign trajectories for the word class *forget* in the Australian Sign Language (ASL) dataset. The vertical axis represents the time.

Each word class has 27 examples of signs with a trajectory point length of 57. A set of 30 word classes has been chosen. We used a supervised form of SOM to determine the classification accuracy for DFT-coefficient feature space trajectory learning. For the required number of word classes, the motion signing data is randomly partioned into equal-sized training and test sets. The codebook vectors are then learnt using the training data. The SOM is initialized with the number of output neurons set equal to the number of word classes present. The weight vectors are initialized with the mean of the trajectories that belong to each class. Then the training data is presented sequentially to the network and the cluster centers for the specific class are updated. Following that the test set is passed to the classifier and the class labels obtained are compared with the ground truth. The experiment is repeated with different numbers and combinations of word classes. Each classification experiment is averaged over 100 runs to reduce any bias resulting from favourable word selection. The classification accuracies are reported in Table 2. Since ground truth is available for the ASL dataset, this experiment gives some indication of motion recognition rates achievable using our trajectory learning system.

In the final experiment, we test the performance of the anomaly detection component of our vision system for trajectory-based motion understanding. The dataset was obtained from tracking people moving around in our laboratory. The tracking algorithm used to collect the data is described elsewhere [34]. Human movements were planned so that object trajectories could be grouped into 4 distinct classes and hence ground truth was known. The trajectory data is classified and the ground truth of each trajectory is known.. The object trajectory dataset superimposed over the background scene is shown in Figure 6.

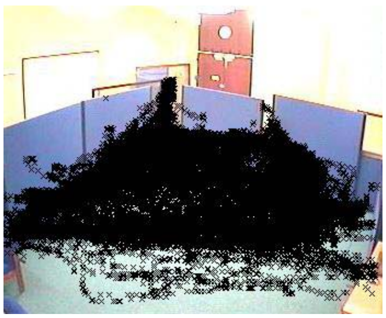

Figure 6. Laboratory background scene with set of object trajectories derived from author's tracking algorithm.

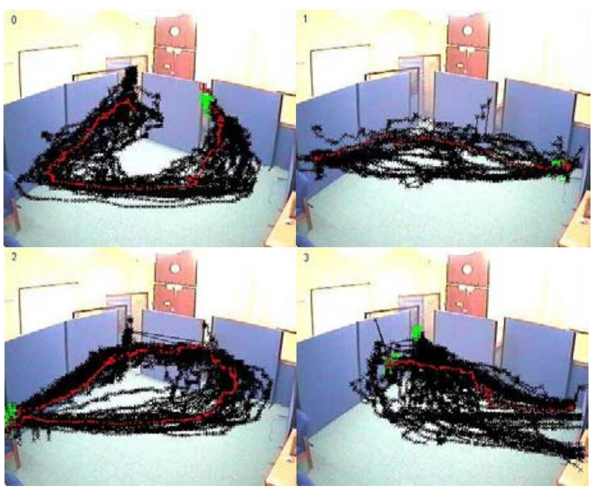

Figure 7. Results of motion trajectory clustering in the DFT coefficient feature space. Colour coding as before (see Fig. 3 caption).

The dataset included some motion activities that varied deliberately from the planned movement patterns. This was excluded from the training data. We randomly selected half the dataset for unsupervised learning and then presented the whole dataset together with the unusual trajectories as a test set. The clustering results are shown in Fig. 7. Visual inspection confirms that qualitatively similar motion trajectories have been grouped together as expected. This can be seen by observing the points marked by green rectangles which indicate the trajectory initial points. Abnormal trajectories which are defined to be sufficiently distant from all the identified classes such that eq.(9) is satisfied for $P < 0.01$ are shown in Figure 8. The system tags these as anomalous trajectories.

Table 2. Trajectory classification results for the Australian Sign Language (ASL) dataset. Trajectories are modeled in the DFT-coefficient feature space.

# classes : # trajectories	% Accuracy
2 : 54	95.7
3 : 81	91.0
4 : 108	89.9
8 : 216	82.1
16 : 432	76.3
24 : 648	70.1

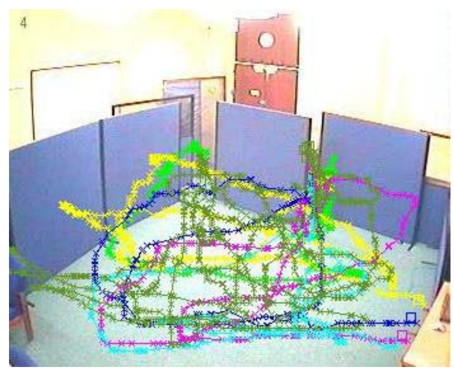

Figure 8. Trajectories categorised by the vision system as anomalous by using Hotelling's test with P<0.01.

6. Discussion and Conclusion

This paper presents a novel vision system component for trajectory-based event detection. Trajectories are modeled as motion time series using DFT coefficients and activity patterns are then learnt in this reduced feature space using a SOM network. Improvements in recognition accuracy and learning efficiency are achieved when compared to point-based trajectory encoding and other clustering techniques. Assuming trajectories are distributed normally in the transformed coefficient feature space, a Mahalanobis classifier can be used to distinguish between normal and abnormal trajectory patterns. Our techniques have been validated using 3 different types of video tracking data.

7. References

[1] S-F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong, "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] S. Jeannin and A. Divakaran, "MPEG-7 visual motion descriptors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 6, pp. 720-724, June 2001.

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

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

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

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

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

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

[10] Y. Jin and F. Mokhtarian, "Efficient video retrieval by motion trajectory", in *Proc. BMVC'04*, 2004.

[11] A. Naftel and S. Khalid, "Classification and prediction of motion trajectories using spatiotemporal approximation", In *Proc. Int. Workshop on Human Activity Recognition and Modelling* (HAREM 2005), Oxford, UK, 2005, pp. 17-26.

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

[13] W. Hu, T. Tan, L. Wang, and S. Maybank, "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] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image Vis. Comput.*, vol. 14, no. 8, pp. 609-615, 1996.

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

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

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

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

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

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

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

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

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

[24] CAVIAR dataset [Online] Available: http://homepages.inf.ed.ac.uk/rbf/CAVIAR.

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

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

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

[28] F. Bashir, A. Khokhar, and D. Schonfeld, "HMM-Based Motion Recognition System using Segmented PCA", In *Proc. IEEE International Conference on Image Processing (ICIP 2005)*, Sept. 2005. Genova, Italy.

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

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

[31] I.F. Sbalzarinii and J. Theriot, "Machine learning for biological classification applications", *Center for Turbulence Research, Proceedings of the Summer Program* 2002.

[32] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, 1998.

[33] KDD archive (2005, Aug. 12). [Online]. Available: http://kdd.ics.uci.edu/databases/auslan2/auslan.data.html.

[34] A. Naftel and S. Khalid, "Video sequence indexing through recovery of object-based motion trajectories," In *Proc. Irish Machine Vision and Image Processing Conference* (IMVIP'04), September 2004, Dublin, Eire. pp 232-239.

0-7695-2506-7/06 $20.00 © 2006 IEEE