A Survey of Vision-Based Trajectory Learning and Analysis for Surveillance

Brendan Tran Morris and Mohan Manubhai Trivedi

***Abstract*—This paper presents a survey of trajectory-based activity analysis for visual surveillance. It describes techniques that use trajectory data to define a general set of activities that are applicable to a wide range of scenes and environments. Events of interest are detected by building a generic topographical scene description from underlying motion structure as observed over time. The scene topology is automatically learned and is distinguished by points of interest and motion characterized by activity paths. The methods we review are intended for real-time surveillance through definition of a diverse set of events for further analysis triggering, including virtual fencing, speed profiling, behavior classification, anomaly detection, and object interaction.**

***Index Terms*—Event detection, motion analysis, situational awareness, statistical learning.**

I. Introduction

SOCIETY is rapidly accepting the use of cameras in a wide variety of locations and applications: live traffic monitoring, parking lot surveillance, inside vehicles, and intelligent spaces. These cameras offer data on a daily basis that need to be analyzed in an efficient manner. Unfortunately, most visual surveillance still depends on a human operator to sift through this video. It is a tedious and tiring job, monitoring for interesting events that rarely occur. The sheer volume of these data impedes easy human analysis necessitating computer vision solutions to help automate the process and assist operators.

Automatic behavior understanding from video is a very challenging problem [1]. It involves extraction of relevant visual information, suitable representation of that information, and interpretation of the visual information for behavior learning and recognition [2]. This is further complicated by wide variability and unconstrained environments. Most monitoring systems are designed for specific environmental situations, such as a specific time, place, or activity scenario. Traditionally, the knowledge structures used for analysis were designed by hand. In these cases, a well-versed expert defined the events of interest for the particular application. By instead using machine learning techniques to automatically construct activity models, it will be better suited for online analysis because it is supported by real data [3].

Manuscript received November 22, 2007; revised March 11, 2008. First published June 17, 2008; current version published August 29, 2008. This work was supported in part by the National Science Foundation and the Technical Support Working Group. This paper was recommended by Guest Editor I. Ahmad.

The authors are with the Computer Vision and Robotics Research Laboratory, University of California, San Diego, CA 92093-0434 USA (e-mail: b1morris@ucsd.edu; mtrivedi@ucsd.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2008.927109

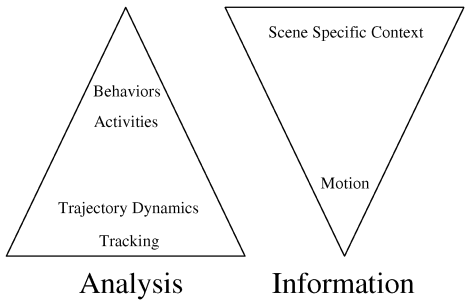


Fig. 1. Relationship between analysis levels and required knowledge: high-level activity analysis requires large amounts of domain knowledge while low-level analysis assumes very little.

II. Problem Description and Definitions

Activity analysis in surveillance is a challenging problem. In addition to the wide range of scenes, there are also many activities of interest each specified by the unique monitoring situation. Homeland security and crime prevention are two hot topics that require monitoring indoors and outdoors of critical infrastructures, highways, parking garages, and other public spaces. The environment must be monitored as well as everything within the scene, such as people, pets, or vehicles. With such a rich activity space, it is difficult to imagine general procedures capable of working over a wide range of scenarios.

The complementary pyramids in Fig. 1 depicts our view of video surveillance. At the bottom of the analysis pyramid are low-level processes such as tracking which uses very little prior information, just motion. Moving up the analysis pyramid provides more complex event representations but also requires the use of more domain-specific knowledge. The most complex behaviors can only be understood in the correct context.

Trajectory dynamics analysis provides a medium between low- and high-level analysis. It is assumed that change, in particular from motion, is the cue of interest for surveillance. In addition, typical motion is repetitive whereas some of the most interesting events rarely occur. This repetition enables event analysis in the context of learned motions. Rather than relying on domain knowledge, low-level cues are used to build an activity analysis procedure. This work formalizes activity analysis in surveillance video based on object tracking. A summary of frequently used terms and their definitions are presented Table I.

Trajectory dynamics analysis seeks to provide low-level situational awareness by understanding and characterizing the behavior of every object in the scene. Each object is identified and tracked to describe their activity and produce video annotations as outlined in the flow diagram of Fig. 2. The set of activities encountered in a surveillance scene reside in a high-dimensional spatio-temporal activity space. The scene model is developed by

TABLE I
TRAJECTORY ANALYSIS TERMS AND DEFINITIONS

Term	Definition
Behavior	A description of activities and events within a specific context.
Activity	A specific action performed by a subject.
Event	The occurrence of an activity in a particular place during a particular time interval.
Activity Pattern/Path/Route	The underlying hidden process that dictates how objects move in a surveillance scene. Pattern is the most general term, routes will signify patterns obtained by clustering, and paths refer to routes that have been explicitly modeled.
Lane	Spatial definition of a path (no dynamics).
Trajectory/Track	A realization of an activity that is extracted by visual tracking.
Typical	Signifies motions that occur frequently. (Note: we will refrain from the use of normal as this indicates some a priori scene knowledge.)
Abnormal/Anomaly/Unusual	Interchangeable terms to denote anything that does not fit into the typical category.

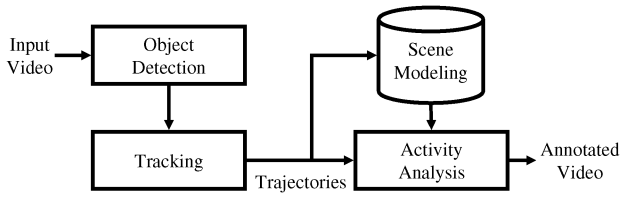


Fig. 2. Activity analysis block diagram: trajectories are used to automatically build a scene model which describes the surveillance situation and provides activity annotation.

first defining points of interest (POIs) as image regions where interesting events occur and the second learning stage defines activity paths (APs) which characterize how objects travel between POIs. By learning the POI and AP, a vocabulary to analyze an arbitrary scene is constructed in an unsupervised fashion based on the data. This vocabulary allows the following key analysis: classification of past and current activity, detection of abnormal activities, prediction of future activities, and characterization of interactions between objects, all of which can be performed in an online fashion on live video, which is paramount for active surveillance.

The main tasks involved in POI/AP learning are:

- activity learning—comparing trajectories in a manner that preserves the intuitive notion of similarity in spite of length mismatch;
- adaption—developing techniques to manage POI/AP models. They must be adapted online for the introduction of new activities, remove old activities with no support, and have model validation;
- feature selection—determining the correct level of dynamic representation necessary for a particular task. It is sufficient to use spatial information to determining what lane a vehicle travels but velocity information might be necessary for accident detection.

The POI/AP framework has successfully been used in many application domains, which are summarized in Table II. The intelligent transportation community has used the AP framework to model road lanes. By learning lanes, highway flow and congestion can be monitored [4], [5], the frequency of lane changes is estimated, the origin and destination (OD) information of vehicles at intersections can be mapped [6], and cameras can even be automatically calibrated [7]. The movements of people both indoors and outdoors have been extensively studied to determine where people tend to travel. Parking lots have been monitored to ensure proper use by detecting unusual driving patterns [8] and people loitering around parked vehicles [9], [10]. New exciting work has come forward to analyze interactions between humans, vehicles, and infrastructure, allowing monitoring of suspicious meetings [1] and luggage drops as well as characterization of conflicts for road safety [12]–[15]. Here suspicious meetings or luggage drops may be monitored as well as characterization of the conflicts for safety on shared roads.

It is important to note that POI/AP techniques require only tracking information, meaning that activity is defined by motion (or lack of motion when an object stops). Ancillary information sources must be used to describe more complex activities (higher level analysis). In order to distinguish a biker from a jogger, appearance information might be needed and environmental knowledge utilized to recognize the difference between people queuing at an ATM and vehicles at a stop sign.

III. AUTOMATIC SCENE MODELING

The following sections describe the semantic scene model introduced by Makris and Ellis [23]. A scene is modeled with a topographical map consisting of nodes and edges (Fig. 3). The nodes of the graph correspond to POIs while the edges, denoted as APs, encode the activity of an object.

A. Tracking

Tracking requires the identity maintenance of each observable object in every frame. An object tracked over T frames generates a sequence of inferred tracking states

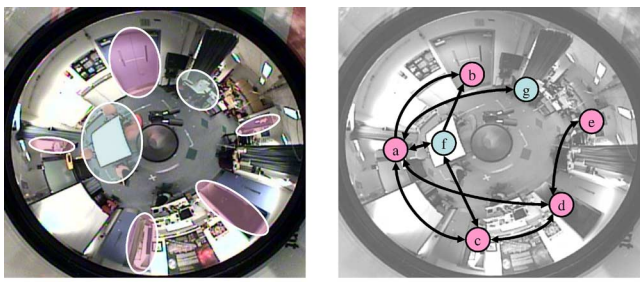
$$S_T = \{s_1, s_2, \ldots, s_T\} \tag{1}$$

where s_t can depict things such as position, velocity, appearance, shape, or other object descriptors. This trajectory information forms the basic building block for further behavioral analysis. Through careful examination of these signatures, activity can be recognized and understood.

Although tracking is well studied, there are still many difficulties due to perspective effects, occlusion, and real-time adaptability to changing conditions. These cause errors in the form of noisy measurements and incomplete or broken trajectories which must be accounted for by the scene learning process.

TABLE II
TRAJECTORY ANALYSIS APPLICATIONS

Application Type	References	Description
People Indoor	[16]-[21]	Walking in office, lab spaces, and hallways. Trajectories strongly influenced by scene configuration (eg. doors and halls).
People Outdoor	[11], [22]-[33]	Movement along sidewalks and other walkways. Less constrained than indoor environments as there are usually less scene obstacles.
Parking Lot	[8]-[10], [34]	Parking lots are analyzed to ensure proper use. This could include detection of suspicious driving patterns or people loitering around vehicles.
Traffic	[4]-[7], [12], [13], [17], [27], [29], [35]-[43]	Significant research done in the intelligent transportation community to study highway flows and congestion, intersection OD mapping, and camera calibration.
Interactions	[12]-[15], [32], [43]-[47]	Analyzes multiple objects (human/vehicle) and how they interact in a scene. Common analysis includes characterizing meetings between people, assessing danger when humans and vehicles are in proximity, conflict analysis at intersection, and surveillance threats such as luggage drops.



(a) (b)

Fig. 3. (a) POIs: pink ellipses indicate doorways (entry/exit zones) and the green ellipses mark desks (stop landmarks). (b) Topological map: POI connected through APs.

For further discussion, the interested reader is directed to recent surveys that categorize existing tracking techniques and identifies new trends [48] and the succinct review of tracking techniques specifically for visual surveillance [49].

B. Interest Points

The first scene modeling task is to discover interesting image regions. These are the nodes of the topographical map that indicate destinations of tracked objects. The two types of nodes considered are the entry/exit zones and stop zones as shown in Fig. 3(a).

The entry and exit zones are the locations where objects either enter or exit the camera field of view (FOV) or where tracking targets appear and disappear. These zones are most often modeled using a 2-D Gaussian mixture model (GMM) $Z \sim \sum_{i=1}^{W} w_i N(\mu_i, \Sigma_i)$ with W components and is learned using expectation maximization (EM) [50]. The entry-point dataset consists of the position from the first tracking state and similarly the exit-point set uses just the last state in a trajectory. The zones are over mixed to encompass all true zones and noise sources. The two can be separated using a density criterion [11]. The density of mixture i is defined as

$$d_i = \frac{w_i}{\pi\sqrt{|\Sigma_i|}} > L_d \tag{2}$$

which measures the compactness of a Gaussian mixture. The threshold

$$L_d = \frac{\alpha}{\pi\sqrt{|\Sigma|}} \tag{3}$$

TABLE III
POINTS OF INTEREST

Type	References	Description
Entry/Exit	[16], [22], [23], [45], [51]-[54]	POI where objects appear or disappear. Scene related examples are doors while view related POI come from the edge of the camera FOV
Stop	[16], [21], [23], [31], [43]	Location where object remain stationary. Objects are stopped (moving very slowly) or remain within a small radius.

indicates the density of an average signal cluster where $0 < \alpha < 1$ is a user-defined weight and Σ is the covariance matrix of all of the points in the zone dataset. Using the threshold L_d, tight mixtures indicate true zones and while wide mixtures imply tracking noise from broken tracks.

The second type of interest point comes from scene landmarks. These are the locations where objects tend to idle or remain stationary for some time, e.g., an office desk. These stop zones are locations that can be defined in two different ways, either as any tracking points with speeds less than a very low predefined threshold [23] or as all the points that remain in a circle of radius R for more than τ seconds [21]. By defining a radius and time constant, this second measure ensures objects actually remain in a particular location while the first could contain points from slow moving targets, as could be the case in a congested scene. The stop-point dataset can be constructed with points generated from the two tests and modeled using a GMM as described above for entry/exit zones. Besides knowing the location, the amount of time spent at each stop zone needs to be known for activity analysis. Yan and Forsyth [31] modeled the duration of stay with an empirical cumulative distribution while Makris [23] noted that stop duration could be adequately approximated with an exponential distribution. Work learning POI is summarized in Table III.

C. Activity Paths

To understand behavior, more than just the POIs are needed. It is necessary to look at time-varying action. The POI can be utilized to filter the training data removing noise from false detections or from broken tracks. Only trajectories that begin in a entry zone and end in an exit zone are retained. Tracks going

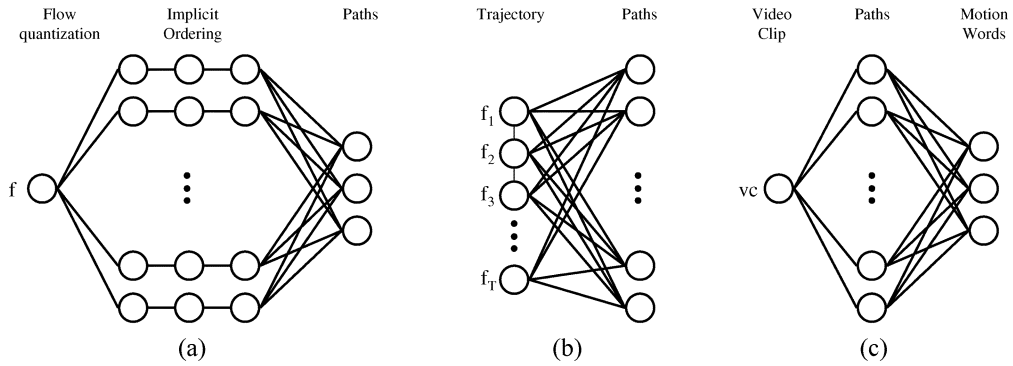


Fig. 4. Trajectory learning schemes. (a) Flow vectors are quantized and the sequence of codebook words represents a path. (b) A full trajectory sequence is used as input and output prototypes represent paths. (c) Video clips are broken into a set of motion words to describe behavior.

through a stop zone are divided into separate trajectories corresponding to an edge into and out of the zone. Activity is thus defined between interest points, in the way an object moves from one point of the topological graph to another.

In a highly structured environment, where the motion is constrained, it is relatively easy to learn the paths. Lane markers have been exploited by the intelligent transportation community for automatic discovery [7], [38], [41]. Because vehicles tend to travel in the middle of a lane, the activity density (changing pixels) has peaks at the lane centers and valleys at the lane dividers. These maxima provide a clear lane definition suitable for lane assignment.

The highway lane problem is a rather simple one because lanes are usually linear and viewed from an advantageous perspective which maintains the lane boundary/center distinction. A general procedure to learn complex and arbitrary paths from just tracking data is desired to remove any reliance on lane structure or geometry. In addition to knowing where lanes are located, activity paths denote behavior which is dependent on higher order dynamic information and temporal characteristics. In order to differentiate between a person walking or running along a sidewalk, temporal dynamics must be included in the path learning procedure to fully describe behavior. Fig. 4 depicts the basic structure of path learning algorithms. The three dominant types differ in the types of inputs, flow vectors, trajectories, or video clips, and the way motion is abstracted. In Fig. 4(a) the input is a single trajectory point at time t, points are connected temporally for implicit ordering into paths. An entire trajectory, Fig. 4(b), can be used as input into the learning algorithm to directly build paths. Fig. 4(c) depicts the video decomposition view of paths. Here, video clips are assigned an activity based on the occurrences of motion words.

IV. Learning Paths

Since a path characterizes how objects move, a raw trajectory can be represented as sequence of dynamical measurements. For example, a common trajectory representation is a flow sequence

$$F_T = \{f_1, f_2, \ldots, f_T\} \tag{4}$$

where the flow vector

$$f_t = \left[x^t, y^t, v_x^t, v_y^t, a_x^t, a_y^t\right]^T \tag{5}$$

represents an objects' dynamics, position $[x, y]$, velocity $[v_x, v_y]$, and acceleration $[a_x, a_y]$, at time t as extracted through visual tracking. Usually, only the position and/or velocity is used because acceleration estimates are typically noisy. Trajectories need not share the same length, even when traveling along a similar route, because objects may move at different speeds leading to mismatch in the number of samples.

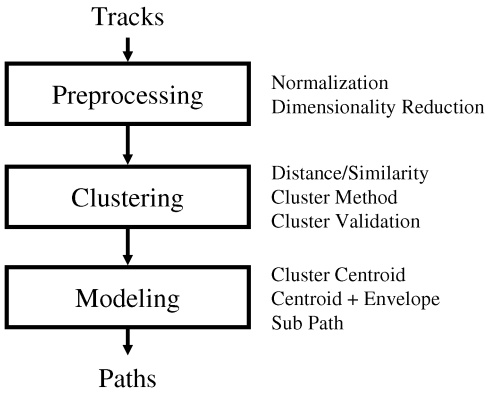


Fig. 5. Trajectory learning steps.

Using only trajectories, it is possible to learn APs in an unsupervised manner following the basic procedure depicted in Fig. 5. The preprocessing step is used to set up the trajectories for clustering which provides a summary and compact representation of a modeled path. Though presented as three sequential tasks, often the path learning steps are blurred with preprocessing, clustering, and modeling occurring in unison.

A. Trajectory Preprocessing

Most of the effort for path-learning research is spent producing trajectory representations suitable for clustering. The main difficulty when dealing with tracks is their time-varying nature which leads to unequal length. Steps must be taken to ensure a meaningful comparison between differing sized inputs. In addition, the trajectory representation should retain the intuitive notion of similarity present in raw trajectories for meaningful clusters.

Most researchers use some combination of trajectory normalization or dimensionality reduction to manipulate raw trajectories in ways that allow use of standard clustering techniques. Table IV summarizes the types of trajectory preprocessing procedures often encountered in literature.

1) Normalization: Normalization ensures that all trajectories have the same length $\hat{T}$. Two simple techniques for length normalization are zero padding [36] and track extension [13], [49].

TABLE IV
TRAJECTORY PREPROCESSING APPROACHES

1) Normalization
   a) Zero-padding [36]
   b) Track-extension [49]
   c) Resampling [16], [22], [23], [37], [39], [53], [55], [56]
   d) Smoothing [25], [39], [57]
2) Dimensionality Reduction
   a) Vector quantization [20], [40], [58], [59]
   b) Polynomial fitting [18], [38]
   c) Multi-Resolution Decomposition [18], [19], [39], [60]
   d) HMM [61]
   e) Subspace Methods [26], [57]
   f) Spectral Methods [20], [39], [57], [61], [62]
   g) Kernel Methods [63], [64]

When zero padding, extra $f_t = 0$ are concatenated to the end of a trajectory while track extension uses the dynamics at the last tracking time T to estimate extra trajectory points as if it had been tracked until time $\hat{T}$ [49]. A large training database is first analyzed and the prototypical size $\hat{T}$ is chosen to be equal to the length of the longest training trajectory. This makes the the trajectory space very large and can be subject to outliers (objects tracked for an unusually long time).

Instead of examining a training database to determine $\hat{T}$, it can be chosen *a priori* with resampling and smoothing techniques. Resampling guarantees all trajectories will be the same length by interpolating the original trajectory. Two popular choices are linear interpolation [22], [53], [55] or subsampling [37] to reduce the number of points. Liao *et al.* [56] used the Douglass–Peucker algorithm to find a minimal set of control points for accurate portrayal of a trajectory's shape by minimizing reconstruction error. Smoothing is used to remove noise from the trajectory and can be accomplished using a number of simple filters [25], [57] or using a low-resolution signal decomposition such as wavelets [39]. The resulting smoothed track can then be interpolated and sampled to a fixed size.

Normalization techniques usually operate on entire tracks, making them ill suited for analyzing incomplete trajectories obtained during live tracking. However, they are computationally simple and keep trajectories in an understandable space, preserving natural intuition.

2) Dimensionality Reduction: The following dimensionality reduction techniques map trajectories into a more computationally manageable space. The new trajectory space is lower dimensional for more robust clustering given less training data. The new space is chosen by assuming a trajectory model and finding parameters that best describe the model.

Vector quantization reduction is achieved by limiting the number of unique trajectories to a finite alphabet of prototypical vectors which symbolize all trajectories [8], [33], [58], [59].

When ignoring trajectory dynamics and relying only on spatial coordinates, trajectories can be treated as a simple 2-D curve. This signal can be approximated by a least-squares polynomial [6], [31]

$$x(t) = \sum_{k=0}^{m} a_k t^k \tag{6}$$

or Chebyshev polynomials [6] of degree m

$$x(t) = \sum_{i=0}^{m} b_k \cos\left(k \cos^{-1}(t)\right). \tag{7}$$

The terms a_k, b_k are used to describe a trajectory in the size m coefficient space.

Wavelet techniques give a representation of a trajectory at different levels of resolution. This inherently smooths the trajectory without destroying its shape or structural relationship. The amount of smoothing is controlled by choosing the appropriate wavelet level. Common wavelet basis functions are the simple square Haar [60] functions or exponentials as used in the discrete Fourier transform (DFT) [18], [19], [39].

Other modeling techniques assume that trajectories are produced by an underlying hidden stochastic process. Continuous Gaussian emission hidden Markov models (HMM) characterize the temporal dependencies between sample points well [61]. Each training trajectory is summarized by the hidden state parameters and the transition matrix which explains how to move from one hidden state to another. Clustering then occurs in the parameter space.

Principal component analysis (PCA) is a popular subpsace method for constructing a new space of lower dimensionality for improved clustering [26], [57]. The new space is spanned by the largest eigenvectors of the training set. Trajectories are then projected onto a subspace that accounts for most of the signal and removes directions or low variance.

Another large class of transformations that have become very popular for trajectory clustering are spectral methods [20], [37], [57], [61], [62]. A similarity matrix S is constructed for the training set where s_{ij} indicates the similarity between trajectories i, j. A new matrix called the Laplacian is formed

$$L = D^{-\frac{1}{2}} S D^{-\frac{1}{2}} \tag{8}$$

where D is a diagonal matrix whose ith diagonal element is the sum of row i of S. The L matrix is decomposed to find its K largest eigenvalues. The eigenvectors are placed into a new matrix whose rows correspond to the transformed trajectories in spectral space. The spectral trajectories are then typically clustered using k-means.

In a similar spirit, kernel methods have been derived from large-margin research to account for nonlinear clusters yet preserve structure and the ordering constraints of tracks [63], [64]. The kernel $K(i, j)$ defines trajectory similarity in a very high dimensional space but provides simple rules to separate clusters.

While clustering is more robust using dimensionality reduction techniques because it avoids the curse of dimensionality, the techniques are only as effective as the trajectory model. If the chosen model does not represent well the tracking process, the resulting clusters will be meaningless.

B. Trajectory Clustering

Clustering is a general machine learning technique to identify structure in unlabeled data. When observing a scene, motion trajectories are collected and are grouped into similar categories we have called routes. In order to generate meaningful

clusters, the clustering procedures must address three main issues: 1) definition of a distance (similarity) measure; 2) cluster update methodology; and 3) cluster validation. Liao presents a complete survey of clustering techniques for time-series data [73] which is applicable to route learning.

1) Distance/Similarity Measures: Clustering techniques rely on definition of a distance (similarity) metric in order to compare tracks. As stated previously, the main difficulty with trajectory clustering is the potential for differing length tracks generated by the same activity. This discrepancy can be combat through the preprocessing techniques or by careful definition of a size independent distance measures.

The Euclidean distance between two trajectories (of equal size) F_i, F_j is computed as

$$d_E(F_i, F_j) = \sqrt{(F_i - F_j)^T (F_i - F_j)}. \tag{9}$$

A simple size-invariant modification of the Euclidean distance to compare two unequal length vectors, F_i, F_j with length m, n, respectively $(m > n)$, uses the last point $f_{j,n}$ to accumulate distortion [30]

$$\bar{d}_{ij} = \frac{1}{m}\left(\sum_{k=1}^{n} d_E(f_{i,k}, f_{j,k}) + \sum_{k=1}^{m-n} d_E(f_{i,n+k}, f_{j,n})\right). \tag{10}$$

Though simple, Euclidean distance performs poorly in the presence of time shifts as only aligned sequences match well. Trajectories can be optimally, aligned at the cost of added computation, before computing distance to deal with mismatched points. Dynamic time warping (DTW) [74] has been used in speech recognition literature to find the distance between unequal length signals. A dynamic program is solved to find an optimal alignment between two trajectories by minimizing the distance between matched points. Longest common subsequence (LCSS) analysis is another popular alignment technique because of robustness to noise [75], [76]. Rather than find a match between all points in each trajectory, outliers may be disregarded and remain unmatched. In a similar spirit, Piciarelli and Foresti [67] defined a new distance measure that does not depend on having the entire trajectory for computation. Assuming trajectory $F_i = \{f_{i,k}\}$, $F_j = \{f_{j,l}\}$ have length T_i, T_j respectively,

$$D_{PF}(F_i, F_j) = \frac{1}{T_i}\sum_{k=1}^{T_i} d_{PF}(f_{i,k}, F_j) \tag{11}$$

where

$$d_{PF}(f_{i,k}, F_j) = \min_{l}\left(\frac{d_E(f_{i,k}, f_{j,l})}{Z_l}\right) \quad l \in \{\lfloor(1-\delta)k\rfloor \ldots \lceil(1+\delta)k\rceil\}. \tag{12}$$

Z_l is a normalization constant that measures the variance of point l. ($D_P(F_i, F_j)$ is used to compare trajectories to existing clusters where a point variance measure makes sense. If comparing two trajectories, one would use $Z_l = 1$.) This defines a distance measure that is the mean of normalized distances from every point to its best match in a sliding temporal window of length 2δ centered on l.

The Hausdorff distance that measures the distance between two unordered sets has also been used to compare trajectories [25]. The distance is defined for unequal length data, but because it does not take into account ordering may incorrectly match dissimilar trajectories (walking different directions on a sidewalk). The Hausdorff distance is symmetrically defined as

$$D_H(F_i, F_j) = \max\left(D_h(F_i, F_j), D_h(F_j, F_i)\right) \tag{13}$$

where

$$D_h(F_i, F_j) = \max_{f_{i,k}}\left(\min_{f_{j,l}} d_E(f_{i,k}, f_{j,l})\right) \quad \forall k, l. \tag{14}$$

A modified Hausdorff distance $h_\alpha(F_i, F_j)$ [62] better suited for trajectory data has been introduced. It respects point ordering by incorporating direction and minimizes the effect of outliers from noisy points

$$h_\alpha(F_i, F_j) = \underset{f_{i,k}\in F_i}{\overset{\alpha}{\text{ord}}}\left(\min_{f_{j,l}\in N(C(f_{i,k}))} d(f_{i,k}, f_{j,l})\right) \tag{15}$$

where $N(f_j)$ is the neighborhood of f_j within F_j, $C(f_{i,k})$ is the set of points in F_j that correspond to a point $f_{i,k}$ in F_i, and $\text{ord}^{\alpha}_{f_{i,k}\in F_i} g(f_{i,k})$ denotes the value of $g(f_{i,k})$ that is larger than the fraction α of all values of g over F_i.

Rather than using the distance calculation directly, a similarity measure [37] can be derived from any of the distance measures as

$$s_{ij} = e^{-D(i,j)/\sigma^2} \tag{16}$$

where σ is a parameter to control how quickly similarity drops with increasing distance $D(i, j)$ between trajectories i and j. This helps abstract the distance metric from data comparison by adding another parameter to tune for performance.

2) Clustering Procedures: Once trajectories have been properly preprocessed, they can be grouped using unsupervised learning techniques. The grouping partitions the trajectory space into perceptually similar clusters called routes. There are a number of different techniques that have been employed for route learning: 1) iterative optimization; 2) online adaptive; 3) hierarchical; 4) neural networks; and 5) co-occurrence decomposition. The primary algorithms along with their strengths and weaknesses for route clustering are summarized in Table V. A more detailed summary of general clustering is available in the survey works of Jain and Berkin [77], [78].

1) Iterative Optimization—The most popular of the clustering techniques because of simple and tractable optimization procedures. Using standard Euclidean distance, closed-form solutions exist to find all of the cluster prototypes in a single iteration. Standard K-means along with its soft variant fuzzy C means (FCM) are the most popular of these techniques, though both require all trajectories be normalized to a fixed length.
2) Online Adaptive—Unlike the iterative optimization techniques, a large training database of trajectories does not need to be collected before building routes. As new tracks are seen they can be incorporated into the model set. In addition, there is no need to set the number of tracks *a priori*, which is difficult to do for a new scene because this is usually unknown. An additional learning parameter must be specified to control the rate of route update. These techniques are of particular interest because they are very well suited to long term, time-varying scenes because clusters are continually updated and adapt to changes.

TABLE V
SUMMARY OF WIDELY USED CLUSTERING TECHNIQUES FOR AUTOMATIC ACTIVITY PATH LEARNING

	Examples	Strengths	Weaknesses
Iterative Optimization	K-Means [57], [62], [65] FCM [36], [37], [55], [66]	Tractable update equations which solve for all cluster prototypes simultaneously. Generally, quite effective despite simplicity.	The number of clusters must be manually specified and data must be of equal length which could destroy dynamic information.
Online Adaption	Similarity Threshold [23], [67] I-kMeans [60]	Well suited for real-time applications because the number of clusters is not specified and adapts to changes over time. There is no requirement for collection or maintenance of a training database.	It may be difficult to specify a criteria for new cluster initialization that prevents outlier inclusion and there may not be any optimality guarantees.
Hierarchical Methods	Agglomerative [26] Divisive [25], [39]	Multi-resolution clustering allowing intelligent choice of the number of clusters. Well suited for graph theoretic techniques such as max-flow/min-cut [68] and dominant set clustering [69], [70] which make binary divisions.	The quality of clusters is dependent on the decisions of how to split (merge) a set and do not usually re-evaluate or adjust these decisions further along the tree.
Neural Networks	SOM [28], [33], [71] Fuzzy SOM [18], [19], [27] SOFM [9], [35]	Complex non-linear relationships are represented in a low-dimensional structure that preserves similarity between close output nodes. The networks can be trained sequentially to update given unseen examples.	Large networks require lots of data and may suffer from long convergence time and difficulties setting neuron weights and learning parameters.
Co-Occurrence Methods	Document-Keyword [29], [40], [59], [72]	Naturally independent of trajectory length because of the use of co-occurrence matrix from a finite vocabulary. Active area because of document retrieval research.	The vocabulary size may be limited for effective clustering and time-ordering is not generally preserved.

3) Hierarchical—There are two main hierarchical clustering variants, agglomerative and divisive, which define similarity relationships between trajectories in a tree-like structure following a bottom-up or top-down procedure, respectively. The root node corresponds to the full dataset while the bottom nodes to individual tracks. The tree structure provides clusters at different resolutions allowing a suitable clustering to be chosen by cutting the tree at a given level without knowledge of the true number of clusters. While multiscaled, each similarity decision is usually "hard" preventing adjustment further along, meaning errors can propagate.
4) Neural Networks—Using the self-organizing map (SOM) introduced by Kohonen [71] trajectories can be clustered in a low-dimensional arrangement preserving topological properties. Each output node of the neural network corresponds to a single route and neighboring nodes correspond to more similar routes. By employing neural networks, highly nonlinear clusters can be learned in trajectory space. These networks can be trained sequentially and easily updated with new examples but may suffer from long convergence time due to the complexity of setting weights and learning parameters as well as the need for large amounts of data.
5) Co-Occurrence Decomposition—The extensive work and success of document retrieval (document-keyword clustering) and natural language processing inspires similar frameworks for route learning. Trajectories are viewed as a bags of words where similar bags contain similar words. A co-occurrence matrix is formed from training data and decomposed to build document subjects (routes). These techniques must define a set vocabulary which may be limited in size.

3) Cluster Validation: The quality of path learned with a clustering algorithm must be verified because the true number of clusters is unknown. Most clustering algorithms require an initial choice for the number of expected clusters which is unlikely to be correct. Morris and Trivedi [16], [55] over clustered the scene and used an agglomerative merge procedure to combine similar cluster prototypes. Other techniques find the correct number of clusters by minimization (maximization) of some optimality criterion. Clustering is performed a number of times by varying the initial number of clusters K. The K that performs best is chosen for the true number of clusters. One such criterion is the tightness and separation criterion (TSC) [36], [37] which measures how close trajectories in clusters are compared to the distance between clusters. Given a training set $D_T = \{F_1 \dots F_M\}$, then

$$TSC(K) = \frac{\frac{1}{M}\sum_{j=1}^{K}\sum_{i=1}^{M} u_{ij}^2 d_E^2(F_i, v_j)}{\min_{i,j} d_E^2(v_i, v_j)} \tag{17}$$

where u_{ij} is the fuzzy membership of trajectory F_i to cluster C_j as represented by prototype v_j. A similar distortion score was adopted by Atev *et al.* [62] and Porikli designed luster validity score [61] for spectral clustering. Other validation criteria have come from information theory, such as Bayesian information criterion (BIC) [8].

C. Path Modeling

Once trajectories have been clustered, the resulting paths are modeled for efficient inferencing. The path model is a compact representation of a cluster partition. Paths have been modeled in two different fashions. The first considers a path in its entirety, from endpoint to endpoint [Fig. 6(a)], while the second decomposes a path into smaller atomic parts called subpaths [Fig. 6(b)]. A summary of the different path models and the research work they can be found in is provided in Table VI.

A path is minimally specified by its centroid. The centroid corresponds to a cluster prototype and specifies how an expected trajectory from the given activity should appear. The path can be further specified by augmenting the centroid with a path envelope. The envelope functions as the lane markers on a road, denoting the extent of a path. This idea can be further extended through probabilistic modeling where the path centroid is the model mean and the envelope represents the variance. The prob-

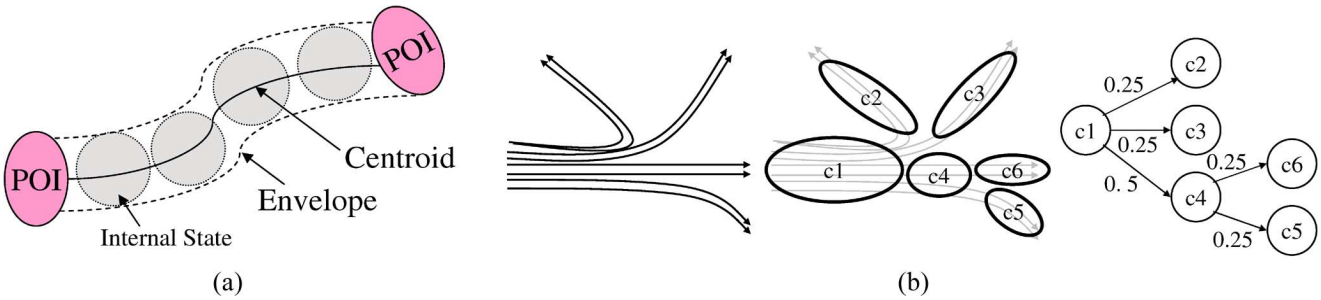


Fig. 6. (a) Full path model: paths have an average center and an envelope denoting path extent with optional internal states to model measurement ordering. (b) Paths represented as a tree of subpaths. The predicted path probability is found by the product of edges from a given node to a leaf node.

TABLE VI
PATH MODELS

Type	References	Description
Centroid	[13], [18], [19], [27], [35], [37]-[39]	Prototypical representation of route summarizing a cluster. The "average" trajectory (a sequence of points) is returned by most clustering algorithms
Envelope	[7], [22], [23], [25], [34], [36], [41], [55]	In addition to centroid the extent of a path is specified. This measures the variation along a path. The two most common representations are the extremal points associated with a route or a Gaussian distribution. HMMs provide an efficient way to learn distributions.
Sub-Paths	[57], [65], [67]	Rather than specify a path in its entirety smaller segments are retained based on similar curvature or when splits occur. Each sub-path is additionally characterized by transitions between one another. This representations work well for lanes (spatial paths) by sharing data.

abilistic models are estimated from the data, where each cluster partition is used to learn an individual path model [36], [55]. Gaussian observation emission HMMs are commonly used because sample ordering is enforced with the transition matrix and effective comparison techniques between trajectories and paths. The standard HMM has been extended to model the duration of time spent in a state [79] to help account for variation in activity time-scales and speed.

In contrast to the full path techniques, subpath methods divide up the trajectory space into atomic elements. Each of the atomic elements represent similar regions of a path such as portions of paths before splits [67] or parts of a path with similar curvature [57]. Subpaths are further defined by their connections with other subpaths. These connections indicate the possible transitions between subpaths and decompose a trajectory into a subpath traversal list.

D. Path Feedback to Low-Level Systems

Though intended for higher level analysis, learned paths are useful as feedback to the lower level functions as well [80]. Shadows suppression was improved on a highway by learning the lane marker positions and removing those cast shadows that fell over a lane line [41]. Tracking robustness was improved by learning the static scene occlusion landscape giving a depth estimate to tracked objects [81]. The paths have also been used as a form of state prediction to direct tracking association [34], [53].

V. AUTOMATIC ACTIVITY ANALYSIS

Once the scene model has been constructed, object behaviors and activities can be analyzed. One of the basic functions of surveillance video is identification of interesting events. In general, it is difficult to define interesting except in a specific context. Parking garage monitoring might be concerned with the availability of open stalls while interactions between people occur in an intelligent conference room. Besides just recognizing typical behavior, all atypical events should be examined as well. By observing a scene over time, the system can learn what is interesting to perform a range of activity analyses such as virtual fencing, speed profiling, path (activity) classification, abnormality detection, online activity analysis, and object interaction characterization. Table VII reviews a list of exemplary activity path modeling systems and their associated activity analyses.

A. Virtual Fencing

A basic event trigger for any surveillance system is perimeter sentry. Monitoring zones or virtual fences are erected in the image plane to issue intelligent alerts when breached. These alarms could be used to initiate control of other cameras in the network such as a high resolution PTZ camera to obtain finer event details for person recognition [82] or vehicle classification [83]. By monitoring entry/exit zones, vehicle counts can be accumulated for traffic flow analysis [83], [84] or detailed OD mapping [6]. When stop zones are characterized by the amount of time objects usually remain idle within them, it is possible to detect loitering people [21], [21], [24] if they remain in an area for an unusually long time.

B. Speed Profiling

Virtual fences only take advantage of spatial information. Tracking also gives dynamics which can be used for speed based alarms. Vehicle velocity measurements have been used to categorize speeding behavior [4] or highway congestion from stalled vehicles or accidents [5]. The bounding boxes in Fig. 7 indicate the speed state of each vehicle with respect to daily averages [55]. Red denotes stopped, yellow slow moving, green the speed of normal travel, and blue to mark a speeding vehicle. Junejo *et*

TABLE VII
COMPARISON OF EXEMPLARY ACTIVITY PATH MODELING TECHNIQUES

Method	Publication	Path Usage	Comments
Flow			
	Johnson 1995 [28]	Classification	Paths learned using a dual-layer SOM connected by leaky neurons which contain the memory of a path and output nodes indicate a path cluster. The trajectories were linearly resampled for consistent point density during training and used a flow vector consisting of both spatial and dynamics information as input.
	Sumpter 2000 [33]	Prediction	The leaky SOM architecture was extended for prediction by incorporating feedback from the output nodes to the leaky neurons.
	Owens 2000 [9]	Classification, Abnormality	Trajectories were smoothed using a temporal moving average window and the resulting flow vectors used as input to a SOFM. Abnormalities were detected using a threshold set as half the maximum distance from a winning neuron in the training set.
	Stauffer 2000 [29]	Classification	Trajectories were quantized into a codebook describing position and velocity used to form a flow co-occurrence matrix. The complete training matrix was hierarchically decomposed in binary tree-like fashion describing a pmf of codewords. The order order of flow codewords is lost in the procedure.
Track			
	Porikli 2004 [61]	Classification	A trajectory was modeled by an HMM whose parameters were used as a feature vector for clustering. Cluster centers were formed using a mutual fitness similarity measure. In addition, number of clusters was estimated using a validity score.
	Junejo 2004 [25]	Classification, Abnormality	Path clusters were formed using the min-cuts graph algorithm with trajectory nodes and associated Hausdorff distance as the edges. The paths consisted of the center plus a spatial envelope and was augmented with a velocity and curvature profile for abnormality characterization.
	Makris 2005 [23]	Classification	Paths with a center and spatial envelope were learned in an online fashion for adaptability to changing conditions. Introduce the an activity-based semantic scene model based on POI/AP.
	Bashir 2005 [57]	Classification, Prediction	Trajectories were broken into atomic sub-trajectories based on curvature. The sub-tracks were projected into a PCA space for k-means clustering. The learned sub-paths formed the states of a Markov model allowing prediction of behavior from sub-path transitions.
	Piciarelli 2006 [67]	Classification, Prediction	Paths learned in an online fashion allowing paths to split into a tree-like structure with shared nodes common to the children. Trajectories were compared without normalization using a novel time-windowed Euclidean distance. The tree representation allowed path prediction by estimating the transitions between nodes and children in the training set.
	Hu 2006 [36]	Classification, Prediction, Abnormality	Paths are learned in a two-stage FCM clustering procedure. The first stage used only spatial information and verified clusters using the TSC. The second clustering stage included velocity and the resulting clusters were modeled by a chain of Gaussian probability distributions.
	Morris 2008 [16]	Classification, Prediction, Abnormality	POI are learned for a scene and used to determine likely paths. Initially, only spatial information was used to over-cluster routes with a merge verification step. The paths were modeled by HMMs with velocity included to handle the normal speed of activities.
Video			
	Zhong 2004 [59]	Classification, Abnormality	A codebook of visual words was created from $m \times m$ motion histograms. The set of words in an activity was learned through bipartite graph co-clustering. The resulting eigen decomposition produces an embedding mapping into the activity space.
	Wang 2007 [40]	Classification, Abnormality	Adapts word-document clustering techniques, LDA and HDP, for visual monitoring. A codebook of flow vectors is defined as the vocabulary and grouped into a set of topics (paths). The distribution of topics in a video clip succinctly described the activity.
	Xiang 2008 [72]	Classification, Abnormality	Trajectory points were clustered using a GMM into a set of activities. The posterior probability of all activities was modeled with a multiple observation HMM (MOHMM) that allows computation of a similarity matrix for spectral clustering, resulting in a composite behavior model for a video clip.

al. [25] used a Gaussian distribution to model the average speed along a path for anomaly detection.

C. Path Classification

The previously discussed behavior analysis tools only relied on the current tracking data to issue alerts which neglects the APs derived from historical motion patterns. The behavior of a novel object is described by finding the maximum *a posteriori* (MAP) path

$$\lambda^{\star} = \arg\max_k p(\lambda_k|F) = \arg\max_k p(F|\lambda_k)p(\lambda_k). \quad (18)$$

This determines which activity path best explains the new datum. The prior path distribution $p(\lambda_k)$ can be estimated from the cluster density or frequency in the training set [36]. The problem can be reduced to a maximum-likelihood (ML) estimation by concentrating on the likelihood $p(F|\lambda_k)$ of path k as

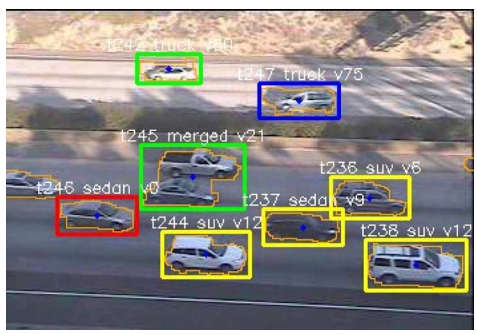


Fig. 7. Speed profiling: {Blue, Green, Yellow, Red} = {Speeding, Normal, Slow, Stopped}. The north speed is ∼65 mph while south is ∼27 mph.

(a) (b) (c)

Fig. 8. Trajectory analysis for abnormality detection. (a) Car crossing lane dividing line causing abnormal trajectory. (b) Abnormal trajectory from car doing 360° loop. (c) More subtle abnormality that might be missed by casual observer. The trajectory shows a person walking along the edge of the room.

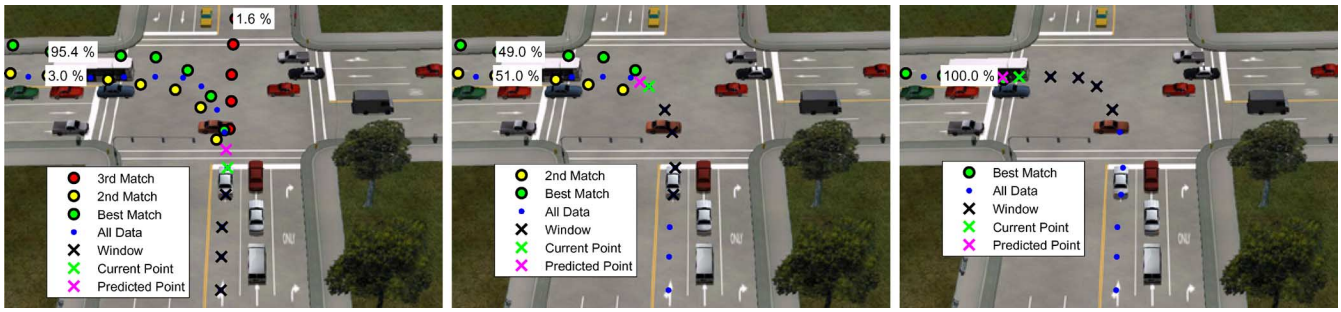


Fig. 9. Trajectory analysis for left turn prediction at intersection showing the probability of the top three best paths.

done with HMMs [16], [22], [55]. When only path prototypes are available, the cluster that best matches the novel trajectory approximates the ML solution. The label of the prototype with minimal distance indicates the best matching path [28], [49], [67].

D. Abnormality Detection

Perhaps the most important task in a surveillance system is the detection of unusual events. Abnormal behaviors can be drastic, as shown in Fig. 8(a) and 8(b), and are easily noticed by human monitoring or they can be subtle as in Fig. 8(c), making manual perception difficult.

Since APs denote typical activities, if a new trajectory does not fit a path well it can be considered an abnormality. Anomalies should rarely occur, thus lack visual support. Abnormal patterns can then be detected by intelligent thresholding

$$p(\lambda^{\star}|F) < L_{\lambda^{\star}} \tag{19}$$

where the most likely AP $\lambda^{\star}$ given a new trajectory F is less than a threshold $L_{\lambda^{\star}}$. The threshold can be tuned for each path individually based on its unique characteristics chosen, for neural network implementations, as half the maximum distance between training samples and a winning neuron (path prototype) [9], [49]. This led to a large number normal tracks misclassified as abnormal. In Junejo's multifeature path model, a path was considered abnormal if it did not fall within the spatial path envelope and fit both a Gaussian velocity and curvature profile [25]. Hu used the minimum probability encountered in a path training set as the threshold [36].

While the previous thresholds were effective, the abnormality threshold should not be static but be adjustable to tune the sensitivity for a given application and control the TP/FP rate [16]. In some places, it will be crucial not to miss any true abnormalities at the cost of false positives while on other occasions false positives may unnecessarily bog down the system.

E. Online Activity Analysis

While it is important to be able to take an entire trajectory and describe motion, it is even more important to recognize and evaluate behavior as it occurs in an online fashion. A real-time system must be able to do behavior inferencing quickly with incomplete data. With online trajectory analysis, the intent of objects can be predicted and tracking anomalies detected.

1) Path Prediction: The tracking data up to time t can be leveraged to infer future behavior and as more information is gathered the prediction is refined. The intent is conditioned on the set of acceptable behaviors (APs) allowing for better long-term prediction. The best three predicted paths are shown for a left turn at an intersection in Fig. 9.

Activity prediction uses an incomplete trajectory

$$\hat{\lambda} = \arg\max_{j} p(\lambda_j | w_t \hat{F}_{t+k}) \tag{20}$$

where w_t is a windowing function and $\hat{F}_{t+k}$ is the trajectory up to the current time t as well as k predicted future tracking states (obtained by extending the motion model k steps into the future). Different temporal windows will affect the tradeoff between the accuracy of prediction based on historical data states and the delay in recognizing a new behavior.

When no window function is used, trajectory data up to the current time is used to make predictions [36]. An object may engage in a number of different behaviors during the course of

tracking, and if all past data is used it is not possible to distinguish them, e.g., lane changes on the highway. A Gaussian window

$$w_t = e^{[(i-t)/t]^2} \tag{21}$$

has been used to decrease the contribution of those older samples [27]. Similarly, a rectangular window [16], [55] has also been used to consider only a short time frame during prediction. This speeds up evaluation in the case of long trajectories whose distant samples are unlikely to contribute to the prediction accuracy.

When a subpath graph is present, path prediction is accomplished by finding the probability of ending in a leaf node [Fig. (6b)]. The transitions between subpaths are learned by approximating probabilities through counting of the training set. Thus, by knowing the current subpath node, the predicted trajectory of the object can be mapped out by following the node transitions with highest probability.

2) Tracking Anomalies: Besides classifying a complete trajectory as abnormal, any out-of-ordinary events should be detected as they occur. This type of alert can be issued during live tracking by substituting $w_t\hat{F}_{t+k}$ for F in (19). The window function w_t does not have to be the same as used for prediction and the threshold $L_{\lambda^\star}$ might need to be adjusted because of less data for practical implementation [13], [42]. As soon as an anomaly occurs, a flag can be raised.

F. Object Interaction Characterization

The last level of automatic analysis seeks to describe object interactions. Like abnormal events, it is difficult to strictly define object interactions because of the wide variety of possible types. Different scenes may have very different types of interactions because of the environment or even the types of objects present. The interactions encountered on a highway are very different than those in a classroom because one monitors vehicles while the other people.

Vehicle collisions have been detected by the intersection of overlapping bounding boxes or 3-D models [13], [42]. This overlap concept can be generalized with the personal space abstraction from psychology. This is the area a person considers to be his territory and is used to define proximity. This surrounding region corresponds to a minimal comfort distance. In the past, this has been defined statically based on object size [43], but really personal space is dependent on environmental and socio-cultural contexts. This insight leads to an adaptable extension to the spatio-temporal personal space that is not static but changes shape due to motion [45]. The white area around a colored object in Fig. 10 indicates the extent of personal space, with it extending further in the direction of travel.

Personal space can be extended for collision and conflict severity analysis where the safety of a site can be assessed such as traffic intersection analysis [12], [15], [47]. In reality collisions are very rare events, the true safety of the scene is better estimated by examining conflicts and avoidance maneuvers [85]. These are instances when an involved target must change its behavior to avoid an accident. These near misses can be detected using the personal space formulation rather than based on a true collision of vehicles. In addition, routes can be extended to postulate when collisions might occur by intersection along predicted paths as shown in Fig. 10. By having accurate path predictions, the safety of an intersection can be gauged by the number of potential collisions.

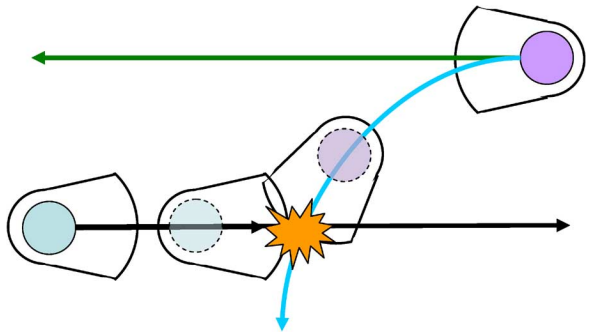


Fig. 10. Collision assessment: using intended paths, a collision likelihood can be assigned to trajectories with predicted intersection between personal space areas.

VI. Future Developments

While the POI/AP framework has been quite successful for a variety of tasks in many surveillance applications, there are still open issues. These range from low-level vision problems such as robust object detection to high-level semantic interpretation as well as developing a concrete definition of what truly is an abnormality.

A. Improved Tracking

Most of the trajectory based analysis systems rely on perfect tracking. Further analysis is meaningless if objects can not be accurately localized and tracked. Algorithms to handle tracking noise, because of broken tracks due to occlusion or crowds, must be developed for a completely unsupervised system in the real-world. Instead of tracking in the image, using normalized ground plane or 3-D coordinates could produce trajectories in world coordinates for better separation and clustering since perspective distortion is minimized.

B. Dynamics and Behaviors

Often, the dynamics are disregarded because trajectories are resampled and, in this process, length, which is a simple indicator of speed, is lost. Many works describe good performance on their datasets, but it is unclear what exactly is being tested. Deeper analysis is needed to define the types of activities spanned using just position versus adding velocity. In addition, more complex activities can be recognized through inclusion of other measurements such as curvature profile used by Junejo *et al.* [25] to detect erratic walking.

C. Clustering Performance

As we have shown, there are many variations to path learning with little agreement on what are the best methods. Zhang *et al.* [86] compared different distance measures using spectral clustering and found that the more complex DTW and LCSS distances did not perform significantly better than Euclidean variants because the shape of their paths were simple. The clustering comparison by Jain *et al.* [87] suggest there may only be

five major classes of clustering techniques to try as others produce similar partitions. A complete comparison of distance/similarity measures, clustering methods, and validation schemes is paramount to furthering trajectory based analysis rather than focusing on the learning methodology.

D. Activity Model Management

A critical feature for a usable surveillance system is the ability to adapt to changing environments for long-term analysis. We presented a few online adaptive techniques [23], [67] but all path learning methods should be able to introduce new models and remove old ones. Things that initially might be considered abnormal could become normal, and old activities that are no longer supported should not add complexity to the analysis evaluation.

E. Path Decomposition

A few works have used subpaths to describe activities because it naturally fits how one might reason about a fork in the road. Using subpaths allows data from differing end activities to share data for more efficient learning and provides a simple means for prediction. However, it is not known whether this truly helps with the activity analysis. Subtle cues may be lost by sharing data. For example, a vehicle making a right turn at an intersection may tend to nudge slightly right and slow down more than a vehicle going straight through. Ways to share data yet retain discriminability need to be explored as well as the tradeoffs between learning complexity, the amount of training data, and prediction capabilities.

F. Activity Analysis Extensions

The techniques described could be extended to a number of different situations. An appealing arena is multicamera networks that can monitor larger areas for better coverage and provide a more complete view of behavior (longer term activity analysis). The cameras would need to establish correspondences between each other and their relative position within the network [51], [88].

Trajectory data need not be produced just from static cameras. Generalized trajectory data could be obtained from moving platforms such as cars through vehicle surround analysis [90] or from aerial video [90]. The ego-motion of the platform itself presents another trajectory in addition to the tracks of surrounding objects. This type of analysis would be quite popular in the intelligent transportation field, specifically for pedestrian safety [91].

The trajectories could also come from articulated objects such as human motion capture. Here activities would need to be defined with respect to groups of trajectories [92], [93].

G. Analysis Evaluation

The most critical step for POI/AP analysis is a clear definition of how to evaluate the differing systems. Currently there is little agreement for evaluation metrics, some report classification accuracy, cluster correctness, or abnormality detection rates. The PETS and CAVIAR databases provide widely used surveillance video but without labels of typical activities or abnormalities. New databases need to be constructed to provide an even field for accurate comparison.

It is noteworthy to consider the problem of masking behaviors or when an abnormal subject masquerades in typical form. This extremely important surveillance situation may require data more diverse than just trajectories to detect necessitating a parallel processing thread in the analysis framework (Fig. 2).

VII. Concluding Remarks

The pervasive use of cameras for a wide variety of general surveillance tasks necessitates systems as easy to setup as the cameras themselves. This paper presents a survey of trajectory-based activity analysis predicated on motion coherence. By observing and tracking objects over time, to generate trajectories, a probabilistic model of typical behavior can be established. A topographical map specifies points of interest connected by activity paths which describe the way objects move. The graph accurately describes observations because it is generated from data and not through hand definition, thus allowing activities to be analyzed in a principled manner. The model is used to focus the surveillance system attention to events of interest: activity classification, detection of abnormalities, activity prediction, and object interaction.

References

[1] P. Remagnino, S. A. Velastin, G. L. Foresti, and M. Trivedi, "Novel concepts and challenges for the next generation of video surveillance systems," *Mach. Vis. Appl.*, vol. 18, no. 3–4, pp. 135–137, Aug. 2007.

[2] M. Shah, "Understanding human behavior from motion imagery," *Mach. Vis. Appl.*, vol. 14, no. 4, pp. 210–214, Sep. 2003.

[3] H. Buxton, "Learning and understanding dynamic scene activity: A review," *Image Vis. Comput.*, vol. 21, pp. 125–136, Jan. 2003.

[4] Y.-K. Jung and Y.-S. Ho, "Traffic parameter extraction using video-based vehicle tracking," in *Proc. IEEE Int. Conf. Intell. Transport. Syst.*, Oct. 1999, pp. 764–769.

[5] S. Kamijo, H. Koo, X. Liu, K. Fujihira, and M. Sakauchi, "Development and evaluation of real-time video surveillance system on highway based on semantic hierarchy and decision surface," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2005, pp. 840–846.

[6] S. Messelodi, C. M. Modena, and M. Zanin, "A computer vision system for the detection and classification of vehicles at urban road intersections," *Pattern Anal. Appl.*, vol. 8, no. 1–2, pp. 17–31, Sep. 2005.

[7] T. N. Schoepflin and D. J. Dailey, "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation," *IEEE Trans. Intell. Transport. Syst.*, vol. 4, no. 2, pp. 90–98, Jun. 2003.

[8] L. Jiao, Y. Wu, G. Wu, E. Y. Chang, and Y.-F. Wang, "Anatomy of a multicamera video surveillance system," *ACM Multimedia Syst.*, vol. 210, no. 2, pp. 144–163, 2004.

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

[10] R. J. Morris and D. C. Hogg, "Statistical models of object interaction," *Int. J. Comput. Vis.*, vol. 37, no. 2, pp. 209–215, Jun. 2000.

[11] N. Robertson and I. Reid, "Behaviour understanding in video: A combined method," in *Proc. IEEE Int. Conf. Comput. Vision*, Beijing, China, Oct. 2005, pp. 808–815.

[12] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi, "Traffic monitoring and accident detection at intersections," *IEEE Trans. Intell. Transport. Syst.*, vol. 1, no. 2, pp. 108–118, Jun. 2000.

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

[14] S. Atev, H. Arumugam, O. Masaoud, R. Janardan, and N. P. Papanikolopoulos, "A vision-based approach to collision prediction at traffic intersections," *IEEE Trans. Intell. Transport. Syst.*, vol. 6, no. 4, pp. 416–423, Dec. 2005.

[15] N. Saunier, T. Sayed, and C. Lim, "Probabilistic collision prediction for vision-based automated road safety analysis," in *Proc. IEEE Conf. Intell. Transport. Syst.*, Seattle, Washington, Sep. 2007, pp. 872–878.

[16] B. T. Morris and M. M. Trivedi, "Learning and classification of trajectories in dynamic scenes: A general framework for live video analysis," *IEEE Trans. Circuits Syst. Video Technol.*, submitted for publication.

[17] M. Brand and V. Kettnaker, "Discovery and segmentation of activities in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 844–851, Aug. 2000.

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

[19] A. Naftel and S. Khalid, "Motion trajectory learning in the DFT-coefficient feature space," in *Proc. IEEE Conf. Comput. Vision Syst.*, Jan. 2006, pp. 47–54.

[20] T. Xiang and S. Gong, "Video behaviour profiling and abnormality detection without manual labeling," in *Proc. IEEE Int. Conf. Comput. Vision*, Beijing, China, Oct. 2005, pp. 1238–1245.

[21] N. Brandle, D. Bauer, and S. Seer, "Track-based finding of stopping pedestrians—A practical approach for analyzing a public infrastructure," in *Proc. IEEE Conf. Intell. Transport. Syst.*, Toronto, ON, Canada, Sep. 2006, pp. 115–120.

[22] D. Makris and T. Ellis, "Path detection in video surveillance," *Image Vis. Comput.*, vol. 20, no. 12, pp. 895–903, Oct. 2002.

[23] D. Makris and T. Ellis, "Learning semantic scene models from observing activity in visual surveillance," *IEEE Trans. Syst., Man, Cybern. B*, vol. 35, no. 3, pp. 397–408, Jun. 2005.

[24] N. D. Bird, O. Masoud, N. P. Papanikolopoulos, and A. Isaacs, "Detection of loitering individuals in public transportation areas," *IEEE Trans. Intell. Transport. Syst.*, vol. 6, no. 2, pp. 167–177, Jun. 2005.

[25] I. N. Junejo, O. Javed, and M. Shah, "Multi feature path modeling for video surveillance," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2004, pp. 716–719.

[26] D. Biliotti, G. Anotonini, and J. P. Thiran, "Multi-layer hierarchical clustering of pedestrian trajectories for automatic counting of people in video sequences," in *Proc. IEEE Workshop Appl. Comput. Vision*, Brekenridge, CO, Jan. 2005, pp. 50–57.

[27] W. Hu, D. Xie, T. Tan, and S. Maybank, "Learning activity patterns using fuzzy self-organizing neural network," *IEEE Trans. Syst., Man, Cybern. B*, vol. 34, no. 3, pp. 1618–1626, Jun. 2004.

[28] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," in *Proc. British Conf. Mach. Vision*, Sep. 1995, vol. 2, pp. 583–592.

[29] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[30] O. Boiman and M. Irani, "Detecting irregularities in images and in video," in *Proc. IEEE Int. Conf. Comput. Vision*, Beijing, China, Oct. 2005, pp. 462–469.

[31] W. Yan and D. A. Forsyth, "Learning the behavior of users in a public space through video tracking," in *Proc. IEEE Workshop Appl. Comput. Vision*, Brekenridge, CO, Jan. 2005, pp. 370–377.

[32] N. Robertson, I. Reid, and M. Brady, "Behavior recognition and explanation for video surveillance," in *Proc. IET Conf. Crime Security*, London, U.K., Jun. 2006, pp. 458–463.

[33] N. Sumpter and A. J. Bulpitt, "Learning spatio-temporal patterns for predicting object behavior," *Image Vis. Comput.*, vol. 18, pp. 697–704, Jun. 2000.

[34] P. Remagnino, A. I. Shihab, and G. A. Jones, "Distributed intelligence for multi-camera visual surveillance," *Pattern Recognition*, vol. 34, no. 7, pp. 675–689, Apr. 2004.

[35] W. Hu, D. Xie, and T. Tan, "A hierarchical self-organizing approach for learning the patterns of motion trajectories," *IEEE Trans. Neural Netw.*, vol. 15, no. 1, pp. 135–144, Jan. 2004.

[36] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 9, pp. 1450–1464, Sep. 2006.

[37] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank, "Semantic-based surveillance video retrieval," *IEEE Trans. Image Process.*, vol. 16, no. 4, pp. 1168–1181, Apr. 2007.

[38] J. Melo, A. Naftel, A. Berardino, and J. Santos-Victor, "Detection and classification of highway lanes using vehicle motion trajectories," *IEEE Trans. Intell. Transport. Syst.*, vol. 7, no. 2, pp. 188–200, Jun. 2006.

[39] X. Li, W. Hu, and W. Hu, "A coarse-to-fine strategy for vehicle motion trajectory clustering," in *Proc. IEEE Conf. Pattern Recognit.*, 2006, pp. 591–594.

[40] X. Wang, X. Ma, and E. Grimson, "Unsupervised activity perception by hierarchical bayesian models," in *Proc. IEEE Comp. Sci. Conf. Comput. Vision Pattern Recognit.*, Jun. 2007, pp. 1–8.

[41] J.-W. Hsieh, S.-H. Yu, Y.-S. Chen, and W. Hu, "Automatic traffic surveillance system for vehicle tracking and classification," *IEEE Trans. Intell. Transport. Syst.*, vol. 7, no. 2, pp. 175–187, Jun. 2006.

[42] H. Veeraraghavan, O. Maoud, and N. Papanikolopulos, "Computer vision algorithms for intersection monitoring," *IEEE Trans. Intell. Transport. Syst.*, vol. 4, no. 2, pp. 78–89, Jun. 2003.

[43] P. Kumar, S. Ranganath, H. Weimin, and K. Sengupta, "Framework for real-time behavior interpretation from traffic video," *IEEE Trans. Intell. Transport. Syst.*, vol. 6, no. 1, pp. 43–53, Mar. 2005.

[44] S. Park and J. Aggarwal, "A hierarchical bayesian network for event recognition of human actions and interactions," *AMC J. Multimedia Syst.*, vol. 10, Special Issue on Video Surveillance, no. 2, pp. 164–179, Aug. 2004.

[45] S. Park and M. M. Trivedi, "Multi-person interaction and activity analysis: A synergistic track- and body-level analysis framework," *Mach. Vis. Appl.*, vol. 18, no. 3, pp. 151–161, Jan. 2007.

[46] S. Atev, O. Masoud, and R. J. N. Papanikolopoulos, "A collision prediction system for traffic intersections," in *Proc. IEEE Conf. Intell. Robots Syst.*, Aug. 2005, pp. 169–174.

[47] C.-Y. Chan, "Defining safety performance measures of driver-assistance systems for intersection left-turn conflicts," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2006, pp. 25–30.

[48] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM J. Comput. Surveys*, vol. 38, no. 4, Dec. 2006.

[49] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey of visual surveillance of object motion and behaviors," *IEEE Trans. Syst., Man, Cybern. C*, vol. 34, no. 3, pp. 334–352, Aug. 2004.

[50] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc.*, vol. 39, pp. 1–38, 1977.

[51] D. Makris, T. Ellis, and J. Black, "Bridging the gaps between cameras," in *Proc. IEEE Comp. Sci. Conf. Comput. Vision Pattern Recognit.*, Jun. 2004, vol. 2, pp. 205–210.

[52] H. M. Dee and D. C. Hogg, "On the feasibility of using a cognitive model to filter surveillance data," in *Proc. IEEE Int. Conf. Adv. Video and Signal Based Surveillance*, Sep. 2005, pp. 34–39.

[53] M. Bennewitz, W. Burgard, and G. Cielniak, "Utilizing learned motion patterns to robustly track persons," in *Proc. IEEE Int. Workshop Visual Surveillance Perform. Evaluation of Tracking and Surveillance*, 2003, pp. 102–109.

[54] C. Stauffer, "Estimating tracking sources and sinks," in *Proc. IEEE Workshop on Event Mining*, Jul. 2003, pp. 35–42.

[55] B. T. Morris and M. M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video," *IEEE Trans. Intell. Transport. Syst.*, to be published.

[56] H.-Y. M. Liao, D.-T. Chen, C.-W. Su, and H.-R. Tyan, "Real-time event detection and its application to surveillance systems," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 509–512.

[57] F. Bashir, W. Qu, A. Khokhar, and D. Schonfeld, "HMM-based motion recognition system using segmented pca," in *Proc. IEEE Comp. Sci. Conf. Comput. Vis. Pattern Recogni.*, Sep. 2005, vol. 3, pp. 1288–1291.

[58] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comp. Sci. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, pp. 246–252.

[59] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activities in video," in *Proc. IEEE Comp. Sci. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2004, vol. 2, pp. 819–826.

[60] J. Lin, M. Vlachos, E. Keogh, and D. Gunopulous, "Iterative incremental clustering of time series," in *Advances in Database Technol.*, 2004, vol. 2992, pp. 106–122.

[61] F. Porikli, "Learning object trajectory patterns by spectral clustering," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jun. 2004, vol. 2, pp. 1171–1174.

[62] S. Atev, O. Masoud, and N. Papanikolopoulos, "Learning traffic patterns at intersections by spectral clustering of motion trajectories," in *Proc. IEEE Conf. Intell. Robots Syst.*, Bejing, China, Oct. 2006, pp. 4851–4856.

[63] T. Jebara, R. Kondor, and A. Howard, "Probability product kernels," *J. Mach. Learning Res.*, vol. 5, pp. 819–844, Dec. 2004.

[64] B. Q. Huang, C. J. Du, Y. Zhang, and M.-T. Kechadi, "A hybrid HMM-SVM method for online handwriting symbol recognition," in *Proc. Int. Conf. Intell. Syst. Design Appl.*, Oct. 2006, pp. 887–891.

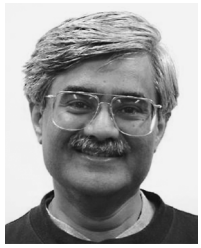
[65] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden markov models," *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1912–1919, Jul. 2007.

[66] M. M. Trivedi and J. C. Bezdek, "Low-level segmentation of aerial images with fuzzy clustering," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-16, no. 4, pp. 589–598, Jul. 1986.

[67] C. Piciarelli and G. L. Foresti, "On-line trajectory clustering for anomalous events detection," *Pattern Recognit. Lett.*, vol. 27, no. 15, pp. 1835–1842, Nov. 2006.
[68] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep. 2004.
[69] M. Pavan and M. Pelillo, "Dominant sets and hierarchical clustering," in *Proc. 9th IEEE Int. Conf. Comput. Vision*, Nice, France, Oct. 2003, pp. 362–369.
[70] M. Pavan and M. Pelillo, "Dominant sets and pairwise clustering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 29, no. 1, pp. 167–172, Jan. 2007.
[71] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 98, no. 9, pp. 1464–1480, Sep. 1990.
[72] T. Xiang and S. Gong, "Video behavior profiling and abnormality detection without manual labeling," in *Proc. IEEE Int. Conf. Comput. Vis.*, Beijing, China, 2005, pp. 1238–1245.
[73] T. W. Liao, "Clustering of time series data—A survey," *Pattern Recognit.*, vol. 38, no. 11, pp. 1857–1874, Nov. 2005.
[74] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[75] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proc. IEEE Conf. Data Eng.*, Feb. 2002, pp. 673–684.
[76] D. Buzan, S. Sclaroff, and G. Kollios, "Extraction and clustering of motion trajectories in video," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2004, pp. 521–524.
[77] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computi. Surveys*, vol. 31, no. 3, pp. 264–323, Sep. 1999.
[78] P. Berkhin, "A survey of clustering data mining techniques," *Grouping Mulidimensional Data*, pp. 25–71, Feb. 2006.
[79] S. Hongeng and R. Nevatia, "Large-scale event detection using semi-hidden markov models," in *Proc. 9th IEEE Int. Conf. Comput. Vision*, Nice, France, Oct. 2003, pp. 1455–1462.
[80] M. S. Ryoo and J. K. Aggarwal, "Hierarchical recognition of human activities interacting with objects," in *Proc. IEEE Comp. Sci. Conf. Comput. Vision Pattern Recognit.*, Jun. 2007, pp. 1–8.
[81] D. Greenhill, J. P. Renno, J. Orwell, and G. A. Jones, "Learning the semantic landscape: Embedding scene knowledge in object tracking," *Real Time Imaging*, vol. 11, no. 3, pp. 186–203, Jun. 2005.
[82] M. M. Trivedi, T. L. Gandhi, and K. S. Huang, "Distributed interactive video arrays for event capture and enhanced situational awareness," *IEEE Trans. Intell. Transport. Syst.*, vol. 20, no. 5, pp. 58–66, Sep. 2005.
[83] R. Khoshabeh, T. Gandhi, and M. M. Trivedi, "Multi-camera based traffic flow characterization and classification," in *Proc. IEEE Conf. Intell. Transport. Syst.*, Seattle, WA, Sep. 2007, pp. 259–264.
[84] A. H. S. Lai and N. H. C. Yung, "Vehicle-type identification through automated virtual loop assignment and block-based direction-biased motion estimation," *IEEE Trans. Intell. Transport. Syst.*, vol. 1, no. 2, pp. 86–97, Jun. 2000.
[85] 2006, Pedestrian and Bicyclist Intersection Saftey indexes Final Report Turner-Fairbank Highway Research Center [Online]. Available: http://www.tfhrc.gov/safety/pedbike/pubs/06125
[86] Z. Zhang, K. Huang, and T. Tan, "Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes," in *Proc. IEEE Conf. Pattern Recognit.*, 2006, pp. 1135–1138.
[87] A. K. Jain, A. Topchy, M. H. C. Law, and J. M. Buhmann, "Landscape of clustering algorithms," in *Proc. IEEE Conf. Pattern Recognit.*, Aug. 2004, pp. 260–263.
[88] O. Javed, Z. Rasheed, K. Shafique, and M. Shah, "Tracking across multiple cameras with disjoint views," in *Proc. 9th IEEE Int. Conf. Comput. Vision*, Nice, France, Oct. 2003, pp. 952–957.
[89] T. Gandhi and M. M. Trivedi, "Vehicle surround capture: Survey of techniques and a novel omni-video-based approach for dynamic panoramic surround maps," *IEEE Trans. Intell. Transport. Syst.*, vol. 7, no. 3, pp. 293–308, Sep. 2006.
[90] Y.-C. Chung and Z. He, "Low-complexity and reliable moving objects detection and tracking for aerial video surveillance with small uavs," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 2670–2673.
[91] T. Gandhi and M. M. Trivedi, "Pedestrian protection systems: Issues, survey, and challenges," *IEEE Trans. Intell. Transport. Syst.*, vol. 8, no. 3, pp. 413–430, Sep. 2007.
[92] J. Min and R. Katsuri, "Activity recognition based on multiple motion trajectories," in *Proc. Int. Conf. Pattern Recognit.*, Aug. 2004, pp. 199–202.
[93] J. Gao, R. T. Collins, A. G. Hauptmann, and H. D. Wactlar, "Articulated motion modeling for activity analysis," in *Proc. IEEE Comp. Sci. Conf. Comput. Vision Pattern Recognit. Workshops*, Jun. 2004, pp. 20–28.

Brendan Tran Morris received the B.S. degree in electrical engineering and computer science from the University of California, Berkeley, in 2002 and the M.S. degree in electrical and computer engineering from the University of California, San Diego, in 2006, where he is currently working toward the Ph.D. degree in intelligent systems, robotics and controls.

He is a member of the Computer Vision and Robotics Research Laboratory with research interests in intelligent surveillance systems and recognizing and understanding activities in video through machine learning.



Mohan Manubhai Trivedi received the B.E. degree (with honors) from the Birla Institute of Technology and Science, Pilani, India, and the Ph.D. degree from Utah State University, Logan.

He is a Professor of electrical and computer engineering and the Founding Director of the Computer Vision and Robotics Research Laboratory, University of California, San Diego. His team designed and deployed the "Eagle Eyes" system on the U.S.–Mexico border in 2006 as a part of Homeland Security project. He served on a panel dealing with the legal and technology issues of video surveillance organized by the Constitution Project in Washington, DC, as well as at the Computers, Freedom and Privacy Conference. He will serve as the General Chair for IEEE AVSS 2008 (Advanced Video and Sensor based Surveillance) Conference. He regularly serves as a consultant to industry and government agencies in the United States and abroad. He is serving as an Expert Panelist for the Strategic Highway Research Program (Safety) of the Transportation Research Board of the National Academy of Sciences.