

# Learning, Modeling, and Classification of Vehicle Track Patterns from Live Video

Brendan Tran Morris, *Student Member, IEEE*, and Mohan Manubhai Trivedi, *Senior Member, IEEE*

**Abstract**—This paper presents two different types of visual activity analysis modules based on vehicle tracking. The highway monitoring module accurately classifies vehicles into eight different types and collects traffic flow statistics by leveraging tracking information. These statistics are continuously accumulated to maintain daily highway models that are used to categorize traffic flow in real time. The path modeling block is a more general analysis tool that learns the normal motions encountered in a scene in an unsupervised fashion. The spatiotemporal motion characteristics of these motion paths are encoded by a hidden Markov model. With the path definitions, abnormal trajectories are detected and future intent is predicted. These modules add real-time situational awareness to highway monitoring for high-level activity and behavior analysis.

**Index Terms**—Anomaly detection, comparative flow analysis, highway efficiency, real-time tracking analysis, trajectory learning and prediction, vehicle type classification.

## I. INTRODUCTION

A KEY GOAL of situational awareness research is to understand the interactions and behaviors present in a scene. This scene awareness is particularly important for visual surveillance systems that must continually monitor a site. Large amounts of data are generated, making it infeasible for a human to accurately process. Activity analysis systems can be employed to filter out relevant data, focusing attention where it is needed most.

Highway traffic management is an important field requiring up-to-date data delivered in real time along with historical data on traffic conditions to design effective control strategies. In California, inductive loop sensors deliver counts (number of vehicles to cross a loop) and occupancy (average fraction of time a vehicle is over a loop) every 30 s from locations all over the state, providing a large data infrastructure. Unfortunately, only about 60% of California loop detectors supply usable data, and they are costly to maintain. Cameras offer an attractive substitute for loops since they can be unobtrusively deployed on the side of a highway and can also be used for other monitoring applications. In addition to providing traffic measurements equivalent to loop detectors, using video to track vehicles in

a scene reveals added information that is difficult to obtain using loop detectors such as origin–destination (OD) maps, travel time, and vehicle type classification. This added analysis provides a more complete highway picture than can currently be provided by loop detectors, allowing the construction of better traffic control strategies.

In addition to collecting traffic statistics, it would be advantageous to have a method of automatically extracting the expected highway behavior. This is particularly important when setting up a large camera network, where it is prohibitive to define every behavior, or when using pan–tilt–zoom (PTZ) cameras, where the view can drastically change, or for monitoring a variety of traffic scenes without tedious supervision. The underlying structure of roads constrains motion and can be leveraged to automatically build up behavior models through careful observation over time. By generating the models from the data, the learned behaviors better reflect what is actually occurring in a scene rather than what is expected. Furthermore, the models allow prediction and detection of unusual or abnormal events. Without *a priori* knowledge, activity analysis is possible in an arbitrary scene just through tracking of objects.

This paper presents two different traffic situational awareness systems. The first system is the visual VEHICLE Classifier and Traffic fLOW analyzeR (VECTOR) [1] module for robust real-time vehicle classification, traffic statistic accumulation, and highway modeling for flow analysis. The second activity analysis module introduced is the path behavior block, which builds a probabilistic scene motion model in an unsupervised manner for activity analysis. This process automatically defines the traffic lanes without manual specification and is used to detect anomalous trajectories and unusual actions, as well as generate long-term path prediction. The efficacy of these behavior modules are demonstrated through analysis of simulated and real-world data.

## II. RELATED RESEARCH

### A. Highway Analysis

Highway analysis requires robust detection of vehicles and tracking. With these two basic tasks, a number of other calculations can be performed, such as vehicle classification, extraction of traffic flow parameters, congestion detection, or a number of other measurements that are useful for traffic management. A major research effort is to build large-scale systems that are able to effectively cover miles of road [2]. Key hurdles associated with this system realization are adaptation to a wide variety of changing environmental conditions [3],

Manuscript received June 14, 2007; revised October 16, 2007 and December 24, 2007. This work was supported in part by the Technical Support Working Group and in part by the University of California Discovery Grant. The Associate Editor for this paper was N. Papanikolopoulos.

The authors are with the Computer Vision and Robotics Research Laboratory, University of California, San Diego, La Jolla, 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/TITS.2008.922970

construction of multiple nodes with cooperating sensors [4], fusion of information between sensor nodes to maintain vehicle identity [5], and development of a database architecture for storage and intelligent retrieval of large volumes of traffic data for event analysis [6].

1) *Vehicle Classification*: Once potential objects have been identified, each foreground region must be classified into a specific object type. An early work used a single dispersedness measure to separate vehicles and humans from other objects by filling a class histogram while tracking [7]. Vehicles in a parking lot have been classified into six general types by linear discriminant analysis (LDA) of blob measurements with improvements gained from tracking-based confidence [8]. A multiclass kernel support vector machine was constructed using vehicle images as input to the nonlinear hierarchical classifier [9]. The classification work in this paper most closely resembles the work in [8] but with real-time classification using a single classifier designed for the entire view.

2) *Track Analysis*: In addition to locating objects, tracking can be used for better scene understanding. When using *a priori* scene knowledge, one must define events of interest. Vehicle counts can be accumulated with the placement of virtual loop detectors in video [10]. Highway congestion warnings have been issued by classifying object motion based on tracked velocities [11]. Even complex parking lot motion behaviors have been defined in a hierarchical structure based on acceleration and velocity tracking profiles [12].

3) *Traffic Analysis Systems*: There are few systems that combine tracking, object classification, traffic parameter extraction, and event detection. Gupte *et al.* developed a vehicle detection and two-type classification system by robustly tracking vehicles after camera calibration [13]. The VISTRAM system [14] classified vehicles into a small set of size-based classes and generated traffic parameters without explicit tracking, but the system did not include any type of event recognition. Kumar *et al.* [15] developed a parking lot monitoring system that tracked objects and classified them into six types using a known Bayesian network. The vehicle behavior at checkpoints was evaluated based on a vocabulary of actions, allowing the detection of abnormal use such as loitering. A zone of influence was defined to represent potentially dangerous interactions between objects. SCOCA [16] is an intersection monitoring system that tracks and performs 3-D model-based classification of objects. The speed of each vehicle is recorded along with its OD information. VECTOR is unique because it has been operating in real time for over a year, robustly conducting a number of analyses.

### B. Path Learning

The second type of track analysis avoids using any scene knowledge and builds event models based on accumulated tracking data. This allows more flexible deployment because the models are learned from the observed data themselves and not defined by a user. Pioneering work by Johnson and Hogg [17] described outdoor motions with a flow vector  $f = [x, y, dx, dy]$  and learned paths using a neural network (NN). Owens and Hunter [18] extended this work using a self-

organizing feature map to learn paths and further detect abnormal behavior after training in a surveillance mode. Stauffer and Grimson [19] learned paths in a hierarchical fashion by building up a co-occurrence of codebook flows. Hu *et al.* [20] sped up the path learning process with a batch NN and presented a method to predict behavior. This work was statistically extended by hierarchically clustering trajectories, first using spatial information and then using temporal information. The paths were modeled by a chain of Gaussian distributions for Bayesian inferencing of anomalies and behavior prediction [21]. Makris and Ellis [22] introduced an online learning method that was able to build paths as new tracks were accumulated rather than learned from a training set. Their scene model defined a Bayesian belief network, where paths were defined not only between the beginning and end of a track but also between places where objects tended to stop. All these path learning techniques leverage ordered and repetitive structure observed in video monitoring.

## III. HIGHWAY ACTIVITY ANALYSIS FRAMEWORK

The general situational analysis system considered is designed to be quite simple and modular. Section IV describes the front-end system consisting of object detection and tracking. Moving objects are detected and tracked, providing the only input data necessary for activity analysis. The two different activity analysis modules are task specific and designed for behaviors of interest.

The VECTOR module described in Section V uses object measurements obtained during tracking to classify the vehicle type, and highway usage statistics are accumulated to build historical models, allowing real-time analysis of traffic flow and speed profiling.

The path behavior block presented in Section VI uses accumulated tracking data to automatically discover and map out the major motion paths in a scene that indicate normal activity patterns. This helps mark in which lane a vehicle is traveling and can signal an alert when a vehicle produces an unusual trajectory; this is also used for real-time indications of anomalous actions and for longer term intent prediction.

## IV. FRONT-END PROCESSING

### A. Object Detection

Foreground pixels belonging to moving objects are quickly determined by using an adaptive background subtraction scheme. Each background pixel is modeled as a single Gaussian process with mean  $\mu$  (time-averaged intensity) and  $\sigma$  (standard deviation of intensity). The Gaussian parameters adapted are

$$\mu_t = (1 - \alpha)\mu_{t-1} + \alpha I_t \quad (1)$$

$$\sigma_t^2 = (1 - \beta)\sigma_{t-1}^2 + \beta(I_t - \mu_t)^2 \quad (2)$$

where  $\alpha, \beta \in [0, 1]$  control how quickly the parameters are updated as each new video frame is received. The foreground is

found by thresholding the background difference image based on past pixel deviations, i.e.,

$$I_{\text{foreground}} = (I_t - \mu_t) > B(\sigma_t + \sigma_0). \quad (3)$$

Here,  $\sigma_0$  is a small constant to suppress noise associated with low-variance scenes typically encountered during low-light and shadowed situations, and  $B$  is the deviation threshold. The threshold is defined for each pixel by its neighborhood  $N$  intensity, i.e.,

$$B_{ij} = \min \left\{ \left( \frac{B_{\max} - B_{\min}}{I_m + I_\sigma} \right) I_N + B_{\min}, B_{\max} \right\}. \quad (4)$$

Equation (4) sets up a threshold that adapts to the local lighting intensity  $I_N$  by a comparison with the mean image intensity  $I_m$  and the standard deviation  $I_\sigma$ . The values of  $B_{\max}$  and  $B_{\min}$  indicate the maximal and minimal deviations that are necessary for detection.

The foreground is further processed to fill in any holes using morphological operations. Each blob is then labeled by connected component analysis, and simple morphological measurements are taken, i.e.,  $m^t = \{\text{area, breadth, compactness, elongation, perimeter, convex hull perimeter, bounding box, best fit ellipse parameters, roughness, centroid}, M_{10}, M_{01}, M_{20}, M_{02}\}$ , as a compact representation of an object's appearance that is suitable for classification.

### B. Tracking

Tracking associates every detected vehicle to an existing track through nearest global matching. To match and update a track, a detection must fit a dynamics model and an appearance constraint.

1) *Kalman Filter*: Vehicle dynamics are modeled by applying a Kalman filter to an object's centroid  $c$ . The tracking state update equation is given by

$$s_{t+1} = As_t + w_t = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ v_x \\ v_y \end{bmatrix} + w_t \quad (5)$$

for velocity  $v$  and time between frames  $\Delta t$ . The tracking state measurement is estimated by a detection's centroid  $\hat{c}$  and velocity  $\hat{v}^t = [\hat{c}_x^t - c_x^{t-1}, \hat{c}_y^t - c_y^{t-1}]^T$ . When a detected object is matched to a track, the measurement  $y_t = [\hat{c}, \hat{v}]^T$  is used to update the Kalman filter. New tracks are initialized by instantiating a Kalman filter with a velocity obtained by a nearest-neighbor match, i.e.,  $\hat{v}^1 = [\hat{c}_x^1 - c_x^0, \hat{c}_y^1 - c_y^0]^T$ . The Kalman filter state prediction is used to match with detections in the next frame.

2) *Track Appearance*: In addition to the dynamic model, a track has an associated appearance model. This appearance model is used to resolve matching ambiguities [23]. These ambiguities mainly arise because of high object density or occlusion. This model guarantees, by enforcing consistency between frames, that the appearance of a vehicle along a trajectory

does not drastically change. The similarity of a detected vehicle  $m_d$  and a track  $m_T$  is given by

$$S_{\text{meas}} = [(m_T - m_d)^T \Sigma^{-1} (m_T - m_d)]^{-1} > T_S \quad (6)$$

where  $\Sigma$  is a diagonal matrix with entries equal to the measurement variance learned during training, and  $T_S$  is a threshold. In a crowded scene, the best match is that which is most similar to the track. This constraint implicitly manages occlusion when objects either merge or split by instantiating a new track. Others have explicitly dealt with occlusion by applying heuristic rules [24] or using spatiotemporal cues [11] to repair occluded tracks. A track can then be updated when there is a consistently matched detection, i.e.,

$$m_T^t = \begin{bmatrix} \eta_0 \\ \vdots \\ \eta_{15} \end{bmatrix} = (1 - \gamma)m_d^t + \gamma m_T^{t-1} \quad (7)$$

given the track and detection measurements at the current time  $t$ . Similar to above,  $\gamma \in [0, 1]$  controls how quickly the object's appearance may change during tracking.

## V. VECTOR

The VECTOR module was designed for highway traffic analysis. In addition to loop detector measurements, VECTOR determines the types of vehicles on the road. Accumulated traffic statistics are used to build a traffic model that is useful for online traffic flow analysis, such as detection of dangerous behavior.

### A. Classification

VECTOR classifies vehicles into eight different types, namely, Sedan, Pickup, SUV, Van, Merged, Bike, Truck, and Semi, as shown in Fig. 1. These vehicles were selected because they were the most often occurring vehicle types from the 2001 National Household Travel Survey conducted by the U.S. Department of Transportation [25]. Although there is no explicit occlusion reasoning during tracking, a merged vehicle class was included to detect occlusions. The block diagram depicting the VECTOR classification scheme is shown in Fig. 2. After object detection, the extracted blob features (7) are LDA transformed and given a vehicle type label  $w_c$  using a weighted  $k$ -nearest neighbor (wkNN) classifier. The frame labels are compiled and incorporated into an improved vehicle track label  $L_T$ .

1) *Training Database Construction*: The nearest-neighbor training set is populated by hand-labeled examples from each of the eight vehicle types  $C$ . Each training vehicle is transformed using Fisher's LDA [26] for better separation, i.e.,  $x_i = W_{\text{LDA}} m_i$ , and placed into its corresponding class set  $D_c$ . The complete training set  $D = \bigcup_{c=1}^C D_c$  is then clustered using fuzzy  $C$ -means (FCM) [27]. The clustering procedure normalizes the number of training examples in  $D_c$  for each class, limiting memory requirements and the number of neighbor comparisons [23].

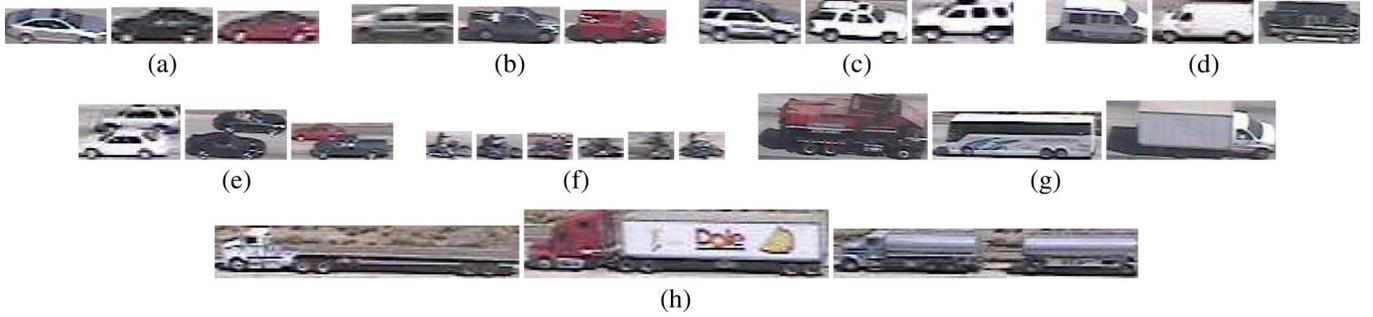


Fig. 1. Sample images from each vehicle class. (a) Sedan. (b) Pickup. (c) SUV. (d) Van. (e) Merged. (f) Bike. (g) Truck. (h) Semi.

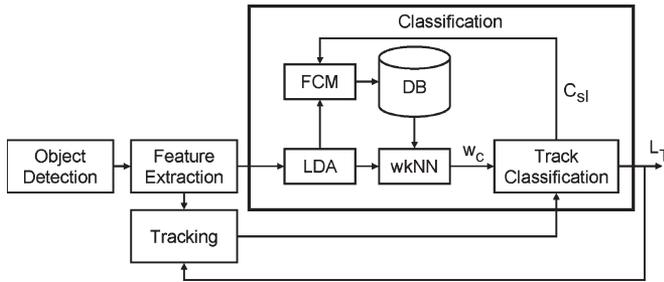


Fig. 2. Block diagram for the tracking-refinement-based wkNN classification scheme.

2) *Frame Classification*: During live tracking, each potential vehicle  $m_d$  is given a soft class membership  $w_c$  for every frame by wkNN comparison [28], i.e.,

$$w_c = \sum_{\substack{i=1 \\ m_i \in D_c}}^K \frac{1}{\|m_i - m_d\|}. \quad (8)$$

This measures the strength of match between a detection and all of the vehicle classes. Match strength is the sum of the similarity (inverse distance) between the  $k$  best matches from each class. While any particular class may have a few closely matching training examples, it is unlikely that there will be a large number of close matches unless it is the true class label. The advantage of the wkNN classification scheme is a soft assignment to every class, for robustness to noise and outliers, rather than the binary indicator usually obtained with NN classifiers.

3) *Tracking Refinement*: Through tracking, an appearance record of a vehicle is accumulated, giving  $T$  object examples over the life of a track. Given these  $T$  samples, a track label is generated by maximum-likelihood estimation. Thus

$$\begin{aligned} L_T &= \arg \max_c \sum_{t=1}^T \ln p(m^t|c) \\ &= \arg \max_c \sum_{t=1}^T \ln \frac{w_c^t}{\sum_c w_c^t}. \end{aligned} \quad (9)$$

The likelihood  $p(x_t|c)$  of class  $c$  is approximated by normalizing the per-frame class weight (8) for each sample  $t$  in a

track. The track class  $L_T$  is refined with each frame, as the track is updated and more information is compiled. The track label estimation leverages the added evidence gathered through tracking to make a decision, rather than just a single-frame measurement that could potentially be corrupted by noise.

4) *Confidence*: The confidence in classification label  $L_T$  can be measured by the sidelobe ratio

$$C_{sl} = \frac{p_1 - p_2}{p_1} \quad (10)$$

where  $p_1$  and  $p_2$  correspond to the probabilities of the first and second best matching vehicle types. The sidelobe ratio gives a measure of how much stronger the class  $L_T$  is than the closest competing class. Highly confident tracks can be used to reinforce the training database, whereas low-confidence samples are rejected.

## B. Traffic Statistic Modeling

Using the system front end, data are collected, and a highway model indicating normal traffic patterns is constructed. The highway model is a time series of fundamental highway usage parameters that are analogous to those obtained from conventional loop detectors. This system delivers flow (number of vehicles/time), density (number of vehicles/distance), and speed (in miles per hour) estimates in 30-s intervals, averaged over a 5-min window. The flow statistic is generated by counting the number of passing vehicles in the 30-s update interval. Density is the average number of vehicles in the camera view normalized by the roadway length. Speed is the average velocity estimate of vehicles. The density and speed calculations are obtained through manual roadway calibration. The lanes were marked and their length measured by tracking a vehicle of known dimension. Fig. 3 shows the southbound lane statistics for U.S. Interstate 5 (I5).

In addition to reproducing loop detector data, video provides a means to extract richer contextual information. Traffic parameters are compiled for each type of vehicle based on the vehicle classifier. Fig. 4(a) plots the flow and Fig. 4(b) speed of different vehicle types on a weekday. These data are useful to understand the different effects of commercial or private vehicles on highway control and to study the environmental impact of emissions. In Fig. 4(a), there are clearly many more sedans on the road than any other class of vehicles, but during

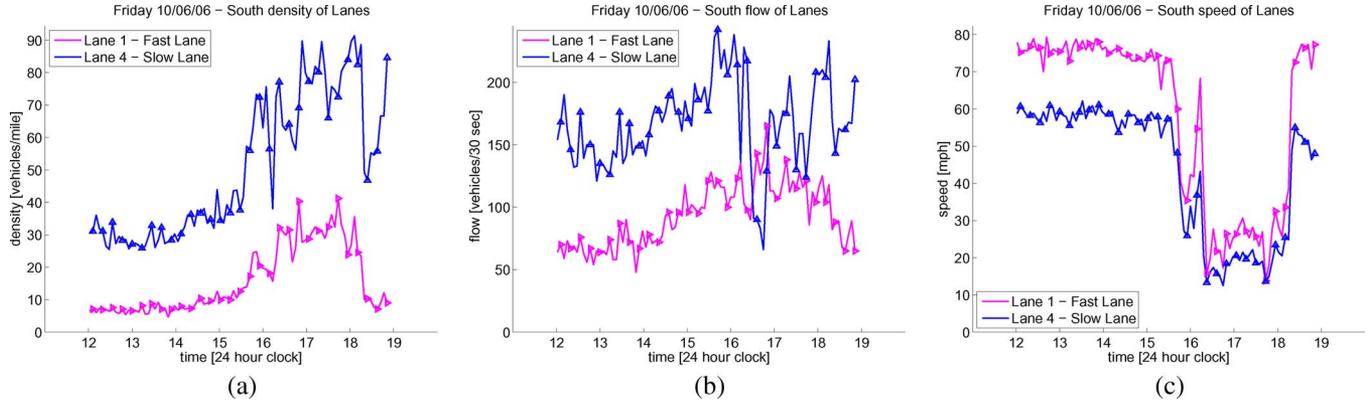


Fig. 3. Individual lane density, flow, and speed for the southbound direction of I5. (a) Lane density. (b) Lane flow. (c) Lane speed.

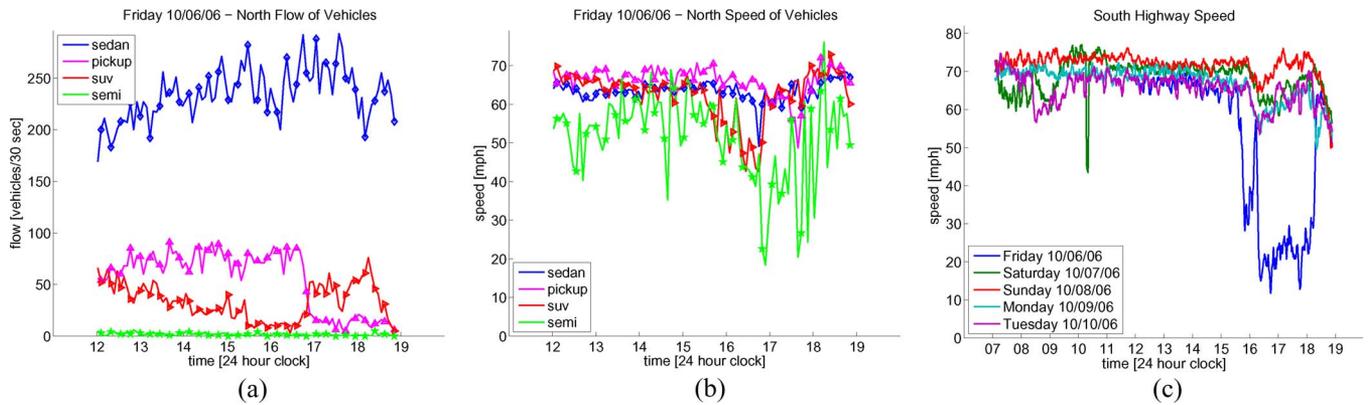


Fig. 4. Traffic statistics separated by vehicle type indicating (a) flow and (b) speed. Semi trucks are rare and travel at noticeable lower speeds. (c) Highway speed profile comparing weekend and weekdays.

the evening commute, the number of pickups and SUVs on the road appear to switch; during the day, there are more pickups, and during rush hour, there are more SUVs. In Fig. 4(b), it is noted that most of the vehicles travel at approximately the same speed (the speed of traffic), but the larger Semi trucks tend to travel slower than passenger vehicles, matching our intuition.

The large amounts of data collected by this system allow usage analysis, not only over the course of a single day but for many days. To build a useful highway model, it is important to incorporate the differences in traffic behavior as a function of time. Fig. 4(c) demonstrates the differences in the speed profile for work and nonwork days. The Friday congestion slowdown between 16:00 and 19:00 is significantly greater than the other weekdays. While the Monday and Tuesday commute is noticeable, it is a more subtle speed disturbance. This demonstrates the need to individually model each day. Seven models are generated by averaging across each specific day.

### C. Flow Analysis

By collecting traffic measurements, models for the expected highway behavior are generated. These models adapt to changing conditions over time and allow for online highway analysis,

such as link efficiency and characterization of the driver's speed.

1) *Highway Efficiency*: Chen *et al.* [29] used flow and speed to show that congestion is not caused by demand exceeding capacity but of inefficient operation of highways during periods of peak demand. Using the accumulated usage statistics, the highway efficiency, at a given time  $t$ , can be estimated by taking into account the changes in flow, i.e.,

$$\hat{\eta}(t) = \frac{\text{flow}(t) \times \text{speed}(t)}{\text{flow}_{\max} \times \text{speed}_{\max}}. \quad (11)$$

Fig. 5 shows the lane efficiency of the north- and southbound directions of the highway. Congestion is evident during the evening commute, as shown by the significant drop in efficiency in Fig. 5(a). It is interesting to note that while the efficiency of the southbound direction drops because of congestion, the northbound highway does not suffer from congestion. The reduced efficiency in the fast lane is actually due to underutilization because the northbound flow is much lower than its average.

2) *Speed Profile*: By using a database of historical speed measurements, a model of daily highway speed patterns can be constructed to incorporate the traffic speed fluctuations over the course of a week [Fig. 4(c)]. The VECTOR system

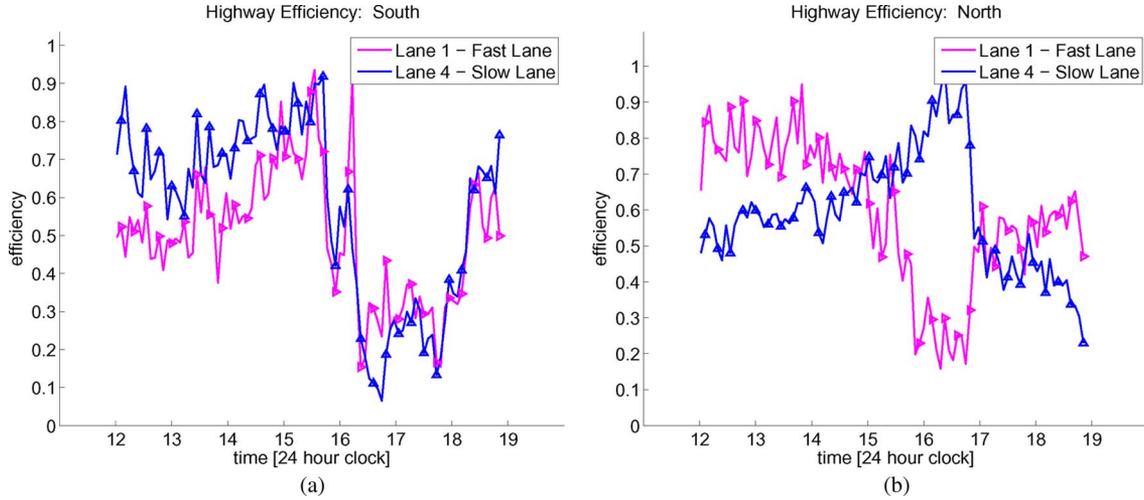


Fig. 5. Highway lane efficiency. (a) South: Low efficiency during evening commute because of congestion. (b) North: Efficiency drop in the fast lane because low flow caused underutilization.



Fig. 6. User-defined speed profiling. (a) Car slowing (yellow box) before stopping on the shoulder. (b) Car coming to rest (red box) on the shoulder of the highway. (c) Friday evening high volume traffic showing merging of vehicles. The north- and southbound directions have different speed profiles because of southbound congestion.

indicates the motion state of each vehicle by the color of its bounding box: {speeding, normal, slow, stopped} = {blue, green, yellow, red}. In Fig. 6(a), a sedan is shown slowing down on the shoulder of the highway before coming to a complete stop in Fig. 6(b). The motion state is defined as

$$S_V(v) = \begin{cases} \text{Stopped,} & 0 \leq v < 0.15V_{\text{avg}}^t \\ \text{Slow,} & 0.15V_{\text{avg}}^t \leq v < 0.6V_{\text{avg}}^t \\ \text{Normal,} & 0.6V_{\text{avg}}^t \leq v < 1.1V_{\text{avg}}^t \\ \text{Speeding,} & 1.1V_{\text{avg}}^t \leq v \end{cases} \quad (12)$$

where  $V_{\text{avg}}^t$  is the average speed at time  $t$ , and  $v$  is the estimated vehicle speed. The speed model currently considers normal speed as the daily average. Fig. 6(c) demonstrates the speed profile during a Friday evening commute. Notice that the northbound direction only contains freely moving vehicles, whereas there are slowly moving vehicles (red and yellow bounding boxes) in the southbound lanes. Congestion causes  $V_{\text{avg}}(\text{South}) \approx 25$  mi/h, whereas the northbound direction retains a faster flow, i.e.,  $V_{\text{avg}}(\text{North}) \approx 70$  mi/h. The speed state can be used as an indicator of dangerous situations because it locates abnormal patterns based on historical data.

## VI. PATH BEHAVIOR ANALYSIS

The path behavior (PB) block is a more general surveillance tool than VECTOR. It might be *a priori* unknown what is expected when monitoring a new scene with an arbitrary camera configuration, but the PB system can learn to look for what is important. This module learns normal motion patterns corresponding to lanes in the road by observing and collecting tracking data, which allows for the detection of anomalous actions and long-term path intent prediction.

A training set of trajectories is acquired by collecting tracking data for a period of time. The set is then clustered to find the major scene spatial routes, which are probabilistically modeled by hidden Markov models (HMMs) and used for activity analysis.

### A. Automatic Path Discovery

By running the tracking software and collecting the trajectories, the system can automatically learn the motion configuration of a location. Object motions map out spatial patterns that are often not random but are drawn from some underlying distribution. The inherent structure and redundancy that are prevalent in a scene can be leveraged to extract typical motion

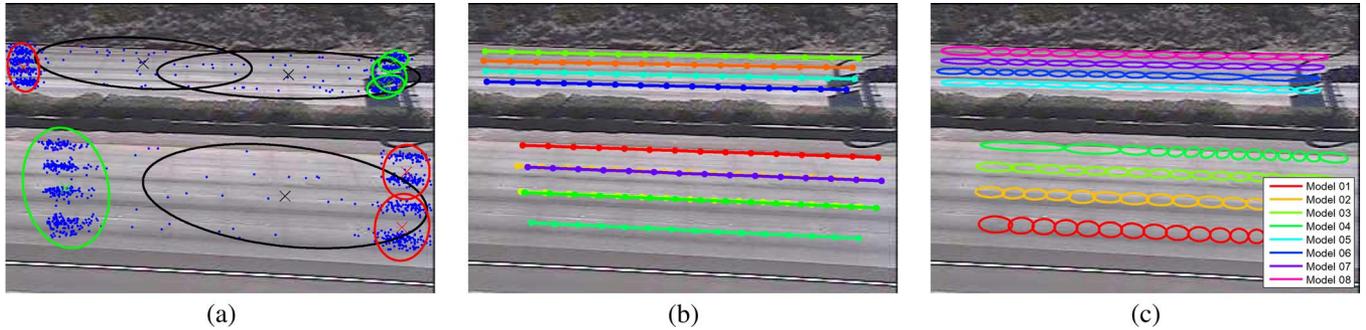


Fig. 7. California I5 automatic path modeling. (a) Entry/exit (green/red) zones with tracking noise (black). (b) Tracking data clustered into routes by FCM and merged into lanes. (c) HMM lane models: Ellipses display just the position portion of the HMM states to two standard covariance deviations.

paths. The paths encode the intent of objects and can be used for better (earlier) behavior prediction than the traditional one-step variety.

The spatial paths are learned by creating a database  $D = \{T_k\}$  of trajectories  $T_k = \{s^t\}_{t=1}^T$  with  $s^t = [x^t, y^t, v_x^t, v_y^t]^T$ . The database is preprocessed to remove incomplete tracks and normalized by resampling before clustering into routes using FCM. Using this approach, the most relevant motions in a video scene can be characterized without *a priori* knowledge or supervision.

1) *Entry/Exit Zones*: Before learning the paths, the tracks are filtered to remove tracking noise. The tracking noise is generated by incorrect tracking due to occlusion; tracks are formed or completed when objects overlap. These irregularities manifest as false start or stop positions distributed throughout the scene. The image entry/exit zones correspond to the true locations where object tracking either begins or ends. Each zone is separately modeled as a mixture of Gaussians and learned using expectation–maximization (EM) for Gaussian mixture models [30]. The zone is overmixed such that the tight components represent the true zone and the wide components represent the tracking noise [22]. Only tracks that begin in an entry zone and end in an exit zone are retained for further processing. All the other filtered tracks are considered tracking failures and are removed from the training set to prevent corruption during path clustering. Fig. 7(a) shows the entry/exit zones learned using 806 training tracks. The black ellipses correspond to the wide noise mixture components. After filtering the training database, 678 tracks remained in the training set.

2) *Track Resampling and Clustering*: After filtering the tracking noise from the training database, the major scene routes can be automatically learned by clustering. Unfortunately, standard clustering algorithms cannot be directly applied to the trajectory data because they are of unequal length. The trajectory length is dependent on the amount of time spent in the camera field of view, which varies from vehicle to vehicle. A trajectory vector suitable for clustering is obtained by linearly resampling each trajectory to a fixed length  $L$ , i.e.,  $T_k = \{s^t\}_{t=1}^L$ . A flow vector [17]  $f = [x_1, y_1, \dots, x_L, y_L]^T$  is constructed by ignoring velocity measures for each track (the dynamics will be handled later). The training set of flow vectors is used as input for FCM clustering into  $N_c$  prototype routes  $\{r_1, \dots, r_{N_c}\}$ , where each route prototype  $r_i$  only corresponds to the  $xy$  location in space.

3) *Path Merging*: The true number of paths  $N_p$  in an arbitrary scene is not *a priori* known and must be estimated. Initially, we cluster with FCM using a large number of paths, i.e.,  $N_c > N_p$ , and then refine the cluster number to the true lane number by merging similar paths. Paths are considered similar if all consecutive points are within a small radius, i.e.,

$$d^t = \sqrt{(r_i^t - r_j^t)^T (r_i^t - r_j^t)} < \epsilon_d \quad \forall t \quad (13)$$

or if the total distance between tracks is small enough, i.e.,

$$D = \sum_{t=1}^L d^t < \epsilon_D = L\epsilon_d. \quad (14)$$

The threshold should be chosen small enough to ensure that adjacent lanes are not merged, which is related to perspective projection foreshortening. Component cluster points are considered close enough for merging in the I5 scene when  $\epsilon_d = 5$  pixels because the northbound lanes are quite close. In practice, FCM tends not to overfit the data but instead finds several very similar clusters, making this simple merge algorithm effective. After initial FCM clustering, the lower southbound lanes have a number of overlapping paths. Fig. 7(b) shows the lanes after cluster merging.

## B. Probabilistic Path Modeling

The FCM procedure spatially locates paths, but this is insufficient for behavior analysis. We need to know not only where objects are located but also the manner in which they travel along a given path. These dynamics are needed to completely characterize a behavior. Using HMMs, the spatiotemporal properties of every path is encoded, differentiating not only the location but also the dynamics.

The advantage of modeling paths by HMMs is the simplicity of training and evaluation. HMMs define a natural procedure to compare different length tracks, as will generally occur, through optimal time normalization. Unlike the FCM clustering, the full unsampled state trajectories containing position and velocity are used to incorporate dynamics.

1) *HMMs*: Each path is compactly represented as  $\lambda_i = (A, B, \pi)$  and is designed to have  $Q$  states. The parameters  $A$

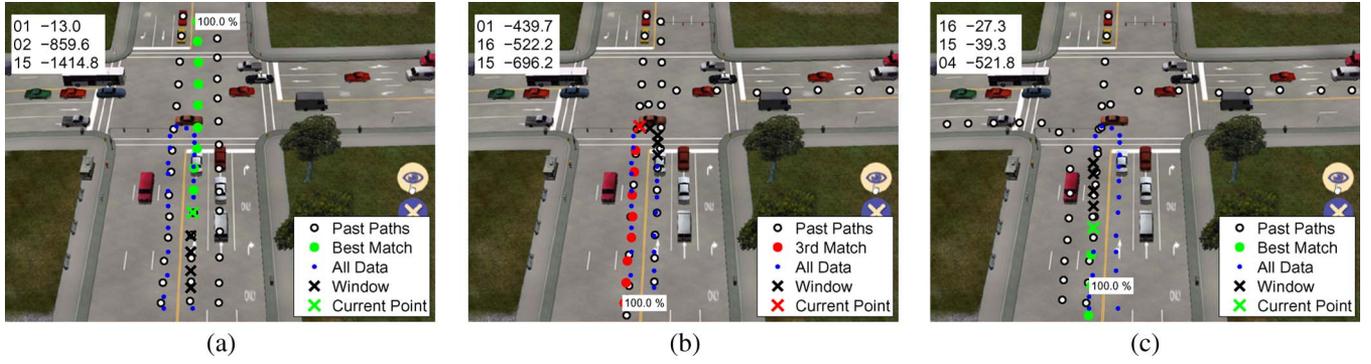


Fig. 8. Unusual tracking behavior during an illegal U-turn. (a) Acceptably (green X) traveling north in Path 1. (b) Suspicious event marked with a red X during the U-turn (still in Path 1). (c) Recovered into acceptable southbound lane (Path 16).

and  $\pi$  can be manually defined by the inherent structure of paths

$$\pi_i = e^{\alpha_p i} \quad (15)$$

$$A_{ij} = \begin{cases} e^{-\alpha_b(I-j+1)}, & j \leq i \\ e^{-\alpha_f(j-I-1)}, & j > i. \end{cases} \quad (16)$$

The rows of  $\pi_i$  and  $A$  are normalized to be valid probabilities. The transition rates are chosen such that  $\alpha_f \ll \alpha_b$  for strong left-right tendencies. The only unknown is the observation distribution  $B$ , which is dependent on the model states  $\{q_j\}_{j=1}^Q$ . The states are assumed to be Gaussian with unknown mean and covariance, i.e.,  $q_j \sim N(\mu_j, \Sigma_j)$ . Each HMM is completely specified after learning the states. The observations used to train the HMM path models incorporate the position and velocity  $O = [x, y, v_x, v_y]^T$ .

An HMM is trained for each route by dividing the training set into  $N_p$  disjoint sets, i.e.,  $D = \bigcup_{i=1}^{N_p} D_i$ . The set  $D_i$  is constructed by collecting all trajectories classified into route  $r_i$ . Only those tracks that fit route  $r_i$  well are retained for training (those with high membership). The  $Q$  states from each of the  $N_p$  HMMs can be efficiently learned using standard methods such as the Baum-Welch method or EM [31]. The set of HMMs learned for a real highway scene is shown in Fig. 7(c).

### C. Behavior Analysis

To analyze the object's behavior, it is necessary to place it into a corresponding path at all times. When objects do not fit into a path model well, it indicates an anomalous event that should be detected. This detection is made more difficult in an online setting because only a portion of the entire track is seen at a given time. Behavior inferencing must be done with incomplete data and must still detect all truly unusual occurrences.

1) *Anomalous Trajectories*: Each trajectory can be placed into a path by a comparison with the set of all HMMs. The path that best explains the test trajectory  $T_n$  (highest likelihood) is the assigned path

$$\lambda^* = \arg \max_i P(T_n | \lambda_i). \quad (17)$$

While every track will be classified into a path  $\lambda^*$ , the quality of this assignment may be low for abnormal trajectories. These

abnormal trajectories can be recognized as those with a likelihood less than a threshold, i.e.,

$$\text{LLT}_p = \beta(\text{LL}_{\text{in}} - \text{LL}_{\text{out}}) + \text{LL}_{\text{out}}. \quad (18)$$

The decision threshold is learned during training by comparing the average likelihood of samples in the training set  $D_i$  to all those outside ( $T_k \in D_j$ , where  $j \neq i$ ), i.e.,

$$\text{LL}_{\text{in}} = \frac{1}{|D_i|} \sum_{k \in D_i} \log P(T_k | \lambda_i) \quad (19)$$

$$\text{LL}_{\text{out}} = \frac{1}{N_T - |D_i|} \sum_{k \notin D_i} \log P(T_k | \lambda_i). \quad (20)$$

The threshold  $\text{LLT}_p$  is linearly related to the log likelihoods, where the sensitivity factor  $\beta \in [0, 1]$  controls the abnormality rate. Larger  $\beta$  values will cause more trajectories to be considered anomalous by increasing the threshold.

2) *Online Trajectory Analysis*: Although it is interesting to analyze complete tracks, it is often more important to recognize and evaluate behavior as it occurs. A new path estimate can be made each time a track is updated and as more information is gathered, the path estimate can be refined, as is done with vehicle classification. Instead of using all the tracking data accumulated up to time  $t$ , only a small window of points is retained. The windowed track consists of past and present measurements, i.e.,  $T_{\text{win}} = \{s_{t-\text{win}}, \dots, s_{t-1}, s_t\}$ . The trajectory is windowed to only consider the recent history as very old samples may not correlate well with the current behavior.

Using the windowed track, the path an object is following at the current time  $t$  can be determined by evaluating (17), with  $T_n$  replaced by its windowed version  $T_{\text{win}}$ . Over the life of a trajectory, path estimates are made, encoding a complete path history  $\{\lambda_1^*, \dots, \lambda_T^*\}$ . A number of consecutive labels indicate a consistent path. Using this transcription allows recognition of lane changes as points when the label changes between consistent paths. The track in Fig. 8 starts in Path 1 and goes through a lane change while making a U-turn before it ends in Path 16.

3) *Unusual Actions*: Similar to abnormal trajectories, unusual events can be detected during tracking. Since only a windowed version of a track is used during tracking, the

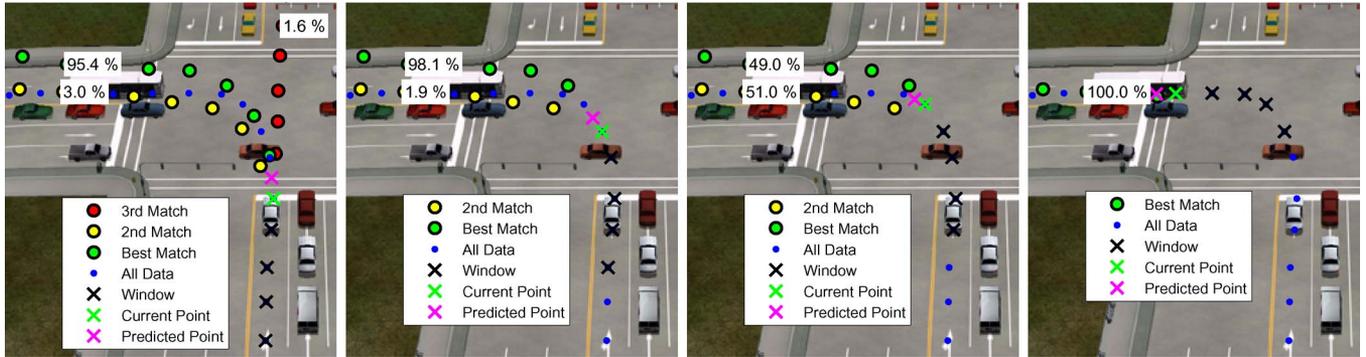


Fig. 9. Left-turn prediction behaves as expected. As more data points are collected, the prediction better matches the true lane.

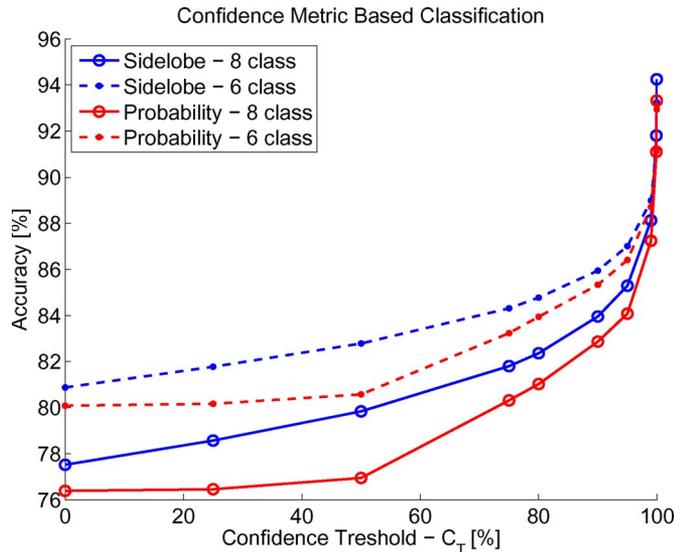


Fig. 10. Classification accuracy using different numbers of classes. The six-class test combines SUV + Van and Semi + Truck. Only vehicles with  $C_{sl} > C_T$  are considered when determining the classification accuracy.

log-likelihood threshold (18) needs to be adjusted. The new threshold is

$$LLT_p^t = \gamma_t [\beta_t(LL_{in} - LL_{out}) + LL_{out}] \quad (21)$$

$$\gamma_t = \frac{\text{win} + 1}{Q}. \quad (22)$$

The abnormality threshold is adjusted with  $\gamma_t$ , averaging the log likelihood into every model state and adjusting for the number of points in  $T_{win}$ . Here,  $\beta_t \in [0, 1]$  is chosen such that  $\beta_t > \beta$  to immediately ensure detection of most suspicious points during tracking. As soon as an object strays from a path model, an unusual event is triggered, allowing for timely detection. Fig. 8 demonstrates tracking abnormality detection for a U-turn maneuver. Initially, the vehicle is traveling in Lane 1; as the U-turn begins, the red X marks an unusual action. After completing the maneuver, the vehicle realigns itself into an acceptable lane.

4) *Path Prediction*: In addition to detecting abnormalities, it is possible to predict behavior using the HMMs. With path models, estimation can be further extended in time than the standard

one-step prediction (Kalman prediction). Accurate path prediction is essential for intersection safety evaluation. A dangerous maneuver can be better assessed by knowing the intent of a driver. Given the top 3 best fitting paths, which are found by evaluating  $P(T_{win}|\lambda_i)$ , prediction is accomplished by determining the probability of remaining in each path. These probabilities are estimated by evaluating  $P(T_{t+1}|\lambda_i)$ , where  $T_{t+1} = \{s_t, \hat{s}_{t+1}\}$  consists of the current and predicted future states. Fig. 9 shows an example of turn prediction. The probability of the correct path improves as the turn progresses, matching our intuition.

## VII. EXPERIMENTAL RESULTS

The following experiments test the accuracy of the VECTOR and path behavior activity analysis modules. VECTOR classification is tested over the course of a single day, and flow analysis is compared to hand-counted flow and with inductive loop detector data from Berkeley's Freeway Performance Measurement Project (PeMS) [32]. The PB automatic path discovery scheme is evaluated on different scenes to characterize the performance of path classification, abnormality detection, and unusual event detection.

### A. Confidence Weighted Vehicle Classification

The wkNN database used for classification was constructed from 10 min of hand-labeled training video. Sixty percent of the detected vehicles in the training video were used to populate the training database. To test the VECTOR classifier, a 5-min clip was saved every hour over a 24-h period, and each vehicle was hand labeled into one of the eight vehicle types. The classification results presented only consist of clips during the daylight hours between 06:00 and 20:00 because object detection failed during low-light conditions. Only the vehicles with confidence greater than the threshold, i.e.,  $C_{sl} > C_T$ , are considered when determining the classification accuracy presented in Fig. 10. The classification improves from 77.5% to 94% while using 6500 to 1336 tracks to compute the accuracy as the confidence threshold increases from 0% and 99.99%. In addition to the full eight-class problem, a smaller six-class merged problem, where often confused vehicle types are grouped (SUV + Van and Semi + Truck), was evaluated. At low confidence, there

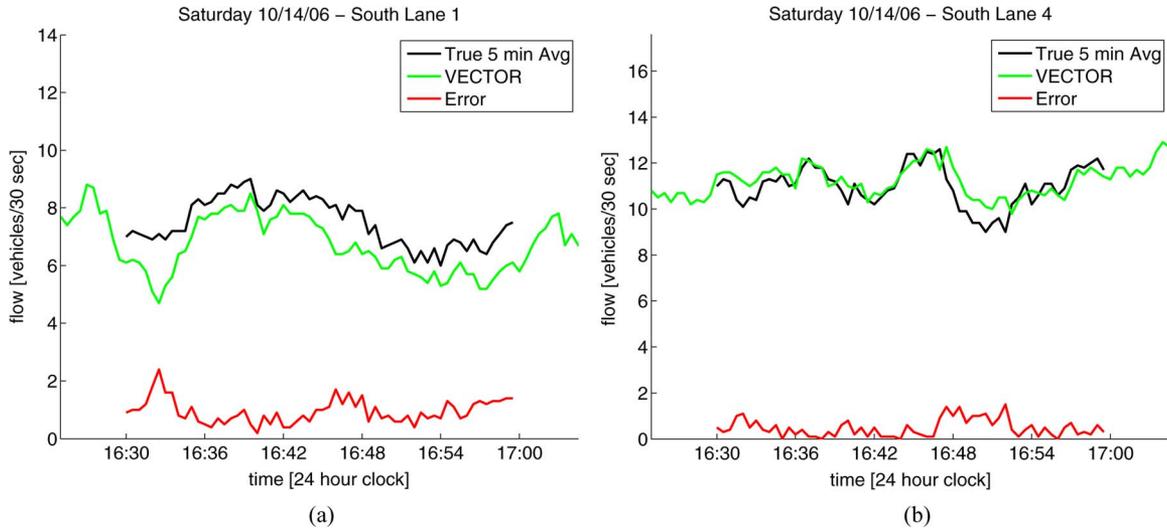


Fig. 11. True lane flows, 5-min average truth, and flow analysis module flow comparison. (a) Lane 1. (b) Lane 4.

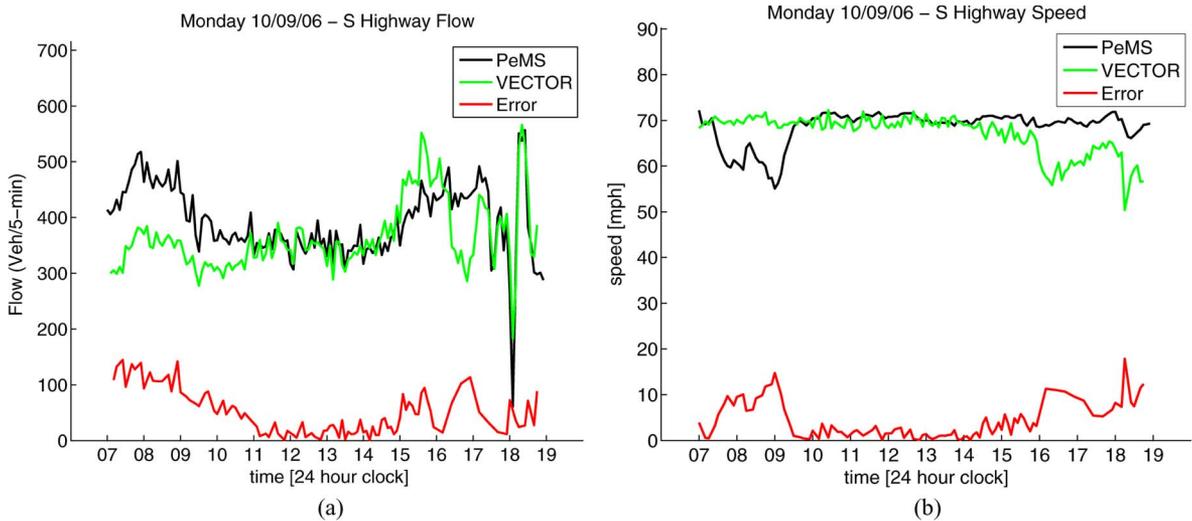


Fig. 12. Comparison between the PeMS data and the flow analysis module shows strong agreement. (a) Flow. (b) Speed.

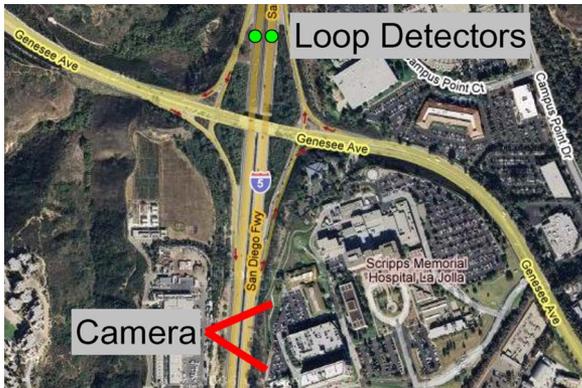


Fig. 13. Camera and PeMS loop detector sensor configuration on opposite sides of the Genesee Ave. ramp.

is a large 4% difference between the curves, but at higher confidence, the gap decreases. This demonstrates the tradeoff between classification accuracy, number of vehicle types, and confidence.

In addition to the sidelobe confidence measure, the confidence was also measured using a normalized probability estimate, i.e.,

$$C_{\text{prob}} = \frac{1}{Z} \max_c \prod_{t=1}^T p(x_t|c), \quad Z = \sum_{c=1}^8 \prod_{t=1}^T p(x_t|c). \quad (23)$$

Fig. 10 shows that the classifier performance is better using sidelobe confidence over class probability. This experiment shows that the VECTOR classification system provides good performance over the wide range of conditions encountered in a day.

*B. Traffic Flow Comparison*

The accuracy of the flow analysis module was tested by comparing the estimated flow with manual hand counts. The true 30-s vehicle counts are averaged in a 5-min sliding window for a direct comparison with the flow analysis output. Fig. 11 plots the 5-min hand count average as a black line, the VECTOR



Fig. 14. Occlusion causes initialization of new trajectories. When the occlusion is of short duration, the tracking module is able to recover the correct trajectory labels.

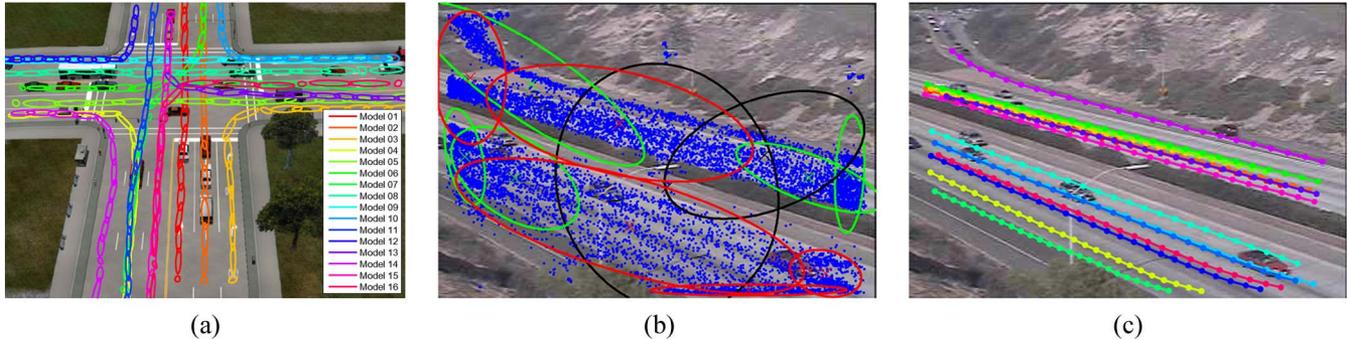


Fig. 15. (a) Path modeling results for a simulated intersection with all routes perfectly located. (b)(c) Path learning on I5 with 3/4 view with many noisy trajectories because of perspective based occlusion. (b) False zones located along the road because of high occlusion rate. (c) Route clusters match lanes well even with large amounts of noise.

output as a green line, and the error as a red line. The ground-truth flow error is usually less than two vehicles in every 30-s window, demonstrating the accuracy of the estimate.

The extracted traffic parameters were also evaluated over a longer time period by a comparison with the loop detector measurements available from the PeMS website. The PeMS data accumulate flow and average speed in 5-min windows rather than the 30-s intervals used by the flow analysis module. Fig. 12 plots the southbound PeMS data along with the 5-min corrected flow analysis module estimates. There is generally good consistency with PeMS, but there are noticeable differences in speed, as shown in Fig. 12(b). There is an early morning drop in PeMS speed, and conversely, there is a significant evening slowing of traffic in the VECTOR plot. The discrepancy in speed measurements comes from the different sensor configurations. The PeMS detectors and camera setup are on opposite sides of the busy Genesee Ave. ramp, as shown in Fig. 13, causing the speed disagreement.

1) *Occlusion Difficulties*: Fig. 12 gives the total link flow, rather than per lane, because of the occlusion difficulties encountered over the course of a day. The true flow data were obtained when traffic density was low and, hence, fewer occlusions to corrupt lane assignment. We incorporate additional counts for merged vehicles. This unfairly overestimates the flow for the lane assigned to the merged detection and misses counts for the other shared lanes, making a PeMS lane-by-lane comparison unfeasible. When the traffic is free flowing, there are very few merged tracks (i.e., 10.7%). However, this merging significantly increases when there is high-density traffic. Examples of occlusions are visible in Fig. 6(c). When there is congestion, 25% of all trajectories arise from merged vehicles, and 20% of those come from merging three or more vehicles. Each merged track was counted as 2.2 vehicles during traffic parameter collection.

Occlusion degrades the accuracy of the VECTOR flow estimate because merged vehicles must be statistically counted. The accuracy will be improved by explicitly handling occlu-

sion. Currently, occlusions cause new tracks to be initialized. If the occlusion is of short duration, the tracking recovers the correct trajectories by keeping trajectories active and updating the tracking state with its Kalman prediction, as shown in Fig. 14. When detections either merge or split, the true trajectories can be recovered through track-based reasoning [24], [33], where temporal characteristics are exploited.

While track-based occlusion recovery is possible in most situations, it will prove difficult when there is congestion. Congestion will cause a high rate of occlusion and potentially many occlusion pairs for each vehicle. The temporal constraints need to be augmented by appearance modeling [34]. By improving object detection with a mean-shift color clustering algorithm [35], shadow removal [36], and inclusion of temporal occlusion modeling, many of the merged vehicles may be resolved into their constituent parts that are suitable for lane-level analysis.

### C. Path Validity

Paths were automatically generated using the FCM procedure for three different scenes: a synthetic intersection and two real scenes. Figs. 8 and 15(c) depict two different I5 views obtained by changing the PTZ camera configuration, from side view to 3/4 view, whereas Fig. 15(a) shows the simulated traffic intersection. The lanes were all correctly discovered in the simulated intersection, as well as in the real highway scenes. Unfortunately, the merge procedure did not perform as well in the real scenes because of larger path variation. Comparing the zones in Figs. 8(a) and 15(a), we can see that there was significantly more tracking noise in the 3/4 view than the side view. The side-view training data were collected in the middle of the day when traffic is low and occlusions are rare. This ensured a large number of complete trajectories, as well as reasonable entry/exit zone discovery. In contrast, the 3/4-view data were taken over the morning commute hours and inherently has more occlusions. During times of high traffic and occlusion, the trajectories will often be broken, making it

TABLE I  
15 LANE CLASSIFICATION RATE (IN PERCENT)

	Lane 1	Lane 2	Lane 3	Lane 4	Total
South	98.7	100	96.2	97.6	98.0
North	100	91.7	84.4	94.6	93.0

difficult to learn the lanes. Only about 20% of the collected trajectories were complete. Yet, even with large amounts of noise, the FCM clustering procedure was able to find most of the lanes [Fig. 15(b)]. Notice that the northbound slow lane next to the off-ramp was not actually discovered. This is because it had little support after entry/exit zone filtering excluded some from the clustering step.

The FCM clustering results are quite promising. Collecting tracking data (something we want to do anyway) long enough to ensure a sufficient number of trajectories from each real lane is adequate for path learning. The difficulty associated with cluster merging may need to be addressed by better camera calibration for better separation of lanes; definition of new similarity metrics that incorporate cluster proximity and membership; adding more tracking features in addition to just the centroid, such as bounding box coordinates; or some combination of all three as robust path discovery is essential for automatic traffic analysis.

#### D. Lane Assignment

The quality of the learned HMMs were evaluated by their ability to assign lanes to 923 hand-labeled tracks from I5 video (this test set only included correctly tracked vehicles). Eight hundred seventy nine were correctly labeled for 95% accuracy. Table I shows the accuracy for each individual lane. Not surprisingly, most errors occur in the northbound lanes, which suffer from perspective distortion, causing the lanes to appear very close in the image plane. In these lanes, the centroid was not always stable and could float into higher lanes.

Using 13 trajectories from the simulated intersection, the detection of abnormal tracks was verified. Each of the 13 test trajectories was classified into a lane, and the log likelihoods were compared with the abnormality threshold (18). Using  $\beta = 0.9$ , all five of the anomalies were correctly identified, and lanes were correctly assigned to the remaining eight tracks.

#### E. Online Analysis

Using the same 13 intersection tracks with known abnormalities, the performance of the online unusual event detector was evaluated. In total, there were 277 tracking points, 46 of which were considered abnormal by manual evaluation. The system was able to correctly locate  $40/46 = 87\%$  of the points while missing only  $6/46 = 13\%$ , using (21) with a temporal window of size 5. Fig. 8 shows an example of an unusual event denoted by a red X. Although six points were completely missed, the abnormality was always detected after a few samples of delay. There were  $3/277 = 1.1\%$  false alarms caused by a vehicle stopping at the intersection because this was not in the training set.

Path prediction cannot be readily evaluated because it is impossible to determine a true path probability. Yet, the PB prediction results (Fig. 9) seemed to produce legitimate estimates.

## VIII. CONCLUSION

A visual activity analysis scheme based on tracking was developed for live highway monitoring and activity analysis. The proposed system observes scene motion to build high-level behavior models. The VECTOR system accurately classifies vehicles, builds a highway model by collecting traffic flow statistics, and categorizes live traffic flow in real time. In contrast to the highway specific VECTOR system, the presented path behavior block is a general tool to extract behaviors in an arbitrary scene. The normal scene motions are automatically learned to build probabilistic models that encode the spatiotemporal nature of activities present in the tracking data. Statistical inferencing based on these models allows for the detection of abnormal trajectories, as well as online analysis, path intent prediction, and usual activity detection. Both of these systems add real-time situational awareness to visual surveillance systems by leveraging motion tracking for enhanced activity and behavioral analysis.

## REFERENCES

- [1] B. T. Morris and M. M. Trivedi, "Real-time video based highway traffic measurement and performance monitoring," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Seattle, WA, Sep. 2007, pp. 59–64.
- [2] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image Vis. Comput.*, vol. 21, no. 4, pp. 359–381, Apr. 2003.
- [3] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, Jun. 1999, pp. 246–252.
- [4] M. Trivedi, I. Mikic, and G. Kogut, "Distributed video networks for incident detection and management," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Dearborn, MI, Oct. 2000, pp. 155–160.
- [5] G. T. Kogut and M. M. Trivedi, "Maintaining the identity of multiple vehicles as they travel through a video network," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Oakland, CA, Aug. 2001, pp. 756–761.
- [6] S. Bhonsle, M. Trivedi, and A. Gupta, "Database-centered architecture for traffic incident detection, management, and analysis," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Dearborn, MI, Oct. 2000, pp. 149–154.
- [7] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," in *Proc. 4th IEEE Workshop Appl. Comput. Vis.*, Princeton, NJ, Oct. 1998, pp. 8–14.
- [8] O. Hasegawa and T. Kanade, "Type classification, color estimation, and specific target detection of moving targets on public streets," *Mach. Vis. Appl.*, vol. 16, no. 2, pp. 116–121, Feb. 2005.
- [9] C. Zhang, X. Chen, and W. bang Chen, "A PCA-based vehicle classification framework," in *Proc. IEEE Data Eng. Workshops*, Apr. 2006, p. 17.
- [10] 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. Transp. Syst.*, vol. 1, no. 2, pp. 86–97, Jun. 2000.
- [11] 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.
- [12] L. Jiao, Y. Wu, G. Wu, E. Y. Chang, and Y.-F. Wang, "Anatomy of a multicamera video surveillance system," *Multimedia Syst.*, vol. 210, no. 2, pp. 144–163, Aug. 2004.
- [13] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 37–47, Mar. 2002.
- [14] Z. Zhu, G. Xu, B. Yang, D. Shi, and X. Lin, "VISTRAM: A realtime vision system for automatic traffic monitoring," *Image Vis. Comput.*, vol. 18, no. 10, pp. 781–794, Jul. 2000.

- [15] P. Kumar, S. Ranganath, H. Weimin, and K. Sengupta, "Framework for real-time behavior interpretation from traffic video," *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 43–53, Mar. 2005.
- [16] 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.
- [17] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," in *Proc. Brit. Conf. Mach. Vis.*, Sep. 1995, vol. 2, pp. 583–592.
- [18] J. Owens and A. Hunter, "Application of the self-organising map to trajectory classification," in *Proc. IEEE Vis. Surveillance*, Jul. 2000, pp. 77–83.
- [19] 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.
- [20] W. Hu, D. Xie, T. Tan, and S. Maybank, "Learning activity patterns using fuzzy self-organizing neural network," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 3, pp. 1618–1626, Jun. 2004.
- [21] 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.
- [22] D. Makris and T. Ellis, "Learning semantic scene models from observing activity in visual surveillance," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 3, pp. 397–408, Jun. 2005.
- [23] B. T. Morris and M. M. Trivedi, "Improved vehicle classification in long traffic video by cooperating tracker and classifier modules," in *Proc. IEEE Int. Conf. Advanced Video Signal Based Surveillance*, Sydney, Australia, Nov. 2006, p. 9.
- [24] J. Melo, A. Naftel, A. Bernardino, and J. Santos-Victor, "Viewpoint independent detection of vehicle trajectories and lane geometry from uncalibrated traffic surveillance cameras," in *Proc. Int. Conf. Image Anal. Recog.*, Oct. 2004, pp. 454–462.
- [25] *2001 National Household Travel Survey*, 2001, Washington DC: U.S. Dept. Transp. (DOT). [Online]. Available: <http://nhts.orl.gov/2001>
- [26] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.
- [27] W. Pedrycz, *Knowledge-Based Clustering: From Data to Information Granules*. Hoboken, NJ: Wiley, 2005.
- [28] B. T. Morris and M. M. Trivedi, "Robust classification and tracking of vehicles in traffic video streams," in *Proc. IEEE Conf. Intell. Transp. Syst.*, Toronto, ON, Canada, Sep. 2006, pp. 1078–1083.
- [29] C. Chen, Z. Jia, and P. Varaiya, "Causes and cures of highway congestion," *IEEE Control Syst. Mag.*, vol. 21, no. 6, pp. 26–32, Dec. 2001.
- [30] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc.*, vol. 39, no. 1, pp. 1–38, 1977.
- [31] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
- [32] *PeMS Performance Measurement System 7.0*. Caltrans, Univ. Calif., Berkeley, PATH, Berkeley, CA, 2006. [Online]. Available: <http://pems.eecs.berkeley.edu>
- [33] Y.-K. Jung and Y.-S. Ho, "Traffic parameter extraction using video-based vehicle tracking," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Oct. 1999, pp. 764–769.
- [34] J. Pan and B. Hu, "Robust occlusion handling in object tracking," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 1–8.
- [35] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, Hilton Head Island, SC, 2000, vol. 2, pp. 142–149.
- [36] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara, "Detecting moving shadows: Algorithms and evaluation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 918–923, Jul. 2003.



**Brendan Tran Morris (S'07)** 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 (UCSD), La Jolla, in 2006. He is currently working toward the Ph.D. degree with specialization in intelligent systems, robotics, and controls at UCSD.

His research interests include intelligent surveillance systems and recognizing and understanding activities in video through machine learning.



**Mohan Manubhai Trivedi (SM'07)** 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 (UCSD), La Jolla. He established the Laboratory for Intelligent and Safe Automobiles (LISA), UCSD, to pursue multidisciplinary research agenda. He served as the Editor-in-Chief of the *Machine Vision and Applications Journal* from 1996 to 2003. He currently serves as an Expert Panelist for the Strategic Highway Research Program (Safety) of the Transportation Research Board of the National Academy of Sciences. Currently, his team is pursuing systems-oriented research in distributed video arrays and active vision, omnidirectional vision, human body modeling and movement analysis, face and affect analysis, intelligent driver assistance, and active safety systems for automobiles.

Prof. Trivedi received Best Paper Awards for two of his recent papers, which he coauthored with his Ph.D. students. He received the Distinguished Alumnus Award from Utah State University, the Pioneer Award (Technical Activities), and the Meritorious Service Award from the IEEE Computer Society. He is currently an Associate Editor for the *IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS*. He served as the Program Chair of the IEEE Intelligent Vehicles Symposium (IV 2006) and will serve as the General Chair of IEEE IV 2010 in San Diego.