Learning Multi-Lane Trajectories using Vehicle-Based Vision

Sayanan Sivaraman, Brendan Morris, and Mohan Trivedi
Computer Vision and Robotics Research Laboratory
University of California, San Diego
http://cvrr.ucsd.edu

Abstract

Safe operation of a motor vehicle requires awareness of the current traffic situation as well as the ability to predict future maneuvers. In order to provide an intelligent vehicle the ability to make predictions, this work proposes a framework for understanding the driving situation based on vehicle mounted vision sensors. Vehicles are tracked using Kalman filtering based on a vision-based system that detects and tracks using a combination of monocular and stereo-vision. The vehicles' full trajectories are recorded, and a data-driven learning framework has been applied to automatically learn surround behaviors. By learning based on observations, the ADAS system is being trained by experience. Learned trajectories have been compared between dense and free-flowing traffic conditions. Preliminary experimental results using real-world multi-lane highways show the basic promise of this approach. Future research directions are discussed.

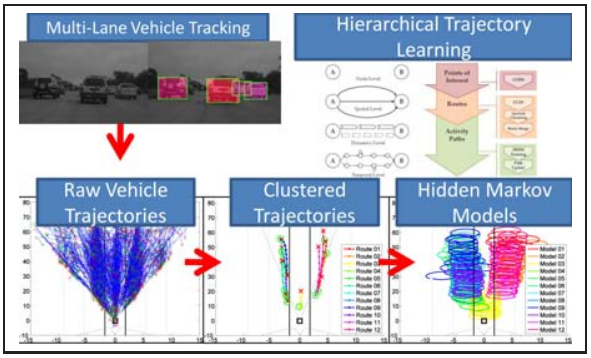


Figure 1: Proposed trajectory learning framework for characterizing vehicle behavior on multi-lane highways. Vehicles are tracked using vision, and full trajectories are archived. A hierarchical, multi-level learning approach is applied, consisting of GMM-based zone learning, clustering, and training hidden Markov models.

1. Introduction

The US National Highway Traffic Safety Administration (NHTSA) report Traffic Safety Facts 2006 [17] found in passenger car crashes that the initial point of impact was the front 49.6%. Heightened surround awareness can directly affect safe driving and manuvering of an automobile. Successful systems currently in vehicles are active cruise control (ACC), which adapts vehicle speed to maintain a safe following distance, and lane assist, which can keep a vehicle in a lane or warn a driver when drifting. This work seeks to aid the development of more advanced surround based safety systems which can give advance warning of impending situations. By predicting the behavior of surrounding vehicles, a driver is able to prepare earlier to ensure safer conditions.

In the literature, various works have addressed the problem of estimating the current motion of other vehicles on the road. In [3], Interacting Multiple Models are used to track oncoming vehicles. The goal is to characterize the vehicles' pose and motion parameters, including yaw, to determine whether the vehicle is coming straight, or turning. In [2], the problem of estimating the vehicle's pose is also addressed. The above approaches model and predict vehicle trajectories by imposing a pose and motion model, and successively update these based on the tracked vehicles. Such approaches require precise measurements of the vehicle dynamics, driver actions, and information about the surround environment [18]. In [13], trajectories were learned for the rear surround of the vehicle using radar sensing.

The next generation of driver assistance systems will need to be able to predict collisions and critical situations [22]. Examples of successful predictive systems include prediction of the ego-vehicle's driver's intent to brake [10], turn [5], or change lanes [11, 8, 7] two seconds ahead of time. The same sort of predictive ability for the behavior of other vehicles would be a great asset to enhancing on-board active safety.

This work intends to learn, understand, and predict typical surround vehicle behavior patterns automatically. Ve-

2011 IEEE International Conference on Computer Vision Workshops
978-1-4673-0063-6/11/$26.00

hicles are detected and tracked using a vision-based system that combines active learning for monocular vehicle detection, and stereo vision for ranging and tracking using Kalman filtering. The vehicles' full trajectories are recorded, and a data-driven approach based on automatic surveillance[14] is utilized to learn behavior patterns from the surround trajectories of detected objects. Surround behaviors are learned in a 3-stage unsupervised hierarchical learning process. Interesting nodes are learned through Gaussian mixture modeling, connecting routes formed through trajectory clustering, and spatio-temporal dynamics of activities probabilistically encoded using hidden Markov models. Learned trajectories have been compared between dense and free-flowing traffic conditions.

2. Multi-Lane Vehicle Detection and Tracking

2.1. Active Learning-Based Vehicle Detection

For detecting vehicles, we utilize monocular image data from the left camera of the stereo rig. We apply a monocular vehicle detector, trained using an active learning framework. This vehicle detector was part of the overall vehicle tracking system reported in [20]. The monocular detector in [20] was initialized using a labeled corpus of training examples. The resulting classifier was evaluated on independent, unlabeled on-road video data. A researcher used an interface for quick and efficient query and archival of informative independent training examples from the unlabeled data, with the learning methodology seeking inclusion of missed vehicle detections and false positives [20].

For the task of identifying vehicles, a boosted cascade of simple Haar-like rectangular features has been used, as was introduced by Viola and Jones [23]. The set of Haar-like features is sensitive to edges, bars, vertical and horizontal details, and symmetric structures [23]. The resulting extracted values are then classified by Adaboost. Modern implementations of the algorithm run at real-time speeds.

$$v_k = \begin{bmatrix} i_k & j_k & w_k & h_k \end{bmatrix}^T \tag{1}$$

Evaluating the vehicle detector on a given frame returns a list of bounding boxes. We denote one bounding box corresponding to a given detected vehicle as v_k, where k is a time-index. Bounding box v_k is parametrized by its $i-j$ pixel coordinate, and the width and height of the box, as given in equation 1.

2.2. Vehicle Localization and Tracking

The calibrated stereo rig is aligned such that the depth image and the left image share the same coordinate system. This allows us to use detections from the monocular vehicle detector to estimate the distance to a given vehicle. Inferring depth from the stereo image is not entirely straightforward, as the main error in stereo measurements is the depth component [1]. In this study, we solve for Z_k, the detected vehicle's depth using equation 2.

$$Z_k = median\ (D(i,j)),\ i,j\ \in v_k \tag{2}$$

Taking the median over the bounding box v_k improves the depth measurement for detected vehicles, but does not require us to smooth depth calculations over the entire image.

We then proceed to calculate X_k, and Y_k, the lateral and vertical positions of the vehicle respectively. We use the following equations, using the $i-j$ pixel coordinates of a given detected vehicle's bounding box, v_k. We first find the centroid of the rectangle, v_k, and then apply equation 3 to solve for X_k, and Y_k. The variables c_i and c_j are the center pixel coordinates in the i and j directions, respectively. Λ_x and Λ_y are scaling constants, intrinsic to the camera.

$$\begin{aligned} X_k &= (i_k + \frac{1}{2}w_k - c_i)\Lambda_x Z_k \\ Y_k &= (j_k + \frac{1}{2}h_k - c_j)\Lambda_y Z_k \end{aligned} \tag{3}$$

The complete measurement for a given detected vehicle, for a given time instant k is then obtained by concatenating the monocular bounding box v_k and the vehicle's full 3D position, as solved in equations 2 and 3. We track each vehicle across both the image plane and the 3D world. Each vehicle is tracked using a Kalman filter [21].

$$\begin{aligned} V_{k+1} &= AV_k + \eta_k \\ M_k &= CV_k + \xi_k \end{aligned} \tag{4}$$

$$\begin{aligned} V_k &= \begin{bmatrix} i_k & j_k & w_k & h_k & Xk & Y_k & Z_k & \Delta X_k & \Delta Y_k & \Delta Z_k \end{bmatrix}^T \\ M_k &= \begin{bmatrix} i_k & j_k & w_k & h_k & Xk & Y_k & Z_k \end{bmatrix}^T + \xi_k \end{aligned} \tag{5}$$

V_k is the full state of the tracked vehicle, and M_k is the observation taken each frame, as detailed in equations 1-5.The full state-space system is given in equation 4, where η_k and ξ_k are the plant and observation noise, respectively. The tracking formulation presented estimates the vehicle's state in both image and 3D coordinates.

3. Hierarchical Trajectory Learning

In this study, a data driven approach is used to model and predict the maneuvers of surround drivers. By observing surrounding and accruing vehicle trajectories over time, it is possible to learn what are typical highway activities, using simple detectors and trackers that operate in real-time, based on experience rather than explicit modeling. The data

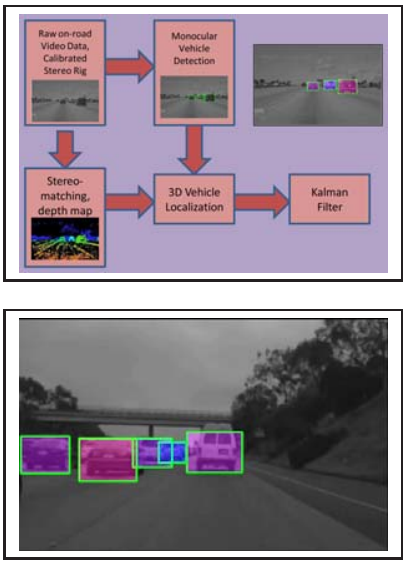


Figure 2: a)Overview of vehicle tracking approach. On-road video frames are grabbed using a calibrated stereo rig. A vehicle detector is applied to the monocular frame. Detected vehicles are then located in the depth map, and 3D coordinates are determined for each vehicle. Vehicles are tracked in both the image plane and real-world coordinates using Kalman filtering [21]. b)Example vehicle tracking results in dense traffic. The vehicles' colors are determined by their longitudinal distance, with respect to the ego vehicle. Tracked vehicles are displayed using a color gradient to indicated longitudinal distance, ranging between red for close vehicles, and blue for far away vehicles.

driven approach relies only on reliably observable features, measured by simple detectors and trackers.

The learning framework we utilize (top-right of Fig. 1) relies on low level motion information contained in tracks [15]. The main inputs for the learning engine are motion trajectories $F = \{f_1, \ldots, f_T\}$ which compactly represents object motion during the time T it is observed. In this work, $f_t = [X, Z, \Delta X_k, \Delta Z_k]^T$ represents the XZ position and velocity in either direction as extracted from the monocular+stereo-vision based sensing system above.

The learning framework provides surround maneuver description at increasing levels of resolution moving down the hierarchy. The coarsest level, the Goal Level, provides important regions in ego-surround where vehicles tend to reside and can be learned from detections. The Route Level incorporates sequential spatial information from observable motion to form initial maneuver description via trajectory clustering. In the third, and final, learning level, dynamics and temporal characteristics are combined with the spatial information to build hidden Markov models (HMMs) for complete description of a maneuver. The HMM formulation conveniently decomposes activities into distinct atomic actions with a specified duration and incorporates vehicle speed and provides the vocabulary to describe driving maneuvering, allowing behavior identification, prediction, and anomaly detection.

The hierarchical learning framework was developed and extensively tested in surveillance settings [14]. Surveillance settings typically offer an elevated vantage point, which minimizes occlusions. Surveillance video sequences lack ego-motion. Surveillance scenes are spatially constrained, and observed tracks are typically approximately temporally constrained as well. By comparison, track learning from an automotive testbed introduces the following additional difficulties:

- Non-stationary sensing and the effects of relative motion
- Variable length tracks: objects may remain in camera's view for short or long time periods
- Measurements and their characteristics are dependent on traffic context
- Behavior in windowed time intervals, and observation of multiple maneuvers executed by the same vehicle
- Limited surround sensing field of view [ie field of view of the stereo rig]
- Frequent occlusions
- Time critical safety implications

3.1. Learning Goals

Goal level learning aims to learn the areas vehicles are expected to appear or disappear from senor range as well as locations where surround vehicles idle. These form prior expectations on surrounding motion which can be used to filter noise (broken) trajectories which arise from unsuccessful tracking. The broken trajectories are removed in the learning process because they do not express a full maneuver.

The three areas of interest are called entry, exit, and stationary following. Entry and exit areas indicate where objects appear and disappear respectively and are defined by the first and last point in a trajectory respectively. The stationary areas correspond to persistent locations, consistent tracking points with very low relative speed within a small radius for a predefined amount of time [4]. Figure 3 plots the stationary following zones of tracked vehicles in dense traffic. The blue dots are vehicles that have zero velocity relative to the ego vehicle. The black ellipses are Gaussian mixture components. The stationary following area directly in front is an indication of the ego-driver's preferred following distance in dense traffic.

Using the training set of trajectories, the distribution of points corresponding to a critical area are learned through a

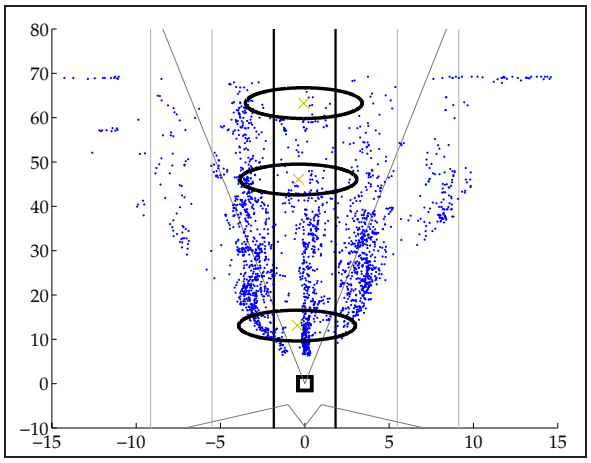


Figure 3: Stationary following regions in dense traffic. Black ellipses are the stationary following zones, where tracked vehicles have low velocity relative to the ego vehicle. Blue dots are the vehicles with low relative velocity.

2D Gaussian mixture modeling (GMM) procedure. An area $\sum_{i=0}^{M-1} m_i G([X,Z]^T, \mu_i, \Sigma_i)$ is composed of Mn Gaussians $G([X,Z]^T, \mu, \Sigma)$ of $[X,Z]^T$ coordinates learned using expectation maximization (EM) [6]. A density criterion [9] is utilized to isolate the most meaningful mixture components. A low density mixtures implies tracking noise since it is not well localized based on tracking points .

3.2. Learning Routes

The second level of maneuver understanding calls for the definition of spatial routes which separate different ways to get between goals. These routes can be learned in unsupervised fashion through track clustering. Clustering automatically extracts the most typical trajectory patterns and only relies on the definition of similarity between tracks.

In this work, spatial routes are learned by first finding the distance between training trajectory pairs using a measure derived from the longest common sub-sequence (LCSS). A similarity matrix is formed from the pairwise distances and spectrally decomposed to generate clusters. Since the number of clusters is not known a priori, the count is estimated through an agglomerative merge procedure.

3.2.1 LCSS Distance

LCSS is an alignment tool for unequal length sequence data that is robust to noise and outliers because not all points need to be matched. This property is well suited for trajectory data because their size varies depending on the amount of time spent in view of sensors. Instead of a one-to-one mapping between points, a point with no good match can be ignored to prevent unfair biasing. The LCSS distance suggested by Vlachos *et al*. [24] is defined as

$$D_{LCSS}(F_i, F_j) = 1 - \frac{LCSS(F_i, F_j)}{\min(T_i, T_j)}, \tag{7}$$

where the $LCSS(F_i, F_j)$ value (6) specifies the number of matching points between two trajectories. $F^t = \{f_1, \ldots, f_t\}$ denotes all the flow vectors in trajectory F up to time t but only utilizes the spatial information (X, Z).

3.2.2 Spectral Clustering

Using the pairwise distances between training trajectories, a rough grouping of maneuvers can be made through spectral clustering. A similarity matrix $S = \{s_{ij}\}$ is constructed from the LCSS trajectory distances using a Gaussian kernel function

$$s_{ij} = e^{-D^2_{LCSS}(F_i,F_j)/2\sigma^2} \in [0,1]. \tag{8}$$

where the parameter σ controls the trajectory neighborhood. S is used to construct the Laplacian

$$L = I - D^{-1/2} S D^{-1/2}. \tag{9}$$

D is a diagonal matrix with elements the sum of the same row in S. Spectral decomposition follows the steps outlined by Ng *et al*. [16] with the difference of using fuzzy C means (FCM) rather than k-means for the final clustering. The advantage of FCM clustering is soft class assignment that minimizes the effects of outliers and the resulting cluster membership values u_{ik} which indicate the quality of assigning a training sample i to cluster k. Fig. 4b shows 10 typical routes found through clustering.

3.2.3 Estimating Typical Maneuver Count

The number of unique highway maneuvers, as observed by the ego-vehicle, is not known a priori and must be learned as well. Initially during spectral clustering, a large number of clusters, N_c, is selected then refines to a smaller number N_p by merging similar clusters.

Trajectories are resampled and a single iteration of FCM is performed to produce route prototypes $\{r_k\}$, $k = 1, \ldots, N_c$ [12]. Two route clusters, r_m and r_n, are considered similar if each consecutive point is within a small radius,

$$d_l(r_m, r_n) = \sqrt{(X_l^m - X_l^n)^2 + (Z_l^m - Z_l^n)^2} < \epsilon_d \quad \forall l, \tag{10}$$

or if the total distance between tracks is small enough,

$$D = \sum_{l=1}^{L} d_l < \epsilon_D = L\epsilon_d. \tag{11}$$

$$LCSS(F_i, F_j) = \begin{cases} 0 & T_i = 0 \mid T_j = 0 \\ 1 + LCSS(F_i^{T_i-1}, F_j^{T_j-1}) & d_E(f_{i,T_i}, f_{j,T_j}) < \epsilon \;\&\; |T_i - T_j| < \delta \\ \max\left(LCSS(F_i^{T_i-1}, F_j^{T_j}), LCSS(F_i^{T}, F_j^{T_j-1})\right) & \text{otherwise} \end{cases} \quad (6)$$

The threshold value ϵ_d controls how far apart trajectories can lie and is chosen experimentally. A cluster correspondence list is created from these pairwise similarities, forming similarity groups $\{V_s\}$. Each correspondence group is combined into a single route (membership moved all into one route and the extra routes removed from the database). [TODO - should we show an example of the merge process aka before and after merge?]

3.3. Dynamics and Temporal Augmentation

A maneuver is not only specified by its location (trace) but also the dynamics in how it evolves over time. Using HMMs, the spatio-temporal properties of every path is encoded probabilistically and provides a natural methodology for comparison.

3.3.1 Maneuver HMM

Hidden Markov models are convenient to represent a maneuver because it provides a means to incorporate both dynamic and temporal information. A maneuver is thus is decomposed into a sequence of smaller events each having its own spatial, dynamic, and temporal characteristics by inclusion of both position and velocity $(X, Z, \Delta X, \Delta Z)$. Each HMM is compactly represented as $\lambda_k = (A_\lambda, B_j, \pi_0)$ and is designed to have Q states. The parameters π_0 represents the prior probability of an maneuver to start with state (event) q. This is naturally defined to favor events that occur earlier in an maneuver.

$$\pi_0(j) = \frac{1}{C} e^{-\alpha_p j} \qquad j = 1, \ldots, Q \quad (12)$$

where α_p controls which states and maneuver is allowed to begin in. The maneuver parameters A_λ and B_j are learned from track data. The transition matrix A_λ describes the sequential evolution of events and models their duration with an exponential distribution across time. The observation distribution B_j describes both the spatial and temporal characteristics of an event (HMM states $\{q_j\}_{j=1}^{Q}$). The observations are modeled as a single Gaussian with unknown mean and covariance,

$$b_j(f) \sim G(f, \mu_{jm}, \Sigma_{jm}). \quad (13)$$

3.3.2 Maneuver HMM Learning

A HMM is trained for each maneuver by dividing the training set into N_p disjoint sets, $D = \bigcup_{k=1}^{N_p} D_k$. The set D_k is the collection of trajectories belonging to route r_k based on membership

$$r_i^* = \underset{k}{\operatorname{argmax}}\, u_{ik} \quad \forall i. \quad (14)$$

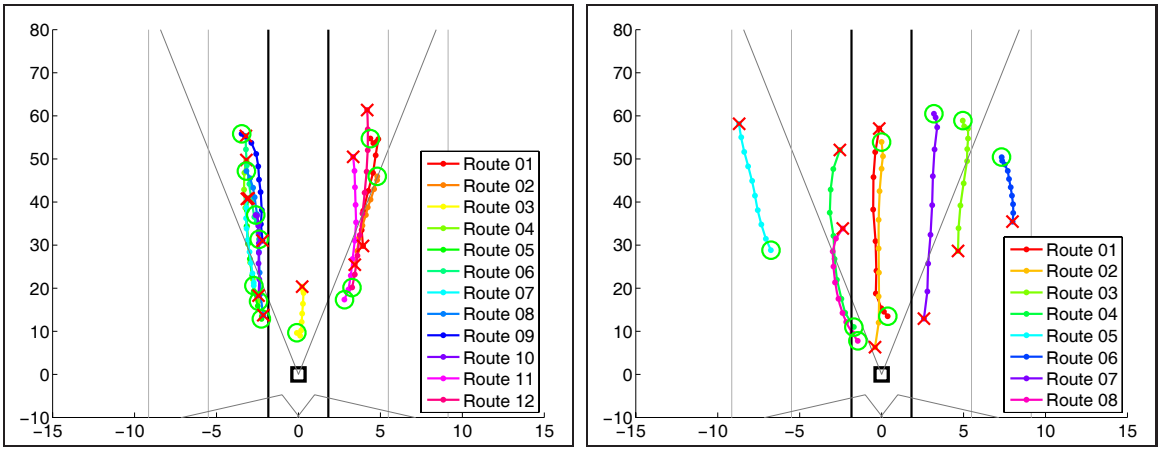
Only those trajectories with membership $u_{ir_i^*} > 0.9$ are retained when creating a path training set because they are typical trajectories and can be confidently placed into route r_i^*, indicating a typical maneuver realization. Using path training set D_k, the N_p HMMs can be efficiently learned using standard methods such as the Baum-Welch method and EM [19]. Unlike the route learning stage, when learning a maneuver HMM a full trajectory, including all tracked information *e.g.* velocity and acceleration, is used as the training input.

4. Examining Vehicle Trajectories: Results and Discussion

The trajectories collected looking out the front of the car in dense traffic are shown in figure 1. This set consists of 1490 trajectories. These trajectories correspond to vehicles that were tracked for a minimum of 15 samples (0.6 sec) along highway segments. For the free flowing dataset, there were 490 individual trajectories collected, again, a minimum of 15 samples. This is due to the fact that in free-flowing conditions, there is less traffic congestion, and fewer vehicles to track.

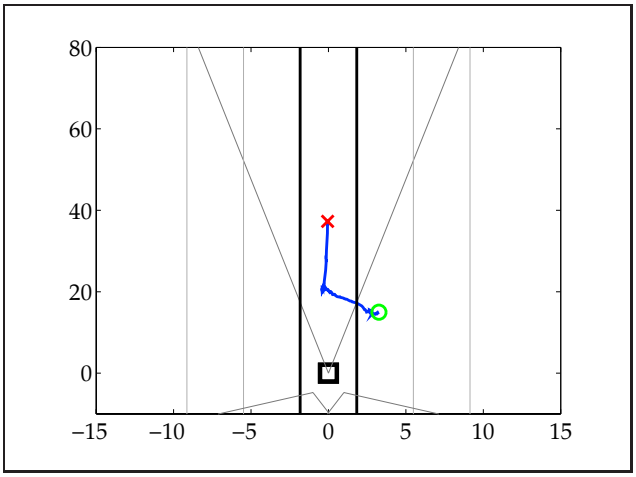
In figure 4a we can make some observations about the typically observed trajectories in congestion. During congestion the wide FOV of stereo is not really utilized because of occlusions. As such, only the side and front lanes are seen. Further, the tracks in the adjacent lanes show vehicles typically occluding each other, evidenced by the number of tracks with significant spatial overlap. In addition, we see that in congestion the ego-lane track is very short. This is indicative of close following and the stop-and-go nature during congestion which limits the effective maneuvering.

In figure 4b we observe that tracks on the driver side [left] correspond to passing vehicles. while everything on the passenger side is the overtake of a surround vehicle. In the ego-lane you can see vehicles that are slow moving (route1 - red) as well as fast (route2 - orange), which correspond to vehicles that the ego vehicle approaches, or vehicles that pull away from the ego-vehicle. During free-flow traffic, there is adherence to standard passing practice; faster vehicles overtake on the left, while in contrast, during dense traffic, passing occurs occurs on either side (eg route5 - dark green).

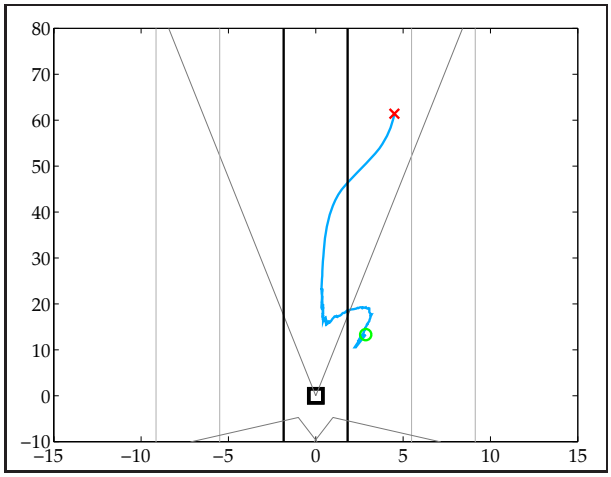


(a) Dense Traffic

(b) Free-flow Traffic



(c) Abnormal Cut-in, Dense Traffic



(d) Abnormal Cut-in, Free-flow Traffic

Figure 4: a)Typical maneuvers learned during dense congestion have limited range due to occlusion of close vehicles (only adjacent lanes visible). The stop-and-go nature of traffic limits the range of the lead vehicle in yellow (Route 3). b) Typical maneuvers learned from free flowing highway data correspond to highway passing. The beginnings of typical routes are marked with a green O, and the ends are marked with a red X. On the driver side (left), are surround vehicles passing (*e.g.* Route 05) while on the passenger side (right) are overtaken vehicles (*e.g.* Route 07) c) A vehicle cuts into the ego lane from the right lane in dense traffic. d) A vehicle cuts into the ego lane from the right lane in free-flow traffic, remains in the ego lane for some time, and then merges into the right lane again. These situations are potentially dangerous.

Abnormal vehicle trajectories are shown in figures 4c and 4d. Figure 4c shows an abnormal trajectory in dense traffic. The vehicle cuts into the ego-vehicle's lane and remains there. Figure 4d shows a vehicle cut into the ego-vehicle's lane in free-flow traffic from the right lane, stay in the go lane for a while, and then changes lanes back into the right lane. Both of these situations were potentially dangerous. The ability to predict situations like these would be an asset to ADAS.

The maneuvers extracted through this data-driven approach are simple, straight pass behaviors because they dominate highway driving. Future work will need to learn from the more infrequent actions, such as lane changes, since they could have more impact on driver safety. At this stage, we can identify abnormal vehicle activity, but we have not specifically learned the dangerous maneuvers.

In this study, we have not modeled the the shape of the road though it may play a key road in predicting behaviors as people will behave differently during curves and straight segments. The trajectories of similar actions may look quite

different depending on road pitch and curvature. In this study, trajectories were associated based on their positions and velocities, with respect to the ego vehicle. By learning the start and exit zones of tracked vehicles, we first associate them spatially. Another approach could associate different vehicles by taking into account motion patterns but not absolute positions with respect to the ego-vehicle. Further experiments can be conducted to test and compare various approaches to learning the trajectories.

5. Concluding Remarks

This work presents an approach to automatically learn behaviors of surround vehicles based on natural observation during driving. Vehicles have been tracked on highways using vision, and the vehicle trajectories have been archived. A hierarchical learning framework is presented to describe behaviors at different resolutions with accompanying unsupervised techniques. Trajectories of surrounding obstacles obtained from the front of an instrumented vehicle are examined and key behaviors which correspond to observable phenomenon, driver overtake and surround vehicle overtake, are automatically discovered demonstrating the value of this data driven approach. Learned trajectories are presented using real-world multi-lane highway. Discussion of future research challenges and possibilities is included.

References

[1] H. Badino, R. Mester, J. Wolfgang, and U. Franke. Free space computation using stochastic occupancy grids and dynamic programming. *ICCV Workshop on Dynamical Vision*, 2007. 2

[2] B. Barrois, S. Hristova, C. Wohler, F. Kummert, and C. Hermes. 3d pose estimation of vehicles using a stereo camera. *IEEE Intell. Veh. Symp.*, 2009. 1

[3] A. Barth and U. Franke. Tracking oncoming and turning vehicles at intersections. *IEEE Intell. Transp. Syst. Conf.*, 2010. 1

[4] 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.*, pages 115–120, Toronto, Canada, Sept. 2006. 3

[5] S. Y. Cheng and M. M. Trivedi. Turn-intent analysis using body pose for intelligent driver assistance. *IEEE Pervasive Computing*, 5(4):28–37, Oct-Dec 2006. 1

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc.*, 39:1–38, 1977. 4

[7] A. Doshi, B. Morris, and M. M. Trivedi. On-road prediction of driver's intent with multimodal sensory cues. *IEEE Pervasive Computing, Special Issue on Automotive Pervasive Computing*, 2011. 1

[8] A. Doshi and M. M. Trivedi. On the roles of eye gaze and head pose in predicting driver's intent to change lanes. *IEEE Trans. Intell Transp. Syst.*, 10(3):453–462, Sept. 2009. 1

[9] D. Makris and T. Ellis. Learning semantic scene models from observing activity in visual surveillance. *IEEE Trans. Syst., Man., Cybernetics. Part B*, 35(3):397–408, June 2005. 4

[10] J. C. McCall and M. M. Trivedi. Driver behavior and situation aware brake assistance for intelligent vehicles. *Proceedings of the IEEE*, 95(2):374–387, Feb. 2007. 1

[11] J. C. McCall, D. Wipf, M. M. Trivedi, and B. Rao. Lane change intent analysis using robust operators and sparse bayesian learning. *IEEE Trans. Intell. Transp. Syst.*, 8(3):431–440, Sept. 2007. 1

[12] B. Morris and M. Trivedi. Learning and classification of trajectories in dynamic scenes: A general framework for live video analysis. In *IEEE Int'l Conf. on Advanced Video and Signal based Surveillance*, pages 154–161, Santa Fe, New Mexico, Sept. 2008. 4

[13] B. Morris and M. M. Trivedi. Unsupervised learning of motion patterns of rear surrounding vehicles. *IEEE Int'l Conf. on Veh. Electronics and Safety*, 2009. 1

[14] B. Morris and M. M. Trivedi. Trajectory learning for activity understanding: Unsupervised, multilevel, and long-term adaptive approach. *IEEE Trans. on Patt. Anal. and Mach. Intell.*, 2011. 2, 3

[15] B. T. Morris and M. M. Trivedi. A survey of vision-based trajectory learning and analysis for surveillance. *IEEE Trans.s on Circuits and Systems for Video Technology*, 18(8):1114–1127, Aug. 2008. Special Issue on Video Surveillance. 3

[16] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, volume 14, pages 849–856, sep 2002. 4

[17] N. H. T. S. A. [NHTSA]. Traffic safety facts 2006. 2006. 1

[18] C. Pantilie and S. Nedevschi. Real-time obstacle detection in complex scenarios using dense stereo vision and optical flow. *IEEE Intell. Transp. Syst. Conf.*, 2010. 1

[19] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb. 1989. 5

[20] S. Sivaraman and M. M. Trivedi. A general active learning framework for on-road vehicle recognition and tracking. *IEEE Trans. Intell Transp. Syst.*, 2010. 2

[21] S. Sivaraman and M. M. Trivedi. Combining monocular and stereo-vision for real-time vehicle ranging and tracking on multilane highways. *IEEE Intell. Transp. Syst. Conf.*, 2011. 2, 3

[22] M. M. Trivedi and S. Y. Cheng. Holistic sensing and active displays for intelligent driver support systems. *IEEE Computer*, 2007. 1

[23] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *IEEE Computer Vision and Pattern Recognition Conference.*, 1(1):511–518, 2001. 2

[24] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. In *Proc. IEEE Conf. on Data Engineering*, pages 673–684, Feb. 2002. 4