Activity classification and anomaly detection using m-Mediods based modelling of motion patterns

Shehzad Khalid *,

Department of Computer Science and Engineering, Bahria University, Islamabad, 44000, Pakistan

Abstract

Techniques for video object motion analysis, behaviour recognition and event detection are becoming increasingly important with the rapid increase in demand for and deployment of video surveillance systems. Motion trajectories provide rich spatiotemporal information about an object's activity. This paper presents a novel technique for classification of motion activity and anomaly detection using object motion trajectory. In the proposed motion learning system, trajectories are treated as time series and modelled using modified DFT-based coefficient feature space representation. A modelling technique, referred to as m-Mediods, is proposed that models the class containing n members with m Mediods. Once the m-Mediods based model for all the classes have been learnt, the classification of new trajectories and anomaly detection can be performed by checking the closeness of said trajectory to the models of known classes. A mechanism based on agglomerative approach is proposed for anomaly detection. Four anomaly detection algorithms using m-Mediods based representation of classes are proposed. These includes: (i) Global Merged Anomaly Detection (GMAD), (ii) Localized Merged Anomaly Detection (LMAD), (iii) Global Un-merged Anomaly Detection (GUAD), and (iv) Localized Un-merged Anomaly Detection (LUAD). Our proposed techniques are validated using variety of simulated and complex real life trajectory datasets.

Key words: Object trajectory, dimensionality reduction, trajectory modelling, event mining, anomaly detection, motion recognition.
PACS:

* Corresponding author.
Email address: shehzad_khalid@hotmail.com (Shehzad Khalid).

1 Introduction

In recent years, there has been a growth of research activity aimed at the development of sophisticated content-based video data management techniques. This development is especially timely given the rapid increase in demand for and deployment of video surveillance systems. General purpose tools are now urgently required for video event mining including discovery and grouping of similar motion patterns, behavior classification and recognition, and detection of anomalous behavior. Behavior can obviously be categorized at different levels of granularity. In far-field surveillance, we are primarily interested in trajectory-based coarse motion description involving movement direction (right/left or up/down) and motion type (walking, running or stopping). These techniques are essential for the development of next generation actionable intelligence surveillance systems.

Much of the earlier research focus, in motion analysis, has been on high-level object trajectory representation schemes that are able to produce compressed forms of motion data [1][2][3][4][5][6][7]. The literature on trajectory-based motion understanding and pattern discovery is less mature but advances using Learning Vector Quantization (LVQ)[8], Self-Organising Maps (SOMs) [9][10], Hidden Markov Models (HMMs) [11][12], and fuzzy neural networks [13] have all been reported. Most of these techniques attempt to model high-level motion behaviour patterns from sample trajectories using discrete point sequences 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.

In different trajectory-based applications, there are four major cornerstones in developing a successful system for automated motion-based event detection and recognition: (i) an effective and low dimensional representation of trajectories that models the underlying characteristics of the original point sequence data space; (ii) learning of motion patterns in the presence of anomalies; (iii) modeling of normal motion patterns (iv) developing a high-accuracy activity classification and anomaly detection system. The issue of trajectory-based motion learning in the presence of anomalies has been addressed in our previous work [14]. This paper is focused on this issue of trajectory-based classification and recognition whilst identifying and filtering the anomalies. We use modified Discrete Fourier Transform (DFT-MOD) based coefficients for low dimensional feature space representation of trajectories. A novel approach for model-based classification of trajectory patterns and anomaly detection is presented. The proposed motion classification and anomaly detection techniques are compared with other methods reported recently in literature, using simulated as well as realistic motion datasets.

The remainder of the paper is organized as follows. We review some relevant background material in section 2. Section 3 briefly describes our coefficient feature space representation of motion trajectory. In section 4, the issue of modeling of normal patterns to be used later for classification of motion activities is addressed. A novel approach for model-based classification of trajectory patterns is also presented. Section 5 presents two variants of a novel mechanism based on agglomerative approach for anomaly detection. In section 6, a localized approach is presented to detect anomalies in the presence of patterns with different orientations and scales. The proposed approach does not require the specification of any manual parameters or thresholds to be used for anomaly detection. Experiments have been performed to show the effectiveness of proposed system for trajectory-based modeling and classification of motion patterns in the presence of anomalous motion samples. These experiments are reported in section 7. The last section summarizes the paper.

2 Background and related work

Trajectory descriptors are known to be useful candidates for compressed representation of object motion in videos. Given a large number of trajectories in motion datasets, the goal of trajectory-based motion learning is to learn a model that is capable of detecting normal motion patterns whilst identifying instances representing anomalous behaviour. In this context, we define anomalies as atypical behaviour patterns that are not represented by sufficient samples in training data and are *infrequently occurring* or *unusual*.

Previous work has sought to represent moving object trajectories through a wide variety of direction schemes, polynomial models and other function approximations. [1][2][3][4][7][15][16][17][18]. It is surprising to find that many of these candidate time series indexing schemes have not yet been applied to the problem of motion data mining and trajectory-based motion classification and recognition. Recent work has either used probabilistic models such as HMMs [19] or discrete point-based trajectory flow vectors (PBF) [8][9][13] as a means of modelling, classification and recognition of motion activity. The problem with PBF vector-encoded trajectory representation is the heavy computational burden making prospects for online learning of motion patterns remote.

Processing of trajectory data for activity classification and recognition has gained significant interest quite recently. Various techniques have been proposed for modeling of motion activity patterns and using the modeled patterns for classification and anomaly detection. These approaches are broadly categorized into statistical and neural network based approaches. Almost all statistical approaches dealing with anomaly detection are based on modelling the density of training data and rejecting test patterns that fall in regions of low density. There are various approaches that use Gaussian mixture models to estimate the probability density of data [20][21][22]. Various techniques based on hidden Markov models (HMM) have also been proposed [23][24][25]. Yacoob [26] and Bashir *et al.* [12][27] have presented a framework for modeling and recognition of human motion based on a trajectory segmentation scheme. A framework is presented to estimate the multivariate probability density function (PDF), based on PCA coefficients of the sub-trajectories, using GMM. Different classes of object motion are modelled by a continuous HMM per class where the state PDFs are represented by GMMs. The proposed technique has been shown to work for sign language recognition. The proposed classification system can not handle anomalies in test data and can only classify samples from normal patterns. Xiang *et al.* [23][24] propose a framework for behavior classification and anomaly detection in video sequences. Natural grouping of behaviour patterns is learnt through unsupervised model selection and feature selection on the eigenvectors of a normalized affinity matrix. A Multi-Observation Hidden Markov Model is used for modelling the behaviour pattern. Hu *et al.* [28][29] and Naftel *et al.* [30] models normal motion patterns by estimating single multivariate gaussian for each class. For anomaly detection in [28], the probability of a trajectory belonging to each motion pattern is calculated. If the probability of association of trajectory to the closest motion pattern is less then a threshold, the trajectory is treated as anomalous. In [31], a semantic event detection technique based on discrete HMMs is applied to snooker videos. Zhang *et al.* [25] propose a semi-supervised model using HMMs for anomaly detection. Temporal dependencies are modelled using HMMs. The probability density function of each HMM state is assumed to be a GMM. Owen and Hunter [10] uses Self Organizing Feature Maps (SOFM) to learn normal trajectory patterns. While classifying trajectories, if the distance of the trajectory to its allocated class exceeds a threshold value, the trajectory is identified as anomalous.

The contribution of this paper is to present a mechanism for modeling of motion patterns from classified training data. A novel approach for model-based classification of trajectory patterns and anomaly detection is also presented. The proposed approach does not require specification of any manual threshold values for anomaly detection. Modeling, classification and the detection of anomalous trajectories is carried out in the parameter space with reduced computational burden.

3 Modified DFT-based trajectory representation

This section provides a brief overview of our trajectory representation scheme based on time series representation and modified DFT (DFT-MOD). Without loss of generality, we consider the projection of a moving object O in the (x, y) image plane. O registers its location (x_i, y_i) in (x, y, t) space at each instant of time $t = t_i$. The object trajectory *T(O)* is defined by the point sequence

$$T(O) = \{(x_1, y_1, t_1), (x_2, y_2, t_2), ..., (x_n, y_n, t_n)\} \tag{1}$$

where n is the sequence length. Hence, trajectories can be treated as motion time series.

In applications to fixed-camera surveillance, it is not necessary to apply shift and scale transformations to the data before model fitting. Trajectories are split into two 1-D time series in (x, t), (y, t) space. In tracking applications, observations are recorded at regular time intervals and hence we assume $t_i = i$ where i is the frame index. $T(O)$ can then be represented as two time series $X = x_i$, $Y = y_i$, $i = 1, ..., n$. We represent the trajectories using DFT-MOD based coefficient feature space representation. DFT-MOD is an extension of DFT [32]. DFT-MOD is generated by augmenting the DFT coefficients-based feature vector with some extra information regarding the length and starting location of the trajectory. These important information are not modelled correctly by DFT if we select only top few DFT coefficients which simply models the mean and trend of motion in the trajectory. All these factors may contribute to the fall-off in retrieval and classification accuracies, using simple DFT based dimensionality reduction, where starting point and duration of motion are important features for distinguishing different trajectories. The derivation of DFT-MOD based feature space representation of trajectories, using DFT coefficients, is specified as follows:

The n-point DFT of $\{x_i\}$, defined as a sequence $\{X_f\}$ of n complex numbers $(f = 0, ..., n-1)$, is given by eq. (2). A similar expression can be defined for $\{y_i\}$ as given in eq. (3).

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

$$Y_f = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} y_i exp(-j2\pi fi/n) \quad f = 0, 1, ..., n-1 \tag{3}$$

where j is the imaginary unit $j = \sqrt{-1}$, and X_f, Y_f are complex numbers with the exception of X_0, Y_0 which are real. Typically, the DFT sequence is truncated after m terms, $f = 0, ..., m-1$. More formally, let a_i and $\widehat{a_i}$ be the real and imaginary part of X and b_i and $\widehat{b_i}$ be the real and imaginary part of Y. Trajectories can be represented in the coefficient feature space by a $2(2m-1)$ dimensional vector of DFT coefficients $\mathbf{F}_{DFT}$, where

$$\mathbf{F}_{DFT} = [a_0, a_1, \widehat{a}_1, ..., a_{m-1}, \widehat{a}_{m-1}, b_0, b_1, \widehat{b}_1, ..., b_{m-1}, \widehat{b}_{m-1}] \tag{4}$$

The DFT-MOD based feature space representation of trajectory is then represented as:

$$\mathbf{F}_{DFT-MOD} = [\omega n, \omega x_0, \omega y_0, \mathbf{F}_{DFT}] \tag{5}$$

where (x_0, y_0) is the starting location approximated by taking the mean of first 10 points of the trajectory , n is the length of trajectory and ω is the scaling factor. When padding the starting point and length information to the DFT coefficients, it is important that these information be scaled down so that they do not dominate the trend information captures by the DFT coefficients. In all of our experiments, we used default scaling factor of $\omega = 0.5$. Trajectories can now be represented in the coefficient feature space by a $2(2m-1)+3$ dimensional vector of DFT-MOD coefficients $\mathbf{F}_{DFT-MOD}$.

4 Mediods-Based Modelling and Classification

In this section, a novel mechanism is proposed for modeling various patterns that are present in motion dataset. A pattern is modeled by a set of cluster centers of mutually disjunctive sub-classes (referred to as mediods) within the pattern. The algorithm for identification of mediods is based on the adaptation of neural gas based learning rule [33]. The resulting models of identified patterns can then be used to classify new unseen trajectory data to one of the modeled classes.

The proposed modelling technique, referred to as m-Mediods modeling, models the class containing n members with m mediods known *a-priori*. Let $DB^{(i)}$ be the classified training samples associated to pattern i and W the weight vector associated to each output neuron. The modeling algorithm comprises the following steps:

(1) Initialise the SOM network with a greater number of output neurons than the desired number of mediods m that we wish to produce. Larger number of output neurons results in high computational complexity whereas having the number of output neurons equivalent to m results in non-unifromly distributed mediods due to the problem of local minima. Based of a series of experiments, using patterns with different statistical properties, we have observed that a good number of output neurons for ini-

tialisation of SOM network can be obtained as:

$$\#_{output} = \begin{cases} \xi & if\, \xi < 150 \;\wedge \xi > (m \times 2) \\ m \times 2 & if \xi < (m \times 2) \\ 150 & if \xi > 150 \end{cases} \tag{6}$$

where $\xi = size(DB^{(i)})/2$.

(2) Initialize weight vectors W_i (where $1 \leq i \leq \#_{output}$) from the PDF $N(\mu, \Sigma)$ estimated from training samples in $DB^{(i)}$.

(3) Sequentially input feature vectors from DB and calculate the Euclidean distance between the training sample and the weight vectors associated to output neurons. Identify k Nearest Weights (k-NW) to input feature vector using:

$$k\text{-NW } (F, \mathbf{W}, k) = \{C \in \mathbf{W} | \forall R \in C, S \in \mathbf{W} - C, \\ \|F - R\| \leq \|F - S\| \wedge |C| = k\} \tag{7}$$

where F is the feature vector representation of training sample, $\mathbf{W}$ is the set of all weight vectors, C is the set of k closest weight vectors and $\|.\|$ is the Euclidean distance function. The value of k determines the number of output neurons that are nearest to F and will be updated in the specific iteration of learning process. For a given training cycle t, $k = \delta(t)$ where $\delta(t)$ is a neighborhood size function whose value decreases gradually over time as specified in eq. (10).

(4) Train SOM network by adjusting the weight vectors so that it starts representing the trend of the data. A subset of the weights (C) are updated using

$$W_c(t+1) = W_c(t) + \alpha(t)\zeta(j)(F - W_c(t)) \quad \forall W_c \in C \tag{8}$$

where W_c is the weight vector representation of output neuron c, j is the order of closeness of W_c to F $(1 \leq j \leq k)$, $\zeta(j,k) = exp(-(j-1)^2/2k^2)$ is a membership function that has value 1 when $j = 1$ and falls off with the increase in the value of j, $\alpha(t)$ is the learning rate of SOM and t is the training cycle index.

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

$$\alpha(t) = 1 - e^{\frac{2(t - t_{max})}{t_{max}}} \tag{9}$$

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

(6) In this step, the neighborhood size is decreased exponentially with training iterations as:

$$\delta(t) = \lceil \delta_{init}(1 - e^{\frac{2(t - t_{max})}{t_{max}}}) \rceil \tag{10}$$

where δ_{init} is the neighborhood size at the start of learning process. A series of experiments were conducted to determine the best value for δ_{init} which comes out to be 5.

(7) Repeat steps 3-5 for all the training iterations.

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

(9) Identify the closest pair of output neurons (i, j) (indexed by (a, b)) given by the condition

$$(a, b) = arg\, min_{(i,j)}\ [(W_i - W_j)^T (W_i - W_j)]^{\frac{1}{2}} \quad \forall i, j \wedge i \neq j \tag{11}$$

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

$$W_{ab} = \frac{mW_a + nW_b}{m + n} \tag{12}$$

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

(10) Iterate through steps 8-9 till the number of neurons gets equivalent to $\#_{mediods}$. Append weight vector W_k to the list of mediods $\mathbf{M}^{(i)}$ modeling the pattern i

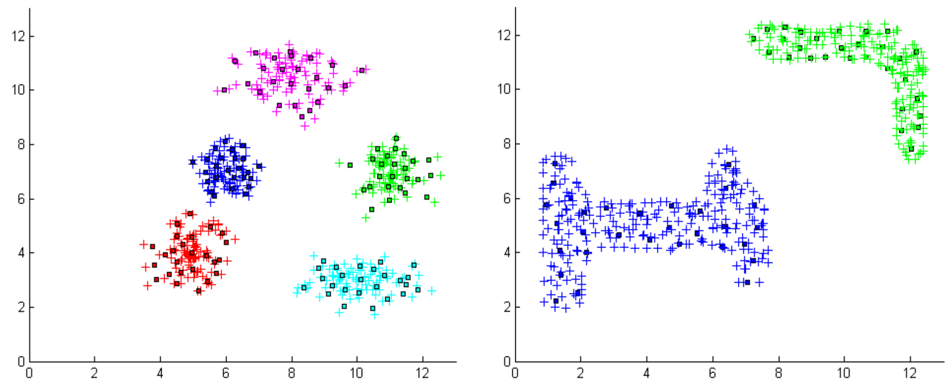
The space complexity of the proposed modeling algorithm is $O(n + \#_{output})$ where n is the number of training samples associated to the modeled pattern and $\#_{output}$ is the initial number of output neurons. In the presence of large number of training samples, $n \gg \#_{output}$ and the space complexity reduces to $O(n)$. The time complexity of our algorithm is $O(t_{max} * \#_{output} * log(\#_{output}))$ where t_{max} is the maximum number of training iterations and $\#_{output} * log(\#_{output})$ is the time complexity of ranking of nodes w.r.t. the closeness to the training sample in each iteration. The complexity $O(t_{max} * \#_{output} * log(\#_{output}))$ is much less than $O(N^2)$ for datasets with large number of training samples.

In order to visualize the modeling process, modeling of patterns using simulated datasets is demonstrated in Fig. 1. In Fig. 1, each point represents an instance from the dataset. Instances belonging to the same class are represented with same color. Squares super-imposed on each group of samples represent the mediods obtained using m-Mediods modeling algorithm to model the patterns.

After modeling the pattern c, the distance array $\mathbf{D}^{(c)}$ corresponding to model $\mathbf{M}^{(c)}$ is pre-computed, to be used later for anomaly detection, as follows:

(1) Identify the closest pair of mediods (i, j) (indexed by (p, q)) from $\mathbf{M}^{(c)}$ as follows:

$$(p, q) = arg\ min_{(i,j)} Dist(M_i, M_j) \qquad \forall i, j \wedge i \neq j \tag{13}$$



(a) (b)

Fig. 1. Mediods-based modeling of patterns in (a) simulated dataset with elliptical clusters (b) simulated dataset with arbitrary shaped clusters

where $Dist(M_i, M_j)$ is the euclidean distance function.

(2) Populate the distance array for the current number of mediods using

$$\mathbf{D}_l^{(c)} = (p, q, Dist(M_p, M_q)) \tag{14}$$

where l is the current number of mediods.

(3) Merge the most similar pair of mediods using

$$M_{pq} = \frac{mM_p + nM_q}{m+n} \tag{15}$$

where m, n are the number of sample trajectories mapped to mediods p and q respectively.

(4) Iterate through steps 1-3 till the number of mediods gets equivalent to 1.

Once the m-Mediods based model for all the classes have been learnt, the classification of new trajectories is performed by checking the closeness of said trajectory to the models of different classes. For this purpose, the trajectory is posed as a query to the entire set of mediods ($\mathbf{M}$) belonging to different classes. Identification of k Nearest Mediods (k-NM) to unseen trajectory can be specified as

$$\begin{aligned} k\text{-NM } (Q, \mathbf{M}, k) = \{ C \in \mathbf{M} | \forall R \in C, S \in \mathbf{M} - C, \\ Dist(\mathrm{Q}, R) \leq Dist(Q, S) \wedge |C| = k \} \end{aligned} \tag{16}$$

where Q is DFT-MOD based feature vector representation of unseen trajectory to be classified and C is the set of k closest mediods. A previously unseen trajectory Q is assigned to the same class, indexed by c, to which the majority of k nearest mediods belong.

5 Merged and Un-Merged Anomaly Detection

After identifying the closest activity pattern (c), it is checked to see if the unseen data is reasonably close to the closest activity pattern or not. In this section, we present two variations of a novel mechanism based on agglomerative approach for anomaly detection.

5.1 Merged Anomaly Detection

The description of merged anomaly detection algorithm is specified as follows:

(1) Initialize index l with the number of mediods (m) used to model a pattern.
(2) Identify the closest pair of mediods and their corresponding distance, for the current number of mediods l, using $\mathbf{D}^{(c)}$ as:

$$(p, q, d_{pq}) = \mathbf{D}_l^{(c)} \tag{17}$$

where d_{pq} contains the distance between mediods indexed by p and q.
(3) Identify the mediod, from $\mathbf{M}^{(c)}$, which is closest to the test sample Q. The closest mediod, indexed by r, is identified using:

$$r = arg\ min_k\ Dist(Q, M_k) \qquad \forall k \tag{18}$$

(4) Test trajectory Q is considered to be a valid member of class c if:

$$Dist(Q, M_r) \leq d_{pq} \tag{19}$$

(5) If the condition specified in eq. (19) is not satisfied, decrement the index l by 1.
(6) Merge the pair of mediods, indexed by (p, q), using

$$M_{pq} = \frac{mM_p + nM_q}{m + n} \tag{20}$$

where m, n are the number of sample trajectories mapped to mediods p and q respectively.
(7) Iterate steps 2-6 till l gets equivalent to the significance parameter τ. If the test trajectory Q has yet not been identified as a valid member of class c, it is considered to be an outlier and deemed anomalous.

The significance parameter τ determines the sensitivity of proposed anomaly detection algorithm to anomalies. Lower value of τ results in acceptance of more unusual data instances as normal members of one of the known classes and *vice versa*. Values of significance parameter τ lies in the range $1 \leq \tau < m$.

5.2 Un-Merged Anomaly Detection

This section specifies the un-merged version of the proposed anomaly detection algorithm. The un-merged algorithm is specifically proposed keeping in mind the patterns with non-convex, complex and arbitrary shaped distributions. In the following, the process of un-merged anomaly detection algorithm is specified:

(1) Identify the closest pair of mediods and their corresponding distance for the number of mediods l, equivalent to the the significant parameter τ, as:

$$(p, q, d_{pq}) = \mathbf{D}_l^{(c)} \tag{21}$$

where d_{pq} contains the distance between mediods indexed by p and q.

(2) Identify the mediod, from $\mathbf{M}^{(c)}$, which is closest to the test sample Q. The closest mediod, indexed by r, is identified using:

$$r = arg\ min_k\ Dist(Q, M_k) \qquad \forall k \tag{22}$$

(3) Test trajectory Q is considered to be a valid member of class c if:

$$Dist(Q, M_r) \leq d_{pq} \tag{23}$$

5.3 Relative Merits of Proposed Anomaly Detection Algorithms

Anomaly detection algorithms can be characterized in terms of the following attributes:

- Time Complexity
- Ability to deal with arbitrary shaped patterns
- Sensitivity to location of mediods
- Sensitivity to the value of significance parameter τ

For the ease in understanding of the comparative analysis, simulation of the working of proposed algorithms for arbitrary shaped patterns, using different values of τ, is presented in Fig. 2. Images on the left of Fig. 2 depicts normality region generated using merged anomaly detection algorithm and images on the right depicts normality regions generated using un-merged anomaly detection algorithm. Test sample is considered to be a normal member of the class if it lies within the normality region, else it is marked as anomalous. Fig. 2(a) depicts the m-Mediods based modeling of simulated dataset with arbitrary shaped clusters. Fig. 2(b)-2(f) represents the normality regions based on values of $\tau = 20, 16, 12$ and 8 respectively.

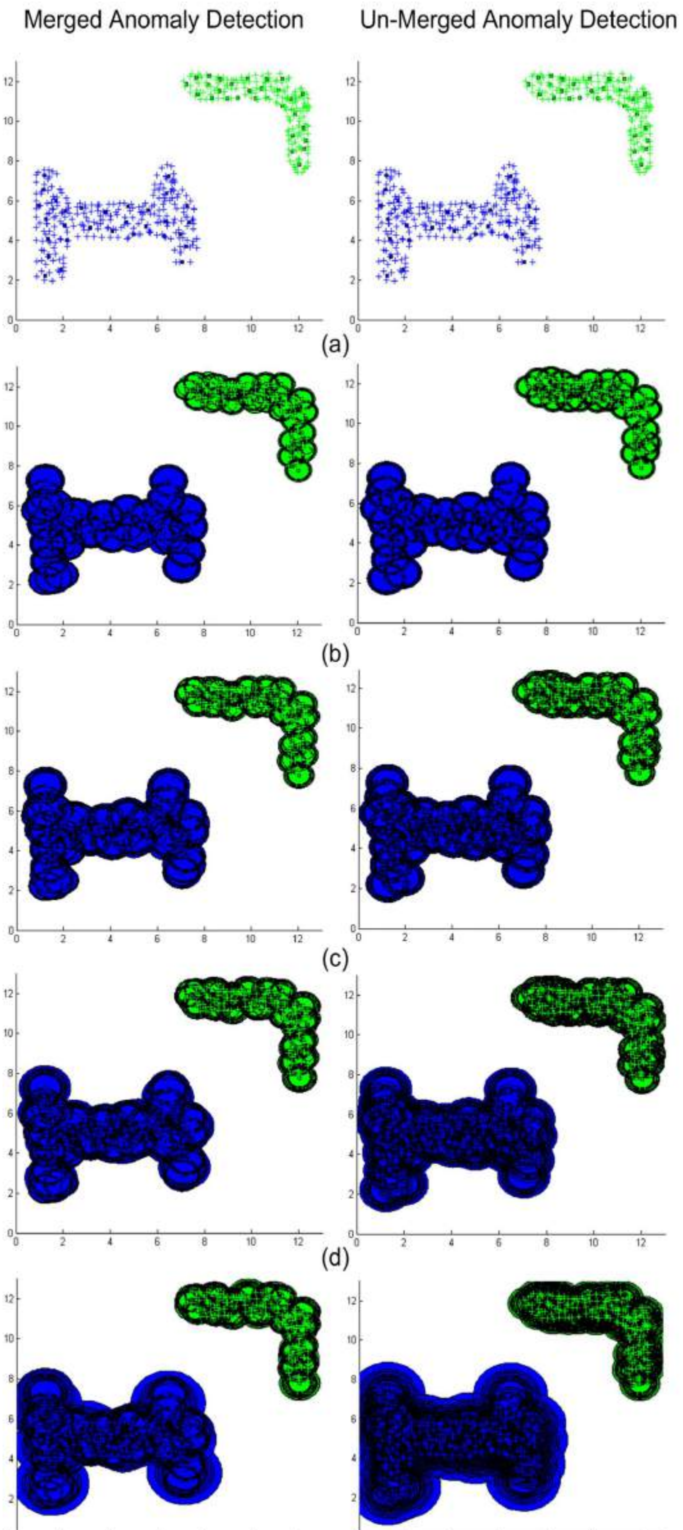


Fig. 2. Simulation of proposed mediods-based anomaly detection for non-convex clusters using different significance parameter τ. (a) m-Mediods modeling. (b)-(e) Normality regions based on values of $\tau = 20, 16, 12$ and 8 respectively.

Un-merged anomaly detection algorithm is very efficient as compared to merged anomaly detection algorithm as it is a non-iterative algorithm. The anomaly detection is carried out by comparing the distance of the test sample from its closest mediod with a single threshold value d_{pq} as identified using eq. (21). On the other hand, merged algorithm is an iterative process and in each iteration, the closest pair of mediods are merged till the normality condition, as specified in eq. (19), is not specified and the number of mediods is less than τ. The time complexity of merged anomaly detection is $O(m * log(m) - \tau * log(\tau))$. Another advantage of un-merged anomaly detection over its competitor is that it is well suited to detect anomalies in the presence of complex and arbitrary shaped patterns.

On the other hand, merged anomaly detection is less sensitive to the location of individual mediods, used to model the pattern, as compared to un-merged anomaly detection. The merging mechanism results in absorbing the merged mediods towards the center of the pattern and hence reducing the sensitivity of the anomaly detection algorithm to the location of mediods. This process also reduces the sensitivity of merged anomaly detection to the value of τ. As it can be visualized from Fg. 2, the normality regions generated using merged anomaly detection do not change significantly by changing the value of τ. On the other hand, normality regions generated using un-merged anomaly detection grow significantly with the decrease in the value of τ.

6 Localized Anomaly Detection

As discussed in section 5, the significance parameter τ determines the sensitivity of proposed anomaly detection algorithm to anomalies. This provides an intuitive way to for selecting possible values for τ. Single value of τ has been used globally for anomaly detection in the presence of different number and orientation of patterns, as visualized in Fig. 2. However, there are still open issues: (i) Anomaly detection in the presence of patterns with different orientations and scales (ii) Selection of appropriate value of τ for anomaly detection. (iii) Manual selection of the value of τ. In this section, a mechanism namely localized anomaly detection is proposed to address these issues. It enables us to automatically select a local significance parameter for each pattern taking into consideration the distribution of individual patterns.

Instead of selecting a single significance parameter τ, we proposed to determine a local significance parameter τ_c for each pattern c. Using a local significance parameter for each pattern enables to incorporate the local statistics of each pattern. Let $\mathbf{M}^{(c)}$ be the list of mediods used to model pattern c, the process of identification of τ_c is outlined as follows:

(1) Initialize significance parameter τ with the number of mediods (m) used to model pattern c.
(2) Sequentially input classified training data belonging to all classes through the anomaly detection system assuming that there is only one pattern c represented by mediods set $\mathbf{M}^{(c)}$.
(3) Increment false positive count FP each time when a classified training sample belongs to pattern c and is identified as anomalous
(4) Increment false negative count FN each time when a classified training sample does not belong to pattern c and is mis-classified to pattern c.
(5) Repeat step 2-4 for all the training samples.
(6) Calculate Significance Parameter Validity Index $(SPVI)$ to check the effectiveness of current value of significance parameter τ. The mathematical expression for SPVI is specified as:

$$SPVI(\tau) = \beta \times FP + (1-\beta) \times FN \qquad 0 \leq \beta \leq 1 \tag{24}$$

where β is a scaling parameter to adjust the sensitivity of classification and anomaly detection system to false positives and false negatives according to specific requirements.
(7) Decrement significance parameter τ by 1.
(8) Repeat step 2-7 till the value of τ gets equivalent to 1. Identify the value of significance parameter τ corresponding to lowest SPVI value as:

$$\tau_c = arg\ min_{\ \tau} \quad SPVI(\tau) \tag{25}$$

where τ_c is the localized significance parameter that will be used to check anomalies against pattern c.

Automatic identification of localized significance parameter τ_c, separately for each pattern c, also enables us to automatically identify the approximate number of mediods m to be used for good modelling of pattern c. The value of τ_c serve as the lower bound on the number of mediods m which can be specified as:

$$m = \tau_c + \wp \tag{26}$$

where $\wp$ is a constant. We ran a series of experiments, using patterns with different statistical properties, to determine the best value for $\wp$ which comes out to be 10. Values of $\wp$ greater than 10 normally results in using redundant mediods to model the pattern. This in turn will add up to the time complexity for merged anomaly detection without having any considerable effect on the accuracy.

7 Experimental Results

We now present some results to demonstrate the effectiveness of the proposed classification and anomaly detection techniques in the coefficient feature space.

7.1 Experimental Datasets

Experiments are conducted on three different synthetic and real life motion trajectory datasets. These include SIM_5 [14], LAB [14][30] and ASL [11][12][14][27][30][37] datasets. The characteristics of these datasets are summarized in Table 1.

Dataset	**Description**	**# of trajectories**	**Extraction method**	**Labelled (Y/N)**
SIM_5	Simulated datasets comprising of two dimensional coordinates generated from Gaussian distributions to form 5 clusters.	arbitrary	Simulation.	Y
LAB	Realistic dataset generated in the laboratory controlled environment for testing purposes. Trajectories can be categorised 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 95 different word classes with 70 samples per word.	6650	Extracting (x, y) coordinates of the mass of right hand from files containing complete sign information.	Y

Table 1
Overview of datasets used for experimental evaluation

7.2 Experiment 1: Evaluation of Proposed Model-based Classification and Anomaly Detection

The purpose of this experiment is to evaluate the performance of proposed model-based approach for classification of unseen data samples to one of the known patterns. The experiment demonstrates the ability of proposed classification system to act as an anomaly detection system. The experiment has been conducted on SIM_5 and LAB datasets. The training data from SIM_5 dataset is shown in Fig. 3. Gaussian parameters used to generate each of the clusters in Fig. 3 is presented in Table 2. The training data is obtained by generating 70 samples from each of the Gaussian distribution. Test data is obtained by generating 500 samples from a uniform distribution such that $(x, y) \in (U(1, 12), U(1, 12))$. On the other hand, LAB dataset is a classified motion dataset and contain anomalous trajectories within the dataset itself. Classified training data for this dataset is obtained by randomly selecting half of the trajectories from each of the normal patterns in the dataset. The remaining half of the trajectories from normal patterns along with anomalous trajectories are extracted and used as a test data.

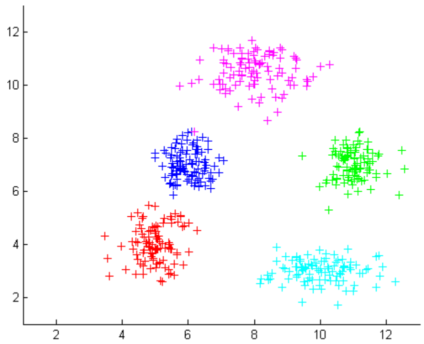


Fig. 3. SIM_5 dataset sampled from five Gaussians

Cluster Colour	Blue	Green	Red	Magenta	Cyan
Mean	(6,7)	(11,7)	(5,5)	(8,10)	(10,3)
Covariance	$\begin{pmatrix} 0.2 & 0 \\ 0 & 0.3 \end{pmatrix}$	$\begin{pmatrix} 0.2 & 0 \\ 0 & 0.3 \end{pmatrix}$	$\begin{pmatrix} 0.3 & 0 \\ 0 & 0.4 \end{pmatrix}$	$\begin{pmatrix} 0.7 & 0 \\ 0 & 0.4 \end{pmatrix}$	$\begin{pmatrix} 0.6 & 0 \\ 0 & 0.2 \end{pmatrix}$

Table 2
Gaussian parameters used to generate 5 clusters

For trajectory-based LAB dataset, trajectories are modeled using DFT-MOD based coefficient feature vectors. For synthetic SIM_5 dataset, the feature vector is composed of the original two dimensional points. Members of each class from the training data are used to generate model of normality associated to each normal pattern, using the algorithm as presented in section 4. Classes are modeled using 30 mediods per class. Once the m-Mediods (with $m = 30$) based model for all the classes have been learnt, classification of samples from the test data is done using the classifier as proposed in section 4. We have used different values of significance parameter τ for SIM_5 datasets. For LAB dataset, classification and anomaly detection is carried out by setting the value of $\tau = 10$. The classification and anomaly detection results for

SIM_5 dataset, using different values of significance parameter τ, are presented in Fig. 4. Training data is represented using '+' marker whereas classified normal samples are represented by small circles. For ease of visualisation, data points belonging to same class are represented with same colour. Samples from test data that are identified as anomalous are represented using black 'x' marker. It is apparent from Fig. 4 that proposed classification system correctly classifies test samples to known classes whilst identifying anomalies in the test data. Another important observation from Fig. 4 is that setting higher values of τ results in acceptance of only those instances as normal that are tightly bounded to normal classes. As τ decreases, we are less likely to detect anomalous patterns because it results in acceptance of more unusual data instances as normal members of one of the known classes. Using different significance levels therefore enables our proposed system to be adaptive to the density of data within a class.

After demonstrating the efficacy of proposed classification and anomaly detection approach on synthetic data, the experiment is then repeated on real life LAB dataset. Classification obtained by applying the proposed approach on LAB dataset is shown in Fig. 5. The matching of classification obtained for each trajectory with its ground truth shows that no trajectory is misclassified. Trajectories identified as anomalous using the value of $\tau = 10$ are shown in Fig. 6. It is clear from Fig. 6 that anomalous trajectories are significantly different from the normal motion patterns as shown in Fig. 5. These experimental results give evidence to the claim that the proposed model-based classification and anomaly detection system is an effective and robust approach that works well with real life motion datasets.

7.3 Experiment 2: Comparison of Proposed Classifier with Competitive Techniques

The purpose of this experiment is to compare the performance of proposed m-Mediods model-based approach for classification with competitive techniques. To establish a base case, we have implemented two different systems for comparison including GMM and Mahalanobis classifier. The experiment has been conducted on real life ASL dataset. Signs from different number of word classes are selected. Classified training data is obtained by randomly selecting half of the trajectories belonging to each of the selected words. The remaining half of the trajectories are then used as test data. Feature vector representation of ASL dataset is obtained as specified in earlier experiment. Patterns are modeled using 20 mediods per pattern. We have used the value of significance parameter $\tau = 8$ for anomaly detection. Modeling of patterns for Mahalanobis classifier is done by estimating a single multivariate Gaussian PDF for each class. Modeling of patterns and classification of unseen samples using GMM

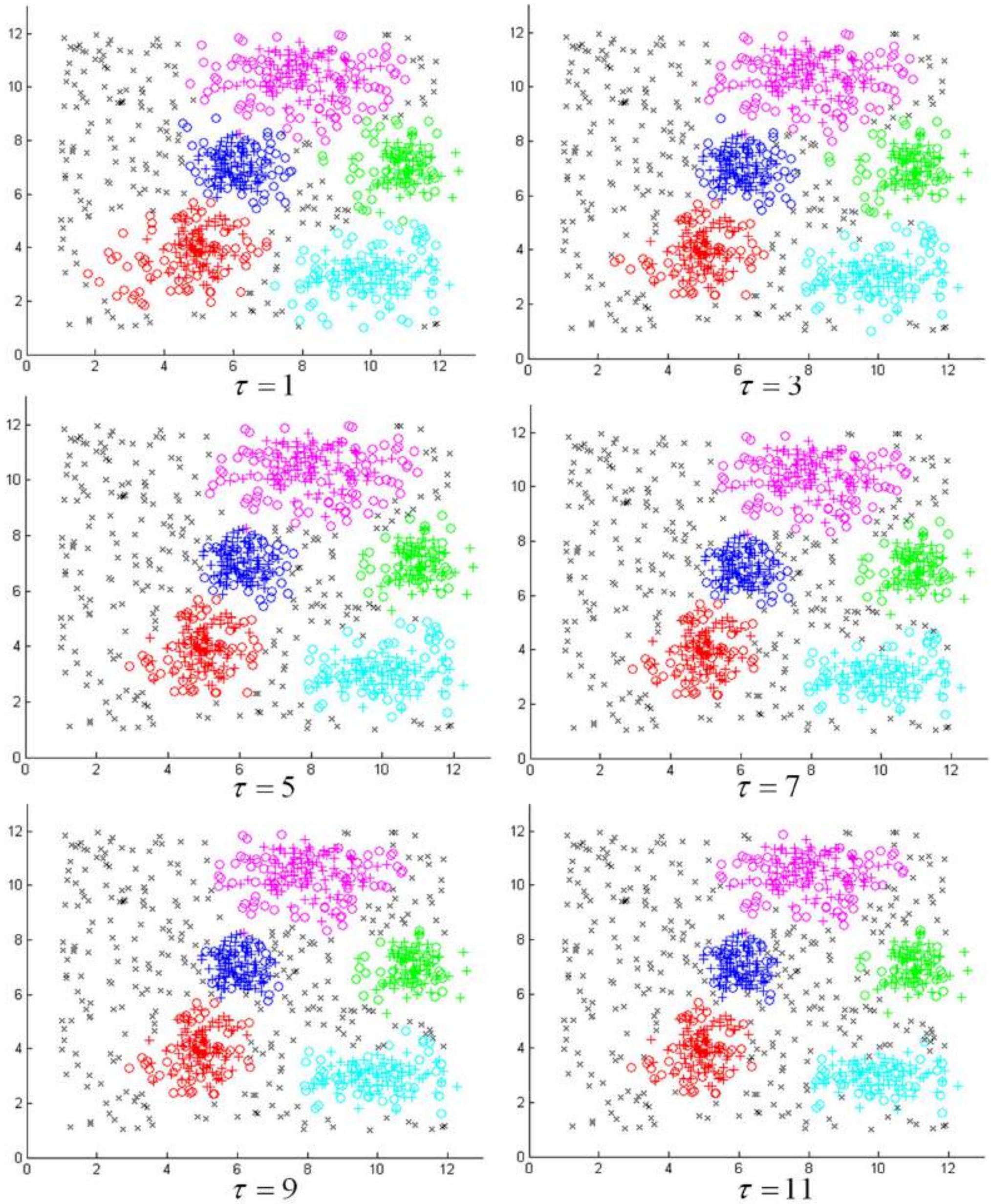


Fig. 4. Classification of test data, based on SIM_5 classes, using different values of significance parameter τ

is based on the approach as described in [37]. We have generated a separate GMM to model each class. The number of modes to be used for GMM-based modeling is automatically estimated using a string of pruning, merging and mode-splitting processes as specified in [37]. Once the models for all the classes have been learnt, the test data is passed to different classifiers 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 50 runs to reduce any bias resulting from favourable word selection.

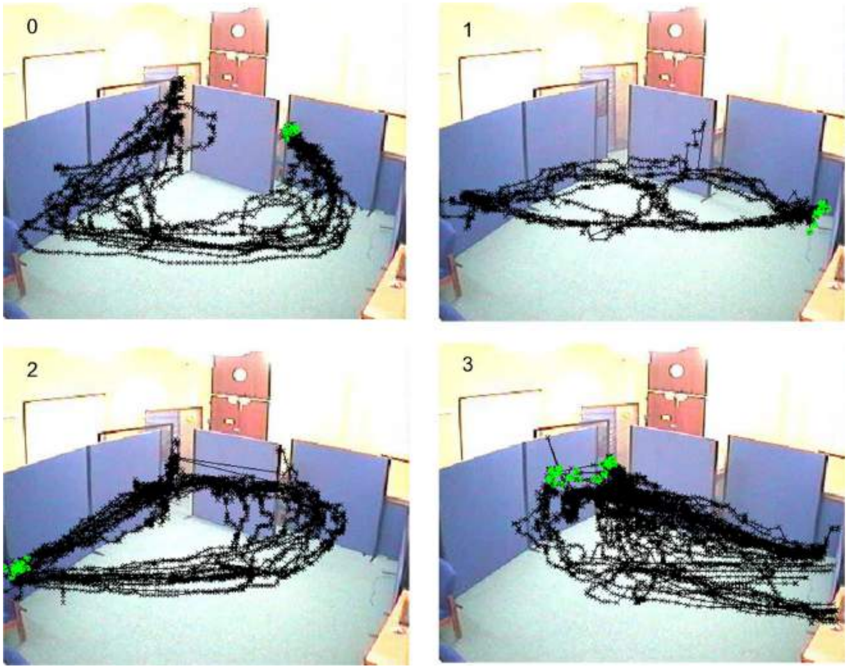


Fig. 5. Classification of test trajectories from LAB dataset

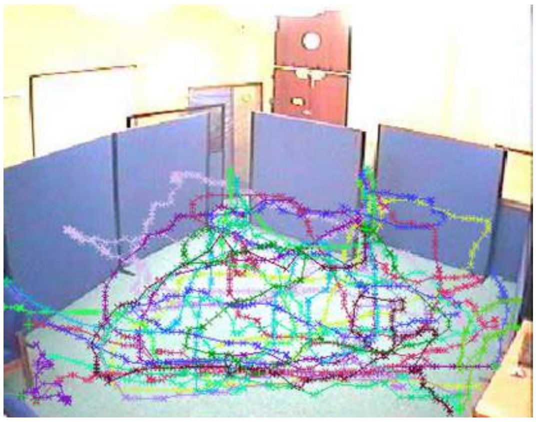


Fig. 6. Trajectories identified as anomalous from LAB dataset using proposed anomaly detection mechanism with $\tau = 10$

The classification accuracies obtained for different classifiers using various numbers of word classes from ASL dataset are shown in Table 3. Based on these results, we see that the proposed m-Mediods model-based classification yields superior classification accuracies. GMM produces good performance for lower number of classes. Increasing the number of classes results in degrading the performance of GMM-based classifier. The proposed classification approach gives better classification accuracies than Mahalanobis classifier as well. From Table 3, it can also be noted that the relative accuracy of the proposed classifier compared with GMM and Mahalanobis classifier increases with an increase in the number of classes; thus making it more scalable for larger number of classes. The superior performance of m-Mediods based modeling and classifi-

	ASL (#classes : #samples)				
	2 : 70	**4 : 140**	**8 : 280**	**16 : 560**	**24:840**
***m*-Mediods**	0.98	0.92	0.88	0.83	0.78
Mahalanobis	0.95	0.88	0.82	0.75	0.71
GMM	0.97	0.92	0.83	0.74	0.69

Table 3
Percentage classification accuracies for different number of classes from ASL dataset

cation mechanism, as compared to its competitors, can be explained by the fact that the proposed approach does not impose any restriction on underlying distribution of modeled patterns. The proposed algorithm can effectively model arbitrary shaped patterns as demonstrated in Fig. 1(b). On the other hand, the competitive approaches have some assumptions on the distribution of patterns (normally gaussian). As a result, these approaches will not generate accurate models of complex patterns thus resulting in lower classification accuracies as compared to the proposed m-Mediods based approach.

Similar experiment with ASL dataset (using similar experimental settings) has been conducted by Bashir *et al.* [27] using their proposed GMM and HMM-based classification system. They reported classification accuracies of 0.96, 0.92, 0.86 and 0.78 for 2, 4, 8 and 16 word classes respectively. Comparing these classification accuracies with the results obtained using our approach, we see that m-Mediods model-based classifier performs better than GMM and HMM-based recognition system [27] even though our proposed classification approach is conceptually simpler and computationally less expensive.

7.4 Experiment 3: Quantitative Evaluation of Proposed Model-based Anomaly Detection Algorithms

The purpose of this experiment is to evaluate and compare the performance of proposed anomaly detection algorithms. These include: (i) Global Merged Anomaly Detection (GMAD), (ii) Localized Merged Anomaly Detection (LMAD), (iii) Global Un-merged Anomaly Detection (GUAD), and (iv) Localized Un-merged Anomaly Detection (LUAD). Experiments using GMAD and GUAD are repeated using different value of global significance parmater τ. The proposed anomaly detection algorithms are compared with statistical test [30] and one-class classifier based anomaly detection [38]. Naftel *et al.* [30] performs anomaly detection through analysis of the covariance structure of patterns. Hotellings 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 T^2 statistic based on the Mahalanobis distance can be calculated as:

$$T^2 = \frac{n}{n+1}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) \tag{27}$$

where n is the number of samples in class i, μ_i is the class mean and Σ_i is the class covariance. Given an input feature vector of dimension p in the coefficient space, a test sample is identified as anomalous if:

$$T^2 > \frac{(n-1)p}{n-p}F_{p,n-p} \tag{28}$$

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 10^{th} percentile of the $F_{p,n-p}$ distribution.

Tax *et al.* [38] performs anomaly detection by generating model of one class (referred to as target class) and distinguishing it from samples belonging to all other classes. There are different options available for generating the model of target class including support vector machine (SVM), gaussian mixture model(GMM) etc. For SVM-based one class classifier (OCC-SVM), we have used RBF kernel for the generation of support vector machine based model of target class. For GMM-based one class classifier (OCC-GMM), we have used the approach as specified in Experiment 2 to generate the GMM-based model.

The experiment has been conducted on the real life ASL dataset. Signs from different numbers of word classes are selected. Classified training data for the ASL dataset is obtained by randomly selecting half of the trajectories belonging to each of the selected words. The remaining half of the trajectories are then used as test data. Trajectories from the ASL dataset are modeled using DFT-MOD based coefficient feature vectors. Feature vectors from the training data are then used to generate models as required by the different classification approaches. The m-Mediods based model of each class is generated using the algorithm as presented in section 4. Patterns are modeled using 20 mediods per pattern.

Once the models for all the classes have been learnt, the test dataset is then passed through the proposed anomaly detection system. 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 experiment is repeated with different numbers and combinations of word classes. Each anomaly detection experiment is averaged over 50 runs to reduce any bias resulting from favorable word selection.

The percentage of instance vectors from ASL dataset, correctly identified as anomalous, are shown in Fig. 7. The anomaly detection accuracies are presented for different anomaly detection algorithms using various number of word classes from ASL dataset. The results summarized in Fig. 7 demonstrate the superiority of anomaly detection using a local significance parameter for each pattern. The anomaly detection accuracies obtained using localized anomaly detection algorithms (LMAD and LUAD) is higher than the global anomaly detection algorithms (GMAD and GUAD). LMAD and LUAD also performs better than SVM and GMM based one-class classifiers (OCC-SVM and OCC-GMM). Another important observation from Fig. 7 is that the merged anomaly detection algorithms yield better accuracies as compared to un-merged anomaly detection algorithms, given a consistent way to select the value of τ. This is to be expected given that the merged anomaly detection algorithms are less sensitive to the location of individual mediods and to the value of significance parameter τ. Also, the accuracies of proposed anomaly detection algorithms are much better as compared to Naftel's method.

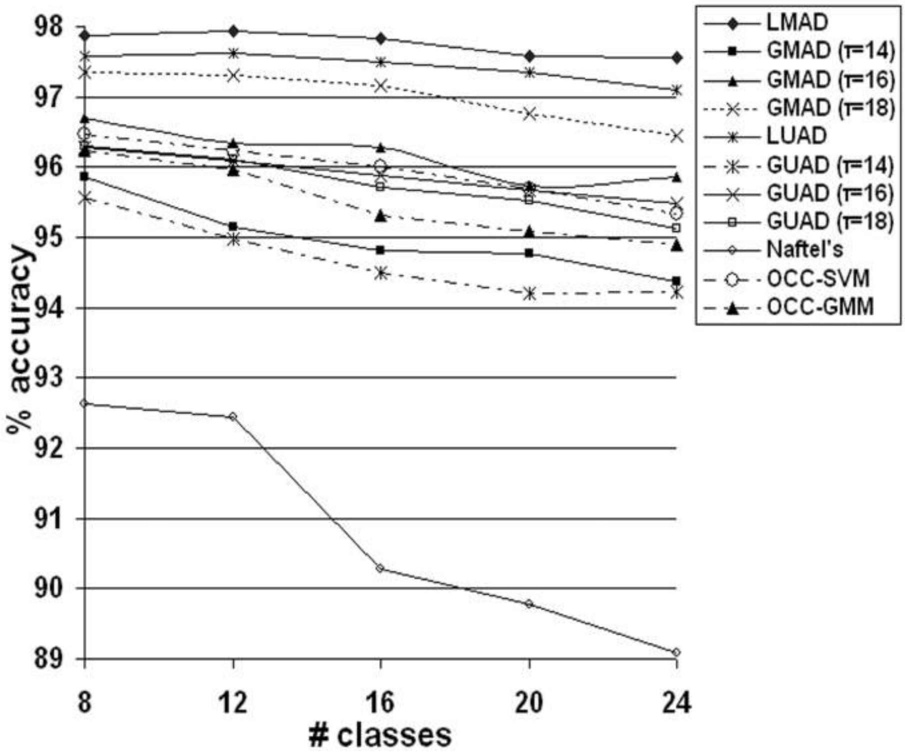


Fig. 7. Percentage anomaly detection accuracies for different number of classes from ASL dataset

8 Discussion and conclusions

In this paper, we have presented a framework for modeling and classification of trajectory-based motion patterns. A novel approach, referred to as m-Mediods modeling, is proposed that models the class containing n members with m-Mediods known *a-priori*. The strength of this technique is its ability to model complex patterns without imposing any restriction on the shape of patterns. Once the m-Mediods model for all the classes have been learnt, the classification of new trajectories and anomaly detection can be performed by checking the closeness of said trajectory to the models of different classes using hierarchical classifier. Four variants of a novel algorithm, namely GMAD, LMAD, GUAD and LUAD, are proposed in this paper for detection of anomalies in motion trajectory datasets.

Experimental results are presented to show the effectiveness of proposed m-Mediods based classification and anomaly detection system. Matching the classification results with labelled training data shows that the test instances are classified correctly and filtered instances (anomalies) are sufficiently distant from all the known classes. Comparison of proposed classifier with competitive techniques demonstrates the superiority of our proposed approach as it performs consistently better than commonly used Mahalanobis, GMM and HMM-based classifiers.

Experiments are also conducted to show the effectiveness of proposed anomaly detection algorithms. Anomaly detection results for different classes of ASL datasets, using different variants of proposed anomaly detection algorithm, are presented. It has been shown that anomaly detection using localized significance parameter τ gives better anomaly detection accuraices as compared to the approach using global value of τ for all the classes. Localized τ enables the anomaly detection system to adapt to the normality distribution of individual classes. Matching the accuracies of merged and un-merged anomaly detection algorithm shows that merged anomaly detection gives more accurate results as compared to its un-merged counterpart. Merged algorithms are robust to the location of individual mediods used to model the patterns and are less sensitive to the value of τ for the reasons outlined in section 5.3. Comparison of proposed anomaly detection algorithms with an existing approach demonstrates the superiority of our approach approach as they consistently perform better for different number of classes.

References

[1] Z. Aghbari, K. Kaneko, A. Makinouchi, Content-trajectory approach for searching video databases, IEEE Transanction on Multimedia, vol. 5, no. 4, December 2003, pp. 516-531.

[2] S.F. Chang, W. Chen, J.M. Horace, H. Sundaram, D. Zhong, A Fully Automated Content based Video Search Engine Supporting Spatiotemporal Queries, IEEE

Transactions on Circuits and System for Video Technology, vol. 8, no. 5, September 1998, pp. 602-615.

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

[4] C.T. Hsu, S.J. Teng, Motion trajectory based video indexing and retrieval, IEEE International Conference on Image Processing, vol. 1, 2002, pp. 605-608.

[5] Y. Jin, F. Mokhtarian, Efficient video retrieval by motion trajectory, Proceedings of British Machine Vision Conference, Kingston, September 2004, pp. 667-676.

[6] S. Khalid, A. Naftel, Evaluation of matching metrics for trajectory based indexing and retrieval of video clips, Proceedings of IEEE WACV, Colorado, USA, January 2005, pp. 242-249.

[7] C. Shim, J. Chang, Trajectory based video retrieval for multimedia information systems, Proceedings of ADVIS, 2004, pp. 372-382.

[8] N. Johnson, D. Hogg, Learning the distribution of object trajectories for event recognition, Proceedings of British Conference on Machine Vision, 1995, pp. 582-592.

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

[10] J. Owens, A. Hunter, Application of the Self-Organising Map for Trajectory Classification, Proceedings of Third IEEE International Workshop on Visual Surveillance, Dublin, Ireland, July 2000, pp. 77.

[11] F.I. Bashir, A.A. Khokhar, D.Schonfeld, View-invariant motion trajectory based activity classification and recognition, ACM Multimedia Systems, special issue on Machine Learning Approaches to Multimedia Information retrieval, 2006, pp. 45-54.

[12] F.I. Bashir, A.A. Khokhar, D.Schonfeld, HMM based Motion Recognition System using Segmented PCA, IEEE International Conference on Image Processing, Genova, Italy, Sept. 11-14, 2005, pp. 1288-1291.

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

[14] S. Khalid, Motion based behaviour learning, profiling and classification in the presence on anomalies, Pattern Recognition, Available online: http://dx.doi.org/10.1016/j.patcog.2009.04.025, 2009.

[15] K. Chan, A. Fu, Efficient time series matching by wavelets, Proc. of International Conference on Data Engineering, Sydney, March 1999, pp. 126-133.

[16] Y. Cai, R. Ng, Indexing Spatio-Temporal Trajectories with Chebyshev Polynomials, ACM SIGMOD/PODS Conference, France, June 13-18, 2004, pp. 599-610.

[17] C. Shim, J. Chang, Content based retrieval using trajectories of moving objects in video databases, Proceedings of IEEE 7th International Conference on Database Systems for Advanced Applications, 2001, pp. 169-170.

[18] R. Agarwal, C. Faloutsos, A. Swami, Efficient Similarity Search in Sequence Databases, 4th International Conference of Foundations of Data Organization and Algorithms, Evanston, Illinois, USA, October 1993, pp. 69-84.

[19] J. Alon, S. Sclaroff, G. Kollios, V. Pavlovic, Discovering clusters in motion time-series data, Proc. IEEE CVPR, vol. 1, June 2003, pp. I-375- I-381.

[20] T. Brotherton, T. Johnson, G. Chadderdon, Classification and novelty detection using linear models and a class dependent elliptical basis function neural network, Proceedings of IJCNN Conference, vol. 2, 4-9 May, 1998, pp. 876-879.

[21] S. Roberts, L. Tarassenko, A probabilistic resource allocating network for novelty detection, Neural Computation, vol. 6. 1994, pp, 270-284

[22] D.Y. Yeung, C. Chow, Parzen window network intrusion detectors, IEEE International Conference on Pattern Recognition, , Canada, 2002, pp. 385 388.

[23] T. Xiang, S. Gong. Video behaviour profiling and abnormality detec- tion without manual labelling, IEEE International Conference on Computer Vision, vol. 2, London, UK, October 2005, pp. 1238 1245.

[24] T. Xiang, S. Gong, Incremental visual behaviour modelling, 6th IEEE Workshop on Visual Surveillance, Graz, May 2006, pp. 6572.

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

[26] Y. Yacoob, M.J. Black, Parameterized Modeling and Recognition of Activities, Computer Vision and Image Understanding, vol. 73 (2), Feb. 1999, pp. 232-247.

[27] F.I. Bashir, A.A. Khokhar, D.Schonfeld, Object Trajectory-Based Activity Classification and Recognition Using Hidden Markov Models, IEEE Transactions on Image Processing, vol. 16, no. 7, 2007, 1912-1919.

[28] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, S. Maybank, A system for learning statistical motion patterns, IEEE Transactions on Pattern Analysis and Machine Learning, vol. 28, no. 9, September 2006, pp. 1450-1464.

[29] W. Hu, D. Xie, Z. Fu, W. Zeng, S. Maybank, Semantic based surveillance video retrieval, In IEEE Transactions on Image Processing, April 2007, pp. 11681181.

[30] S. Khalid, A. Naftel, Classifying Spatiotemporal Object Trajectories using Unsupervised Learning in the Coefficient Feature Space, Multimedia Systems, Vol. 12(3), Dec. 2006, pp. 227-238.

[31] N. Rea, R. Dahyot, A. Kokaram, Semantic Event Detection in Sports through motion understanding, Proceedings of Conference on Image and Video Retrieval, Dublin, Ireland, July 21-23, 2004.

[32] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, Fast Sub-sequence Matching in Time-Series Databases, Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data, 1994, pp. 419-429.

[33] T. M. Martinetz, S. G. Berkovich, K. J. Schulten, Neural-gas network for vector quantization and its application to time-series prediction, IEEE Transactions on Neural Networks, vol. 4, no. 4, 1993, 558-569.

[34] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, New York, 1995.

[35] J. Owens, A. Hunter, Novelty Detection in Video Surveillance Using Hierarchical Neural Networks, Proceedings of ICANN, Madrid, Spain, August 28-30, 2002, pp. 1249-1254.

[36] CAVIAR test sequences [Online]. Available: http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/

[37] F. I. Bashir, A. A. Khokhar, D. Schonfield, Automatic object trajectory based motion recognition using gaussian mixture models, In IEEE International Conference on Multimedia and Expo, Netherland, July 6-8, 2005, pp. 1532-1535.

[38] D.M.J. Tax, One-class classification, PhD thesis, Delft University of Technology, http://ict.ewi.tudelft.nl/davidt/thesis.pdf, June 2001.