Multi-feature object trajectory clustering for video analysis

Nadeem Anjum and Andrea Cavallaro

Abstract— **We present a novel multi-feature video object trajectory clustering algorithm that estimates common patterns of behaviors and isolates outliers. The proposed algorithm is based on four main steps, namely the extraction of a set of representative trajectory features, non-parametric clustering, cluster merging and information fusion for the identification of normal and rare object motion patterns. First we transform the trajectories into a set of feature spaces on which Mean-shift identifies the modes and the corresponding clusters. Furthermore, a merging procedure is devised to refine these results by combining similar adjacent clusters. The final common patterns are estimated by fusing the clustering results across all feature spaces. Clusters corresponding to reoccurring trajectories are considered as normal, whereas sparse trajectories are associated to abnormal and rare events. The performance of the proposed algorithm is evaluated on standard data-sets and compared with state-of-the-art techniques. Experimental results show that the proposed approach outperforms state-of-the-art algorithms both in terms of accuracy and robustness in discovering common patterns in video as well as in recognizing outliers.**

I. Introduction

Video object trajectories are important elements for the analysis and the representation of behaviors ([1], [2]). Clustering is a key component of trajectory analysis when instead of modeling and analyzing the motion of an individual object, multiple trajectories are processed together to discover the inherent structures of activities in a video. This process aims to classify trajectories into two major classes, namely normal trajectories, which belong to common patterns, and outliers, which exhibit a deviant behavior. Clustering groups unlabeled data in such a way that elements in a single cluster have similar characteristics, and elements in different clusters have the most dissimilar characteristics ([3], [4], [5]). After transforming the trajectories into an appropriate feature space, trajectories are grouped together based on a proximity measure (i.e., a similarity or dissimilarity measure).

In general trajectory analysis algorithms use only one feature space for clustering ([6], [7], [8], [9], [10], [11]). Even when more features are used, they are not processed simultaneously ([12], [13]). This can result in a coarse cost function defined by the proximity measure, thus leading to a local minima problem. One way to overcome this problem is to use a stochastic optimization algorithm. However, the convergence properties (such as the radius of convergence) of such algorithms are limited [14].

In this paper, we propose a partitional trajectory clustering framework that combines internally a fuzzy clustering approach based on multiple features before generating a final crisp partition. We use multiple feature spaces simultaneously to obtain a higher degree of descriptiveness of the trajectories as opposed to using one feature space only. Each feature space is then regarded as the empirical probability density function (*pdf*) of the represented parameter and modes in each space correspond to the maxima of the *pdf*. Once the modes are determined, the members can be associated to each mode to form the clusters. We use Mean-shift in each feature space for mode-seeking and clustering. The clustering results of Mean-shift in each space are then refined by applying a cluster merging procedure. The final clustering is obtained by analyzing the clustering results from each feature space. The process results in the definition of the clusters' structures along with the fuzzy membership of a trajectory to the final clusters. The clusters with small number of associated elements and the trajectories that are far from the clusters' center are considered as outliers. Figure 1 shows the flow diagram of the proposed approach.

The rest of this paper is organized as follows: Sec. II covers the state-of-the-art for trajectory representation. Section III details the proposed approach. Section IV describes the experimental setup, discusses the results achieved by the proposed approach and compares the proposed approach with existing algorithms. Finally, in Sec. V we draw the conclusions.

II. Trajectory representation: prior work

The choice of a suitable pattern representation provides the core for a clustering algorithm. This section discusses and compares existing approaches for trajectory representation, which are summarized in Table I. Pattern representations can be divided into two groups, namely *supervised* and *unsupervised*. Supervised techniques rely on the information supplied by training samples. A Hidden Markov Model (HMM) is a supervised trajectory representation approach, in which the state transition matrix represents the dynamics of a trajectory. Porikli [15] performs trajectory clustering using eigenvector analysis on the HMM parameter space. Alon *et al.* [16] allows each sequence to belong to more than a single HMM with some probability and the hard decision about the sequence class membership is deferred until a later stage for final clustering. Parameterized-HMMs [17] and coupled-HMMs [18] are also used to recognize more complex events such as moving object interactions. Although HMMs are robust to

The authors acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC), under grant EP/D033772/1. N. Anjum and A. Cavallaro are with the Multimedia and Vision Group, Queen Mary University of London, e-mail: {nadeem.anjum,andrea.cavallaro}@elec.qmul.ac.uk

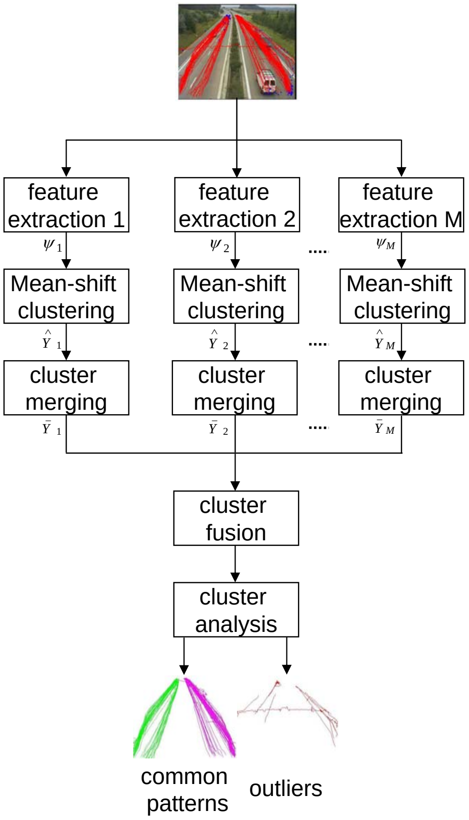


Fig. 1: Flow diagram of the proposed object trajectory clustering algorithm

dynamic time warping, the structures and probability distributions are highly domain-dependent. Moreover, the parameter space increases considerably with the complexity of the events, as more hidden states are required for modeling. Statistical model-based approaches ([19], [8]) are also supervised methods that associate different statistical properties to each trajectory class. These approaches tend to be sensitive to the initial choice of the model parameters that may lead to poor cluster models. The approaches also require prior knowledge of the number of clusters.

Unlike supervised techniques, unsupervised representations do not require training samples. Self-organizing maps (SOMs) are one of such examples that use unsupervised learning to represent high-dimensional data patterns in a low dimensional space, while keeping the topological properties of the input space ([20], [9], [21]). Once SOM nodes are organized, all the data associated with a given node may be made available via that node. Learning Vector Quantization (LVQ) is an alternative approach to SOM, where classes are represented by prototypes and a new input trajectory is associated with the corresponding nearest prototype [22], [23]. For real motion sequences, the convergence of these techniques is slow and the learning phase is usually carried out off-line. The Trajectory Directional Histogram (TDH) is another representation to encode the statistical directional distribution of the trajectories [12]. However, this feature alone does not suffice because it does not encode spatial information. Therefore, two trajectories that are far on the image plane shall be clustered together if they have similar directional histograms. In the clustering literature, Principal Components Analysis (PCA) has been used extensively to reduce the dimensionality of the data set prior to clustering while extracting the most important data variations. Bashir *et. al* ([13], [24]) represent trajectories as a temporal ordering of sub-trajectories. These sub-trajectories are then represented by their PCA coefficients for optimally compact representation. In [25], the authors extracted the sub-trajectories from points of curvature change and again represented the sub-trajectories with PCA coefficients. PCA works well for data with a single Gaussian distribution. For mixture of Gaussian distributions, Independent Components Analysis (ICA) is used to obtain a compact representation. Antonini *et. al* [7] transform the input trajectories using ICA and then use the Euclidean distance to find the similarities among trajectories. Both PCA and ICA require an accurate estimation of the noise covariance matrix from the data, which is generally a difficult task. Furthermore, in their standard form, they do not contain high-order statistical information and therefore the analysis is limited to second-order statistics.

TABLE I: State-of-the-art methods for trajectory representation (Key. *HMMs*: Hidden Markov Models; *PRMs*: Probabilistic and Regression Models; *STFA*: Spatio-temporal Function Approximations; *PCA*: Principal Components Analysis; *ICA*: Independent Components Analysis; *TDH*: Trajectory Directional Histogram)

Category	**Rep.**	**Application**	**Ref.**
Supervised	HMMs	Vehicle tracking	[15]
		Nose tracking	[16]
		Gesture recognition	[17]
		Behavior analysis	[18]
	PRMs	Hand tracking	[8]
		ECG and cyclone trajectories	[19]
Unsupervised	STFA	Pedestrian tracking	[9]
		Behavior analysis	[20]
		Pedestrians scene	[23]
		Speech signal analysis	[22]
		Vehicle tracking	[21]
	PCA	Hand tracking,	[25]
		sports and	[24]
		traffic analysis	[13]
	ICA	Pedestrians counting	[7]
	TDH	Vehicle tracking	[12]

III. Proposed approach

A. Feature extraction

To overcome the problems associated with single-feature clustering discussed in Sec. I, we propose a multi-feature trajectory clustering algorithm that improves the overall clustering performance by exploiting the descriptiveness of multiple feature spaces. Let a trajectory T_j be represented as $T_j = \{(x_j^i, y_j^i); i = 1, \ldots, N_j\}$, where (x_j^i, y_j^i) is the estimated position of the j^{th} target on the image plane, N_j is the number of trajectory points and $j = 1, ..., J$. J is number of trajectories. Note that the trajectories are likely to have different length. Let $F_m(.)$ be a feature extraction function defined as $F_m(.) : T_j \rightarrow \Psi_m^{d_m}$, with $m = 1, ..., M$, where M is total number of feature spaces. F_m maps every

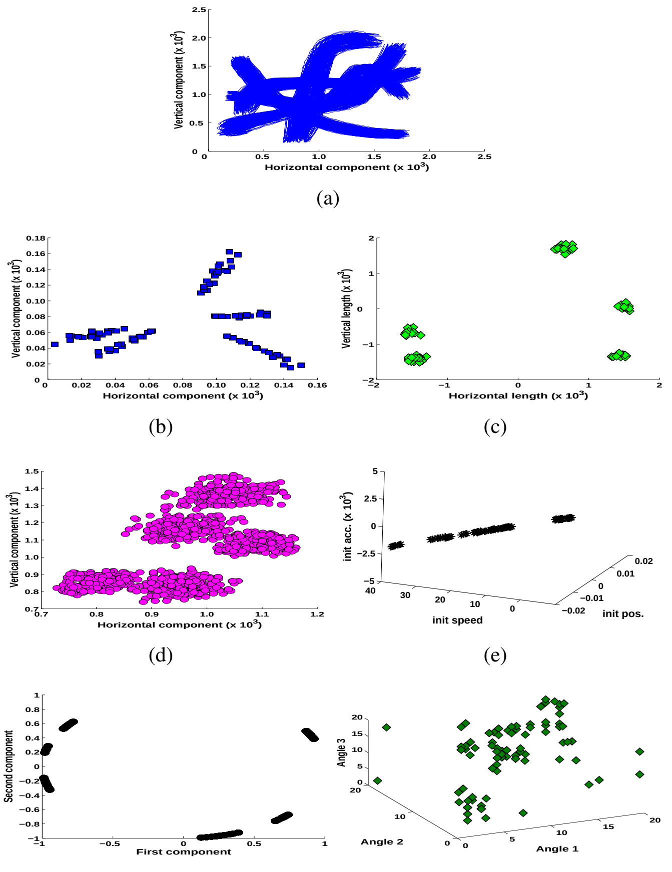


Fig. 2: (a) Sample set of 1100 trajectories and their projections on the following feature spaces: (b) average velocity, (c) directional distance, (d) trajectory mean, (e) combination of initial position, speed and acceleration, (f) principal components and (g) trajectory turns (three dominant angles)

trajectory T_j to a d_m-dimensional feature space $\{\Psi_m^{d_m}\}_{m=1}^{M}$. The feature spaces are treated independently in order to avoid the need for normalization, which is required if features are processed together, and to help in analyzing multi-domain (spatial and angular) non-orthogonal feature spaces together. Moreover, it also provides a framework for parallel clustering using different features and integrates them together to avoid problems comparing non-similar features.

Feature selection depends upon the application and each feature space helps in finding the coarse structures from the input data, which are then integrated for fine-grained clustering. For this purpose, we investigate spatial and angular trajectory representations, namely (a) the average target velocity, (b) the directional distance, (c) the target trajectory mean, (d) the combination of the initial target position, its speed and its acceleration, (e) the PCA of the trajectory points, and (f) trajectory turns.

The average target velocity, $\overline{\mathbf{v}}_\mathbf{j}$, describes the rate of change of the j^{th} object position. This feature helps separating the trajectories of objects moving at varying pace. $\overline{\mathbf{v}}_\mathbf{j}$ is defined as

$$\overline{\mathbf{v}}_j = \frac{1}{N_j - 1} \sum_{i=1}^{N_j - 1} \left(x_j^{i+1} - x_j^i, y_j^{i+1} - y_j^i \right). \qquad (1)$$

The directional distance, $\mathbf{d_j}$, of the j^{th} object is considered as the second feature to extract the horizontal and vertical length of a trajectory. Moreover $\mathbf{d_j}$ also encodes the direction of motion (moving toward or away from the camera). This feature helps distinguishing longer trajectories from shorter ones and also trajectories in opposite directions. $\mathbf{d_j}$ is calculated as

$$\mathbf{d_j} = \left(x_j^{N_j} - x_j^0, y_j^{N_j} - y_j^0 \right). \qquad (2)$$

The third spatial feature encodes the horizontal and vertical components of j^{th} trajectory mean ($\mathbf{m_j}$). This feature works well to distinguish the trajectories belonging to different regions on the image plane and is calculated as

$$\mathbf{m_j} = \frac{1}{N_j} \sum_{i=1}^{N_j} \left(x_j^i, y_j^i \right). \qquad (3)$$

In order to model the shape of the j^{th} trajectory irrespective of its length and sample points we use polynomial regression as fourth feature space. The matrix notation for the model estimation of T_j is written as

$$\mathbf{y_j}^{'} = \left(\begin{array}{ccccc} 1 & \mathbf{x_j} & (\mathbf{x_j})^2 & ... & (\mathbf{x_j})^\rho \end{array} \right) (\beta_0 \beta_1 ... \beta_\rho)^T + \epsilon, \qquad (4)$$

where the first term of the R.H.S is a N_jxρ matrix with $\mathbf{x_j} = \{x_j^i\}_{i=1}^{N_j}$, the second term is ρx1 vector and the last term is a N_jx1 vector. The output vector is also N_jx1 vector. The goal here is to find the optimal values of β_i for which $\epsilon = |\mathbf{y}^{'} - \mathbf{y}|$ becomes minimum. The process requires an inherent trade-off between accuracy and efficiency. As the degree of the polynomial increases, the fit grows in accuracy but only up to a point. We find the appropriate degree by starting with a first degree polynomial and continually monitoring the fit to see if the degree needs to be increased. If so, the regression is restarted with the degree incremented by one. Here we have fixed $\rho = 2$ as an increase of the value of ρ does not affect the overall accuracy. The three coefficients $(\beta_0, \beta_1, \beta_2)$ maps to the initial position, speed and acceleration of the object.

In order to consider the variation information of each trajectory, we apply PCA on sample points of each trajectory. For simplicity of notation, let $\mathbf{x_j} = (x_j, y_j)$, then T_j can be rewritten as $T_j = \{\mathbf{x_j^i}\}_{\mathbf{i=1}}^{\mathbf{N_j}}$. After subtracting the trajectory mean ($\mathbf{m_j}$) from each trajectory point,

$$\widetilde{T}_j = \{\mathbf{x_j^i} - \mathbf{m_j}; i = 1, ..., N_j\}, \qquad (5)$$

we consider the covariance matrix as

$$\mathbf{\Xi_j} = \frac{1}{N_j} \widetilde{T}_j \widetilde{T}_j^T. \qquad (6)$$

The eigenvalue decomposition of Ξ_j results in eigenvalues, $\alpha = \{\alpha_i\}_{i=1}^{N_j}$, and corresponding eigenvectors, $\varphi = \{\varphi_\mathbf{i}\}_{i=1}^{N_j}$. After sorting α in descending order, we consider the first two $\varphi_\mathbf{k}, \varphi_\mathbf{l} \in \varphi$, corresponding to the top two eigenvalues, $\alpha_k, \alpha_l \in \alpha$, as most of the variation lies in these two components.

Lastly, to consider the sharpness of turns in T_j, the directional histograms ($\hbar_j$) are calculated using the method presented in [12] as

$$\mathbf{\hbar_j} = H(\theta_\mathbf{j}^\mathbf{i}), \qquad (7)$$

where $H(.)$ is a histogram function calculated over the directional angles ($\theta_{\mathbf{j}}^{\mathbf{i}} = tan^{-1}(y_j^{i+1} - y_j^i / x_j^{i+1} - x_j^i)$). We take the indices of the top three peaks of $\bar{\mathbf{h}}_{\mathbf{j}}$ as they describe the dominant angles in the trajectory. Figure 2 shows the relative placement of each trajectory of a given data-set in each feature space.

Once the trajectories are transformed into multiple feature spaces, the next step is to analyze each feature space to form clusters. Without prior knowledge on the type of target we are observing, we consider all the features equally important and give them equal weights for the final clustering. The detailed discussion about the clustering process is given in the next section.

B. Clustering

Each feature space is regarded as the empirical probability density function (*pdf*) of the distribution of the trajectories in a particular feature space [26]. We use Mean-shift on the normalized feature spaces to find the modes of the *pdf* and then we associate each trajectory with the nearest mode to form the clusters.

Mean-shift is a clustering technique that climbs the gradient of a probability distribution to find the nearest dominant mode or peak ([27]). Let $\chi_l \in \Psi_j^{d_j}$; $l = 1, ..., L$ be a set of L data points. The multivariate density estimator $\hat{f}(x)$ is defined as

$$\hat{\mathbf{f}}(\mathbf{x}) = \frac{1}{Lh_m^{d_m}} \sum_{l=1}^{L} K\left(\frac{x - \chi_l}{h_m^{d_m}}\right), \tag{8}$$

where $h_m^{d_m}$ is the bandwidth of the kernel, $K(.)$. The choice of $h_m^{d_m}$ plays an important role in Mean-shift clustering. We employ an incremental procedure to select this value. Initially, $h_m^{d_m}$ is set to 10% of each dimension of m^{th} feature space and it iteratively increases to 80%. The lower bound prevents clusters containing a single trajectory, while the upper bound avoids a cluster with all trajectories grouped together. Although a smaller $h_m^{d_m}$ produces less biased density estimator, it increases the variance. In order to find the compromise between the two quantities we used Mean Integrated Squared Error (MISE) [27], defined as

$$MISE(x) = \int E((\chi_l - \hat{\mathbf{f}}(\mathbf{x}))^2)dx, \tag{9}$$

The value of $h_m^{d_m}$ for which $MISE(x)$ is minimum is considered to be the optimal one. Moreover, $K(.)$ in Eq. (8) is defined as

$$K(x) = \left\{ \begin{array}{ll} \frac{1}{2V_m^{d_m}}(d_m + 2)(1 - x^T x) & \text{if } x^T x < 1 \\ 0 & \text{otherwise} \end{array} \right. , \tag{10}$$

where $V_m^{d_m}$ represents the volume of a d_m*-dimensional* unitary sphere. The density gradient estimate of the kernel can be written as

$$\hat{\nabla}\mathbf{f}(\mathbf{x}) = \nabla\hat{\mathbf{f}}(\mathbf{x}) = \frac{1}{Lh_m^{d_m}} \sum_{l=1}^{L} \nabla K\left(\frac{x - \chi_l}{h_m^{d_m}}\right). \tag{11}$$

Equation (11) can be re-written as

$$\hat{\nabla}\mathbf{f}(\mathbf{x}) = \frac{d + 2}{h_m^{d_m} V_m^{d_m}} \left(\frac{1}{L_c} \sum_{\chi_l \in S(x)} (\chi_l - x)\right), \tag{12}$$

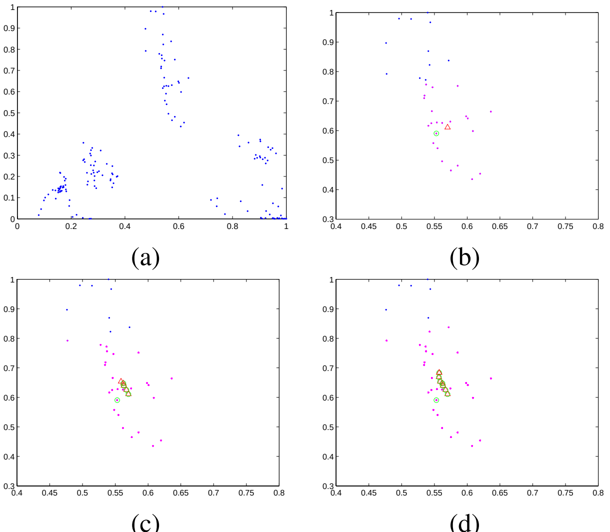


Fig. 3: Sample Mean-shift clustering results on the first two principal components of the highway traffic sequence $S3$. Each point represents a trajectory. (a) Initial trajectory representation. Results (zoom) of the (b) 1^{st}, (c) 5^{th}, and (d) 8^{th} iteration of the mode seeking procedure. (Key. Blue: unprocessed points. Magenta: points within the kernel bandwidth. Green: mode-seeking path. Triangles: mode at each iteration)

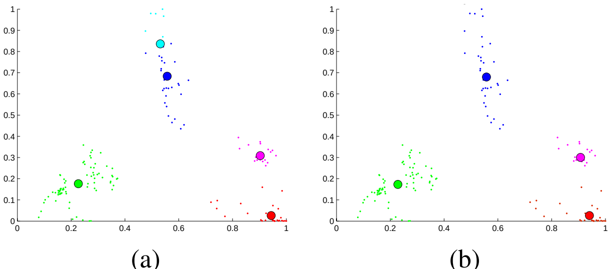


Fig. 4: Sample cluster merging results. (a) Initial trajectory clustering result (5 clusters); (b) final clustering result after cluster merging (4 clusters)

where *S(x)* is a hypersphere of radius $h_m^{d_m}$, with volume $h_m^{d_m} V_m^{d_m}$, centered in x and containing L_c data points. The Mean-shift vector, $\zeta_{\mathbf{h}}(\mathbf{x})$, is defined as

$$\zeta_{\mathbf{h}}(\mathbf{x}) = \frac{1}{L_c} \sum_{\chi_l \in S(x)} (\chi_l - x), \tag{13}$$

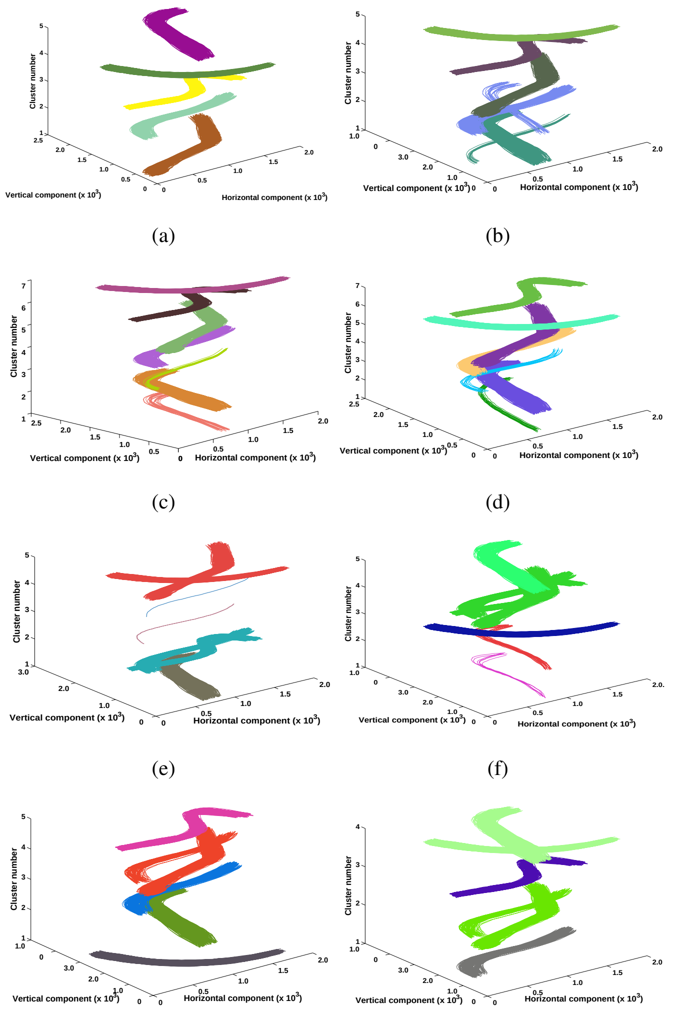
and, using Eq. (12), we can express $\zeta_{\mathbf{h}}(\mathbf{x})$ as

$$\zeta_{\mathbf{h}}(\mathbf{x}) = \frac{h_m^{d_m} V_m^{d_m}}{d_m + 2} \frac{\hat{\nabla}\mathbf{f}(\mathbf{x})}{\hat{\mathbf{f}}(\mathbf{x})}. \tag{14}$$

The output of the Mean-shift procedure is the set of data points associated to each mode. This process is illustrated in Fig. 3. Initially, the mode seeking process starts by fixing a trajectory as a seed point; then after the Mean-shift process converges to the local mode, all the trajectories within the bandwidth, $h_m^{d_m}$, of the kernel, $K(.)$, are assigned to that mode [28]. These trajectories are then not considered for future iterations. The next seed point is selected randomly from the unprocessed trajectories. The process terminates when all trajectories are assigned to a corresponding local mode.

Algorithm 1 Generalized cluster fusion process

```
$\xi$ = {$\aleph_1$,$\aleph_2$,...,$\aleph_M$}: number of clusters for each feature space
$C_j^i$: $j^{th}$ cluster in the $i^{th}$ space;
$\aleph$: number of final clusters;
Compute: $\aleph$
    $\aleph$ = median($\xi$)
Compute: $C^f$
    $l \leftarrow$ find $\xi$ =$\aleph$
$C_i^f = C_i^l : i = 1$,...,$\aleph$
for $n = 1$ to $M$ do
    if $n \neq l$
    for $i = 1$ to $\aleph$ do
        find $\hat{\nu} = \arg\max_j |C_i^l \cap \{C_j^n\}_{j=1}^{\xi_n}|$
        $C_i^f = (C_i^f \cap C_{\hat{\nu}}^n)$
    end for
    end if
end for
```



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 5: Comparison of single and multiple feature clustering results: (a) independent multiple features (proposed), (b) concatenated multiple feature, (c) average velocity only, (d) directional distance only, (e) trajectory mean only, (f) combination of initial position, speed, acceleration only, (g) principal components only and (h) trajectory turns (the three dominant angles) only

A small bandwidth causes an increase in the number of modes and a larger variance, which results in unstable variations of local density. This artifact can be eliminated by merging the closely located modes [29]. In this work, we merge adjacent clusters if their modes are apart by less than $h_m^{d_m}$+$0.1\%(h_m^{d_m})$. A sample result of cluster merging is shown in Fig. 4, where two modes in Fig. 4(a) (upper region: dark and light blue) are located within the threshold value and are merged as shown in Fig. 4(b).

C. Cluster fusion

The final partitioning of the trajectories is obtained after analyzing the refined clustering results from each feature space. The integration of the clusters consists of three steps, namely the estimation of the final number of clusters, the establishment of the correspondence between clusters in different feature spaces, and the association of each trajectory to a final cluster.

Let $\xi = \{\aleph_k\}_{k=1}^{M}$ be the set containing the number of clusters for each feature space $\Psi_k^{d_k}$. The final number of clusters $\aleph$ is selected as the median value of the set ξ.

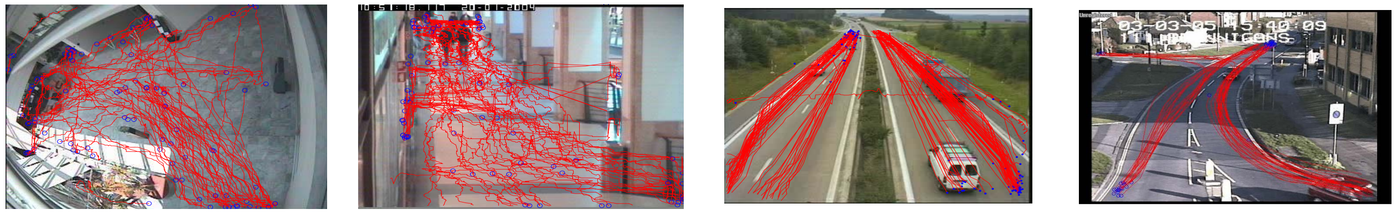
After selecting the final number of clusters, we estimate the structure of clusters as characterized by a single mode; we model each cluster with a univariate Gaussian with bandwidth of the kernel defining the variance of the cluster itself. In order to find the structure of each cluster, we start the process with a feature space $\Psi_l^{d_l} \in \{\Psi_k^{d_k}\}_{k=1}^{M}$ such that $\xi_l = \aleph$. The initial parameters of the final clusters ($C_i^f; i = 1, ..., \aleph$) are those defined by $\Psi_l^{d_l}$. To refine the parameters according to the results of the other feature spaces, we find the correspondence of $\Psi_l^{d_l}$ with all the other feature spaces $\Psi_n^{d_n}$ with $\Psi_n^{d_n} \in \{\Psi_k^{d_k}\}_{k=1}^{M}$ and $n \neq l$. Let $\hat{\nu}$ be the index of the cluster in $\Psi_n^{d_n}$ that has the maximum correspondence (maximum number of overlapping elements) with the i^{th} cluster of $\Psi_l^{d_l}$:

$$\hat{\nu} = \arg\max_j \left|C_i^l \cap C_j^n\right|, \qquad (15)$$

where $i = 1, ..., \aleph$ and $j = 1, ..., \aleph_n$. Also, C_i^l and C_j^n represent the i^{th} and j^{th} clusters in $\Psi_l^{d_l}$ and $\Psi_n^{d_n}$, respectively. Then C_i^f is updated by taking the overlapping elements, $C_i^f = (C_i^f \cap C_{\hat{\nu}}^n)$. This process continues for all features spaces (see Algorithm 1). This results in $\aleph$ clusters consisting of all the trajectories that are consistent across all feature spaces, and are therefore considered to represent a reliable structure for each cluster. At this point we associate to the final clusters the trajectories ($T' \subseteq \{T_j\}_{j=1}^{J}$) which are not consistent across all the feature spaces. To this end, we calculate a conditional probability of $T_o \in T'$ generated from the given cluster model as

$$p(T_o|C_k^f) = \frac{1}{M}\sum_{i=1}^{M}\frac{1}{\sqrt{2\pi}\sigma_{f,k}}e^{-(\frac{\mu_{i,j}-\mu_{f,k}}{\sigma_{f,k}})^2}, \qquad (16)$$

where $\mu_{i,j} = \frac{1}{|C_j^i|}\sum_{j=1}^{|C_j^i|}\mathbf{m_j}$ and $\mu_{f,k} = \frac{1}{|C_k^f|}\sum_{k=1}^{|C_k^f|}\mathbf{m_k}$ (see Eq. (3)) are the mean values of C_j^i and C_k^f respectively; $\sigma_{f,k} = \frac{1}{|C_k^f|}\sum_{k=1}^{|C_k^f|}\mathbf{\Xi_k}$ (see Eq. (6)) is the standard deviation of C_k^f. The motivation behind these parameters is that each cluster is characterized by a single mode with the spread of the cluster



(a) (b) (c) (d)

Fig. 6: Trajectory data-sets: CAVIAR video sequences (a) S1 and (b) S2, Highway traffic video sequences (c) S3 and (d) S4

can be represented by σ. Note that the decision is made at the cluster level and not at the feature space level, thus removing the dependence on dimensionality or normalization. T_o will be assigned to C_k^f if

$$p(T_o|C_k^f) > p(T_o|C_l^f), \tag{17}$$

where l=1, ..., $\aleph$ and $l \neq k$.

Figure 5 shows a comparison of trajectory clustering results using single and multiple (concatenated and independent) features. Single features (Fig. 5(c-h)) are not always capable of providing an effective data representation as, for example, the average velocity encodes directional information only, without providing any spatial relationship information among the trajectories. This results in grouping trajectories even if they are located far from each other. On the other hand, features such as the directional distance and trajectory mean contain spatial information only, without encoding any variation information. For this reason, these features do not generate clusters of trajectories with similar motion patterns that are not spatially close. The integration of the different features and their properties improves the overall trajectory clustering results by generating a more meaningful grouping, as shown in Fig. 5(b). However, in Fig. 5(b) there are still 33 wrongly clustered trajectories. The proposed approach further improves the performance by first post-processing the results (cluster merging) at feature space level and then by fusing the post-processed results at later stage (Fig. 5(a)).

D. Outlier detection

An outlier trajectory is the one that deviates from other trajectories as it is a result of an abnormal. In this work we focus on identifying two types of outlier trajectories: those existing in dense regions but exhibiting a different behavior from the common pattern; and those located in sparse regions.

To detect the first type of outliers, we use a distance-based approach. If a trajectory $T_j^{'} \in C_k^f$, with trajectory mean $\mu_{T_j^{'}}$, lies far from the center ($\mu_{f,k}$) of the cluster it belongs to, then it is considered as outlier, i.e.,

$$\frac{|\mu_{T_j^{'}} - \mu_{f,k}|}{\sigma_{f,k}} > \tau, \tag{18}$$

where $\tau = 0.95$.

To detect the second type of outliers, we identify trajectories belonging to sparse regions by considering the size of their cluster. If a cluster has few associated trajectories and can not be merged with a nearby cluster, then it is considered to be a set of outliers. Here the threshold value is set to the 10% of the cardinality of the cluster containing the median number of associated elements.

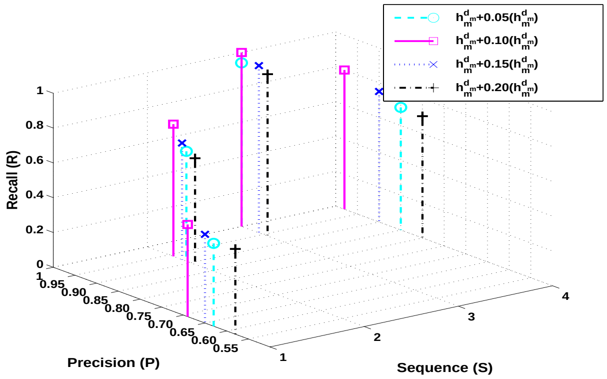


Fig. 7: Sensitivity analysis at the variation of the cluster merging criterion

IV. Experimental results

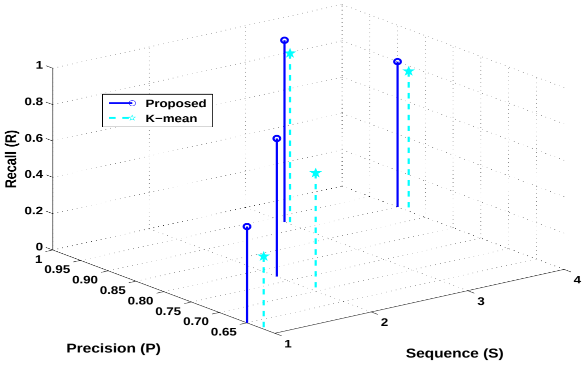
A. Experimental setup

Four standard video sequences are used to evaluate the proposed approach and compare it with state-of-the-art trajectory clustering algorithms. The first two sequences are from the *CAVIAR* data-set[1]. These videos are captured in a lobby ($S1$, 90 trajectories, 384 x 288 pixels, 25 Hz) and in a corridor of a shopping center ($S2$, 84 trajectories, 384 x 288 pixels, 25 Hz). Two traffic sequences are from the MPEG–7 ($S3$, 134 trajectories[2], 352 x 288 pixels, 25 Hz) and from the CLEAR [30] ($S4$, 47 trajectories, 720 x 480 pixels, 25 Hz) data-sets, respectively. Figure 6 shows the cumulated trajectories superimposed on a key-frame of each test sequence.

For $S1$, we are interested in clustering the trajectories into four main groups consisting of 18, 13, 22 and 10 trajectories and representing, respectively, the trajectories belonging to the upper-region on the image plane (starting from the top-right and ending at top-left), the trajectories starting and ending at the bottom-right of the image plane, the trajectories from the center to the bottom-right of the image and, finally, the trajectories starting and ending at the center-left region of the image. For $S2$, we are interested in finding five dominant regions consisting of 15, 28, 10, 9 and 10 trajectories and representing, respectively, the trajectories from the center-left to the bottom-right on the image plane, the trajectories on the top-left corner, the trajectories from the bottom-right to the center-left, the trajectories from the top-left corner on the image plane to the bottom region and, finally, the trajectories that start and end in the bottom-half of the image. For the $S3$ traffic sequence the objective is to determine two dominant

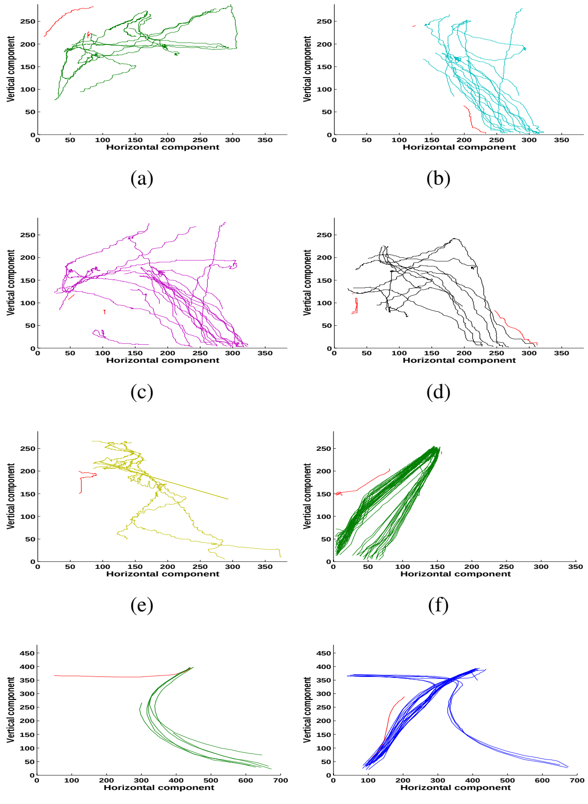
[1] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/

[2] http://www.tele.ucl.ac.be/PROJECTS/MODEST/



(a)

Fig. 8: Comparison of the proposed approach and K-mean



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Fig. 9: Sample outlier detection improvements by applying the proposed outlier detection criteria on the SOM clustering results (red: new outliers identified in a cluster)

patterns consisting of 55 and 47 trajectories. For $S4$, the goal is to cluster trajectories into three groups consisting of 20, 17 and 7 trajectories. The remaining trajectories are considered as outliers. Note that in S3 and S4 there are true outliers e.g., a person crossing the highway (S3) and a vehicle stopping on the road (S4). In S1 and S2, the outliers are short trajectories generated by window shoppers (S1) or people which are very far from the cameras (S4).

To objectively assess the clustering performance we use *Precision* (P) and *Recall* (R). For the i^{th} final cluster, C_i^f, P is calculated as

$$P = \frac{|C_i^f \cap \Gamma_i|}{|C_i^f|}, \tag{19}$$

and R as

$$R = \frac{|C_i^f \cap \Gamma_i|}{|\Gamma_i|}, \tag{20}$$

where $|.|$ is the cardinality of a cluster.

In addition to the evaluation of the results obtained using the original trajectories, we perform a series of robustness tests in which we evaluate the clustering results after corrupting the input data (missing data and noisy data). In the first test, we reduce the trajectory sampling rate by 2, 3, 4 and 5 steps. In the second test, we add Gaussian noise with standard deviation equal to 5%, 10%, 15% and 20% of the longest trajectory of each set. In all further experiments proximity measure for cluster merging is set to $h_m^{d_m} + 0.1\%(h_m^{d_m})$ (See Fig. 7).

B. State-of-the-art approaches

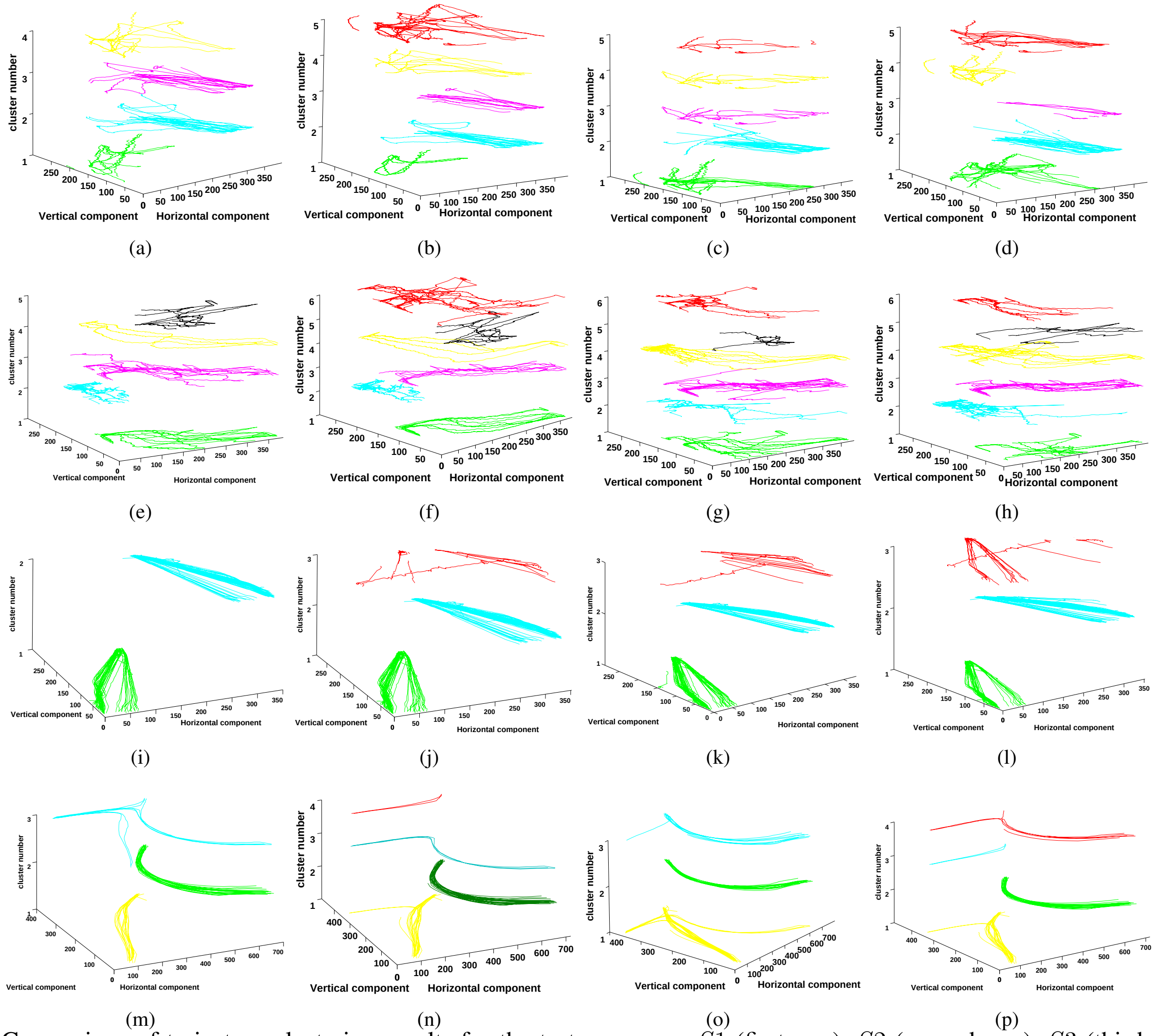
As a baseline test, initially, we compare the performance of the proposed approach with the standard K-mean clustering [31]. Afterward, we compare it with two state-of-the-art trajectory clustering algorithms, based on Self-Organizing Maps [9] and on Trajectory Directional Histograms [12].

C. Performance evaluation

Figure 8 shows the comparison of the proposed approach and K-mean. The graph shows the average P and R results for the four test sequences. The results show that the proposed approach has 3% and 13% higher precision and recall for $S1$. For $S2$, it has 7% and 9% higher precision and recall compared to K-mean. Similarly, its P and R results are better by 1% and 6% for $S3$. Finally, for $S4$ the proposed approach has higher P and R results by 2% and 3%, respectively. The subsequent discussion focuses the performance comparison of the proposed approach with SOM and TDH based approaches.

Figure 10 compares the clustering results of the proposed approach, SOM and TDH on the four test sequences ($S1$, $S2$, $S3$ and $S4$). To facilitate the visualization, the trajectory clusters are shown in 3D, where each vertical layer corresponds to a separate cluster. Results for the three approaches are shown next to the ground truth. In each plot, the top-most layer corresponds to the detected outliers. Furthermore, the P and R results for the three approaches are also compiled in Table II (hereinafter ”the Table”).

In Fig. 10 (b-d), for the first group of trajectories in $S1$, unlike the proposed approach, SOM and TDH generate clusters which contain additional trajectories that either starts from the center or bottom regions on the image plane. For the second group, the clustering results for the proposed approach outperform SOA and TDH. In particular, the clustering results of TDH and SOM for the third group are contaminated by additional trajectories that either start from the bottom or the center-left image plane regions. From the Table it is possible to notice that for the first three clusters, the proposed approach not only clusters the trajectories accurately (R values) compared to SOM and TDH, but also avoids the additions of other trajectories into a group (P values). However, for C_4, the accuracy of TDH is 10% better than that of the proposed approach, though it adds more outliers and trajectories belonging to other groups. The precision values are important as they give an indication of how a particular



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Fig. 10: Comparison of trajectory clustering results for the test sequences $S1$ (first row), $S2$ (second row), $S3$ (third row) and $S4$ (fourth row). First column: ground truth. Second column: results of the proposed approach. Third column: SOM results. Fourth column: TDH results. The top-most layer ('red') shows the detected outliers for each approach. Detailed results for each cluster are available at http://www.elec.qmul.ac.uk/staffinfo/andrea/traj.html

algorithm treated the outliers: the larger P, the higher the relevant information contained and therefore fewer outliers are part of a cluster. P shows that in C_4 TDH has 20% extra trajectories compared to the proposed approach. The Table also shows that, compared to SOM and TDH, the proposed approach is more robust to missing data and noisy trajectories. On average, the overall degradation in accuracy (recall value) is 3% for the proposed approach when the trajectory is reduced by 5 sampling steps. On the other hand, the degradations for SOM and TDH are 8% and 12.5%, respectively. Similarly, for noisy trajectories, on average the proposed approach degrades by 9.5% when 20% noise is added to the trajectories. For SOM and TDH, the degradation is 13.5% and 20.5%, respectively. The complexity of the trajectories affects SOM negatively. The reduced sampling and noisy trajectories affect TDH the most, as its performance degrades substantially compared to the other two approaches.

For $S2$, Fig. 10 (g-h) shows that for the first type of motion the clustering results of both SOM and TDH contain extra trajectories that either start from the bottom or top regions. Similar results can be observed for other clusters. On average, the proposed approach (Fig. 10 (f)) outperforms SOM and TDH by 32.6% and 31.8%, respectively. The robustness tests also reveal that the proposed approach efficiently clusters the trajectories compared to SOM and TDH, especially in C_2 and C_5 for SOM, and in C_4 and C_5 for TDH.

For $S3$, Fig. 10 (j) shows that the proposed approach correctly identifies the clusters. There are noticeable errors in the SOM results (Fig. 10 (k)), for example the left-most outlier trajectory is considered as part of the 1^{st} cluster. Similarly, for TDH (Fig. 10 (l)), there are several trajectories that should be part of the 1^{st} cluster, but are treated as outliers. For C_1, the precision of the proposed approach is 96%. Both SOM and TDH have a lower precision for the same cluster. SOM tends to accept outliers as part of a normal cluster compared to the proposed approach and TDH. For C_2 the precision and recall values for the proposed approach are 90% and 100%, respectively. For SOM, 9% of the trajectories in C_2

TABLE II: Clustering precision and recall comparison for $S1$, $S2$, $S3$ and $S4$

	Proposed										SOM [9]										TDH [12]									
	C_1		C_2		C_3		C_4		C_5		C_1		C_2		C_3		C_4		C_5		C_1		C_2		C_3		C_4		C_5	
	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R
S1	.62	.50	.56	.77	1	.54	.43	.30	-	-	.36	.44	.33	.54	.23	.23	.13	.20	-	-	.45	.50	.36	.61	.44	.18	.21	.40	-	-
↓ 2	.60	.50	.54	.76	1	.52	.38	.29	-	-	.34	.41	.30	.52	.21	.21	.11	.18	-	-	.44	.48	.36	.56	.42	.18	.21	.40	-	-
↓ 3	.60	.49	.54	.76	1	.52	.36	.29	-	-	.33	.39	.28	.50	.19	.19	.11	.15	-	-	.40	.45	.34	.52	.40	.16	.20	.36	-	-
↓ 4	.54	.47	.52	.75	.95	.51	.36	.28	-	-	.31	.37	.26	.48	.18	.16	.09	.13	-	-	.38	.42	.32	.43	.38	.14	.19	.30	-	-
↓ 5	.50	.47	.48	.75	.90	.47	.35	.27	-	-	.29	.35	.24	.43	.15	.15	.06	.10	-	-	.38	.39	.29	.40	.36	.12	.18	.28	-	-
5%	.58	.49	.53	.71	.89	.45	.36	.29	-	-	.30	.38	.27	.50	.17	.20	.10	.14	-	-	.40	.41	.32	.42	.40	.14	.19	.36	-	-
10%	.50	.48	.50	.70	.81	.42	.32	.26	-	-	.28	.24	.24	.49	.14	.18	.08	.11	-	-	.37	.36	.29	.39	.37	.11	.17	.30	-	-
15%	.47	.46	.48	.68	.80	.39	.29	.26	-	-	.25	.30	.20	.45	.14	.15	.06	.09	-	-	.34	.32	.27	.36	.35	.09	.15	.28	-	-
20%	.43	.45	.46	.67	.80	.37	.29	.24	-	-	.20	.25	.17	.42	.12	.14	.06	.06	-	-	.31	.28	.25	.31	.32	.46	.14	.22	-	-
Avg	**.54**	**.48**	**.51**	**.73**	**.91**	**.46**	**.35**	**.28**	-	-	**.30**	**.35**	**.25**	**.48**	**.17**	**.18**	**.09**	**.13**	-	-	**.38**	**.40**	**.31**	**.40**	**.38**	**.11**	**.19**	**.32**	-	-
S2	.80	.57	.80	.90	.86	.64	.63	.94	.75	.75	.54	.50	.12	.10	.46	.54	.45	.78	.17	.25	.42	.36	.26	.90	.54	.64	.07	.06	.25	.25
↓ 2	.80	.53	.80	.90	.84	.63	.61	.94	.74	.75	.52	.50	.12	.10	.46	.52	.45	.78	.16	.24	.38	.35	.24	.90	.51	.60	.07	.05	.23	.24
↓ 3	.79	.51	.78	.89	.84	.60	.61	.94	.72	.73	.52	.48	.10	.08	.42	.48	.42	.76	.14	.23	.35	.32	.21	.86	.48	.58	.05	.04	.21	.20
↓ 4	.76	.49	.76	.86	.80	.57	.59	.90	.70	.69	.48	.46	.09	.07	.40	.46	.39	.73	.12	.21	.31	.26	.19	.80	.44	.55	.03	.02	.18	.16
↓ 5	.75	.45	.75	.83	.80	.54	.56	.87	.68	.67	.44	.42	.06	.06	.36	.44	.36	.69	.10	.20	.27	.21	.17	.76	.40	.50	.03	.02	.17	.12
5%	.78	.50	.78	.88	.85	.61	.62	.90	.73	.72	.50	.44	.10	.09	.41	.51	.34	.65	.13	.21	.38	.28	.20	.73	.44	.58	.06	.05	.16	.20
10%	.76	.47	.74	.85	.83	.58	.60	.84	.79	.72	.47	.36	.08	.07	.39	.46	.28	.62	.09	.18	.27	.21	.17	.68	.38	.52	.04	.04	.14	.16
15%	.74	.44	.71	.81	.79	.58	.57	.87	.68	.68	.40	.30	.07	.06	.37	.42	.26	.59	.08	.17	.24	.18	.14	.58	.36	.48	.01	.02	.10	.14
20%	.68	.39	.69	.78	.74	.51	.53	.84	.64	.61	.38	.26	.06	.01	.34	.41	.22	.55	.06	.14	.20	.15	.12	.48	.31	.40	.01	.01	.06	.10
Avg	**.76**	**.48**	**.76**	**.86**	**.82**	**.58**	**.59**	**.89**	**.71**	**.70**	**.47**	**.41**	**.09**	**.07**	**.40**	**.47**	**.35**	**.68**	**.12**	**.20**	**.31**	**.26**	**.18**	**.71**	**.43**	**.54**	**.04**	**.03**	**.17**	**.17**
S3	.96	1	.90	1	-	-	-	-	-	-	.72	1	.98	.91	-	-	-	-	-	-	.83	1	.86	.79	-	-	-	-	-	-
↓ 2	.94	1	.90	1	-	-	-	-	-	-	.72	.98	.98	.91	-	-	-	-	-	-	.82	.90	.86	.76	-	-	-	-	-	-
↓ 3	.91	1	.90	1	-	-	-	-	-	-	.71	.98	.98	.90	-	-	-	-	-	-	.81	.86	.86	.73	-	-	-	-	-	-
↓ 4	.87	.97	.89	1	-	-	-	-	-	-	.71	.97	.97	.90	-	-	-	-	-	-	.81	.82	.83	.73	-	-	-	-	-	-
↓ 5	.75	.96	.89	1	-	-	-	-	-	-	.70	.88	.98	.89	-	-	-	-	-	-	.84	.68	.89	.71	-	-	-	-	-	-
5%	.92	1	.90	1	-	-	-	-	-	-	.72	1	.98	.90	-	-	-	-	-	-	.81	.90	.81	.63	-	-	-	-	-	-
10%	.87	1	.89	1	-	-	-	-	-	-	.75	.96	.98	.88	-	-	-	-	-	-	.79	.86	.81	.57	-	-	-	-	-	-
15%	.81	.95	.89	1	-	-	-	-	-	-	.74	.91	.97	.85	-	-	-	-	-	-	.79	.80	.80	.51	-	-	-	-	-	-
20%	.75	.91	.85	1	-	-	-	-	-	-	.72	.87	.96	.81	-	-	-	-	-	-	.67	.77	.77	.42	-	-	-	-	-	-
Avg	**.86**	**.98**	**.89**	**1**	-	-	-	-	-	-	**.72**	**.95**	**.97**	**.88**	-	-	-	-	-	-	**.79**	**.84**	**.83**	**.64**	-	-	-	-	-	-
S4	.87	1	.84	1	1	.40	-	-	-	-	.70	1	.87	.65	.35	.12	-	-	-	-	.93	.65	.90	1	1	.21	-	-	-	-
↓ 2	.86	1	.96	1	.99	.31	-	-	-	-	.68	.89	.86	.65	.31	.11	-	-	-	-	.81	.42	.86	.81	1	.19	-	-	-	-
↓ 3	.86	1	.94	1	1	.25	-	-	-	-	.68	.76	.81	.38	.21	.11	-	-	-	-	.75	.39	.83	.63	1	.15	-	-	-	-
↓ 4	.86	1	.90	1	1	.21	-	-	-	-	.56	.62	.80	.12	.19	.10	-	-	-	-	.71	.28	.78	.51	1	.10	-	-	-	-
↓ 5	.85	1	.84	1	1	.19	-	-	-	-	.26	.32	.75	.05	.19	.08	-	-	-	-	.67	.20	.75	.38	1	.10	-	-	-	-
5%	.98	1	.89	1	.97	.41	-	-	-	-	.70	.80	.67	.60	.41	.10	-	-	-	-	.93	.65	.92	.81	1	.12	-	-	-	-
10%	.97	1	.87	1	.96	.39	-	-	-	-	.61	.62	.59	.51	.15	.08	-	-	-	-	.92	.63	.90	.65	.99	.10	-	-	-	-
15%	.96	1	.86	1	1	.33	-	-	-	-	.55	.31	.51	.43	.13	.06	-	-	-	-	.89	.63	.90	.61	1	.10	-	-	-	-
20%	.96	1	.81	1	.99	.29	-	-	-	-	.44	.08	.52	.34	.12	.05	-	-	-	-	1	.60	.89	.59	1	.09	-	-	-	-
Avg	**.92**	**1**	**.88**	**1**	**.99**	**.31**	-	-	-	-	**.58**	**.60**	**.71**	**.41**	**.23**	**.09**	-	-	-	-	**.84**	**.42**	**.86**	**.66**	**1**	**.12**	-	-	-	-

are misclassified, whereas for TDH the misclassification is 21%. As for the robustness test results, the proposed approach misclassifies 4% of trajectories in C_1 when the sampling rate is reduced from 2 to 5 steps and all the trajectories are successfully grouped together for C_2. By comparison, the clustering results of SOM degrade by 12% and 2% for C_1 and C_2, respectively. Likewise, the degradation of the results for TDH is 32% and 8% for C_1 and C_2. In the case of the maximum added noise, +20%, the trajectory classification results are reduced by 9% in C_1 and 0% in C_2 for the proposed approach. However, this reduction is 13%, and 10% in C_1, and C_2 for SOM, and 23%, and 37% in C_1, and in C_2 for TDH. The proposed approach is more robust to outliers up to ↓ 4 and +15% in the generation of the final clusters.

Figure 10 (n-p) shows the results for $S4$. SOM could not detect any outlier. If the proposed outlier detection criteria are applied on the SOM results, its performance can be improved. Figure 9 shows few examples where the proposed outlier detection criteria has improved the clustering results. The Table shows that although for C_1 and C_2, the proposed approach identified the patterns correctly, for C_3 its performance is poorer due to varying behavior of the trajectories in that cluster and also to a strong similarity between the trajectories and the outliers. However, the comparison with the other two approaches reveals that the results of the proposed approach are better by 35% for C_2 and 43% for C_3 over SOM, and by 35% and 31% for C_1 and C_2 over TDH. The robustness test results for $S4$ show that the proposed approach does not modify its results for C_1 and C_2, in case of sampling rate reduction. However, there is a 21% degradation for C_3. On the other hand, the degradation for SOM is 68%, 60%, 4% in C_1, C_2 and C_3, respectively. Furthermore, this degradation for TDH is 45%, 62% and 11%, respectively. Moreover, for the noisy data, the proposed approach is more stable and the only reduction in performance is 14% for C_3. In the same conditions, there are relevant classification errors for SOM (92%, 31%, 19%) and TDH (5%, 41%, 3%).

V. Conclusions

We proposed a clustering algorithm for video object trajectories analysis. The algorithm transformed object trajectories into distinct feature spaces that represented complementary characteristics of the motion pattern of a video object. Features were the average target velocity, the target directional distance, the trajectory mean, the initial target position, its speed and its acceleration, the principal components of trajectory points, and the degree of turn. Next, we used Mean-shift to estimate clusters in each space and adjacent clusters in a feature space were merged to refine the initial results. We associated a fuzzy membership of a trajectory to the final clusters and crisp clusters were then obtained based on the maximum membership using information from all the feature spaces. Finally, the clusters with few associated trajectories and trajectories that were far from the clusters' center were considered outliers. The proposed algorithm was validated on standard test sequences and compared with state-of-the-art approaches. The results demonstrated that the proposed algorithm is more robust to noise and to missing object observations. Our future work is focused on event modeling using trajectory clustering. This modeling is aimed at recognizing structured or unstructured activities of interest in a multi-camera network.

References

[1] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," in *Proc. of British Conf. on Machine vision, Birmingham, UK*, Sep. 1996.

[2] R. Rosales and S. Sclaroff, "Trajectory guided tracking and recognition of actions, computer science department. university of boston, Tech. Rep. 1999-002, Sep. 1999. (online: http://citeseer.ist.psu.edu/rosales99trajectory.html, last accessed: 28 Feb, 2008).

[3] A. R. Webb, *Statistical Pattern Recognition, 2nd Edition*. John Wiley and Sons Ltd., Jul. 2002.

[4] Y. Zeng and J. Starzyk, "Statistical approach to clustering in pattern recognition," in *Proc. of 33rd southeastern Symp. on System Theory, Athens, Ohio, USA*, Mar. 2001.

[5] Y. Gdalyahu, D. Weinshall, and M. Werman, "Self-organization in vision: stochastic clustering for image segmentation, perceptual grouping, and image database organization," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1053–1074, Oct. 2001.

[6] F. Amasyali and S. Albayrak, "Fuzzy c-means clustering on medical diagnostic systems," in *Proc. of Intl. twelfth Turkish Symp. on Artificial Intelligence and Neural Networks (TAINN2003), Canakkale, Turkey*, Jul. 2003.

[7] G. Antonini and J. Thiran, "Counting pedestrians in video sequences using trajectory clustering," *IEEE Trans. on Circuit and Systems for Video Technology*, vol. 16, no. 8, pp. 1008–1020, Aug. 2006.

[8] S. Gaffney and P. Smyth, "Trajectory clustering with mixtures of regression models," in *Proc. of Intl. Conf. on Knowledge Discovery and Data Mining, San Diego, USA*, Aug. 1999.

[9] S. Khalid and A. Naftel, "Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space," in *Proc. of third ACM Intl. workshop on Video Surveillance and Sensor Networks, Singapore*, Sep. 2005.

[10] A. Naftel and S. Khalid, "Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space," *Trans. on Multimedia Systems*, vol. 12, no. 3, pp. 227–238, Dec. 2006.

[11] Z. Fu, W. Hu, and T. Tan, "Similarity based vehicle trajectory clustering and anomaly detection," in *Proc. of IEEE Intl. Conf. on Image Processing, ICIP, Genova, Italy*, Sep. 2005.

[12] X. Li, W. Hu, and W. Hu, "A coarse-to-fine strategy for vehicle motion trajectory clustering," in *Proc. of Intl. Conf. on Pattern Recognition (ICPR), Hong Kong, China*, Aug. 2006.

[13] F. Bashir, A. Khokhar, and D. Schonfeld, "Real-time motion trajectory-based indexing and retrieval of video sequences," *IEEE Trans. on Multimedia*, vol. 9, pp. 58–65, Jan. 2007.

[14] S. Andradottir, "A new algorithm for stochastic optimization," in *Proc. of 22nd Conf. on Winter simulation, USA*, Dec. 1990, pp. 364 – 366.

[15] F. Porikli, "Trajectory pattern detection by HMM parameter space features and eigenvector clustering," in *Proc. of 8th European Conf. on Computer Vision (ECCV), Prague, Czech Republic*, May 2004.

[16] J. Alon, S. Sclaroff, G. Kollios, and V. Pavlovic, "Discovering clusters in motion time-series data," in *Proc. of IEEE Conf. on Computer vision and pattern recognition (CVPR), Madison, Wisconsin*, Jun. 2003.

[17] A. D. Wilson and A. F. Bobick, "Recognition and interpretation of parametric gesture," in *Proc. of IEEE 6th Intl. Conf. on Computer Vision, Bombay, India*, Jan. 1998.

[18] N. Oliver, B. Rosario, and A. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831–843, Aug. 2000.

[19] D. Chudova, S. Gaffney, and P. Smyth, "Probabilistic models for joint clustering and time warping of multi-dimensional curves," in *Proc. of 19th Conf. on Uncertainty in Artificial Intelligence, Acapulco, Mexico*, Aug. 2003.

[20] N. Sumpter and A. J. Bulpitt, "Learning spatio-temporal patterns for predicting object behaviour," in *Proc. of British Conf. on Machine vision, Southampton, UK*, Sep. 1998.

[21] W. Hu, D. Xie, T. Tan, and S. Maybank, "Learning activity patterns using fuzzy self-organizing neural network," *IEEE Trans. on Systems, Man and Cybernetics, Part B*, vol. 34, no. 3, pp. 334–352, Jun. 2004.

[22] J. Baras and S. Dey, "Combined compression and classification with learning vector quantization," *IEEE Trans.on Information Theory*, vol. 45, no. 6, Sep. 1999.

[23] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image and Vision Computing*, vol. 14, no. 8, pp. 609–615, Aug. 1996.

[24] F. Bashir, A. Khokhar, and D. Schonfeld, "Segmented trajectory based indexing and retrieval of video data," in *Proc. of Intl. Conf. on Image Processing (ICIP), Barcelona, Catalonia, Spain*, Sep. 2003.

[25] ——, "Object trajectory-based activity classification and recognition using hidden markov models," *IEEE Trans. on Image Processing*, vol. 16, pp. 1912–1919, Jul. 2007.

[26] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603 – 619, May 2002.

[27] ——, "Distribution free decomposition of multivariate data," *Trans. on Pattern Analanlysis and Applications*, vol. 2, no. 1, pp. 22 – 30, 1999.

[28] D. Comaniciu, V. Ramesh, and P. Meer, "The variable bandwidth mean shift and data-driven scale selection," in *Proc. of IEEE Intl Conf. on Computer Vision (ICCV), Vancouver, Canada*, 2001.

[29] H. Wang and D. Suter, "False-peaks-avoiding mean shift method for unsupervised peak-valley sliding image segmentation," in *Proc. of Intl. Conf. on Digital Image Computing: Techniques and Applications (DICTA), Sydney, Australia*, 2003.

[30] R. Kasturi, "Performance evaluation protocol for face, person and vehicle detection & tracking in video analysis and content extraction, vace-ii," Jan. 2006.

[31] G. A. F. Seber, *Multivariate Observations*. John Wiley and Sons, New York, 1984.

Nadeem Anjum received his M.Sc degree (with distinction) in digital signal processing from Queen Mary, University of London, United Kingdom, in 2006. Since 2006, he is a research student and assistant tutor at the Electronic Engineering department of Queen Mary, University of London. His research interests include multi-sensor network localization and video analysis. Mr. Anjum has served as a reviewer for the IEEE Transactions on Circuits and Systems for Video Technology and the ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion.

Andrea Cavallaro received the M.Sc. degree (summa cum laude) from the University of Trieste, Trieste, Italy, in 1996 and the Ph.D. degree from the Swiss Federal Institute of Technology (EPFL), Lausanne, in 2002, both in electrical engineering. From 1998 to 2003, he was a Research Assistant with the Signal Processing Laboratory, EPFL. Since 2003, he has been a Lecturer with Queen Mary, University of London (QMUL), London, U.K. He is the author of more than 70 papers, including five book chapters. Dr. Cavallaro was the recipient of a Research Fellowship with BT, the Royal Academy of Engineering Teaching Prize in 2007, and was coauthor for two student paper prizes at IEEE ICASSP 2005 and 2007. He was general chair of IEEE AVSS 2007, technical chair of EUSIPCO 2008, Guest Editor of two journal special issues on object tracking, and is an elected member of the IEEE Multimedia Signal Processing Technical Committee.