

Event Detection by Eigenvector Decomposition Using Object and Frame Features

Fatih Porikli Tetsuji Haga
Mitsubishi Electric Research Laboratories
fatih@merl.com

Abstract

We develop an event detection framework that has two significant advantages over past work. First, we introduce an extended set of time-wise and object-wise statistical features including not only the trajectory coordinates but also the histograms and HMM based representations of object's speed, orientation, location, size, and aspect ratio. These features enable detection of events that cannot be detected with the existing trajectory features reported so far. Second, we introduce a spectral clustering algorithm that can automatically estimate the optimal number of clusters. First, we construct feature-wise affinity matrices from the pairwise similarity scores of objects using the extended set of features. To determine the usual events, we apply eigenvector decomposition and obtain object clusters. We show that the number of eigenvectors used in the decomposition is proportional to the optimal number of clusters. Unlike the conventional approaches that try to fit predefined models to events, we analyze the conformity of objects using affinity matrices to find the unusual events. We improve the feature selection process by incorporating feature variances. We prove that the clustering stage is not adversely affected by high dimensionality of data space. Our simulations with synthetic and real data reveal that the proposed detection methods accurately detect usual and unusual events.

1. Introduction

Event detection requires interpretation of the “semantically meaningful object actions” [3]. To achieve this task, the gap between the numerical features of objects and the symbolic description of the meaningful activities needs to be bridged.

Past work on event detection has mostly consisted of extraction of trajectories followed by a supervised learning. For example, an activity recognition method that is based on view-dependent template matching was developed in [1]. Action is represented by a temporal template, which is computed from the accumulative motion properties at each pixel. Davis [2] represents simple periodic events (e.g., walking) by constructing dynamic models of periodic pattern of people's movements. Hogg [6] clusters the distri-

butions of object trajectories. Stauffer [16] estimates a hierarchy of similar distributions of activity based using co-occurrence feature clustering. Zelnik [18] defines events as temporal stochastic processes and targets a temporal segmentation of video. Their dissimilarity measure is based on the sum of χ^2 divergences of empirical distributions, which requires off-line training. The number of clusters is pre-set in event detection. Starner[15] uses a Hidden Markov Model (HMM) to represent a simple event and recognize this event by computing the probability that the model produce the visual observation sequence. In [8], HMM is used for intrusion detection. Existing HMM's based approaches require off-line training of events. However, it is not viable to foresee every possible event. Besides, the nature of event varies depending on the application, thus event modeling becomes even more challenging.

There are related praiseworthy work on spectral clustering by Ng [12] and Meila [11]. We can extend this list to Marx [9], Kamvar [7], even back to Fiedler [4]. However, these methods address different issues. For instance, Ng uses k-means clustering. Unlike us, they do not investigate the relation between the optimal number of clusters and the number of largest eigenvectors. Meila extends Ng to generalized eigenvalue representation. Although they use multiple eigenvectors, the number of eigenvectors is fixed. Kamvar addresses supervisory information, which we do not require. Marx develops coupled-clustering with fixed number of clusters. One main disadvantage of these approaches is that they are all limited to the equal duration trajectories since they depend on the coordinate correspondences.

Although the extraction of trajectories is well studied, little investigation on the secondary outputs of a tracker has been done. Medioni [10] uses eight constant features which include height, width, speed, motion direction, and the distance to a reference object. Visual features were also addressed by Zelnik [18] and Stauffer [16]. Zelnik uses spatiotemporal intensity gradients at different temporal scales. Stauffer uses co-occurrence statistics of coordinate, speed and size.

Since existing trajectory-based features are insufficiently expressive, they cannot be used to identify certain events.

We are thus motivated to develop more expressive features that we then employ to detect events we were not able to detect with conventional features. In addition to trajectory, we introduce statistical features including the histograms and parameter representations of tracked objects and frames. We find however that our proposed features have high dimensionality. Since conventional learning methods are adversely affected by high dimensionality, we are motivated to develop a new approach to clustering that is much more robust to increase in the dimensionality of the feature space and has lower complexity than the conventional approaches.

Unlike the past work cited above, we employ an unsupervised learning method. It is based on eigenvector decomposition of the feature similarity matrices. We show that the number of clusters governs the number of eigenvectors used to span the feature similarity space. We are thus able to automatically compute the optimal number of clusters.

Our method does not require definition of what is usual and what is not. We define usual as the high recurrence of events that are similar. As a result, unusual is the group of events that are not similar to the rest. This enables us to detect multiple unusual events.

The rest of the paper is organized as follows. In the Section 2, the tracking features are introduced. Section 3 explains the formation of affinity matrices and the clustering algorithm. Section 4 discusses the simulations.

2. Trajectories to Features

Types of the events and their indicative features vary depending on the applications. However, the features that we propose below characterize most of the available low-level properties of objects. All features are presented here are acquired automatically using an object tracker presented in [14]. Although we extensively make use of 2D trajectories, the same analogy can be extended for 3D trajectories and higher dimensions, e.g. a sequence of multi-dimensional vectors, etc.

A trajectory is a time series of coordinates representing the motion path of an object over the duration (lifetime), i.e. number of frames that object exists. These coordinates correspond to marked positions of object shape in consecutive frames. A marked position often indicates either the center-of-mass, the intersection of main diagonals (for ellipsoid model), or the average of minimum and maximum on perpendicular axes (for bounding box model) of object region. We adopt the following notation $T : \{p_n\} : \{(x_1, y_1, t_1), (x_2, y_2, t_2), \dots, (x_N, y_N, t_N)\}$ where N is the duration (number of constituting samples) of the trajectory.

We propose tracking features that can be classified into two groups as depicted in Fig.1. The first set of features describes the properties of individual objects such as motion, color, shape, etc. The second set of features represents

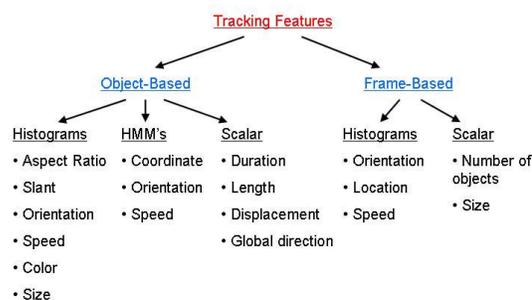


Figure 1: Object tracker provides object and frame features.

frame-wise attributes such as the number of objects, aggregated speed, orientation, etc. of objects existing in those frames.

Certain object properties are dynamic and change their values from frame to frame during the tracking process, e.g. the speed of an object. Instead of depending only on the instantaneous values, using normalized histograms as features enables to capture the history of these dynamic features. A histogram in fact corresponds to the probability density distribution, thus it sustains statistical characteristics such as mean, variance and other higher order moments. However, since histograms discard temporal ordering for time-sequences, we develop temporal model based representations. The model based representations are more expressive than the histograms in case the relation between the consecutive values of time-series (coordinate, orientation, speed) possess important discriminating information. Furthermore, since most higher level analysis require features to have equal dimensions, the model based representations, such as HMM's, enable transferring of the variable length features into a common parameter space.

2.1. Object Based Features

There are several scalar features that describe an object. In spite of its simplicity, the duration (lifetime) is one of the distinctive features. For instance, in a surveillance setting the suspicious event may be a left behind unattended bag, which can be easily detected using the duration feature since other objects do not usually stay for extended periods of time. The total length of the trajectory is defined as $\sum \Delta p_n$. Note that, this is slightly different from the total displacement of the object, which is defined as $|p_1 - p_N|$. Depending on the camera setup, the length based descriptors may be used to differentiate different paths. The length/duration ratio gives the average speed. A general orientation descriptor records the direction of the object between its first and last appearance. Figure 2 illustrates some of the object features.

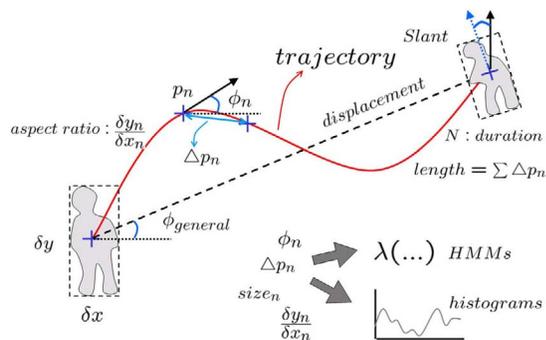


Figure 2: Object based features.

The probability densities of the dynamic properties, such as orientation, aspect ratio, slant (angle between a reference axis and the main diagonal of object), size, instantaneous speed, location, and color, are approximated by histograms. The location histogram keeps track of the image coordinates of the object trajectory and it can be used for zone monitoring. Using the size histogram, we can distinguish an object moves along the camera focal direction (size will get larger) from another object that moves parallel. Often, an object moves at different speeds during its lifetime.

The speed histogram gains more importance in such scenarios that motion properties is the main source of information. This histogram can interpret the regularity of the movement to detect the erratically moving objects. A vehicle involved in a crash accident can be distinguished since its speed histogram will have dual modalities at the very high and very low speeds rather than being distributed uniformly or having a single peak. The orientation histogram makes it possible to distinguish objects moving on certain paths. We can find a vehicle that first backs up on a wrong lane and continues in the right direction, which may not be detected using only an ordinary orientation representation. The aspect ratio is a good descriptor to differentiate human objects and vehicles. The aspect ratio histogram can be effectively used to determine the gait of a person.

The color properties may be represented by a conventional histogram or by a few number of dominant colors with an additional computational cost. For low-resolution scenarios such that sufficiently large faces (for human objects) and license plates (for vehicles) are difficult to extract, the color histogram remains as the only choice to identify objects. For instance, the person who gets dressed in a different color from the rest of the workers may be the object of interest at a factory surveillance setting.

Since histograms fall short of capturing the temporal properties of time-sequences, we also use temporal model based representations. Certain detection tasks require similarity measurements based on the form of the trajectories

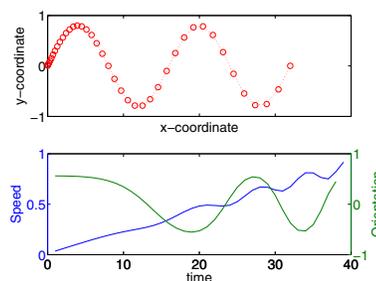


Figure 3: Top: a sample coordinate sequence consists of 40 points. Bottom: corresponding speed and orientation sequences computed from the coordinate sequence. It is more effective to understand (thus to evaluate computationally) acceleration of an object from the speed sequence, and direction changes from the orientation sequence.

independent of the coordinate values and initial time. In addition to the conventional time-coordinate sequence representation, we define two sequences using the instantaneous orientation and speed values as illustrated in Fig. 3. These sequences can evaluate the trajectory form similarity even if there is spatial and temporal translation between the trajectories.

2.2. Frame Based Features

On the other hand, frame-wise features specify the characteristics of objects existing within the same frame. These features become more distinctive as the number of the visible objects in the frame increases. We use frame-wise features to determine the instances of the events as opposed to finding the particular objects.

The number of objects detected at the current frame is one frame-wise feature. This feature may give important clues about the ambiguous situations, e.g. an *unexpectedly* high number of people in a room in case the room is *usually* empty. The total size of the objects is another feature that indicates the density of objects. An aggregated location histogram shows where objects stay with respect to the image plane. The dominant orientation measures the aggregated orientation of all objects in the frame. The histogram of the instantaneous orientations captures the distribution of orientation within the frame. This feature can detect the changes of the motion flow direction. For instance, it may indicate which team is on attack at a soccer game. The speed histogram of the objects also defines the magnitude distribution of the motion in the current frame. This feature can distinguish frames in where an object has different speed from the rest. The frame-wise histograms of the aspect ratios and size are defined similarly.

2.3. HMM Representations

We transfer the coordinate, orientation, and speed sequences into a parameter space λ that is characterized by a set of HMM parameters.

An HMM is a probabilistic model composed of a number of interconnected states in a directed graph, each of which emits an observable output. Each state is characterized by two probability distributions: the transition distribution over states and the emission distribution over the output symbols. A random source described by such a model generates a sequence of output symbols. Since the activity of the source is observed indirectly, through the sequence of output symbols, and the sequence of states is not directly observable, the states are said to be hidden.

We replace the trajectory information as the emitted observable output of the above directed graph. The hidden states then capture the transitive properties of the consecutive coordinates of the spatiotemporal trajectory. The state sequence that maximizes the probability becomes the corresponding model for the given trajectory.

A simple specification of an K -state $\{S_1, S_2, \dots, S_K\}$ continuous HMM with a Gaussian observation is given by:

1. A set of prior probabilities $\pi = \{\pi_i\}$ where $\pi_i = P(q_1 = S_i)$ and $1 \leq i \leq K$.
2. A set of state transition probabilities $B = \{b_{ij}\}$, where $b_{ij} = P(q_{t+1} = S_j | q_t = S_i)$ and $1 \leq i, j \leq K$.
3. Mean, variance and weights of mixture models $\mathcal{N}(O_t; \mu_j, \sigma_j)$ where μ_j and Σ_j are the mean and covariance of the state j .

Above, q_t and O_t are the state and observation at time t . For each trajectory T , we fit an H -mixture HMM $\lambda = (\pi, B, \mu, \Sigma)$ that has left-to-right topology using the Baum-Welch algorithm. We chose the left-to-right topology since it can efficiently describe continuous processes. We train a HMM model using the trajectory as the training data. As a result, each trajectory is assigned to a separate model.

The optimum number of states and mixtures depend on the complexity and duration of the trajectories. To provide sufficient evidence to every Gaussian of every state in the training stage, the lifetime of the trajectory should be much larger than the number of models in the mixture times the number of states $N \gg H \times K$. On the other hand, a state can be viewed as a basic pattern of the trajectory, thus depending the trajectory the number of states should be large enough to conveniently characterize distinct patterns but small enough to prevent from overfitting.

3. Features to Events

An *event* is defined as "something that happens at a given place and time". We detect two types of events using the

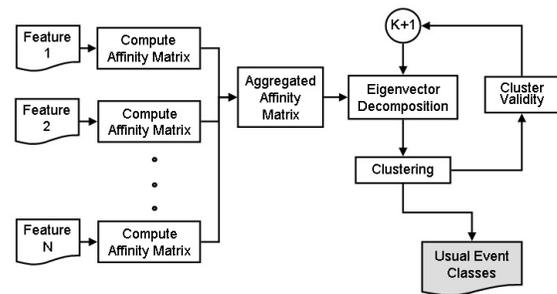


Figure 4: Usual events is detected using affinity matrices.

above features: 1) object domain events, 2) frame domain events. An object domain event is obtained by evaluating the mutual similarities (unusual) and clustering objects (usual) using their properties. Similarly, a frame domain event is derived from the frame features and it corresponds to a particular time instance. Therefore we define events as either as clusters of parameter space components (usual events) or outliers (unusual events). We present two methods to detect unusual and usual events. An unusual event is associated with the distinctness of the activity. For instance, a running person where everybody walks is interpreted as unusual as well as a walking person where the rest run. A usual event indicates the commonality, e.g. a path that most people walks. Flow diagrams of the detection process for usual and unusual events are given in Figures 4 and 6.

To detect the usual events, we find object clusters by analyzing the affinity matrices. For each feature, an affinity matrix is computed using the pair-wise object similarities. Then, matrices are added and normalized to obtain an aggregated matrix. We apply eigenvector decomposition to find the optimal number of clusters. We use the decomposed matrix and then thresholded values to assign objects in the clusters. We associate feature weights, which can be adapted to specific applications using priori information.

To determine the unusual events, we analyze each affinity matrix. Objects are ordered with respect to their conformity scores. These scores are multiplied by the weights to inject the priori information. Finally, the objects are re-ordered with respect to the total conformity scores, and the objects that have low scores are identified as unusual events. The same analogy is valid for the frame domain events.

Thus, we define an unusual event in relation to other observed behavior, which can be done either incorporating only the simultaneously existing objects (group of frames) or all detected objects (all frames) in the history.

Why Spectral Clustering?

Note that, it is possible to compute pair-wise distances for unequal duration trajectories, which are very common for object tracking applications, but it is not possible to map all the trajectories into a uniform data space where the vector dimension is constant. Therefore, the ordinary clustering methods that require uniform feature size are not applicable for this purpose. Thus, we developed the following spectral clustering based method.

3.1. Affinity Matrix

For each feature, an affinity matrix A is constructed. The elements a_{ij} of this matrix are equal to the similarity of the corresponding objects i and j , $i, j = 1, \dots, M$. The similarity is defined as $a_{ij} = e^{-d(i,j)/2\sigma^2}$, where $d(i, j)$ is distance, and σ^2 is a constant scaler. Note that matrix $A \in \mathcal{R}^{n \times n}$ is a real semi-positive symmetric matrix, thus $A^T = A$.

In case of the HMM parameter based features, the distance $d(i, j)$ is measured using a mutual fitness score of the models and input features. We define the distance between two trajectories in terms of their HMM parameterizations as

$$d(T^a, T^b) = |L(T^a; \lambda^a) + L(T^b; \lambda^b) - L(T^a; \lambda^b) - L(T^b; \lambda^a)| \quad (1)$$

which corresponds the cross-fitness of the trajectories to each other's models. A detailed analysis of HMM distance measures can be found in [13].

The $L(T^a; \lambda_a), L(T^b; \lambda_b)$ terms indicate the likelihood of the trajectories to their own fitted model, i.e. we obtain the maximum likelihood response for the models. The cross terms $L(T^a; \lambda_b), L(T^b; \lambda_a)$ reveal the likelihood of a trajectory generated by the other trajectories model. In other words, if two trajectories are identical, the cross terms will have a maximum value, thus eq. 1 will be equal to zero. On the other hand, if trajectories are different, their likelihood of being generated from each others model will be small, thus the distance will be high.

3.2. Detection of Usual Events

First, the affinity matrices are decomposed using a certain number of the largest eigenvectors.

Eigenvector Decomposition

The decomposition of a square matrix into eigenvalues and eigenvectors is known as eigenvector decomposition.

Although spectral clustering [5], [17], [12], [11] is addressed before in the literature, to our knowledge no one has established the relationship between the optimal clustering of the data distribution and the number of eigenvectors that

should be used for spanning. Here we show that the number of eigenvectors is proportional to the number of clusters.

Let $V \equiv [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_M]$ be a matrix formed by the columns of the eigenvectors. Let D be a diagonal matrix $diag[\lambda_1, \dots, \lambda_M]$. Lets also assume eigenvalues are $\lambda_1 \geq \lambda_2 \geq \dots \lambda_M$. Then the generalized eigenvalue problem is

$$(A - I)V = [A\mathbf{v}_1 \ \dots \ A\mathbf{v}_M] = [\lambda_1\mathbf{v}_1 \ \dots \ \lambda_M\mathbf{v}_M]D = VD \quad (2)$$

and $A = VDV^{-1}$. Since A is symmetric, the eigenvectors corresponding to distinct eigenvalues are real and orthogonal $VV^T = V^T V = I$, which implies $A = VDV^T$.

Let a matrix P_k be a matrix in a subspace \mathcal{K} that is spanned by the columns of V such as $P_k = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_k, 0]$ where V is the orthogonal basis satisfies $A = VDV^T$. Now, we define vectors \mathbf{p}_n as the rows of the truncated matrix P_k as

$$P_k = \begin{bmatrix} \mathbf{p}_1 \\ \vdots \\ \mathbf{p}_M \end{bmatrix} = \begin{bmatrix} v_{11} & \dots & v_{1k} & 0 & \dots \\ \vdots & & \vdots & & \vdots \\ v_{M1} & \dots & v_{Mk} & 0 & \dots \end{bmatrix} \quad (3)$$

We normalize each row of matrix P_k by $p_{ij} \leftarrow p_{ij} / \sqrt{\sum_j^k p_{ij}^2}$. Then a correlation matrix is computed using the normalized rows by $C_k = P_k P_k^T$. For a given P_k , the value of p_{ij} indicates the degree of similarity between the object i and object j . Values close to one correspond to a match whereas negative values and values close to zero suggest that objects are different. Let ϵ be a threshold that transfers values of matrix C_k to the binary quantized values of an association matrix W_k as

$$w_{ij} = \begin{cases} 1 & c_{ij} \geq \epsilon \\ 0 & c_{ij} < \epsilon \end{cases} \quad (4)$$

where $\epsilon \approx 0.5$. The clustering is then becomes grouping the objects that have association values equal to one $w_{ij} = 1$.

To explain why this works, remember that eigenvectors are the solution of the classical extremal problem $\max \mathbf{v}^T A \mathbf{v}$ constrained by $\mathbf{v}^T \mathbf{v} = 1$. That is, find the linear combination of variables having the largest variance, with the restriction that the sum of the squared weights is 1. Minimizing the usual Lagrangian expression $\mathbf{v}^T A \mathbf{v} - \lambda(\mathbf{v}^T \mathbf{v} - 1)$ implies that $(I - A)\mathbf{v} = \lambda I \mathbf{v}$. Thus, \mathbf{v} is the eigenvector with the largest eigenvalue.

As a result, when we project the affinity matrix columns on the eigenvector \mathbf{v}_1 with the largest eigenvalue and span \mathcal{K}_1 , the distribution of the a_{ij} will have the maximum variance therefore the maximum separation. Keep in mind that a threshold operation will perform best if the separation is high. To this end, if the distribution of values have only two distinct classes then a balanced threshold passing through the center will divide the points into two separate clusters. With the same reasoning, the eigenvector \mathbf{v}_2 with the second largest eigenvalue, we will obtain the basis vector that

gives the best separation after normalizing the projected space using the \mathbf{v}_1 since $\mathbf{v}_1 \perp \mathbf{v}_2$. Thus, we deduce the following lemma:

Cluster & Eigenvector Lemma: The number of largest eigenvalues (in absolute value) to span subspace is one less than the number of clusters.

As opposed to using only the largest or first and second largest eigenvectors (also the generalized second minimum which is the ratio of the first and the second depending the definition of affinity), the correct number of eigenvectors should be selected with respect to the target cluster number. Using only one or two does fail for multiple clusters scenarios.

The values of the thresholds should still be computed. We obtained projections that gives us the maximum separation but we did not determine the degree of separation i.e. maximum and minimum values of projected values on the basis vectors. For convenience, we normalize the projections i.e. the rows of current projection matrix (V_k) as $\mathbf{p}^T \mathbf{p} = 1$ and then compute the correlation $V_k^T V_k$. Correlation will make rows that their projections are similar to get values close to 1 (equal values will give exactly 1), and dissimilar values to 0. By maximizing the separation (distance) between the points in different clusters on an orthonormal basis, we pushed for the orthogonality of points depending their clusters; $\mathbf{p}_i \mathbf{p}_j \approx 1$ if they are in the same cluster, and $\mathbf{p}_i \mathbf{p}_j \approx 0$ if they are not.

Estimating the Number of Clusters - Ad Hoc Method

After each eigenvalue computation of matrix A , we compute a validity score α_k using the clustering results as

$$\alpha_k = \sum_c \frac{1}{M_c} \sum_{i,j \in Z_c} p_{ij} \quad (5)$$

where Z_c is set of objects included in the cluster c , M_c number of objects in Z_c . The validity score gets higher values for the better fits. Thus, by evaluating the first local maximum of this score, we determine the correct cluster number automatically. Figure 5 demonstrates the assessment of the proposed score with well-defined clusters. As shown, the validity score accurately chooses the the optimal number of clusters at each time. Thus, we answer the natural question of clustering; "what should be the total cluster number?"

As a summary, the clustering for a given maximum cluster number k^* includes

1. Compute A , approximate eigenvectors using Ritz values $\lambda_k \simeq \theta_k$, find eigenvectors v_k for $k = 1, \dots, k^*$,
2. Find $P_k = V_k V_k^T$ and Q_k for $k = 1, \dots, k^*$,
3. Determine clusters and calculate α_k ,

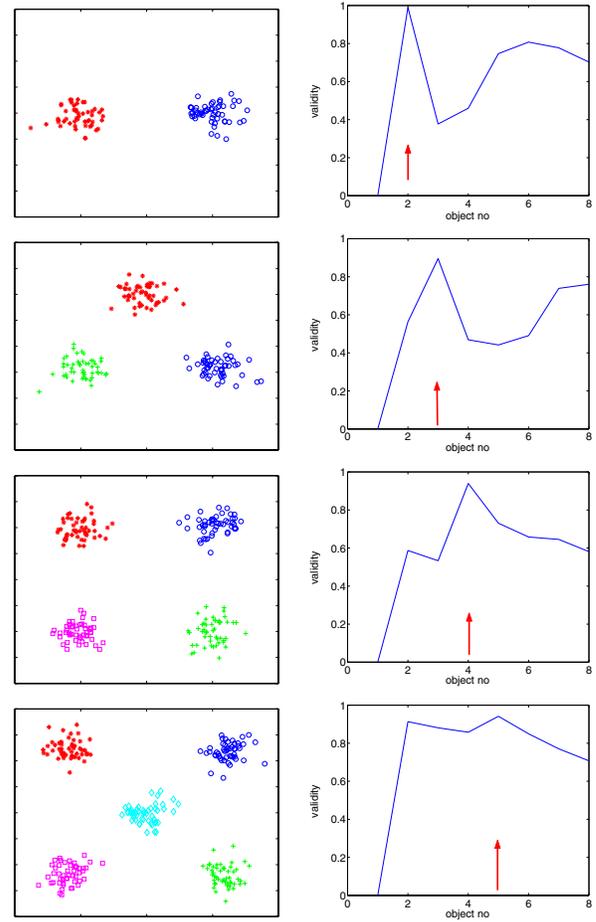


Figure 5: Performance evaluation of the validity score. As visible, the validity score reaches its maximum value at the optimal cluster number at each case.

4. Compute $\alpha' = d\alpha/dk$ and find local maxima.

The maximum cluster number k^* does not affect the determination of the fittest cluster; it is only an upper limit.

Comparison with K-means

A question arise that why we preferred the eigenvector clustering to the ordinary k-means?

Most importantly, a 'mean' or a 'center' vector cannot be defined for trajectories that have different durations. We only have pair-wise distances. In eigenvector decomposition, mutual inter-feature distance as opposed to center-distance is used.

Ordinary k-means may oscillate between cluster centers, and different initial values may cause completely dissimilar clusters. Besides, k-means can stuck to local optima. Therefore, k-means based cluster number estimation is not

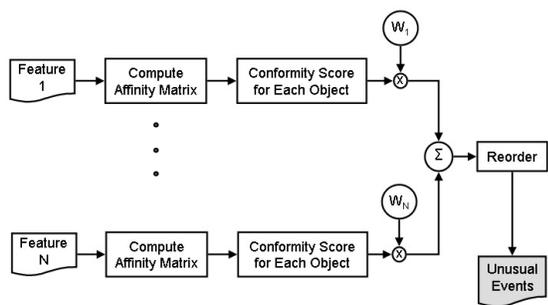


Figure 6: Unusual events is found using conformity scores.

always accurate. Furthermore, the computational complexity of k-means increases with the larger sizes of the feature vectors. Although the eigenvector decomposition is $\mathcal{O}(\frac{2}{3}M^3 + \frac{1}{2}kM^2)$, it is not exponentially proportional to the size of feature vector s . (Note that we do not claim the eigenvector computation cannot be done more efficiently than $\mathcal{O}(M^3)$, e.g. for m eigenvectors, the complexity reduces to $\mathcal{O}(mM^2)$). In case the $s \approx M$, k-means algorithm, which has complexity of $\mathcal{O}((k \log M)^s + Jk^2M)$ becomes much more demanding than eigenvector decomposition (J is the required iterations necessary for convergence).

3.3. Detection of Unusual Events

Using the affinity matrices, conformity scores of the objects are computed. The conformity score of an object i for a given feature f is the sum of the corresponding row (or column) of the affinity matrix that belong that feature $\beta_f(i) = \sum_n a_{in}$. To fuse the responses of different features, we propose a simple weighted sum approach. We obtain a total conformity score for an object as

$$\beta(i) = \frac{1}{M} \sum_f w_f \beta_f(i), \quad (6)$$

where $w_f = 1$ for equivalently important features. Then, we order each object with respect to its total conformity score. The object that has the minimum score corresponds to most different, thus most unusual event.

One distinct advantage of the conformity score is that it does not assert unusuality in case all events are similar. Figure 7 demonstrates the conformity scores for two different cases: (7-a) data consists of well-defined clusters where all points are similar and no significant variance exists, and Fig. (7-b) same data with two additional outlier points. Figure 7-c shows the conformity scores of the points within first well-defined data. As visible, the scores are close to each other. However, for the second data set where two outlier points exists, the conformity of the outlier points are significantly lower as given in Fig.7-d. Furthermore, more a point

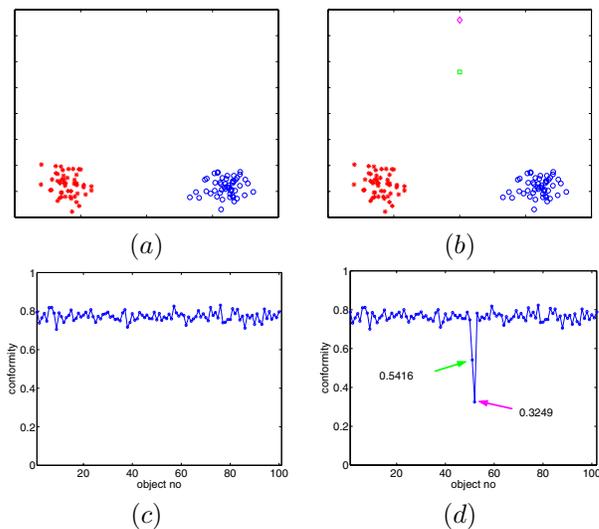


Figure 7: (a) Data that consists of two clusters. As visible, all points belong to one of the clusters. (b) Same data with two additional points that are located significantly far away from the main clusters. (c) Corresponding conformity scores of the points in a. (d) Conformity scores of the points in b. Note that, the conformity scores decreases as the points become more dissimilar.

becomes dissimilar to the rest of the points, lower its score becomes. This shows that the proposed conformity score effectively determines the degree of the ambiguity as well as it distinguishes the unusual event from the ordinary.

Feature Selection and Adaptive Weighting

It is also possible to select most discriminating features before the clustering stage. However feature selection requires priori knowledge about the application and understanding of the nature of events. Thus, we preferred to let the clustering module to determine the prominent features instead of a preselection of such features. Moreover, we will show that truncation of the eigenbasis amplifies unevenness in the distribution of features by causing features of high affinity to move towards each other and other to move apart.

Our simulations show that the feature variance is an effective way to select the above feature weights w_i . The feature variance is calculated from the corresponding affinity matrix. In case the feature supplies distinctive information the variance will have a higher value. The opposite is also true. Thus, we assign the fusion weights as

$$w_f = \frac{1}{n^2} \sum_i \sum_j (a_{ij} - \mu_f)^2 \quad (7)$$

where a_{ij} is an element of the affinity matrix A_f for the feature f . This enables emphasizing important features.

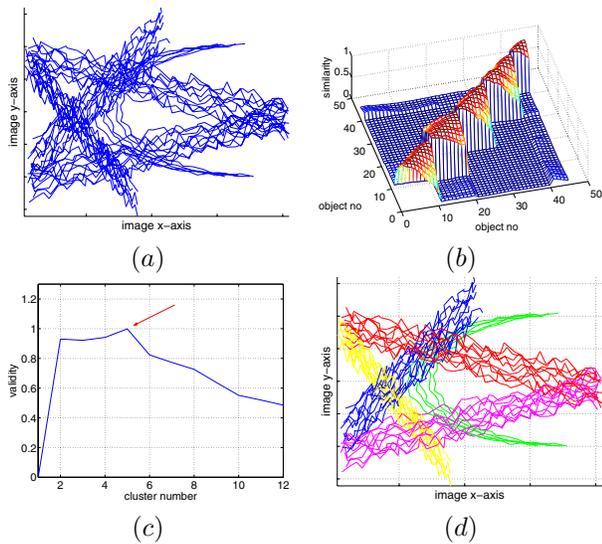


Figure 8: (a) Set of trajectories, (b) corresponding affinity matrix, (c) validity score, (d) result of automatic clustering.

4. Experiments

We conducted experiments using both synthetic and real data. For HMM representation of the coordinate, speed, and orientation. We used the same number of models and number of states. To make the simulations more challenging we contaminated the trajectories with noise.

Fig.11 shows three different simulated scenarios for detection of unusual events: 1) an object moving in opposite direction to the rest, 2) a waiting object where other objects moves, 3) a fast moving object. All of these scenarios may corresponds real unusual suspicious events, for instance the first scenario corresponds to a wrong-lane entry, the second scenario is a browsing or stalking activity, and the third scenario may be a running person in an airport where everybody walk. The trajectories for each case are depicted in Fig.11-a. The second column (Fig.11-b) shows the fused affinity matrices using the weights w_f . We compute the conformity scores $\beta(i)$ from the affinity matrices, which are given Fig.11-d. As visible, the conformity score is found the most unusual event accurately at each time (Fig.11-e).

We can also find a list of most unusual events using the conformity scores as shown in Fig.10 where the most unusual events were 1) a person moving across an highway, 2) a car backing up on the shoulder, 3) a person getting out of the car and leaving the scene, and a car slowing down in the shoulder. We can extent this list. Note that, we didn't adapt the weights or define models, the algorithm found the events automatically.

We simulated usual event detection using the trajectories given in Fig.8-a. In this set, there are 5 distinct pathways

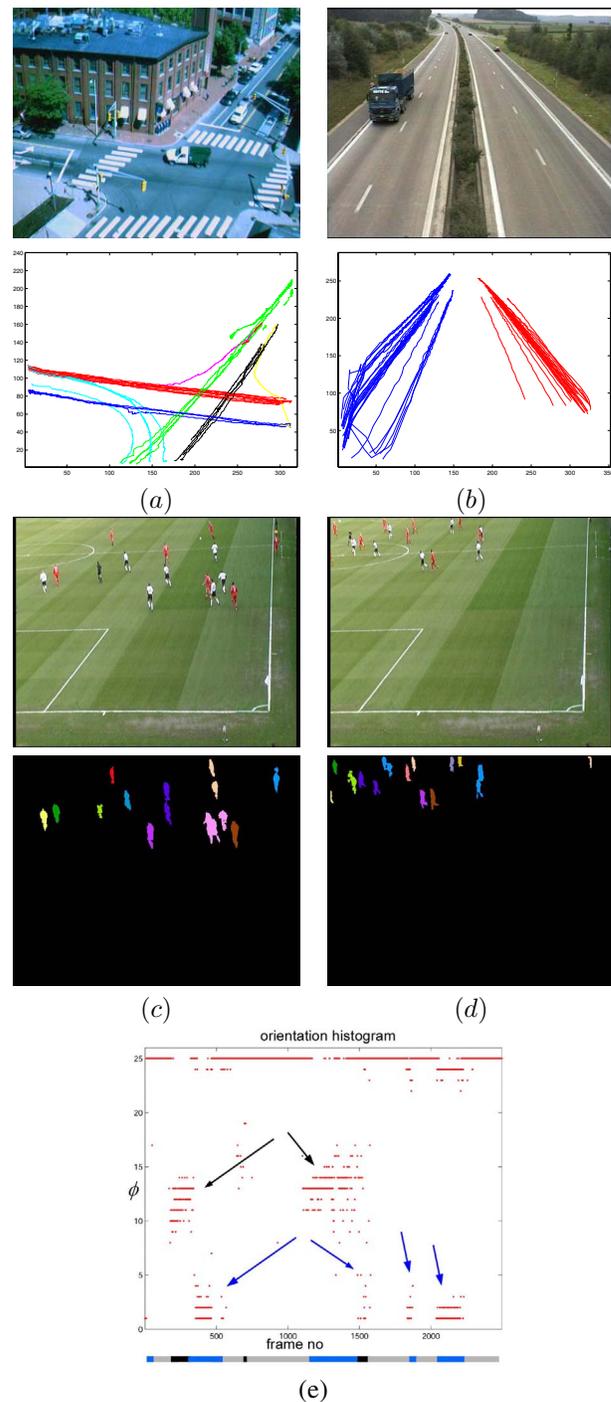


Figure 9: (a-b) Traffic lanes are learned using the HMM's for the object-wise coordinate and orientation sequences. (c-d) The team on attack is accurately determined using the frame-wise orientation histogram. (e) Orientation histogram and clustering results (marked under the horizontal axis) Black frames show group of frames in which the red team is on attack, blue shows indicate the white team is on attack, and gray indicates neutral situation.

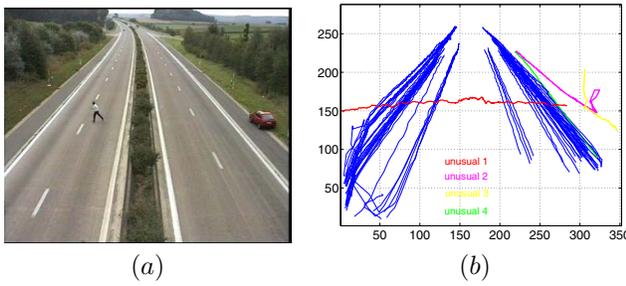


Figure 10: Automatic unusual event detection results: (a) frame shows the most unusual object who is crossing over the highway, (b) other 124 trajectories and unusual objects. No priori information is used.

exist. Fig.8-b shows the aggregated affinity matrix. We determined the optimal number of clusters using the validity score α as shown in Fig.8-c. The maximum validity score is obtained for $k = 5$ which is same as the ground truth. The clustered trajectories are given in Fig.8-d. As visible, the proposed method successfully found the correct clusters.

We applied the usual event detection to real data and used HMM's of coordinate and orientation sequences. As depicted in Fig. 9-a,b, we obtained the usual events as the different traffic lanes. We used frame-wise orientation histogram features for soccer video. We observed that the proposed algorithm automatically detected the usual events as the team on the attack. In Fig.9-c,d two frames that correspond to the two different usual events are shown.

Since the extended set of features are more expressive, we are able to detect events that cannot be detected using the features that have been reported so far. Our technique thus offers substantial improvement over the existing techniques in both computational simplicity and enhanced functionality. Our experiments also prove the proposed methods are stable.

5. Discussion

We proposed a new set of more expressive features that enable detection of events that could not be detected using conventional descriptors. We developed an unsupervised clustering framework based on the above and successfully applied it to event detection. This framework is not adversely affected by increases in feature dimensionality.

We achieve clustering of variable length trajectories by pair-wise affinities as opposed to unstable interpolation based approaches. We described a feature selection criteria to amplify the contribution of discriminative features. We also showed that the number of largest eigenvalues (in absolute value) to span subspace is one less than the number of clusters.

References

- [1] J. Davis and A. Bobick, "Representation and recognition of human movement using temporal templates", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997.
- [2] L. Davis, R. Chelappa, A. Rosenfeld, D. Harwood, I. Haritaoglu, and R. Cutler, "Visual Surveillance and Monitoring", *Proc. DARPA Image Understanding Workshop*, 73-76, 1998.
- [3] A. Ekinci, A. M. Tekalp, "Generic event detection in sports video using cinematic features", *Proc. IEEE Workshop on Detection and Recognizing Events in Video*, 2003.
- [4] M. Fiedler, "A property of eigenvectors of non-negative symmetric matrices and its application to graph theory", *Czechoslovak Mathematical Journal*, 25:619672, 1975.
- [5] G.L. Scott and H. C. Longuet-Higgins, "Feature grouping by relocalisation of eigenvectors of the proximity matrix" *Proc. British Machine Vision Conference*, 103-108, 1990.
- [6] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition", *Proc. British Machine Vision Conference*, 583592, 1995.
- [7] S. Kamvar, D. Klein, and C. Manning, "Interpreting and Extending Classical Agglomerative Clustering Algorithms using a Model-Based Approach", *Proc. ICML*, 2002.
- [8] V. Kettner, "Time-dependent HMMs for visual intrusion detection", *Proc. IEEE Workshop on Detection and Recognizing Events in Video*, 2003.
- [9] Z. Marx, I. Dagan, and J. Buhmann, "Coupled Clustering: a Method for Detecting Structural Correspondence", *Proc. International Conference on Machine Learning*, 353-360, 2001.
- [10] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, "Event detection and analysis from video streams", *IEEE Trans. on PAMI*, 23(8), 873-889, 2001.
- [11] M. Meila and J. Shi, "Learning Segmentation by Random Walks", *Proc. of Advances in Neural Information Processing Systems*, 2000.
- [12] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm", *Proc. of Neural Information Processing Systems*, 2001.
- [13] F. Porikli, "Trajectory Distance Metric Using Hidden Markov Model based Representation", *Proc. of European Conf. on Computer Vision*, Prague, 2004.
- [14] F. Porikli and O. Tuzel, "Human body tracking by adaptive gaussian models and mean-shift analysis", *Proc. Int'l Conf. on Computer Vision Systems*, Graz, 2003.
- [15] T. Starner and A. Pentland, "Visual recognition of american sign language using hidden markov models", *Proc. Int'l Workshop Automatic Face- and Gesture-Recognition*, 1995.
- [16] C. Stauffer and W.E. Grimson, "Learning patterns of activity using real-time tracking", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8), 747-757, 2000.
- [17] Y. Weiss, "Segmentation using eigenvectors: a unifying view", *Proc. of IEEE Int'l Conf. on Computer Vision*, 975-982, 1999.
- [18] L. Zelnik-Manor and M. Irani, "Event-Based Video Analysis", *Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, 2001.

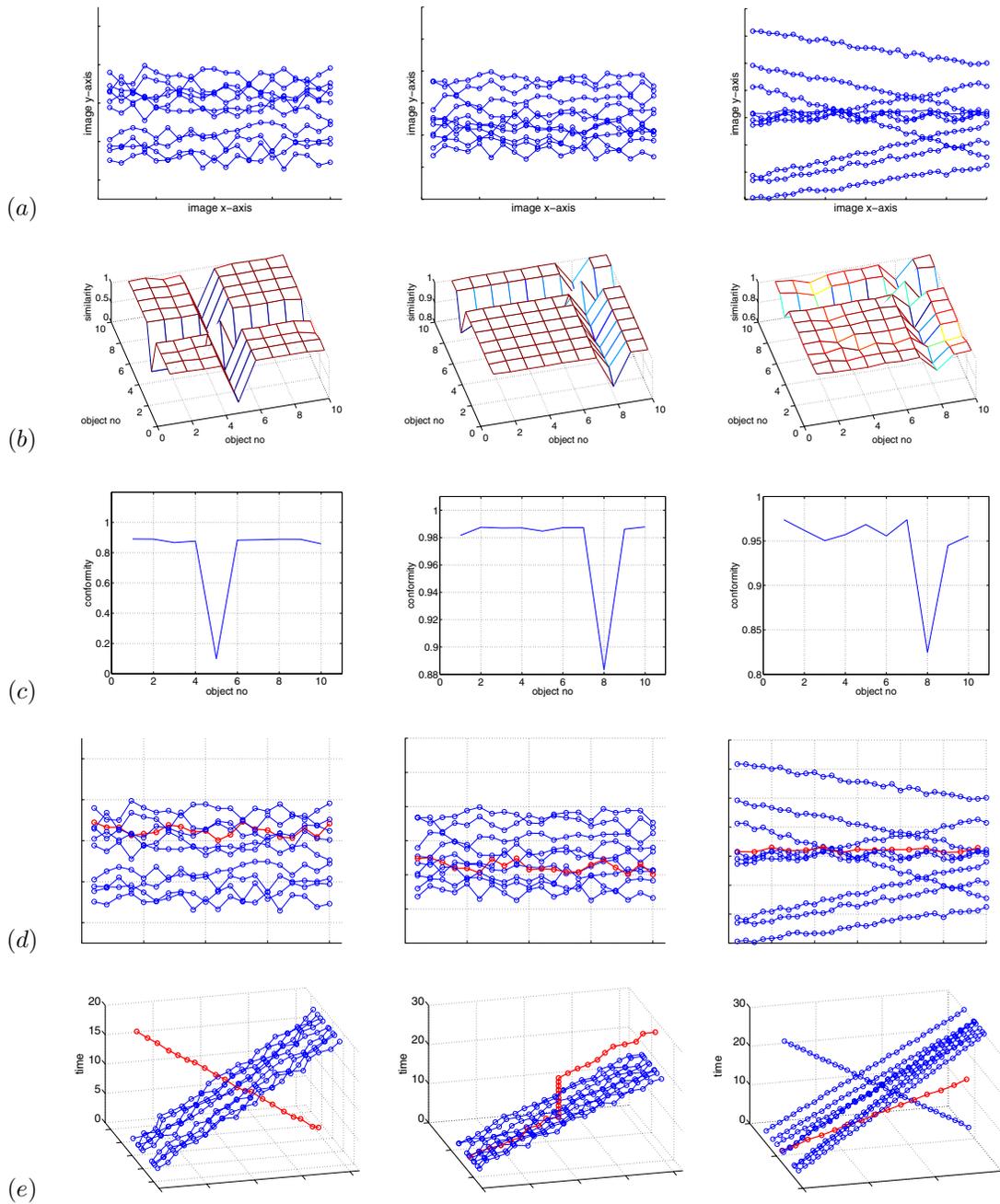


Figure 11: Unusual event detection: (a) input trajectories, (b) affinity matrices, (c) conformity scores (lowest score shows the most unusual), (d) detected most unusual trajectory (red), and (e) results in spatiotemporal space. First row simulates the wrong lane entry, second row simulates waiting, third row simulates running.