Dynamic Probabilistic Drivability Maps for Lane Change and Merge Driver Assistance

Sayanan Sivaraman, *Member, IEEE*, and Mohan Manubhai Trivedi, *Fellow, IEEE*

***Abstract*—In this paper, we present a novel probabilistic compact representation of the on-road environment, i.e., the dynamic probabilistic drivability map (DPDM), and demonstrate its utility for predictive lane change and merge (LCM) driver assistance during highway and urban driving. The DPDM is a flexible representation and readily accepts data from a variety of sensor modalities to represent the on-road environment as a spatially coded data structure, encapsulating spatial, dynamic, and legal information. Using the DPDM, we develop a general predictive system for LCMs. We formulate the LCM assistance system to solve for the minimum-cost solution to merge or change lanes, which is solved efficiently using dynamic programming over the DPDM. Based on the DPDM, the LCM system recommends the required acceleration and timing to safely merge or change lanes with minimum cost. System performance has been extensively validated using real-world on-road data, including urban driving, on-ramp merges, and both dense and free-flow highway conditions.**

***Index Terms*—Dynamic programming, lane change, merge, predictive driver assistance, probabilistic modeling.**

I. Introduction

THE National Highway Transportation Safety Administration reports that 49% of fatal crashes feature a lane or roadway departure and that a majority of crashes feature more than one vehicle [1]. Of particular concern are complex maneuvers such as lane changes and merges (LCMs), which require the driver to maintain awareness of the vehicles and dynamics in multiple lanes. According to the NHTSA, lane-change crashes account for some 500 000 crashes per year in the United States [2]. Merge maneuvers at highway ramps account for far more crashes per mile driven than other highway segments [3].

Lane changes are a common driving maneuver, during which the ego-vehicle transitions from its current lane to an adjacent lane, either on the right or left side. A driver may execute a lane change for a variety of reasons, including traffic flow or congestion, navigation, or preference. Lane changes commonly take place in highway driving, as shown in Fig. 1(b), and in urban driving, as shown in Fig. 1(a). Merges take place during the transition from urban to highway driving, during which a temporary merge lane exists for the vehicles to rapidly accelerate up to highway speeds. A typical merge scenario is shown in Fig. 1(c).

In recent years, there has been great progress in sensing and computation for intelligent vehicles. Sensors have become higher in fidelity and cheaper over time. Computation has become cheaper and faster, whereas the advent of multicore architectures and graphical processing units (GPUs) allows for parallel processing. Research efforts that utilize intelligent vehicles, equipped with advanced sensing and computing technology, have proliferated in recent years, resulting in robust environmental perception using computer vision [4]–[6], radar [7], and lidar [8]. Using the perception modules available, researchers have begun to address decision-making and assistance for complex maneuvers, while keeping the driver in the loop.

Until recently, decision-making for driver assistance lane changes has fit a binary decision paradigm, the systems based on fundamental on-road perception answering a yes/no question. Many decision systems for lane changes have focused on when a lane change is infeasible, with sensors monitoring the vehicle's blind spots [9], [10]. When a driver is being assisted, the feedback is delivered as *negative* human-machine interface (HMI), communicating that the maneuver is *not* feasible. Commercially available active safety systems such as lane departure warning (LDW) or side warning assist (SWA) typically feature negative HMI warnings.

In this paper, we develop predictive driver assistance for LCMs, with an eye toward *positive* HMI. The system introduced in this paper is intended to communicate that the maneuver *is* feasible, as well as *when* and *how* to execute the LCM. The vehicle maintains full awareness, standing ready and available to help the driver navigate the on-road environment, when the driver requests it.

Fig. 2 illustrates the full spectrum of maneuver-based decision systems in intelligent vehicles. At one end, we have fully manual driving. Cooperative driving will integrate predictive driver assistance systems and autonomous driving, allowing for seamless transfers of control between the driver and the vehicle. At the far end of the spectrum is fully autonomous driving, which remains an active research area [11].

In this paper, we introduce a novel compact representation of the on-road environment, i.e., the dynamic probabilistic drivability map (DPDM), and demonstrate its utility in predictive driver assistance for LCMs. The DPDM is a data structure that contains spatial information, dynamics, and probabilities of drivability, readily integrating measurements from a variety of sensors. In this paper, we develop a general predictive LCM assistance system, which efficiently solves for the minimum-cost maneuver, using dynamic programming over the DPDM.

Manuscript received May 7, 2013; revised January 3, 2014; accepted February 23, 2014. Date of publication March 20, 2014; date of current version September 26, 2014. The Associate Editor for this paper was A. Eskandarian.

The authors are with the Laboratory for Intelligent and Safe Automobiles, University of California, San Diego, La Jolla, CA 92093-0434 USA (e-mail: ssivaram@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.2014.2309055

(a) (b) (c)

Fig. 1. Lane changes commonly take place in both (a) urban driving and (b) highway driving. Merges take place in the (c) transition from urban to highway driving.

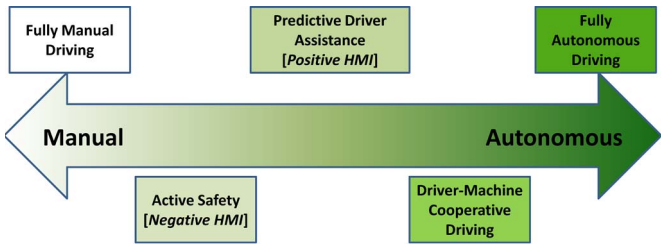
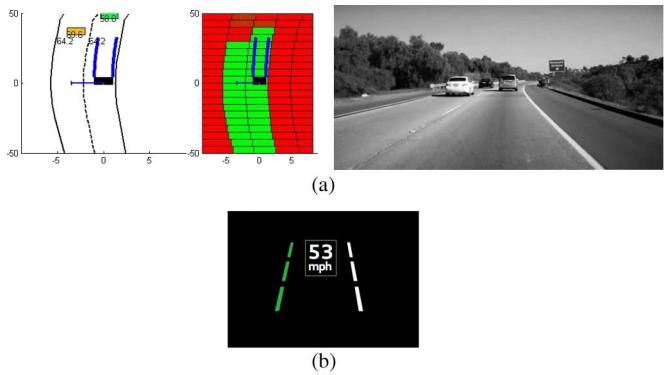


Fig. 2. Full spectrum of maneuver-based decision systems in intelligent vehicles, with implications for driving. At one end, there is fully manual driving. Active safety systems, such as LDW and SWA, are already becoming more commercially available. Predictive driver assistance remains an open area of research. Cooperative driving will integrate predictive systems and seamlessly allow handoffs of control between the driver and autonomous driving. At the far end of the spectrum is fully autonomous driving, with no input from the driver.



(a) (b)

Fig. 3. (a) DPDM. (Left) Lane-change recommendations. The probability of drivability is indicated by the color of the map cell, with green areas carrying a high probability and red areas carrying a low probability of drivability. The DPDM integrates information from lidar, radar, and vision-based systems, including lane estimation and vehicle tracking. Recommendations for lane changes are made using this information. (Right) Camera view of the road. (b) HMI concept is displayed to the driver via head-up display, during merge and lane-change maneuvers.

The full system provides timing and acceleration recommendations, which are designed to advise the driver *when* and *how* to merge and change lanes. The LCM system has been extensively tested using real-world on-road data from urban driving, dense highway traffic, free-flow highway traffic, and merge scenarios.

Fig. 3(a) shows the DPDM from a typical highway segment, whereas Fig. 3(b) shows the HMI concept, which communicates recommendations to the driver via head-up display. The full system has been fully implemented in C++ and runs in real time on the road, in the AUIA Audi A8 instrumented automotive testbed. The remainder of this paper is structured as follows: Section II provides a brief review of related work in the research literature. Section III details the theoretical formulation for the DPDM. Section IV details LCM assistance based on the DPDM. Section V features experimental results. Finally, Section VI offers concluding remarks and discussion.

II. Related Research

Here, we discuss related work in the research literature. In particular, the work presented in this paper relates to compact representations of the on-road environment and to decision-making and assistance during LCM maneuvers. In the following subsections, we discuss these areas.

A. Compact Representations

Compact representations of the on-road environment have been widely used in the literature. Mainly used for processing raw sensor data, compact representations often comprise the lowest level of data representation and have been widely used for data coming from stereo vision [12], radar [13], and lidar [14]. The most common compact representations are variations of Bayesian occupancy filters [15], which are often simply referred to as occupancy grids. The Bayesian occupancy filter is a grid-based representation of sensor data, initially proposed for processing lidar data, in which cells of fixed dimensions comprise a grid structure, each cell carrying a probability of being occupied. Raw data points from lidar scans are placed within the grid, and probabilities are propagated over time using recursive Bayesian filtering [15].

Occupancy grids in the tradition of Bayesian occupancy filter have been used in various studies in the intelligent vehicles domain. In [13], GPU processing was used to implement efficient occupancy grid computations on lidar and radar data, with applications to road boundary detection. In [16], a pyramid subsampling scheme was used to increase the efficiency of occupancy grid computations using lidar data. In [14], a 3-D occupancy structure was used to interpret raw lidar data. In [17], sequential likelihood ratios were computed for stereo-vision occupancy grids. In [8], occupancy grid methods were used to detect pedestrians using lidar. In [18], a three-state model for cells included representation as occupied, hidden, or free. In [19], a weighted sum of lidar and stereo-vision observations was used, the weight based on the confidence in the sensor's measurement.

A second major approach to occupancy grid computation has been based on Dempster–Shafer belief mass theory. Instead of computing the occupancy of a cell using probabilities, the

occupancy of a cell is represented by a belief mass, often a weighted sum of historical and currently observed data. The belief mass approach is used in [12] for occupancy grid computation using stereo-vision data and in [20] for lidar data. In [21], lidar and geo-referenced mapping data were fused for generating and refining the occupancy grid.

In [5], the stereo-vision data are represented in an occupancy grid composed of particles. Each cell is a particle, in the tradition of particle filtering, and carries a probability as well as a velocity. This representation enables low-level tracking from the raw occupancy data itself. In [22], stereo-vision data are represented as an elevation map, which is a compact representation that models the ground surface and surrounding obstacles more explicitly.

In this paper, we use a compact representation for high-level reasoning and decision-making. Rather than populate our compact representation with raw sensor data, we use a compact representation to efficiently interpret and access high-level information from on-board perception systems.

B. Decision-Making During Maneuvers

Early work in intelligent vehicles focused on fundamental perception problems and straightforward safety applications. Lane estimation research [23] has been applied to LDW applications. Blind-spot detection of vehicles using radar [7] and vision [10], [24] has been used for SWA applications. In recent years, there has been movement toward more sophisticated applications for negotiating on-road maneuvers.

Lane-change maneuvering has been a topic of great research interest. In [25], vision was used for detection of lanes and drivable area, and automatic lane changes were executed using a fuzzy controller on a scaled-down model test track. There has been also interest in automatic control of vehicles in a manner that approximates driver maneuvering [26]. In [27], statistics were collected on real-world driver maneuvering and dynamics during lane changes. These statistics were used to generate realistic lane-change trajectories. In [28], a general criticality criterion was defined, and lane-change maneuvering was suggested using simulators, but specific dynamics were not suggested. In [29], a set of lane-change trajectories was generated and evaluated, with a controller actuating a safe lane change trajectory.

In [9] and [30], a decision for whether to change lanes was made using Bayesian decision graphs, which are a variant on dynamic Bayesian networks. The Bayesian network served to propagate measurement uncertainty into the decision-making process. In [31], this work was augmented by computing the expected utility, which is a measurement derived from Shannon entropy, of changing lanes. In [32], collisions were mitigated by computing TTC times and planning evasive maneuvering.

Lane-change research has also focused on the driver. In [33], development of HMI for lane change was explored, using four predefined open-loop maneuvers, including constant-velocity "lane change," "lane change with acceleration," "lane change with deceleration," and "no lane change." Identifying and predicting the driver's intent to change lanes on highways has been also an area of research interest [34].

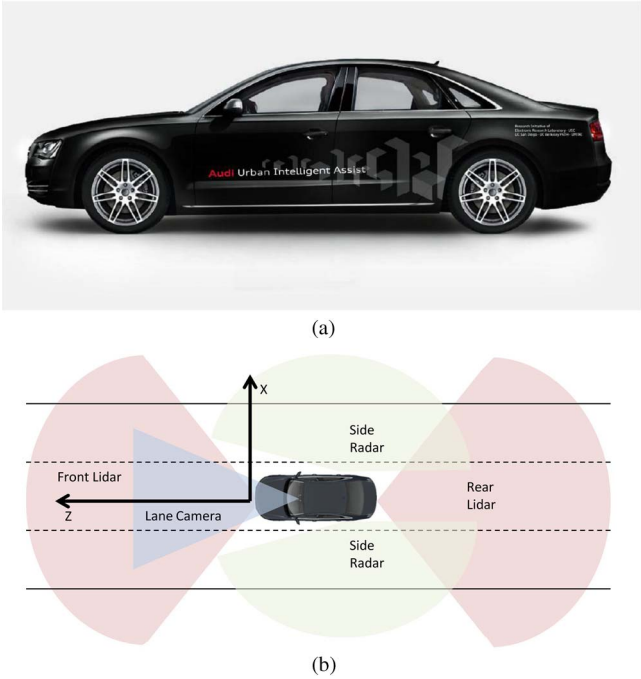


Fig. 4. (a) Audi automotive testbed used in this study. (b) Depiction of the sensing capabilities of the instrumented testbed.

Most prior work dedicated to merge maneuvers has included infrastructure-based system integration. In [35], the oncoming and merging vehicles have a communication channel via local infrastructure, i.e., the vehicle-to-infrastructure (V2I) communication node allowing the vehicles to share dynamics information to help time the merge, with a fuzzy controller actuating the maneuver. In [36], V2I channels are also used to share dynamics between vehicles. It is shown that this approach increases throughput in simulation.

III. DPDMs

In this paper, we introduce the DPDM, a compact representation for the on-road environment. The DPDM represents the ego-vehicle's surround in terms of drivability in accordance with spatial, dynamic, and legal constraints. Unlike many compact representations, which are used to represent raw low-level sensor data, the DPDM is used for high-level interpreted data. Instead of serving as a tool to facilitate object detection and tracking, the DPDM readily integrates data from on-road tracking modules, sensors, and maps, in order to compute the drivability of the ego-vehicle's surround. Here, we detail the DPDM, including theoretical basis, assumptions, and the observation sensor modules used in this study. First, we briefly describe the instrumented automotive testbed and its sensing capabilities, as shown in Fig. 4(a) and (b).

1) External Vision: For looking out at the road, the AUIA experimental testbed features a single forward-looking camera, captured at 25 Hz. This camera is capable of object detection and tracking both as a standalone unit [4] and as part of sensor-fusion setups [37]. In this paper, we use the camera for lane marker detection and lane tracking. The right half of Fig. 3 shows the camera view from the forward-looking camera's field of view.

2) Radar: For tracking vehicles on the sides of the ego-vehicle, we employ two medium-range radars, 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, i.e., one facing forward and one facing backward. 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.

A. Drivability Cell Geometry

The DPDM is comprised of an array of cells that characterize the drivability of a defined geometric region. Physically, the drivability cells are convex quadrilaterals, adapting their geometry to that of the lanes. The length of a drivability cell is fixed to 5.0 m, chosen to spatially represent drivability in terms of "car lengths." The choice of a fixed length for a drivability cell provides an intuitive measure for drivers and implies that a drivable cell should fully accommodate the ego-vehicle. Partitioning the drivable space also allows us to discretize the computation for LCMs. We parametrize a drivability cell using the four points that serve as vertices for the convex quadrilateral and, dually, the four line segments that connect them.

We use video from the vehicle's forward-looking camera to perform lane estimation via an embedded system, including detection of up to four lane boundaries, corresponding to the ego lane, left adjacent lane, and right adjacent lane. The drivability cell's geometry is derived from a piecewise-linear approximation to the road geometry. We model the lane geometry using a clothoid model [38], as shown in

$$L_i(Z) = \frac{1}{6}C_{1,i}Z^3 + \frac{1}{2}C_{0,i}Z^2 + \tan(\psi)Z + L_{0,i}, \quad i \in \{1, \ldots, N\}. \quad (1)$$

We parametrize the lane boundaries as a function of longitudinal distance Z, curvature C_0, derivative of curvature C_1, the ego-vehicle's angle with respect to the lane boundaries ψ, and the lateral position of each lane marking L_0 for lane markings $i \in \{1, \ldots, N\}$. We use the clothoid model for up to 25 m from the ego-vehicle, beyond which we use a linear approximation based on the Taylor series expansion of the clothoid model, as given in

$$\begin{aligned} Z_0 &= \pm 25 \\ m_i &= \tan(\psi) + C_{0,i}Z_0 + \frac{1}{2}C_{1,i}Z_0^2 \\ L_i(Z) &= m_i(Z - Z_0) + L_i(Z_0). \end{aligned} \quad (2)$$

Fig. 5 shows an example of the DPDM adapting its geometry to that of the road. We note the curvature of the DPDM, as estimated by the lane tracker module, in accordance with (1) and (2).

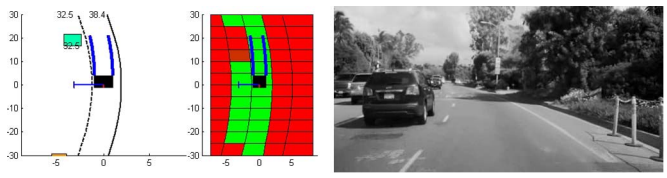


Fig. 5. Drivability cells are physically modeled as convex quadrilaterals, which adapt their geometry to the geometry of the roads and lanes. Their width adapts to the lane width, and they also adapt to accommodate lane curvature.

B. Drivability Cell Probabilities

Beyond geometry, drivability cells carry a probability of drivability. Lane information comes from the lane estimation module, which tracks the lanes using the on-board forward-looking camera. Vehicles and obstacles are detected and tracked using a sensor fusion system based on lidar and radar sensors.

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, i.e., $p(V_k) = N(\mu_k, \Sigma_k)$, where μ_k and Σ_k represent the expected value and covariance, respectively. For a tracked vehicle, the motion estimates consist of the vehicle's lateral X_k and longitudinal Z_k position, its width W_k and length L_k, its lateral ΔX_k and longitudinal ΔZ_k velocities, orientation ψ_k, and yaw rate $\Delta\psi_k$. We also predict the state of tracked vehicles Δt ahead of time using a linearized motion model, where w is a noise parameter, i.e.,

$$\begin{aligned} E[V_k] &= [X_k \quad \Delta X_k \quad Z_k \quad \Delta Z_k \quad W_k \quad L_k \quad \psi_k \quad \Delta\psi_k]^T \\ E[V_{k+\Delta t}] &= AE[V_k] + w_{k+\Delta t} \\ \Sigma_{k+\Delta t} &= A\Sigma_k A^T + E[w_{k+\Delta t} w_{k+\Delta t}^T]. \end{aligned} \quad (3)$$

Ego-motion is compensated to solve for absolute motion using (4). We model the motion of the ego-vehicle using measurements from the inertial sensors, accessed via the CAN bus. Given current velocity v_{ego} and yaw rate $\dot{\psi}$, the ego-vehicle moves as follows during time interval Δt. The Z-direction represents longitudinal distance, and the X-direction represents lateral distance, as shown in Fig. 4(b) [39]. Thus

$$\begin{bmatrix} X_{\text{ego}}(\Delta t) \\ Z_{\text{ego}}(\Delta t) \end{bmatrix} = v_{\text{ego}} \begin{bmatrix} 1 - \cos(\dot{\psi}\Delta t) \\ \sin(\dot{\psi}\Delta t) \end{bmatrix}. \quad (4)$$

Vehicle tracking information influences drivability of a given cell, as the presence of a vehicle within the cell's boundaries yields a low probability of drivability. In addition to the presence of any vehicles or obstacles within the cells, the drivability cells store their positions, dimensions, velocities, and orientations.

Lane estimation influences the drivability of a given cell for both physical and legal reasons. The physical dimensions of a drivability cell are adapted to the estimated lane geometry. The recognized lane markings indicate the legality of crossing a given lane boundary. The lane estimation module can detect solid boundaries, dashed lines, and Bott's dots. Using local traffic laws, we model the drivability probability cells that lie beyond the lane boundaries.

We define the space of sensor observations Y into tracked vehicles and objects V and lane marker information L.

The observation due to lane markings is integrated geometrically in the dimensions and spacing of the drivability cells

and probabilistically using the detected lane markings of the boundary. We define the observation based on lane markings as

$$L_k = \begin{cases} 0, & \text{if marker type is solid or not detected} \\ 1, & \text{if marker type is dashed or Bott's dot.} \end{cases} \quad (5)$$

We interpret the lane observation probabilistically, attaching observation probabilities to the observed L_k, given the current state of the cell's probability of drivability. We use the observation probabilities $P(L_k|D_k)$ to compute the cell's drivability based on lanes.

We define the observation based on vehicles based on the placement of the vehicle, testing whether the vehicle lies within the boundaries of the drivability cell by testing each corner of the vehicle. Testing whether a given point lies within a convex polygon can be efficiently computed by taking the inner product of the point with each of the line segments that define the polygon, as shown in Algorithm 1. The observation is based on any of the four corners of a tracked vehicle or detected object lying within the boundaries of a drivability cell, as shown in (6).

$$V_k = \begin{cases} 0, & \text{if vehicle/object observed partially inside cell} \\ 1, & \text{otherwise.} \end{cases} \quad (6)$$

Algorithm 1 Test Whether a Point Lies Within a Convex Quadrilateral

$result = 1$
$Cell = \{l_1, \ldots, l_4\}$ ▷ Line segments that parametrize the cell
$P = [X \quad Z \quad 1]^T$
for $i = 1 \rightarrow 4$ **do**
  $l_i = [a \quad b \quad c]^T$
  $dot = P^T l_i$ ▷ Dot product of point and line segment
  if $dot < 0$ **then**
    $result = 0$
    $break$
  end if
end for
return $result$

We interpret this observation probabilistically, attaching observation probabilities to the observed V_k, given the cell's current probability of drivability. We use the observation probability $P(V_k|D_k)$ to compute the cell's drivability based on vehicles and obstacles.

At time k, we compute the probability of drivability for a given cell $P(D_k|Y_k)$, given the observations, using (7). We note that this step is equivalent to evaluating the forward algorithm [40]. We compute $P(D_k|Y_k)$ separately given V and given L and take the minimum probability of drivability, i.e.,

$$\begin{aligned} Y_k &= [V_k \quad L_k]^T \\ P(D_k|V_k) &= \frac{P(V_k|D_k)P(D_k)}{P(V_k)} \\ P(D_k|L_k) &= \frac{P(L_k|D_k)P(D_k)}{P(L_k)} \\ P(D_k|V_k, L_k) &= \min\left\{P(D_k|V_k), P(D_k|L_k)\right\}. \end{aligned} \quad (7)$$

TABLE I
DRIVABILITY CELL ATTRIBUTES

Attribute	Units/Equations	Description
p1, p2, p3, p4	(X, Z), meters	Points that parametrize the convex quadrilateral
l1, l2, l3, l4	$aX+bZ+c=0$, meters	Lines that parametrize the convex quadrilateral
Position	(X, Z), meters	Position of vehicles/objects within cell
Velocity	$(\Delta X, \Delta Z)$, meters/second	Velocity of vehicles/objects within cell
Size	(W, L), meters	Width and length of vehicles/objects within cell
Orientation, Yaw Rate	$(\psi, \Delta\psi)$ Degrees, degrees/second	Orientation of vehicle/objects within cell
$P(D)$	Probability	Probability that the cell is drivable

We maintain the probability of drivability by representing the time series as a two-state Markov chain, shown in (8). We assume each cell's probability of drivability to be spatially independent, allowing the observations in Y to implicitly encode dependence values. The state transition probability Π determines the probability of drivability in the next time instant $k+1$, given the probability in the current time instant k [41]; W_{k+1} is a martingale increment process, i.e.,

$$\begin{aligned} P(D_{k+1}|D_k) &= \Pi \\ D_{k+1} &= \Pi D_k + W_{k+1}. \end{aligned} \quad (8)$$

Table I summarizes the attributes of the drivability cell. The drivability cell is implemented as an abstract data class and contains a drivability probability, geometric parameters, and the estimated parameters of tracked objects that lie within their boundaries.

IV. LCM RECOMMENDATIONS

In this paper, we make recommendations for LCM maneuvers. The recommendations consist of recommended accelerations and timings to execute the maneuver, specifying *how* and *when* to change lanes or merge into traffic, which is a problem that features a high number of variables. There are myriad combinations of surround vehicles, lane configurations, obstacles, and dynamics to consider. The DPDM allows us to compactly encode spatial, dynamic, and legal constraints into one probabilistic representation, which we use to compute the timing and accelerations necessary to execute a maneuver.

As shown in Table I, each cell of the DPDM carries a probability of drivability $P(D)$, as well as the position, dimensions, and dynamics of any tracked vehicles or obstacles that lie within the cell's boundaries. We display each cell's current probability of drivability encoded in color, with high-probability regions shown in green and low-probability regions shown in red, as shown in Fig. 3. We use a total of 100 cells in a 20×5 map, for a 50-m longitudinal and five-lane-width range.

We solve for the lowest cost recommendation to get into the adjacent lane, by formulating the problem as a dynamic programming solution over the DPDM. Dynamic programming breaks down a large problem into a series of interdependent smaller problems [42]. Using dynamic programming over the

DPDM serves two key roles. First, the DPDM decomposes and discretizes the search space, providing computational convenience while enabling an intuitive representation for the driver's understanding. Second, the use of dynamic programming assures that the algorithm's solutions preserve directionality in both space and dynamics, while enhancing efficiency by caching the calculations from spatially dependent cells.

A. Cost Function

We formulate the cost of a given maneuver, decomposing the cost into spatial, distance, and dynamic components. The spatial component of the cost is based on a given cell's probability of drivability. The distance cost is based on the acceleration necessary to arrive at a given cell location after a given time period. The dynamic cost is based on the necessary acceleration to safely execute the maneuver, given the other vehicles in the surround.

1) Spatial Cost: Integrating a spatial cost into the merge and lane-change recommendation system addresses two main concerns. The spatial cost ensures that recommended merge and lane-change maneuvers do not result in collisions with vehicles and obstacles in adjacent lanes. It also ensures that recommended maneuvers are not illegal, the spatial cost often indicating when and where a given maneuver is *valid*, i.e.,

$$\text{Spatial Cost}_{i,j} = K\left(1 - P(D_{i,j})\right). \tag{9}$$

We derive the spatial cost from the probability of drivability, which is stored in the DPDM. For a given cell, the spatial cost is proportional to the probability that the cell is not drivable. We set $K = 100$. Cells with a low probability of drivability carry a high spatial cost. This formulation allows us to seamlessly integrate the DPDM into the recommendation computation.

2) Distance Cost: In addition to the spatial cost, each maneuver carries a dynamic cost, based on its necessary acceleration and timing for successful execution. We probe each DPDM cell location in the adjacent lane, within 25 m of the ego-vehicle. We make use of the fact that it takes a typical driver 4–6 s to change lanes and base the initial timing of the maneuver on this fact [43]. The acceleration necessary for a vehicle to end up at distance D_j after time t can be derived from Newtonian kinematics, i.e.,

$$D_j = \frac{1}{2} a_j t^2 \tag{10}$$

where j denotes the index of a given DPDM cell, and D_j denotes its longitudinal position in the DPDM's moving frame-of-reference. Thus, we compute the acceleration and distance relative to keeping the current velocity constant, i.e.,

$$\text{Dist. Cost}_{i,j} = a_j D_j. \tag{11}$$

We compute the distance cost, deriving it from the Newtonian expression for work, i.e., $ma \cdot D$. We exclude the mass of the ego-vehicle, instead setting it to identically 1, as in (11).

3) Dynamics Cost: The above expression computes the necessary acceleration for the ego-vehicle to travel a distance in a given time. However, additional computation is required to accommodate the dynamics of the surround vehicles. We initially filter the surround vehicles, based on the current TTC or time-to-collision, computed from a given vehicle's longitudinal position x_o and velocity v_o, i.e.,

$$TTC = \frac{x_o}{v_{\text{ego}} - v_o}$$

$$a_{\text{safe}} = \begin{cases} \frac{3}{2}\frac{(v_o - v_{\text{ego}})^2}{v_o t_s - x_o}, & \text{if } TTC < \tau \\ 0.0, & \text{otherwise} \end{cases}$$

$$t_{\text{safe}} = \begin{cases} \frac{v_o - v_{\text{ego}}}{a}, & \text{if } TTC < \tau \\ 0.0, & \text{otherwise} \end{cases}$$

$$D_{\text{safe}} = \begin{cases} \frac{1}{2} a_{\text{safe}} t_{\text{safe}}^2, & \text{if } TTC < \tau \\ 0.0, & \text{otherwise} \end{cases}$$

$$\text{Dyn. Cost}_{i,j} = a_{i,j,\text{safe}} D_{i,j,\text{safe}}. \tag{12}$$

We only take into account vehicles with a TTC lower than a threshold τ, which we have set to 5.0 s. For these vehicles, we solve for the minimum safe acceleration a_{safe}, timing, and distance [9]. We then compute the dynamic cost from these parameters. Filtering surround vehicles' based on the TTC allows the system to compute the dynamics, based on the surround vehicles that are most pertinent. While all vehicles feature spatially in the DPDM and consequent spatial cost computation, only vehicles with appropriate dynamics and spacing are considered for dynamic cost computation.

TABLE II
MERGE DATA, 50 MERGES

Merge Attribute	Measurement
Segment Length, Mean	12.6 seconds
Segment Length, Std. Dev	4.5 seconds
Ego-vehicle Speed, Mean	25.0 $\frac{m}{s}$
Ego-vehicle Speed, Std. Dev	4.7 $\frac{m}{s}$
Ego-vehicle Acceleration, Mean	.32 $\frac{m}{s^2}$
Ego-vehicle Acceleration, Std. Dev	.85 $\frac{m}{s^2}$
Recommended Deceleration	20.5%
Recommended Acceleration	51.9%
Recommend Maintain Current Velocity	27.6%

Algorithm 2 Min-Cost Maneuver Recommendations

$Cost(0,0) = a_{0,0,\text{safe}} D_{0,0,\text{safe}}$
$Cost(1,0) = a_{1,0,\text{safe}} D_{1,0,\text{safe}}$
for $i = 0 \rightarrow 1$ **do**
  for $j = 0 \rightarrow 5$[25 m ahead] **do**
    $A = cost(i, j-1)$ ▷ Cost of staying in lane
    $B = cost(i-1, j)$ ▷ Cost of switching lanes here
    $cost(i,j) = \min(A, B)$
    $cost(i,j) += a_{i,j} D_{i,j}$ ▷ Distance Cost
    $cost(i,j) += a_{i,j,\text{safe}} D_{i,j,\text{safe}}$ ▷ Dynamics cost
    $cost(i,j) += K(1 - P(D_{i,j}))$ ▷ Spatial cost
  end for
end for
$\min\_cost = \min_j cost(1,j)$
$a_{\min} = a_{i,j_{\min}},\ t_{\min} = t_{i,j_{\min}}$
return $\min\_cost, a_{\min}, t_{\min}$

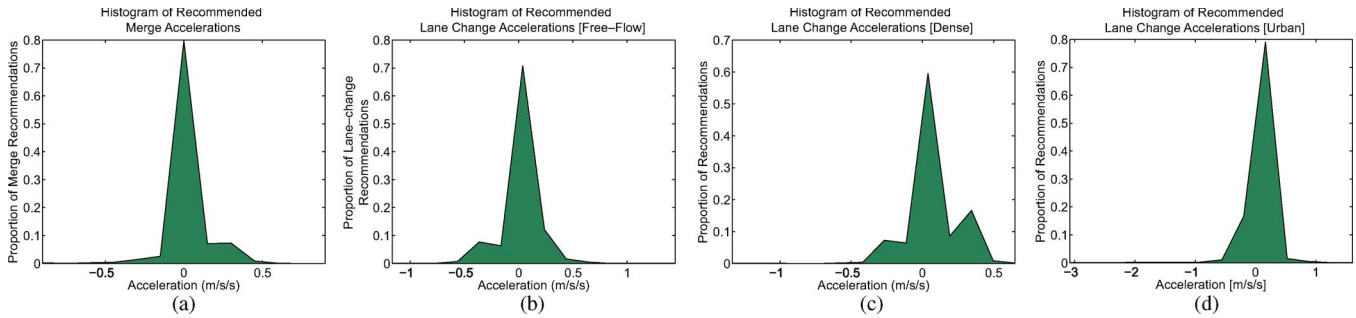


Fig. 6. Histograms of recommended accelerations during maneuvers. (a) Merges. Most of the recommendations require the ego-vehicle to accelerate, which is to be expected during merge maneuvers. (b) Free-flow highway lane changes. Most of the recommendations involve constant-velocity lane changes or decelerating to safely accommodate slower vehicles. (c) Dense highway lane changes. Most of the recommendations require a positive acceleration. This is due to the fact that there is often lane-specific congestion in dense traffic, which results in high relative velocities between adjacent lanes. (d) Urban lane changes. Urban driving features a roughly equal proportion of acceleration, deceleration, and constant-velocity lane changes.

B. Min-Cost Solution via Dynamic Programming

At each possible DPDM cell location within 25 m of the ego-vehicle, we compute a cost derived from the spatial cost, distance cost, and dynamics cost, as detailed in the previous subsection. The system then recommends the maneuver that carries the lowest cost to merge or change lanes into the adjacent lane. We efficiently solve for the lowest cost solution via dynamic programming over the DPDM.

Algorithm 2 details the dynamic programming steps to compute the cost of merging and the recommended accelerations. We compute the spatial, distance, and dynamics costs at each cell location in the ego and adjacent lanes. We perform the cost computation in the forward and rearward directions and recommend the maneuver with lower cost for acceleration/deceleration.

Using dynamic programming allows us to efficiently and correctly identify the lowest cost path into the adjacent lane, allowing the system to identify overtaking and undertaking paths around a vehicle in the adjacent lane. The returned recommendation consists of the minimum cost, the recommended acceleration, and recommended timing for the maneuver. If the cost exceeds a threshold, we term the recommendation *invalid*, and no recommendation is returned by the system.

V. Experimental Results

We evaluate the performance of the LCM system using real-world data, captured on roads and highways in San Diego. We test the system performance during four classes of maneuver: merge, free-flow highway lane changes, dense highway lane changes, and urban lane changes.

In each of the scenarios, we provide the following performance metrics. We provide statistics that describe the ego-vehicle's dynamic state during the sequence and the proportion of recommendations that require acceleration, deceleration, or constant-velocity maneuvering. We include histogram plots of the recommended accelerations. We highlight data from select sequences, providing time-series plots of recommendations' accelerations, costs, and plots of the DPDM and camera footage during the sequences.

A. Merges

We evaluate the system over 50 separate merge events, captured on San Diego area highways. Merges occur during the transition from urban to highway driving and take place as the ego-vehicle enters highway traffic. The sequences are taken from a number of separate data capture drives and take place at various times of the day and during various months of the year. We annotate the merge sequence to include roughly 10 s of captured data prior to the merge itself.

Table II features statistics on the merges sequences used in this study. During merge maneuvers, most of the recommendations entail acceleration. This makes sense, as vehicles typically need to accelerate up to highway speeds as they enter highway traffic. As shown in Fig. 6(a), while most of the merge recommendations feature acceleration, there is a peak at 0 $(\mathrm{m/s^2})$, which comprises merge scenarios where the system recommends constant velocity.

We highlight a merge sequence in Fig. 7, during which there are no pertinent vehicles in the surround, as described by (12). As such, the only contributions to the cost evaluation come from the spatial cost of the DPDM, in this case, exclusively from the solid lane boundary. As shown in Fig. 7(b), at the beginning of the sequence, the solid lane boundaries render the DPDM drivability probabilities quite low in the left cells, which correspondingly contributes high cost to the computation, as shown in Fig. 7(a). Once the left-lane boundary changes from solid to dashed, the DPDM probabilities become high in the left-lane cells, and the spatial cost significantly reduces. At this point, the system recommends a constant-velocity merge.

In Fig. 8, we highlight a merge in which the system recommends an acceleration. The top image in Fig. 8 shows the cost, recommended acceleration, and actual vehicle velocity over time, whereas Fig. 8(b) and (c) shows the DPDM early and later in the maneuver, respectively. Early in the maneuver, the system recommends an acceleration, due to a vehicle in the left lane. The ego-vehicle accelerates, whereas the surround vehicle decelerates, and the cost and recommended acceleration reduce to 0.

B. Highway Lane Changes: Free-Flow Traffic

We evaluate the system for assistance during lane changes using 25 sequences captured in free-flowing highway traffic. We distinguish free-flow from dense highway traffic based on the speed of the ego-vehicle during the segment and on the number of surround vehicles within a 50-m longitudinal

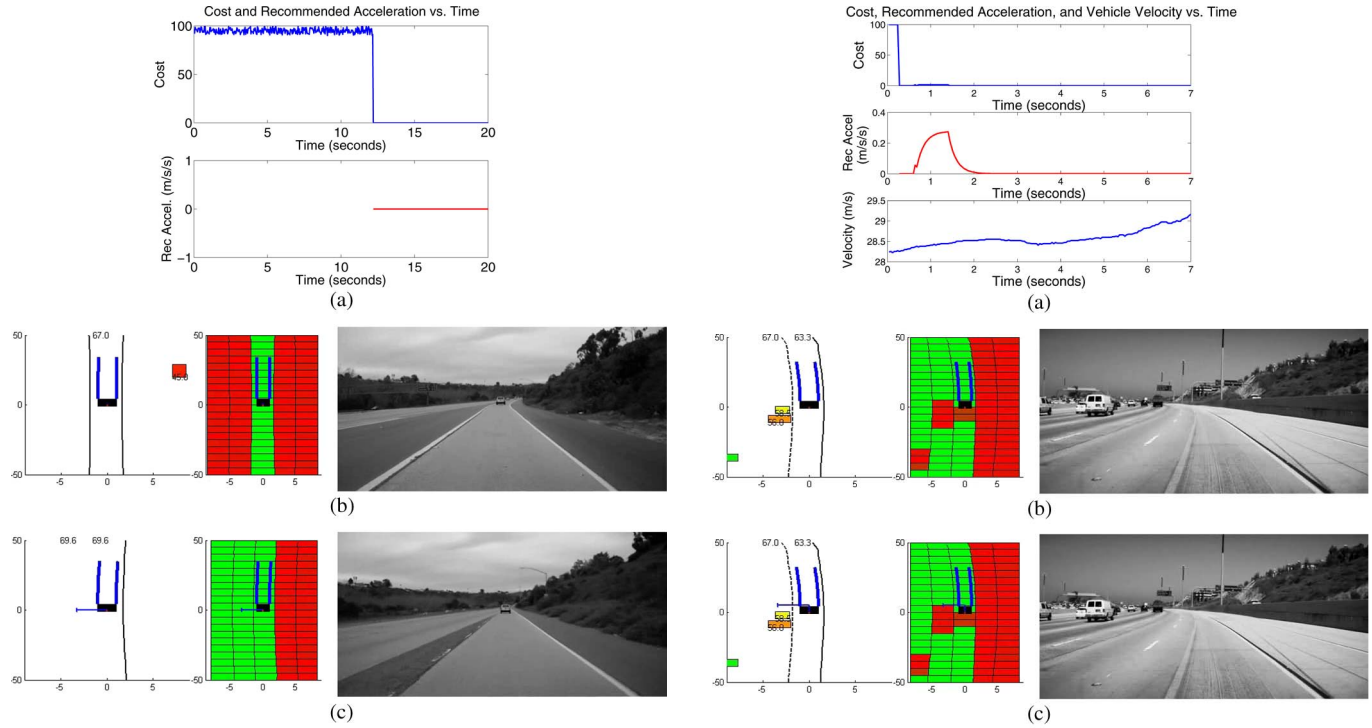


Fig. 7. We demonstrate the system recommendations in a merge scenario with no pertinent vehicles in the surround. (a) We plot the cost of merging to left versus time and the system's recommended acceleration during the sequence. Given the lack of surround vehicles, the only on-road constraints to consider are the lane markings. (b) At the beginning of the merge sequence, the DPDM cells to the left of the ego-vehicle have a low probability of drivability, because of the solid lane boundary. This coincides with a very high cost and a null recommendation to merge. (c) After the lane boundary has transitioned to dashed markings, the system recommends a constant-velocity merge, with very low cost.

Fig. 8. We demonstrate the system recommendations during a merge scenario that requires acceleration. (a) We plot the cost, recommended acceleration, and vehicle velocity versus time during the merge sequence. (b) At the beginning of the sequence, the DPDM cells to the left carry a low probability of drivability, as there is a vehicle in the left blind spot. (c) As the sequence progresses, the system recommends an acceleration in order to create a safe distance between the ego-vehicle and vehicle to the left. (d) Once the ego-vehicle has adequately accelerated, the recommended acceleration drops to 0.

distance from the ego-vehicle. In free-flow traffic, the ego-vehicle typically travels at the driver's preferred speed, unconstrained by congestion.

Fig. 6(b) shows a histogram of the recommended accelerations for free-flow highway lane changes, whereas Table III presents statistics on the same data set. A large portion of recommendations involve deceleration in free-flow traffic, mainly to change lanes behind slower-moving vehicles. We note that the average speed during free-flow highway segments is 29 (m/s), or roughly 65 mi/h, which is the speed limit on southern California highways. There is also a peak in the histogram of recommended accelerations at 0 $(\mathrm{m/s^2})$, for constant-velocity lane changes.

Fig. 9 examines a sequence during which the system recommends a deceleration in order to change into the right lane. Early in the sequence, there is a slower vehicle in the right lane, ahead of the ego-vehicle. The lowest cost maneuver to change into the right lane is a deceleration, as shown early in Fig. 9(a). As the ego-vehicle passes the slower vehicle, the lowest cost maneuver becomes a constant-velocity lane change. The slower-moving vehicle exits the freeway and does not show up in the DPDM in Fig. 9(b).

TABLE III
FREE-FLOW HIGHWAY LANE CHANGES, N = 25

Lane-Change Attribute	Measurement
Segment Length, Mean	11.5 seconds
Segment Length, Std. Dev	8.9 seconds
Ego-vehicle Speed, Mean	29.0 $\frac{m}{s}$
Ego-vehicle Speed, Std. Dev	2.1 $\frac{m}{s}$
Ego-vehicle Acceleration, Mean	.06 $\frac{m}{s^2}$
Ego-vehicle Acceleration, Std. Dev	.56 $\frac{m}{s^2}$
Left Lane Changes	52%
Right Lane Changes	48%
Recommended Deceleration	40.3%
Recommended Acceleration	29.7%
Recommend Maintain Current Velocity	30.0%

C. Highway Lane Changes: Dense Traffic

We evaluate the lane-change recommendations using 25 segments captured in dense highway traffic. Fig. 6(c) plots the histogram of accelerations for changing lanes in dense traffic, whereas Table IV provides statistics. The average vehicle speed in dense traffic is slower than in free-flow traffic. A large portion of lane change recommendations include acceleration, due to the high relative velocities found in dense traffic, due to lane-specific congestion.

We examine a segment from the dense traffic data set, during which the system makes recommendations to change into the

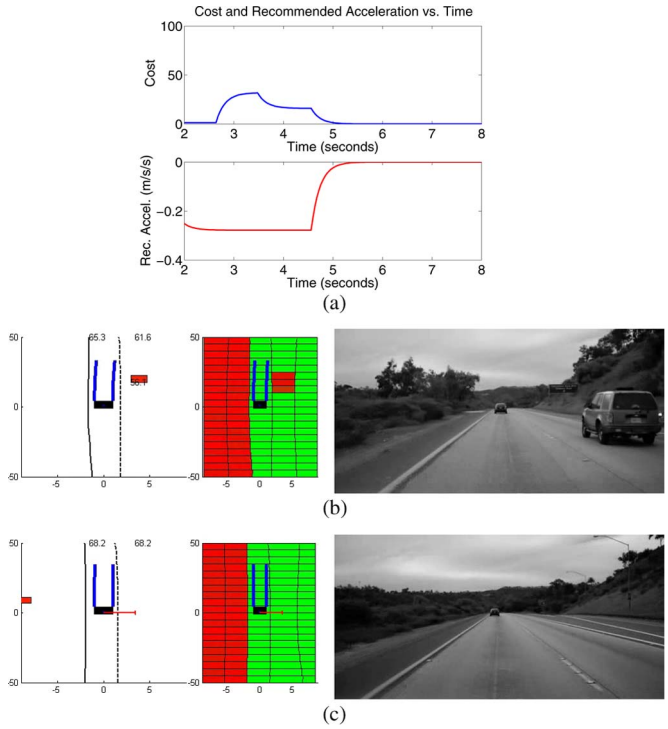


Fig. 9. Lane-change recommendations as the (right) ego-vehicle overtakes a slower vehicle in free-flowing traffic. (a) At the beginning of the sequence, the right-lane recommendation involves deceleration with some cost. As the ego-vehicle overtakes the slower vehicle, the cost and required acceleration both go to zero, and a constant-velocity lane change is possible. (b) The slower vehicle is in front of the ego-vehicle in the right lane. (c) The ego-vehicle has overtaken the slower vehicle, which has subsequently exited from the highway.

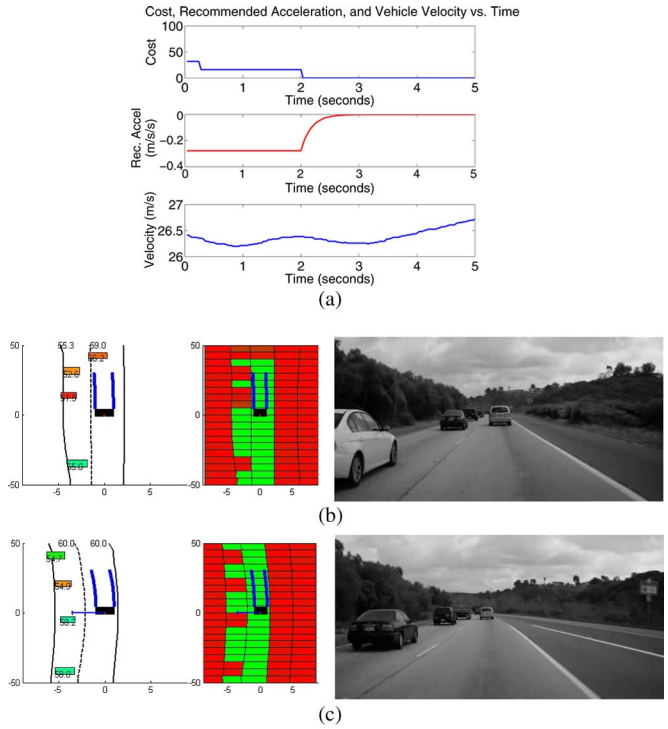


Fig. 10. Dense highway segment. (a) Cost, recommended acceleration, and velocity versus time. (b) DPDM from the beginning of the segment. (c) DPDM from the end of the segment.

TABLE IV
DENSE HIGHWAY LANE CHANGES, N = 25

Lane-Change Attribute	Measurement
Segment Length, Mean	6.2 seconds
Segment Length, Std. Dev	3.2 seconds
Ego-vehicle Speed, Mean	25.9 $\frac{m}{s}$
Ego-vehicle Speed, Std. Dev	2.6 $\frac{m}{s}$
Ego-vehicle Acceleration, Mean	-.03 $\frac{m}{s^2}$
Ego-vehicle Acceleration, Std. Dev	.67 $\frac{m}{s^2}$
Left Lane Changes	58.3%
Right Lane Changes	41.7%
Recommended Deceleration	30.3%
Recommended Acceleration	43.1%
Recommend Maintain Current Velocity	26.7%

left lane. Fig. 10 plots the cost, recommended acceleration, and DPDM plots from this sequence. At the beginning of the segment, the system recommends a deceleration, in order to fit behind a slower vehicle in the left lane. We note that the ego-vehicle's velocity stays roughly constant for the first 3 s of the segment. As the ego-vehicle passes the slower vehicle, the recommendation becomes a constant-velocity maneuver to change into the left lane.

D. Urban Lane Changes

We evaluate the performance of the lane-change recommendations using 50 instances captured in urban driving scenarios. Fig. 6(d) plots the histogram of lane-change recommendations, whereas Table V provides statistics on the data set. In urban driving, recommendations for deceleration, acceleration, and constant-velocity lane changes all occur with roughly equal frequency. This is due to the fact that overall, urban speeds are slower, and urban driving features a greater range of vehicle speeds, as shown in Table V. Urban driving also features discrete-like stoplights, intersections, and driveways.

TABLE V
URBAN LANE CHANGES, N = 50

Lane-Change Attribute	Measurement
Segment Length, Mean	7.9 seconds
Segment Length, Std. Dev	5.3 seconds
Ego-vehicle Speed, Mean	16.5 $\frac{m}{s}$
Ego-vehicle Speed, Std. Dev	4.5 $\frac{m}{s}$
Ego-vehicle Acceleration, Mean	-.1 $\frac{m}{s^2}$
Ego-vehicle Acceleration, Std. Dev	.9 $\frac{m}{s^2}$
Left Lane Changes	56%
Right Lane Changes	44%
Recommended Deceleration	34.5%
Recommended Acceleration	35.5%
Recommend Maintain Current Velocity	30.0%

Fig. 11 shows an urban sequence, during which the LCM system recommends acceleration to change into the left lane. At the beginning of the sequence, the system recommends a constant-velocity change to the left lane. As the ego-vehicle slows, we see the cost contribution increase due to required dynamics, and the required acceleration to change lanes increases. As the ego-vehicle's speed increases, the required acceleration and cost both fall to 0, and the system recommends a constant-velocity lane change.

VI. CONCLUDING REMARKS AND FUTURE WORK

In this paper, we have introduced a novel compact representation for the on-road environment, i.e., the DPDM, and demonstrated its utility in driver assistance during LCMs. The

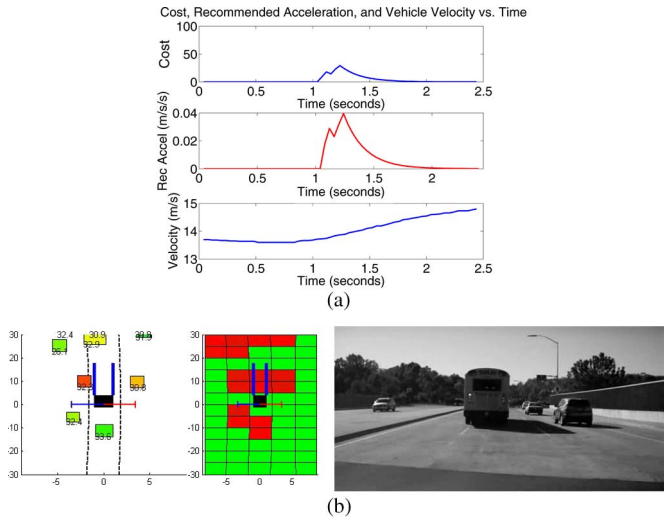


Fig. 11. Urban sequence during which the system recommends acceleration to change into the left lane. (a) We plot cost, recommended acceleration, and velocity versus time. (b) A vehicle approaches the ego-vehicle with positive relative velocity in the left lane. As the ego-vehicle speeds up, the recommended acceleration and cost go to 0.

DPDM interprets the vehicle's surround as a map of probabilities and geometrically adapts to the lane geometry. The DPDM compactly encodes spatial, dynamic, and legal information from a variety of sensing modalities. We efficiently compute minimum-cost maneuvers by formulating maneuver assistance as a dynamic programming problem over the DPDM. In this paper, we have demonstrated the utility of the DPDM for driver assistance during merges and lane changes in highway and urban driving. The full system has been implemented in C++ and runs in real time. An HMI concept for relaying the maneuver recommendations to the driver via head-up display, as shown in Fig. 3, has been also developed. It uses simplified iconography and intuitive colors to assist the driver during merge and lane change maneuvers. On-road user interactivity studies are under way.

ACKNOWLEDGMENT

The authors would like to thank colleagues from Laboratory for Intelligent and Safe Automobiles, University of California, San Diego, La Jolla, CA, USA, and Audi Electronics Research Laboratory, Belmont, CA, for their support and collaboration.

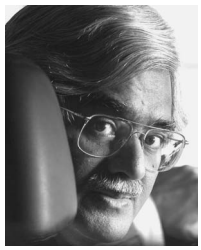
REFERENCES

[1] *Traffic Safety Facts*, Department of Transportation, National Highway Traffic Safety Administration, Washington, DC, USA, 2011.
[2] *General Estimates System*, National Automotive Sampling System, National Highway Traffic Safety Administration, Washington, DC, USA, 2007.
[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," *J. Safety Res.*, vol. 35, no. 1, pp. 107–114, 2004.
[4] S. Sivaraman and M. M. Trivedi, "Active learning for on-road vehicle detection: A comparative study," in *Mach. Vis. Appl*, 2011, Special Issue on Car Navigation and Vehicle Systems. DOI: 10.1007/s00138-011-0388-y.
[5] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1331–1342, Dec. 2011.
[6] S. Sivaraman and M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773–1795, Dec. 2013.
[7] X. Mao, D. Inoue, S. Kato, and M. Kagami, "Amplitude-modulated laser radar for range and speed measurement in car applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 408–413, Mar. 2012.
[8] S. Sato, M. Hashimoto, M. Takita, K. Takagi, and T. Ogawa, "Multilayer lidar-based pedestrian tracking in urban environments," in *Proc. IEEE IV Symp.*, Jun. 2010, pp. 849–854.
[9] R. Schubert, K. Schulze, and G. Wanielik, "Situation assessment for automatic lane-change maneuvers," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 607–616, Sep. 2010.
[10] J. D. Alonso, E. R. Vidal, A. Rotter, and M. Muhlenberg, "Lane-change decision aid system based on motion-driven vehicle tracking," *IEEE Trans. Veh. Technol.*, vol. 57, no. 5, pp. 2736–2746, Sep. 2008.
[11] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving?" in *Proc. IEEE CVPR*, Providence, RI, USA, Jun. 2012, pp. 3354–3361.
[12] T.-N. Nguyen, B. Michaelis, A. Al-Hamadi, M. Tornow, and M. Meinecke, "Stereo-camera-based urban environment perception using occupancy grid and object tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 154–165, Mar. 2012.
[13] F. Homm, N. Kaempchen, J. Ota, and D. Burschka, "Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection," in *Proc. IEEE IV Symp.*, Jun. 2010, pp. 1006–1013.
[14] A. Azim and O. Aycard, "Detection, classification and tracking of moving objects in a 3D environment," in *Proc. IEEE IV Symp.*, Jun. 2012, pp. 802–807.
[15] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière, "Bayesian occupancy filtering for multitarget tracking: An automotive application," *Int. J. Robot. Res.*, vol. 25, no. 1, pp. 19–30, Jan. 2006.
[16] E. Richter, P. Lindner, G. Wanielik, K. Takagi, and A. Isogai, "Advanced occupancy grid techniques for lidar based object detection and tracking," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Oct. 2009, pp. 1–5.
[17] M. Perrollaz, J.-D. Yoder, A. Nè andgre, A. Spalanzani, and C. Laugier, "A visibility-based approach for occupancy grid computation in disparity space," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1383–1393, Sep. 2012.
[18] T. Yapo, C. Stewart, and R. Radke, "A probabilistic representation of lidar range data for efficient 3D object detection," in *Proc. IEEE CVPRW*, Jun. 2008, pp. 1–8.
[19] J. Adarve, M. Perrollaz, A. Makris, and C. Laugier, "Computing occupancy grids from multiple sensors using linear opinion pools," in *Proc. IEEE ICRA*, May 2012, pp. 4074–4079.
[20] J. Moras, V. Cherfaoui, and P. Bonnifait, "Credibilist occupancy grids for vehicle perception in dynamic environments," in *Proc. IEEE ICRA*, May 2011, pp. 84–89.
[21] J. Moras, F. S. A. Rodriguez, V. Drevelle, G. Dherbomez, V. Cherfaoui, and P. Bonnifait, "Drivable space characterization using automotive lidar and georeferenced map information," in *Proc. IEEE IV Symp.*, Jun. 2012, pp. 778–783.
[22] F. Oniga and S. Nedevschi, "Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection," *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1172–1182, Mar. 2010.
[23] J. McCall and M. Trivedi, "Video-based lane estimation and tracking for driver assistance: Survey, system, and evaluation," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20–37, Mar. 2006.
[24] 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," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 737–747, Jun. 2012.
[25] W. He, X. Wang, G. Chen, M. Guo, T. Zhang, P. Han, and R. Zhang, "Monocular based lane-change on scaled-down autonomous vehicles," in *Proc. IEEE IV Symp.*, Jun. 2011, pp. 144–149.
[26] T. Lee, B. Kim, K. Yi, and C. Jeong, "Development of lane change driver model for closed-loop simulation of the active safety system," in *Proc. 14th IEEE ITSC*, Oct. 2011, pp. 56–61.
[27] W. Yao, H. Zhao, F. Davoine, and H. Zha, "Learning lane change trajectories from on-road driving data," in *Proc. IEEE IV Symp.*, Jun. 2012, pp. 885–890.
[28] C. Rodemerk, S. Habenicht, A. Weitzel, H. Winner, and T. Schmitt, "Development of a general criticality criterion for the risk estimation of driving situations and its application to a maneuver-based lane change assistance system," in *Proc. IEEE IV Symp.*, Jun. 2012, pp. 264–269.
[29] G. Xu, L. Liu, Y. Ou, and Z. Song, "Dynamic modeling of driver control strategy of lane-change behavior and trajectory planning for collision prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1138–1155, Sep. 2012.

[30] R. Schubert and G. Wanielik, "Empirical evaluation of a unified Bayesian object and situation assessment approach for lane change assistance," in *Proc. 14th IEEE ITSC*, Oct. 2011, pp. 1471–1476.
[31] R. Schubert, "Evaluating the utility of driving: Toward automated decision making under uncertainty," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 354–364, Mar. 2012.
[32] J. Hillenbrand, A. M. Spieker, and K. Kroschel, "A multilevel collision mitigation approach—Its situation assessment, decision making, and performance tradeoffs," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 4, pp. 528–540, Dec. 2006.
[33] S. Habenicht, H. Winner, S. Bone, F. Sasse, and P. Korzenietz, "A maneuver-based lane change assistance system," in *Proc. IEEE IV Symp.*, Jun. 2011, pp. 375–380.
[34] A. Doshi, B. Morris, and M. M. Trivedi, "On-road prediction of driver's intent with multimodal sensory cues," *IEEE Pervasive Comput.*, vol. 10, no. 3, pp. 22–34, Jul.–Sep. 2011.
[35] V. Milanes, J. Godoy, J. Villagra, and J. Perez, "Automated on-ramp merging system for congested traffic situations," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 500–508, Jun. 2011.
[36] D. Marinescu, J. Curn, M. Bouroche, and V. Cahill, "On-ramp traffic merging using cooperative intelligent vehicles: A slot-based approach," in *Proc. 15th IEEE ITSC*, Sep. 2012, pp. 900–906.
[37] S. Matzka, A. Wallace, and Y. Petillot, "Efficient resource allocation for attentive automotive vision systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 859–872, Jun. 2012.
[38] S. Sivaraman and M. Trivedi, "Integrated lane and vehicle detection, localization, and tracking: A synergistic approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 906–917, Jun. 2013.
[39] U. Franke, C. Rabe, H. Badino, and S. Gehrig, "6D-vision: Fusion of stereo and motion for robust environment perception," in *Proc. DAGM*, 2005, pp. 216–223.
[40] R. J. Elliot, L. Aggoun, and J. B. Moore, *Hidden Markov Models: Estimation and Control*. New York, NY, USA: Springer-Verlag, 1995.
[41] P. G. Hoel, S. C. Port, and C. J. Stone, *Introduction to Stochastic Processes*. Boston, MA, USA: Houghton Mifflin, 1972.
[42] S. Dasgupta, C. Papadimitriou, and U. Vazirani, *Algorithms*. New York, NY, USA: McGraw-Hill, 2007.
[43] *California Driver Handbook-Safe Driving Practices: Merging and Passing*, California Department of Motor Vehicles, Sacramento, CA, USA, 2012.

Sayanan Sivaraman (M'07) received the B.S. degree from University of Maryland, College Park, MD, USA, in 2007 and the M.S. and Ph.D. degrees from University of California, San Diego, La Jolla, CA, USA, in 2009 and 2013, respectively, all in electrical engineering, with a focus on intelligent systems, robotics, and controls.

He is with Volkswagen Group of America, Electronics Research Laboratory, Belmont, CA. His research interests include computer vision, machine learning, intelligent vehicles, and mobility.



Mohan Manubhai Trivedi (F'08) received the B.E. degree (with honors) in electronics from Birla Institute of Technology and Science, Pilani, India, in 1974 and the M.S. and Ph.D. degrees in electrical engineering from Utah State University, Logan, UT, USA, in 1976 and 1979, respectively.

He is a Professor of electrical and computer engineering and the Founding Director of the Computer Vision and Robotics Research Laboratory and the Laboratory for Intelligent and Safe Automobiles, University of California, San Diego, La Jolla, CA, USA. He is a coauthor of a number of papers that have won him Best Paper awards. He and his team are currently pursuing research in machine and human perception, machine learning, human-centered multimodal interfaces, intelligent transportation, driver assistance, and active safety systems. His team has played a key role in several major research collaborative initiatives.

Dr. Trivedi received the Distinguished Alumnus Award from Utah State University, the Pioneer Award (Technical Activities) and the Meritorious Service Award from IEEE Computer Society, and the 2013 Outstanding Research Award from IEEE Intelligent Transportation Systems Society. He is on the Board of Governors of IEEE Intelligent Transportation Systems Society and on the Editorial Board of IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He is a Fellow of the International Association for Pattern Recognition and the International Society for Optics and Photonics (SPIE).