2013 IEEE Intelligent Vehicles Symposium (IV)
June 23-26, 2013, Gold Coast, Australia

# Merge Recommendations for Driver Assistance: A Cross-Modal, Cost-Sensitive Approach

Sayanan Sivaraman[1], Mohan M. Trivedi[1], Mario Tippelhofer[2], and Trevor Shannon[2]

***Abstract*—In this study, we present novel work focused on assisting the driver during merge maneuvers. We use an automotive testbed instrumented with sensors for monitoring critical regions in the vehicle's surround. Fusing information from multiple sensor modalities,we integrate measurements into a contextually relevant, intuitive, general representation, which we term the Dynamic Probabilistic Drivability Map [DPDM]. We formulate the DPDM for driver assistance as a compact representation of the surround environment, integrating vehicle tracking information, lane information, road geometry, obstacle detection, and ego-vehicle dynamics. Given a robust understanding of the ego-vehicle's dynamics, other vehicles, and the on-road environment, our system recommends merge maneuvers to the driver, formulating the maneuver as a dynamic programming problem over the DPDM, searching for the minimum cost solution for merging. Based on the configuration of the road, lanes, and other vehicles on the road, the system recommends the appropriate acceleration or deceleration for merging into the adjacent lane, specifying when and how to merge.**

***Index Terms* - Active Safety, Driver Assistance, Real-time Vision, Machine Learning.**

## I. Introduction

The National Highway Transportation Safety Administration reports that automotive collisions in urban environments accounted for 43% of fatal crashes in the United States in 2009. Tens of thousands of peoples are killed on the roads each year, and most fatal crashes feature more than one vehicle [1]. Urban driving environments present a variety of critical situations that challenge the driver, and can result in collisions, injuries, and deaths. Examples of difficult urban driving scenarios include intersections, roundabouts, and merges.

Merges are of a particular concern, as they take place during the transition from urban driving to highway driving. Over a short distance, a merging vehicle must accelerate from urban driving to highway speeds, and enter oncoming traffic while maintaining adequate spacing on all sides of the vehicle. While traffic agencies provide general recommendations for merging into highway traffic [2], the maneuver is often challenging for the driver. As a result, entrance ramps are the sites of far more crashes per mile driven than other highway segments [3], with the most common accidents during merges are rear-end collisions, front-end collisions, and side-swipes.

In this work, we detail recent work oriented towards assisting the driver during merge maneuvers, as part of the Audi Urban Intelligent Assist research project. Using a uniquely instrumented automotive testbed, we capture multi-modal data in real-world on-road driving environments. The on-board environmental sensing includes cameras, radars, and lidar sensing, for perception of lanes, vehicles, and other obstacles.

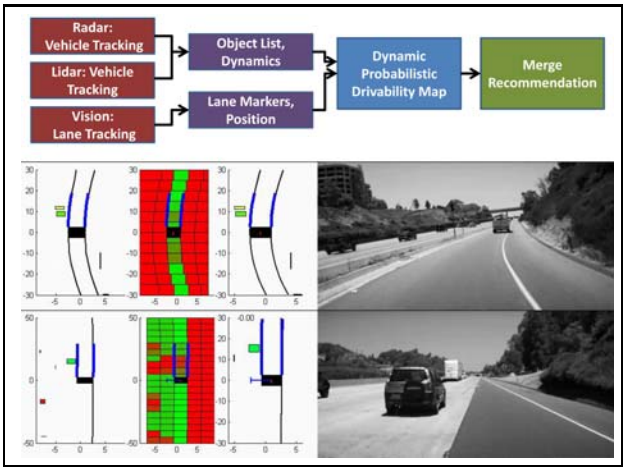



Fig. 1. Visualizing the DPDM, and merge recommendations. The DPDM takes high-level tracking information from a multi-modal sensor suite, including vision, radar, and lidar. Integrating this information into a compact probabilistic representation, the system makes recommendations for acceleration or deceleration for merge maneuvers.

We construct a compact representation of the on-road environment, which we term the Dynamic Probabilistic Drivability Map [DPDM], which takes into account obstacles, dynamics, vehicles, and legal considerations. We then provide recommendations on how to merge into the adjacent lane, based on a cost-sensitive analysis of the-road environment, solved using dynamic programming over the DPDM.

Figure 1 summarizes the system presented in this work. The top row shows the information flow, from raw sensor data, to tracking information, to the DPDM representation, and finally to merge recommendations. The second and third rows show example system output. In the left half, we show a bird's-eye view of the sensor information, DPDM representation, and recommended acceleration for merging. The bold blue lines show the predicted ego-motion over the next 1.0 second. The right-most axis shows a zoomed-in bird's-eye view complete with recommended acceleration drawn with a blue arrow. In the right half, we show video frames from the vehicle's forward-looking camera.

The main contributions of this work consist of the following. Using an instrumented testbed and its respective perception and tracking modules, an compact probabilistic

1. Sayanan Sivaraman and Mohan M. Trivedi are with the Laboratory for Intelligent and Safe Automobiles, University of California, San Diego, USA.
2. Mario Tippelhofer and Trevor Shannon are with Volkswagen Group of America's Electronics Research Laboratory in Belmont, CA, USA.

representation of the on-road environment is constructed and dynamically updated. Using this representation, the system makes recommendations on how to merge into the adjacent lane. We note that our system tells the driver not just *whether* he can merge, but *when* and *how* to merge.

The remainder of this paper consists of the following. Section 2 briefly describes related work in the field, regarding maneuver assistance and data representations. Section 3 describes the testbed build-up and sensor configuration, lays the theoretical foundations for the Dynamic Probabilistic Drivability Map, and describes solving for merge recommendations using dynamic programming. Section 4 presents experimental evaluation, and Section 5 provides concluding remarks.

## II. Related Research

The work presented in this study relates to driver assistance, maneuver assistance, environmental perception, and occupancy grids. In this section we briefly discuss related works in the intelligent vehicles literature that address these topics.

Extensive research has been conducted on assisting drivers during critical maneuvers, including turns at intersections [4] and lane changes on highways [5]. Driver-centric approaches have aimed to understand the driver's maneuvering and body language, in order to predict the driver's intent to execute a given maneuver up to 2 seconds ahead of time, allowing the vehicle to preemptively offer the appropriate assistance.

Providing assistance during maneuvers requires an understanding of the driver, the vehicle's dynamics, and the surround environment. Assistance for lane change maneuvers have tried to determine the safety of executing the maneuver by tracking vehicles in the ego-vehicle's blind spot using radar [6] or vision [7], [8]. Commercially-available radar systems are now available, sold as Side-Warning Assist [SWA] systems. By tracking vehicles in the critical regions, and computing the time-to-collision [TTC], systems of this sort provide a warning if execution of the maneuver is deemed unsafe [9].

Maneuver assistance has been taken beyond computation of TTC in [10]. Using a sensor-suite consisting of radars and vision, the system propagates measurement uncertainty into the decision process to change lanes, by using Bayesian Networks [10]. This work is further developed in [11], in which the best maneuver is selected by the Bayesian Network, and communicated to the driver via Human-Machine Interface [HMI].

In this work, we use a representation called the Dynamic Probabilistic Drivability Map, which bares some similarity to occupancy grids. Occupancy grids are a low-level representation of sensor data, typically consist of an array of cells, each of which has a probability of begin occupied by an obstacle. Occupancy grids have been widely used for sensor perception, including vision [12], [13] and lidar [14]. The grids are used to segment and detect objects by merging occupied cells, and to detect the free space in the sensor's field of view [14]. In [12], the occupancy grid is enhanced, such that the cells feature a probability of occupancy, as well as a velocity, so that occupancy particles are segmented and tracked in one framework.

Unlike occupancy grids, our DPDM formulation is not a low-level representation of sensor data, but rather a high-level representation of the driving environment. Instead of populating the map with raw sensor readings, we use high-level tracking data from vision, radar, and lidar sensors, including lane information and object lists. We further detail our approach in the following section.

## III. Assisting the Driver During Merge Maneuvers

In this section, we detail our system oriented towards assisting the driver during merge maneuvers. We briefly describe the sensor configuration of the automotive testbed. We then provide the models for on-road estimation, including ego-motion, lane models, and object tracking. We present the formulation for the Dynamic Probabilistic Drivability Map [DPDM]. We then describe how we solve for merge recommendations via dynamic programming over the DPDM. The end result is a recommendation to the driver, detailing *when* and *how* to merge into highway traffic. The current implementation computes the recommended acceleration in $\frac{m}{s^2}$, which can be presented to the driver as a highlighted recommended range of speeds in the instrument cluster.

### A. Testbed Buildup

As part of the Audi Urban Intelligent Assist [AUIA] project, we utilize a uniquely instrumented experimental testbed. Built on a 2011 Audi A8, the automotive testbed has been outfitted with extensive auxiliary sensing for the research and development of advanced driver assistance technologies. The testbed is shown in figure 2.

In this subsection, we briefly detail the environmental sensing, which consists of vision, radar, and lidar sensing. Figure 3 depicts the environmental sensing configuration. The goal of the testbed buildup is to provide a near-panoramic sensing field of view for experimental data capture. Currently, the experimental testbed features robust computation in the form of a dedicated PC for development, which taps all the available data from the on-board vehicle systems. Sensor data from the radars and lidars are fused into a single object list, with object tracking and re-identification handled by a sensor-fusion module developed by Audi [15].

*1) External Vision:* For looking out at the road, the AUIA experimental testbed features a single forward-looking camera, captured at 25Hz. This camera is capable of object detection and tracking both as a standalone unit [16] and as part of sensor-fusion setups [15]. In this study we use the camera solely for lane marker detection and lane tracking. The right half of figure 1 shows the camera view from the forward-looking camera.

Fig. 2. Instrumented Audi A8 automotive testbed used in this study.

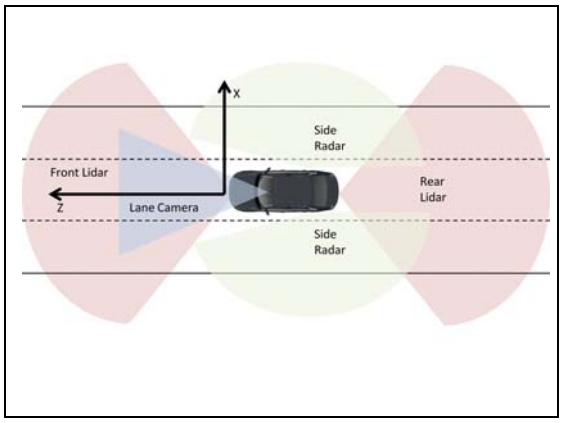



Fig. 3. Sensor configuration for the Audi Urban Intelligent Assist experimental testbed. The sensor suite includes lidars monitoring the front and rear of the vehicle, radars monitoring the sides, and a camera monitoring the lanes. The axes show the global coordinate system used throughout this work.

*2) Radar:* For tracking vehicles on the sides of the ego-vehicle, we employ two medium-range radars [MRR], which have been installed behind the rear-side panels on either side of the vehicle. The radars are able to detect and track vehicles as they overtake the ego vehicle on either side.

*3) Lidar:* The AUIA testbed features two lidar sensors, one facing forwards and one facing backwards. We use these sensors for detecting and tracking vehicles, as well as detecting obstacles such as guardrails and curbs. The lidars provide high-fidelity sensor information, and are able to estimate parameters such as vehicle length, width, and orientation, as well as position and velocity.

### B. Ego-Motion Model

We model the motion of the ego-vehicle using measurements from the inertial sensors, accessed via the CANbus. Given current velocity $v_{ego}$, and yaw rate $\dot{\psi}$, the ego vehicle moves as follows during time interval $\Delta t$. The $Z$ direction represents longitudinal distance, and the $X$ direction represents lateral distance, as shown in figure 3. [17].

$$\begin{bmatrix} X_{ego}(\Delta t) \\ Z_{ego}(\Delta t) \end{bmatrix} = \frac{v_{ego}}{\dot{\psi}} \begin{bmatrix} 1 - cos(\dot{\psi}\Delta t) \\ sin(\dot{\psi}\Delta t) \end{bmatrix} \tag{1}$$

### C. Lane Model

Using the forward-looking camera, lane markings are detected and lane geometry estimated. The system can detect lane markings corresponding to up to four lane boundaries, which comprise the ego-vehicle's lane, and the two adjacent lanes. The system differentiates between dashed and solid lane markings. We model the lanes using a standard clothoid model, given in equation 2. Using the global coordinate system shown in figure 3 [18], we parametrize the lane boundaries as a function of longitudinal distance $Z$, the curvature $C_0$, the derivative of curvature $C_1$,the ego-vehicle's angle with respect to the lane boundaries $\psi$, and the lateral position of each lane marking $L_0$ for lane markings $i \in \{1, 2, 3, 4\}$.

$$\begin{aligned} L_i(Z) =& \frac{1}{6}C_{1,i}Z^3 + \frac{1}{2}C_{0,i}Z^2 + tan(\psi)Z + L_{0,i} \\ & i \in \{1, 2, 3, 4\} \end{aligned} \tag{2}$$

We use the clothoid model for up to 25m from the ego-vehicle, beyond which we use a linear approximation based on the Taylor series expansion of the clothoid model.

### D. On-Road Vehicle Tracking

For vehicle tracking, we use a constant-velocity motion model. Each tracked vehicle's state $v_k$ at time instant $k$ is represented by a normal distribution, $p(v_k) = N(\mu_k, \Sigma_k)$, where $\mu_k$ and $\Sigma_k$ represent the expected value and covariance respectively. For a tracked vehicle, the motion estimates consist of the vehicle's lateral and longitudinal position, its width and height, and its lateral and longitudinal velocities, its orientations, and its yaw rate, relative to the ego vehicle. Ego-motion is compensated to solve for absolute motion using equation 1.

$$E(v_k) = \begin{bmatrix} X_k & \Delta X_k & Z_k & \Delta Z_k & W_k & H_k & \psi_k & \Delta\psi_k \end{bmatrix}^T \tag{3}$$

The sensor-fusion module, developed by Audi [15], tracks and re-identifies vehicles across the sensor modalities, allowing for hand-offs between the lidar and radar systems, tracking vehicles in a consistent global frame of reference.

### E. Dynamic Probabilistic Drivability Map

We base our recommendations for merging into the adjacent lanes on a compact representation of the on-road environment, the Dynamic Probabilistic Drivability Map [DPDM], an array of cells that carry $P(D)$, the probability that the area covered by the cell is *drivable*. We define drivability to mean that the ego-vehicle is able to, and legally allowed to drive in this area. This means that the area is free of obstacles, other vehicles, and is legally traversable.

Each cell in the DPDM is dynamic in space. Physically, the cells are each quadrilaterals, with lengths fixed at a car-length, approximately 5 meters, which is the standard length for passenger vehicles. The lateral positions of the are adjusted according the clothoid model in equation 2, so that a vertical column of cells corresponds to a single lane on the road, as shown in the second column of figure 1. To this end, the cells can approximate the curvature of the road.

Each cell is also dynamic in time, carrying the probability of drivability $P(D)$. The probability of drivability is determined from observing the tracking information from the environmental sensors. Consider the set of sensor measurements $Y = \{Lanes, Vehicles, Obstacles\}$. We determine the probability that the cell is drivable $P(D|Y)$ based on set of measurements $Y$, decomposing $Y$ into the lane and object tracking measurements. As the dimensions of the cells are based on the measured lane dimensions, the probability $P(D|Lanes)$ is based on the lane marking type; solid lane boundaries are illegal to traverse, while dashed markings are legal.

In determining the probability of drivability given vehicles and obstacles $P(D|Vehicles, Obstacles)$, we place the vehicles and obstacles into the DPDM, testing to see whether they lie within a given cell. Placement of the vehicles and obstacles is efficiently computed using a hash function. Upon placing vehicles and obstacles in the cell, we also store their positions and velocities for efficient computation of merge recommendations, as detailed in the following subsection. We take $P(D|Y)$ to be the min value of $P(D|Lanes)$ and $P(D|Vehicles, Objects)$.

$$P(D|Y) = \min\{P(D|Lanes), P(D|Vehicles, Obstacles)\} \tag{4}$$

We maintain the probability of drivability by representing the time series as a 2-state Markov chain, shown in equation 5. We assume each cell's probability of drivability spatially independent, allowing the observations in $Y$ to implicitly encode dependencies. The state transition probability $\Pi$ determines the probability of drivability in the next time instant $k+1$, given the probability in the current time instant $k$ [19]; $W_{k+1}$ is a martingale increment process.

$$\begin{aligned} P(D_{k+1}|D_k) &= \Pi \\ D_{k+1} &= \Pi D_k + W_{k+1} \end{aligned} \tag{5}$$

In figure 1, the middle axis shows the DPDM. The shape and dimension of the cells dynamically adapt to the road geometry, based on the lane tracking. The color of each cell indicates its current probability of drivability, with high probability regions shown in green, and low probability regions shown in red. We use a total of 100 cells in a $20 \times 5$ map, for 50m longitudinal and 5 lane-width range.

### *F. Dynamic Programming for Merge Recommendations*

As a vehicle approaches a merge situation, the driver must maintain an awareness of the vehicles in the surround, the lanes, the merge distance, and his own vehicle's dynamics. Based on the configuration of the on-road environment, the driver must make a decision on how, and when, to merge into the adjacent lane. He may choose to accelerate, decelerate, or if possible, maintain constant velocity while merging into the adjacent lane.

This problem features a high number of variables to consider, and the real-world driving environment features noisy measurements and ample uncertainty; a globally-consistent closed-form solution based solely on Newtonian kinematics is intractable. In this work, we tackle this problem by solving for the lowest-cost merge recommendation via dynamic programming over the DPDM. Dynamic programming breaks down a large problem [How should the driver merge into the adjacent lane?] into a series of inter-dependent smaller problems [20]. We use the DPDM to decompose the task into a discrete number of computations, computing the cost of accelerating and of decelerating to each possible cell location within 25m of the ego-vehicle. We choose the merging route with the lowest cost; if this cost falls below a certain threshold, we recommend this to the driver.

**Algorithm 1** Cost of Merging

$Cost(0,0) = 0$ ▷ Keep const velocity
$accel(0,0) = a_{\text{curr}}$
**for** $i = 0 \rightarrow 1$ **do**
  **for** $j = 0 \rightarrow 5$[25m ahead] **do**
    $A = cost(i, j-1) + (a_j + a_{\text{safe},i})D_j$
    $+100(1 - P(D_{i,j}))$ ▷ Cost of staying in lane
    $B = cost(i-1, j) + (a_j + a_{\text{safe},i})D_j$
    $+100(1 - P(D_{i,j}))$ ▷ Cost to change lanes
    $cost(i,j) = min(A, B)$
  **end for**
**end for**
$min\_cost = \min_j Cost(1, j)$
$a_{min} = a_{j_{min}}$
**return** $min\_cost$, $a_{min}$

Algorithm 1 details the dynamic programming steps to compute the cost of merging, and the recommended accelerations. $a_j$ is the necessary acceleration to end up at cell $j$ over time period $T$. $a_{\text{safe},i}$ is the acceleration necessary to maintain a safe distance from preceding and following vehicles in the given lane position $i$. We perform the cost computation in the forward and the rear direction, and recommend the maneuver with lower cost for acceleration/deceleration.

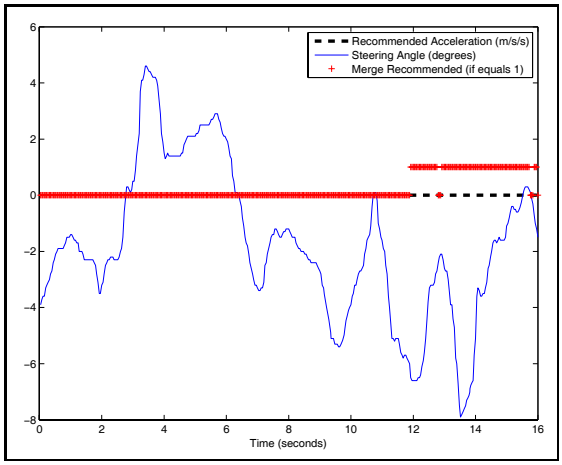
## IV. Experimental Results

To validate our approach, we evaluate the system over 42 separate merge events, captured on San Diego area highways, over the course of 3 months. Table I features statistics on the merges sequences used in this study. It takes the typical driver roughly 4 seconds to merge into traffic [2], and we chose sequences that preceded the merge event by roughly 10 seconds. The mean vehicle speed during the merge sequences was roughly 25 $\frac{m}{s}$, and the average acceleration over the sequences was roughly 0.32 $\frac{m}{s^2}$.

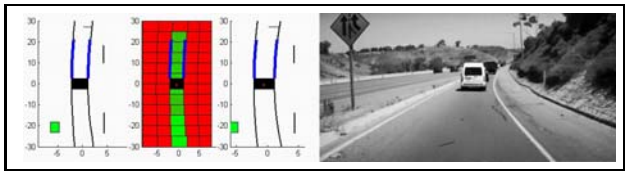
Figure 4 depicts the system performance over a merge sequence. Figure 4 a) plots the vehicle's steering angle over time in blue. As the sequence progresses, the value goes negative, indicating steering towards the left, to merge into the left lane. In red, we plot the recommendation to merge, as a 0-1 signal. For the first part of the sequence, the system does

TABLE I
MERGE DATA, 42 MERGES

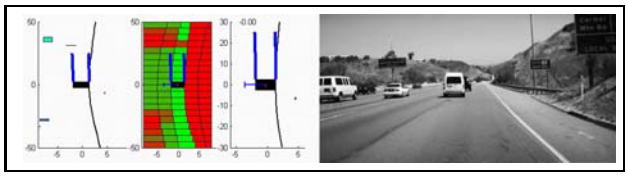
| Merge Attribute | Measurement |
|---|---|
| Segment Length, Mean | 13.6 seconds |
| Segment Length, Std. Dev | 3.4 seconds |
| Ego-vehicle Speed, Mean | 24.8 $\frac{m}{s}$ |
| Ego-vehicle Speed, Std. Dev | 4.9 $\frac{m}{s}$ |
| Ego-vehicle Acceleration, Mean | .32 $\frac{m}{s^2}$ |
| Ego-vehicle Acceleration, Std. Dev | .85 $\frac{m}{s^2}$ |




(a)



(b)

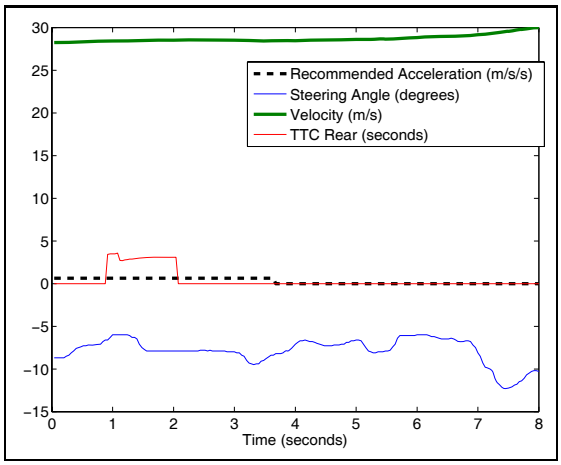


(c)

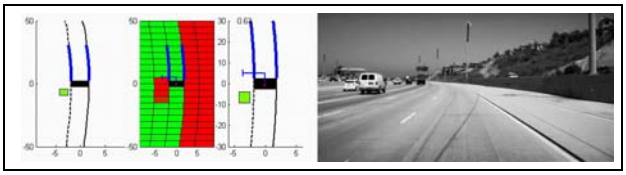
Fig. 4. System performance over a merge sequence. a) Plotting vehicle steering angle [blue], merge recommendation [red], and recommended acceleration over time [black]. Early in the sequence, a merge is not recommended, as the lane boundary is solid, as shown in b). Late in the sequence, a constant-velocity merge is recommended, as this is safe and the lowest-cost way to merge. b) As the lane boundaries are solid, the only drivable area is the ego-lane. c) The drivable area is larger, and the system recommends a constant-velocity merge into the left lane.

not recommend merging. This is because the lane marking detector identifies solid lane boundaries, indicating that a merge into the left lane is not allowed. This is shown in figure 4 b). Towards the end of the sequence, the red plot goes to 1, indicating that a lane change is recommended. The black plot in figure 4 shows a recommended acceleration of 0 at this point, meaning that the vehicle can safely merge at its current velocity, and this is the lowest-cost merge.

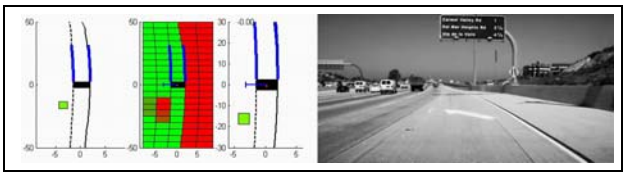
Figure 5 shows another merge sequence. We plot the vehicle's velocity in green, the steering angle in blue, the




(a)



(b)



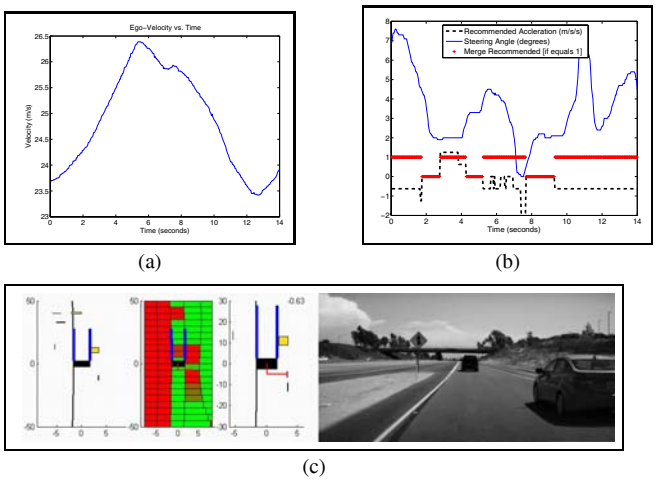
(c)

Fig. 5. A merge with recommended acceleration. a) Plotting velocity [green], steering angle [blue], recommended acceleration [black], and rear TTC [red] over time. b) The system recommends acceleration to merge, as the rear vehicle in the left lane is too close, as shown in the bird's eye view. c) The system recommends a constant-velocity merge, as the rear vehicle is no longer a threat.

recommended merge acceleration in black, and the TTC for rear collision in red. Early in the sequence, the system recommends acceleration in order to merge, because of a vehicle behind the vehicle in the left lane, as shown in figure 5 b). Later in the sequence, after the driver has accelerated, the system recommends a constant-velocity merge, as the rear vehicle is no longer a threat.

Figure 6 shows a merge sequence in which the ego-vehicle needs to merge into the right lane. Given the configuration of other vehicles, the system recommends a deceleration in order to merge. Figure 6 a) plots the ego-vehicle's velocity over time. We note that the vehicle decelerates in order to merge. In 6 b), we plot the steering angle in blue, the merge recommendation in red, and the recommended acceleration for merge in black. Due to vehicles overtaking the ego-vehicle on the right, the merge recommendation alternates between positive and negative. Towards the end of the sequence, there is a recommended deceleration in order to merge into the right lane, shown in black.

Table II features aggregate statistics for the performance of the system over the set of 42 merge sequences. The




(a) (b)

(c)

Fig. 6. A merge into the right lane, with recommended deceleration. a) Plotting vehicle velocity over time. b) Plotting steering angle [blue], merge recommendation [red], and recommended acceleration [black]. c) The system recommends a deceleration to merge into the right lane.

TABLE II
MERGE RECOMMENDATIONS, 42 MERGES

| Recommendation Attribute | Measurement |
|---|---|
| Minimum Timegap, Mean | 0.97 seconds |
| Minimum Timegap, Std. Dev | 2.94 seconds |
| Recommended Deceleration | 9.5% |
| Recommended Acceleration | 19.1% |
| Recommend Maintain Current Velocity | 71.4% |

mean minimum timegap over all sequences was approximately one second. In the vast majority merge instances, the recommended acceleration was zero, meaning the system recommended a constant-velocity merge. This follows our formulation, which seeks the lowest-cost safe merge recommendation. In about 9.5% of merges, the system recommends deceleration, about half as often as acceleration. In some 19% of instances, the system recommended acceleration, which makes sense for merging onto the highway.

## V. CONCLUDING REMARKS AND FUTURE WORK

In this study, we have presented a novel, cost-sensitive, approach for assisting drivers during merge maneuvers. Based on a compact probabilistic representation of the on-road environment, the system computes the lowest-cost safe merge maneuver, recommending to the driver *when* and *how* to merge into highway traffic. In this work we have presented theoretical formulation, and quantitative results evaluated on over 40 real-world merge sequences. Future work will include testing in different settings, such as urban lane changes and highway lane changes. We will also explore HMI concepts and interactivity with the driver [21].

## VI. ACKNOWLEDGMENTS

This work was sponsored in part by the UC Discovery Grant. The authors would like to thank colleagues at Audi Research and the Electronics Research Laboratory in Belmont, California for their support and collaboration.

## REFERENCES

[1] D. of Transportation National Highway Traffic Safety Administration, "Traffic safety facts," 2011. [Online]. Available: http://www-nrd.nhtsa.dot.gov

[2] *California Driver Handbook-Safe Driving Practices: Merging and Passing*. CA Dept. of Motor Vehicles, 2012. [Online]. Available: http://www.dmv.ca.gov/pubs/hdbk/merg_pass.htm

[3] A. T. McCartt, V. S. Northrup, and R. A. Retting, "Types and characteristics of ramp-related motor vehicle crashes on urban interstate roadways in northern virginia," *Journal of Safety Research*, vol. 35, no. 1, 2004.

[4] S. Cheng and M. Trivedi, "Turn-intent analysis using body pose for intelligent driver assistance," *Pervasive Computing, IEEE*, vol. 5, no. 4, pp. 28 –37, oct.-dec. 2006.

[5] A. Doshi and M. Trivedi, "On the roles of eye gaze and head dynamics in predicting driver's intent to change lanes," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 453 –462, sept. 2009.

[6] X. Mao, D. Inoue, S. Kato, and M. Kagami, "Amplitude-modulated laser radar for range and speed measurement in car applications," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 408 –413, march 2012.

[7] J. D. Alonso, E. R. Vidal, A. Rotter, and M. Muhlenberg, "Lane-change decision aid system based on motion-driven vehicle tracking," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 5, pp. 2736 –2746, sept. 2008.

[8] B.-F. Lin, Y.-M. Chan, L.-C. Fu, P.-Y. Hsiao, L.-A. Chuang, S.-S. Huang, and M.-F. Lo, "Integrating appearance and edge features for sedan vehicle detection in the blind-spot area," *Intelligent Transportation Systems, IEEE Transactions on*, 2012.

[9] J. Hillenbrand, A. M. Spieker, and K. Kroschel, "A multilevel collision mitigation approach–its situation assessment, decision making, and performance tradeoffs," *IEEE Trans. Intell. Trans. Syst.*, vol. 7, no. 4, pp. 528–540, 2006.

[10] R. Schubert, K. Schulze, and G. Wanielik, "Situation assessment for automatic lane-change maneuvers," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 3, pp. 607 –616, sept. 2010.

[11] R. Schubert, "Evaluating the utility of driving: Toward automated decision making under uncertainty," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 1, pp. 354 –364, march 2012.

[12] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1331 –1342, dec. 2011.

[13] M. Perrollaz, J.-D. Yoder, A. N andgre, A. Spalanzani, and C. Laugier, "A visibility-based approach for occupancy grid computation in disparity space," *Intelligent Transportation Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –11, 2012.

[14] S. Sato, M. Hashimoto, M. Takita, K. Takagi, and T. Ogawa, "Multilayer lidar-based pedestrian tracking in urban environments," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, june 2010, pp. 849 –854.

[15] S. Matzka, A. Wallace, and Y. Petillot, "Efficient resource allocation for attentive automotive vision systems," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 2, pp. 859 –872, june 2012.

[16] S. Sivaraman and M. M. Trivedi, "Active learning for on-road vehicle detection: A comparative study," *Machine Vision and Applications*, 2011.

[17] U. Franke, C. Rabe, H. Badino, and S. Gehrig, "6d-vision: Fusion of stereo and motion for robust environment perception," in *Proc. of DAGM*, 2005.

[18] S. Sivaraman and M. Trivedi, "Integrated lane and vehicle detection, localization, and tracking: A synergistic approach," *Intelligent Transportation Systems, IEEE Transactions on*, 2013.

[19] P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to Stochastic Processes*. Houghton Mifflin, 1972.

[20] S. Dasgupta, C. Papadimitriou, and U. Vazirani, *Algorithms*. McGraw-Hill, 2007.

[21] B. Morris and M. Trivedi, "Vehicle iconic surround observer: Visualization platform for intelligent driver support applications," in *Intelligent Vehicles Symposium (IV), 2010 IEEE*, 2010, pp. 168–173.