



**HAL**  
open science

## Probabilistic Analysis of Dynamic Scenes and Collision Risk Assessment to Improve Driving Safety

Christian Laugier, Igor Paromtchik, Mathias Perrollaz, Yong Mao,  
John-David Yoder, Christopher Tay, Kamel Mekhnacha, Amaury Nègre

► **To cite this version:**

Christian Laugier, Igor Paromtchik, Mathias Perrollaz, Yong Mao, John-David Yoder, et al.. Probabilistic Analysis of Dynamic Scenes and Collision Risk Assessment to Improve Driving Safety. *Its Journal*, 2011, 3 (4), pp.4-19. 10.1109/MITS.2011.942779 . hal-00645046

**HAL Id: hal-00645046**

**<https://inria.hal.science/hal-00645046v1>**

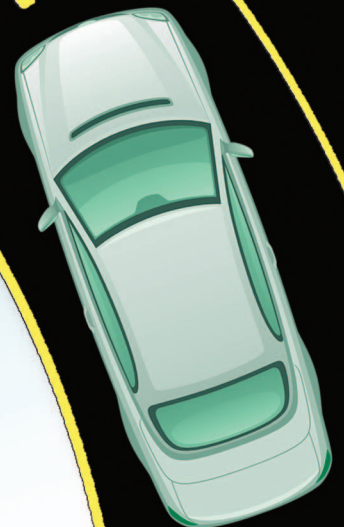
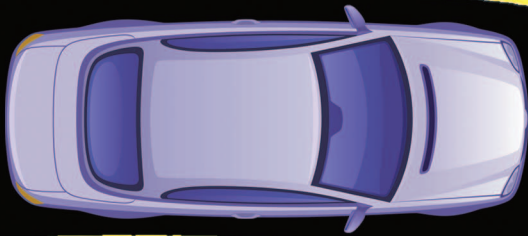
Submitted on 25 May 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

**Abstract**—The article deals with the analysis and interpretation of dynamic scenes typical of urban driving. The key objective is to assess risks of collision for the ego-vehicle. We describe our concept and methods, which we have integrated and tested on our experimental platform on a Lexus car and a driving simulator. The on-board sensors deliver visual, telemetric and inertial data for environment monitoring. The sensor fusion uses our Bayesian Occupancy Filter for a spatio-temporal grid representation of the traffic scene. The underlying probabilistic approach is capable of dealing with uncertainties when modeling the environment as well as detecting and tracking dynamic objects. The collision risks are estimated as stochastic variables and are predicted for a short period ahead with the use of Hidden Markov Models and Gaussian processes. The software implementation takes advantage of our methods, which allow for parallel computation. Our tests have proven the relevance and feasibility of our approach for improving the safety of car driving.

**Index Terms**—Intelligent vehicle, sensor fusion, probabilistic filter, collision risk, driver assistance.



**Christian Laugier,  
Igor E. Paromtchik,  
Mathias Perrollaz, and Mao Yong**

*INRIA Grenoble Rhône-Alpes,  
38334 Saint Ismier, France  
Christian.Laugier@inrialpes.fr*

**John-David Yoder**  
*Ohio Northern University, OH, USA*

**Christopher Tay and Kamel Mekhnacha**  
*ProBayes, 38334 Saint Ismier, France*

**Amaury Nègre**  
*CNRS, Laboratory of Informatics of Grenoble, France*

*Digital Object Identifier 10.1109/MITS.2011.942779  
Date of publication: 28 October 2011*

CARS:©ISTOCKPHOTOS.COM/ELENA BELYAKOVA

# Probabilistic Analysis of Dynamic Scenes and Collision Risks Assessment to Improve Driving Safety

## I. Introduction

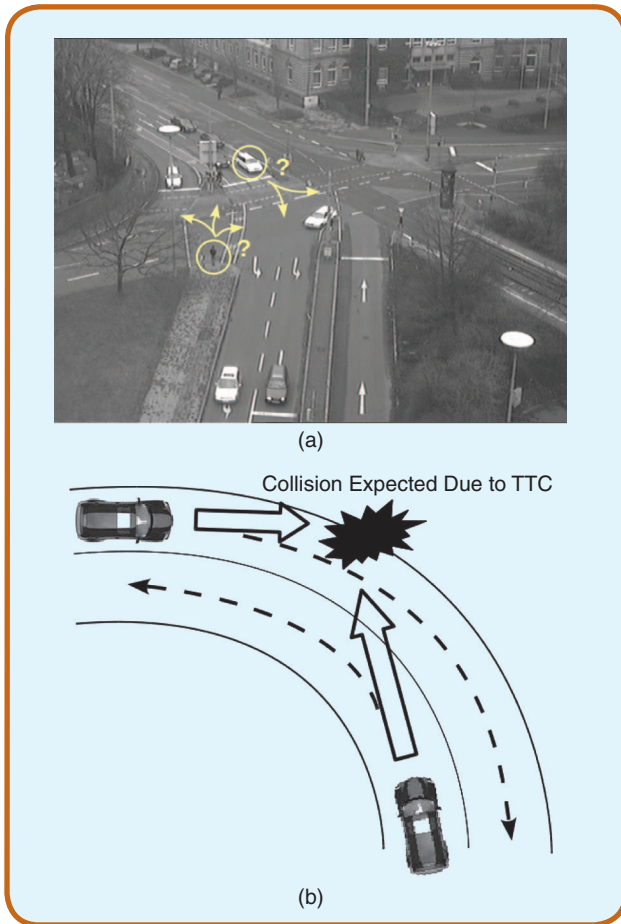
### A. Automotive and Technological Context

Driving a car assumes some level of risk of collision in any traffic scenario. The modern technologies help mitigate the effects of accidents (e.g. seat belts, airbags, safety glass, energy absorbing frames) or reduce their likelihood (e.g. anti-lock braking system, dynamic stability control). Concurrently, the exploitation of the synergies between mechatronics, drive-by-wire, perception, real-time data processing and communication facilitates the risk management by traffic participants toward *zero-collision driving*. The key problem is to correctly interpret the traffic scene by means of processing information from a variety of sensors.

Improvement of driving safety remains a highly relevant topic, with significant accomplishments being reported, from obstacle detection and driver warning to active response leading to modifying the driving parameters when a collision becomes imminent. Automated maneuvering represents a strong advantage over manual driving since it reduces the required reaction time (in comparison to a human driver) to avoid collisions or mitigate their impact. Various successful automated maneuvers were reported, using world-first prototypes such as: an automated platooning system in the scope of the PATH Program [1], or the automated parking systems at INRIA [2, 3]. Recent promising results in more complex scenarios are the CityCars concept [4], the DARPA Urban Challenge [5, 6], or the Google Cars [7].

Another approach to reduce driving accidents is to assist the driver in avoiding collisions, e.g. a parking assistance by producing audible alarms to the human driver while the car approaches obstacles [8]. More advanced systems anticipate the car motion on the parking lot and display the situation to the driver during a human-driven parking [9]. Such warning signals along with the modification of velocity and steering angle during the maneuvers assist the driver to smoothly perform the maneuver while avoiding collisions. Other devices allow for alerting the driver when drifting out of the traffic lane, or provide an adaptive cruise control function [10].

Assessing the risks of collision in real driving situations is a challenging problem. At present, the common measure of collision risk is time-to-collision (TTC) [11]. It is calculated by assuming all objects being detected, their positions being computed and their velocities remaining constant relative to the ego-vehicle. For a time horizon within a second on a straight road, the TTC is used effectively in accident mitigation systems (e.g.



**FIG 1** Limitation of TTC as a risk measure: (a) under-estimation of collision risk at intersection with stopped cars, (b) over-estimation of collision risk on a curved road.

pretension of seat belts) just before impact. However, it becomes less efficient for the time horizon of a few seconds. Fig. 1 illustrates its limitation as the sole estimate of risk. If all of the cars are stopped at the intersection, as shown in Fig. 1-a, the TTC is calculated as infinite. Thus, a case of relatively high-risk (many accidents occur in intersections) is estimated as very low-risk, i.e. this is a false negative. The situation on a curved road in Fig. 1-b gives an example of the TTC calculated as very low, implying a very high risk. But because the most likely result is the both cars moving in their lanes, this case is a false positive, i.e. reporting very high risk when in fact the risk is relatively low.

The above scenarios indicate that TTC alone is insufficient as a risk indicator for managing complex situations. The road context (road shape, speed limit, intersection layout, etc.) can add relevant information. Predicting the future actions (behaviors) of other traffic participants, like a car or a pedestrian in Fig. 1-a, can further improve estimation of collision risk. Since these future behaviors can never be known exactly in advance, their probabilistic prediction is required [12].

## B. State of the Art

Much research has focused on directly modeling and detecting objects in the scene and various approaches have been proposed, depending on the sensor involved: telemeter like radar [13] or laser scanner [14], cooperative detection systems [15], or vision systems. Most monocular vision approaches suppose recognition of specific objects, like vehicles or pedestrians, using a model which is generally built through statistical learning [16]. Stereo-vision is particularly suitable for generic obstacle detection [17, 18] and object classification [19], because it provides a three-dimensional representation of the road scene.

The computational complexity has been a critical feature of stereo-vision, but recent algorithms, like [20], as well as progress in Graphics Processing Unit (GPU) computing [21] now allow for processing of stereo images in real time. Advanced approaches combine stereo-vision and movement to make perception more robust [22]. It should also be noted that most of the successful vehicles in the DARPA Urban Challenge used a three-dimensional laser scanner Velodyne to assist in finding obstacles [23].

Many approaches rely on sensor fusion to attain sufficient reliability for automotive applications, with some methods being designed for particular sensors [24, 25], or offering a generic framework [26]. Most of them are at the object level and must therefore deal with the difficult task of data association. Rather than start with obstacle models, various approaches take advantage of a grid representation of the scene [27, 28, 29]. In order to work efficiently with occupancy grids, we have previously developed a probabilistic framework with the Bayesian Occupancy Filter (BOF) [30, 31, 32], which provides filtering, data fusion, and velocity estimation capabilities while allowing for parallel computation. The Fast Clustering and Tracking Algorithm (FCTA) [33] is then used to identify and track individual objects. The BOF is designed with the intent of its implementation in hardware as a system-on-chip. Like other grid based approaches, the BOF framework performs sensor fusion at the cell level [30].

Collision risk assessment employs the information from sensor fusion and uses models in order to estimate potential threats [34, 35, 36]. The information about the road geometry and the communication between the vehicles and with the infrastructure provide to improve risk assessment [37, 38]. In addition to the knowledge about an object detected at a certain location at a specific time in the traffic scene, the prediction of its likely future behavior leads to more adequate interpretation of its possible impact on the ego-vehicle [12, 39, 40, 41].

## C. Problem Statement and Approach

This article focuses on the probabilistic modeling and analysis of dynamic traffic scenes by means of *sensor data*

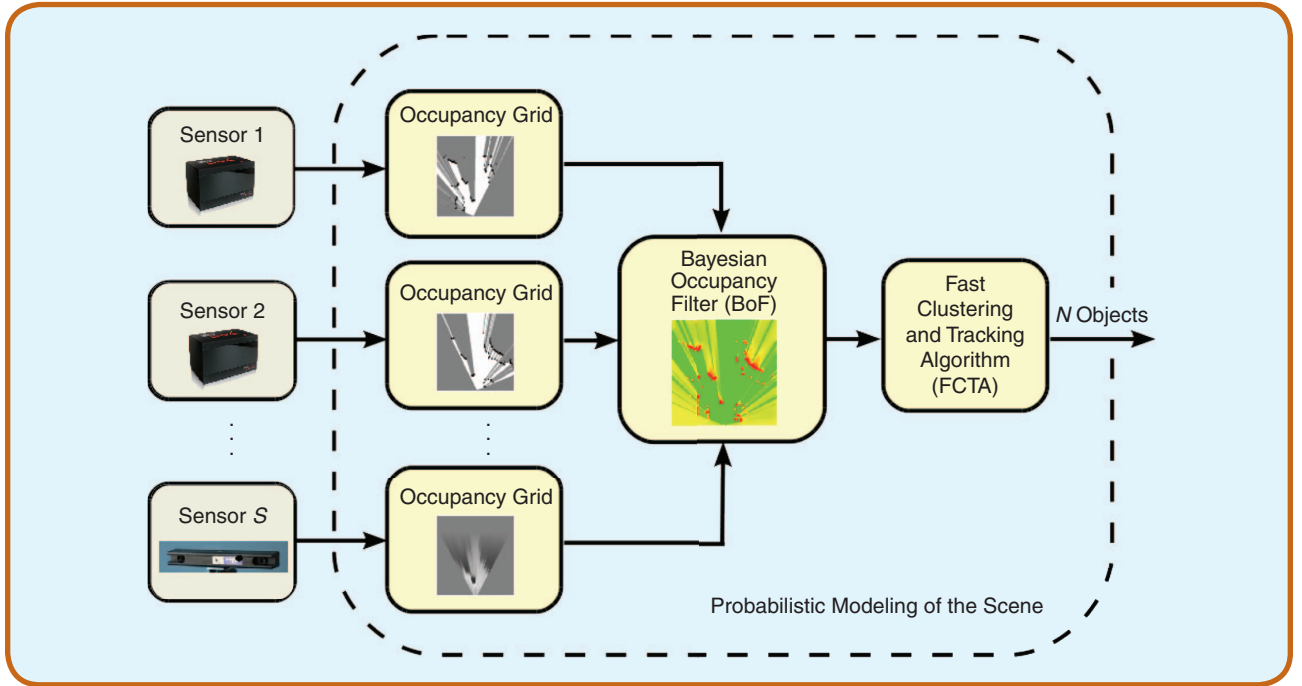


FIG 2 Architecture of the environment modeling module.

fusion from on-board sensors and continuous assessment of collision risk [42]. Among the relevant sensors for monitoring the local environment, we use stereo-vision and lidars, mounted on-board of the ego-vehicle [45]. The environment is represented by a grid, and the fusion of sensor data is accomplished by means of the BOF [50, 51]. The BOF evaluates probabilities of both *cell occupancy* and *cell velocity* for each cell in a four-dimensional spatio-temporal grid. The monitoring of traffic scenes includes detection and tracking of objects by the FCTA [33]. The collision risks are considered as stochastic variables. Hidden Markov Model (HMM) and Gaussian process (GP) are used to estimate and predict collision risks and the likely behaviors of multiple dynamic agents in road scenes.

The main contribution of this article is to present two main components of our conceptual framework: traffic scene modeling and collision risk assessment for the ego-vehicle. To deal with uncertainties (e.g. possible noise on sensor data) and with the complexity of road scenes, these functionalities have been developed in our probabilistic framework ProBT<sup>®</sup>. The corresponding methods are implemented into the software that runs on our experimental platform on a Lexus car and Toyota's driving simulator allowing for damage-free collision situations.

The article is organized as follow: Section II describes our approach to model and monitor the dynamic traffic environment, Section III explains the approach to collision risk assessment, Section IV discusses our experimental results and Section V lists conclusions and indicates our ongoing work.

## II. Online Traffic Scene Modeling and Monitoring

An overview of our environment-modeling module is shown in Fig. 2. The inputs to this module are sensor data. The combined use of two lidars and stereo-vision helps mitigate uncertainty and allows for detection of partially occluded objects. The output of the module is an estimation of the position, velocity and associated uncertainty of each observed object, which are used as input to the risk assessment module.

### A. Occupancy Grid from Lidar

An occupancy grid from lidar data is constructed using a beam-based probabilistic sensor model, similar to that described in [28], where each beam in a lidar detection frame is considered as independent from the other beams. Static and dynamic entities of the environment are separated, by using a local-SLAM based algorithm. Let  $m$  denote the map,  $Z_{0:t} = (z_0, \dots, z_t)$  be the sensor observations, where  $z_t$  is the frame of observation at time step  $t$ ,  $U_{1:t} = (u_1, \dots, u_t)$  denote the odometer data and  $X_{0:t} = (x_0, \dots, x_t)$  be the vehicle states. The objective of a full sequential SLAM algorithm is to estimate the posterior  $P(m, x_t | Z_{0:t}, U_{1:t}, x_0)$ . Since we build a map of a local area, which moves with the ego-vehicle, we are not concerned with the precision of the vehicle's global states. Therefore, we use a maximum likelihood localization and apply a log-odds filtering scheme to update the map. The maximum likelihood state can be estimated for the localization step as:

$$x_t = \operatorname{argmax}_{x'_t} P(z_t | m, x'_t) P(x'_t | x_{t-1}, u_t), \quad (1)$$



where  $x'_t$  denotes a state sample at time  $t$ . Then, based on the new state of the vehicle, the occupancy map is updated. Let

$$l(m_i | X_{0:t}, Z_{0:t}) = \log \frac{P(m_i | X_{0:t}, Z_{0:t})}{P(-m_i | X_{0:t}, Z_{0:t})} \quad (2)$$

denote the log-odds value of a cell  $m_i$  in  $m$ . The update formula can be written as:

$$l(m_i | X_{0:t}, Z_{0:t}) = l(m_i | x_t, z_t) - l(m_i) + l(m_i | X_{0:t-1}, Z_{0:t-1}), \quad (3)$$

where  $l(m_i)$  is the prior value, and  $l(m_i | x_t, z_t)$  is obtained from the beam-based inverse sensor model.

Given the occupancy grid map and the current state of the vehicle, the laser impacts generated from stationary objects or moving objects can be discriminated. For this purpose, clusters are created from connected regions of the grid, and clusters with a high average occupancy probability are classified as static. Additionally, while using a multi-layer laser scanner, only cells containing multiple laser impacts are considered as occupied. This intends to remove laser hits on the road surface.

### B. Occupancy Grids from Stereo-Vision

Our stereoscopic sensor is equipped with two cameras in a “rectified” geometrical configuration, where the image planes are assumed to be perfectly parallel and aligned.



**FIG 3** Computation of the occupancy grid in the u-disparity plane: (a) left image from the stereo camera, (b) obstacle u-disparity image, (c) road u-disparity image, (d) occupancy grid in the u-disparity plane.

To compute the disparity map, we use a double correlation framework, which exploits different matching hypotheses for vertical and horizontal objects [44, 45]. Horizontal hypotheses are generated by applying an homography to the right image of the pair, according to the road surface. This method provides an immediate classification of pixels during the matching process, resulting in two disparity images containing disparity pixels from obstacles and the road surface.

### The u-Disparity Representation

The u-disparity (similar to v-disparity [17]) is computed by projecting the disparity map along the columns with accumulation, see an example in Fig. 3. If the disparity map provides a representation of the scene in the disparity space, then the u-disparity representation is equivalent to a bird-eye view in this space, as illustrated in Fig. 3-b. Vertical objects appear as portions of lines in this image, e.g. the rear of the vehicle.

Working in the u-disparity plane has three main advantages. First, it allows us to make use of equally-spaced measurement points, with constant pixel-wise uncertainty. By contrast, moving to a Cartesian space before processing the data would give a varying density of measurements. Second, the u-disparity calculation is computationally efficient and allows for highly parallel processing. Third, it allows for optical directions to be seen as parallel during the calculation. In fact, a set of rays belonging to the same vertical plane is represented by a column in the u-disparity image. Consequently, it allows us to consider the visible and occluded portions of the image directly, similar to the common approaches used for range finders.

### Occupancy Grid in u-Disparity

To benefit from the above advantages of the u-disparity representation, we compute the occupancy grid in the u-disparity plane. This type of approach has been studied in [46], but our approach is novel in providing a probabilistic management of the visible and occluded areas of the scene and in using the information given by the road/obstacle pixel classification. Here we give an overview of the approach, while its detailed description can be found in [47].

Let us denote a detection plane in Cartesian coordinates as  $\mathcal{P}$ , which is the support of the grid.  $\mathcal{P}$  is chosen to be parallel to the plane defined by the camera baseline and the optical axes. This configuration provides a direct invertible transformation between a cell  $U = (u, d)$  in the u-disparity plane and a cell  $X \subset \mathcal{P}$ .

For building the grid, our objective is to obtain a probability  $P(O_U)$  for a cell  $U$  being occupied by an obstacle in the u-disparity plane. This probability depends on the visibility,  $V_U$ , and the confidence of observation,  $C_U$ , where  $V_U$  and  $C_U$  are binary random variables (e.g. either it is visible or not).  $P(V_U)$  and  $P(C_U)$  are calculated for a given cell, and the laws of probability are used to obtain the full decomposition of  $P(O_U)$ :

$$P(O_U) = \sum_{v,c} P(V_U = v) P(C_U = c) P(O_U | V_U = v, C_U = c). \quad (4)$$

The probability density function  $P(O_U | V_U, C_U)$  is obtained from  $P(V_U)$  and  $P(C_U)$  and the following boundary conditions:

$$\begin{aligned} P(O_U | V_U = 0, C_U) &= 0.5, \\ P(O_U | V_U = 1, C_U = 1) &= 1 - P_{FP}, \\ P(O_U | V_U = 1, C_U = 0) &= P_{FN}, \end{aligned} \quad (5)$$

where  $P_{FP}$  and  $P_{FN}$  are respectively the probability of a false positive or a false negative in the matching process. These are assumed to be constant and known empirically. Tuning these parameters sets the confidence we have in the stereoscopic data. While the range of occupancy values is reduced as these parameters increase, the overall solution (relative values of one cell compared to another) does not change significantly.

Because of the separation of the disparity image into obstacle and road images, we can further improve the occupancy grid by taking into account the road pixels. Call  $P(T_U)$  the total occupancy probability for cell  $U$ , considering both road and obstacle pixels, and  $R_U$  the binary random variable meaning that cell  $U$  only belongs to the road surface. We begin with the logical assertion that the cell is totally occupied if it is occupied by an obstacle and not by the road surface:

$$P(T_U) = P(O_U)(1 - P(R_U)). \quad (6)$$

To compute  $P(R_U)$ , we consider both obstacle and road pixels. This is because road pixels are often found at the base of obstacles, meaning that  $P(R_U)$  must remain low when  $P(O_U)$  is high. This formulation allows us to be more certain that regions of the image where the road can be seen are not occupied.

The basic application of this algorithm is depicted in Fig. 3. The fronts of obstacles result in (mostly) straight black lines, as shown in Fig. 3-b. The road u-disparity image (c), meanwhile, shows much more detail where there is dense information on the road, such as the crosswalk. You can see that in the occupancy grid in Fig. 3-d where black color represents high probability of occupancy, and white color corresponds to very low probability. This grid maintains strong information about the obstacles (they remain black lines), while a cell is empty (white) in areas where the road was detected. Most areas behind obstacles are unknown, meaning they are assigned a value of  $P(T_U) = 0.5$  and are represented by gray color.

#### Occupancy Grid in Cartesian Space

The Cartesian occupancy grid requires the calculation of those pixels in the u-disparity grid, which affect a given cell of the Cartesian grid. Let us define the surface  $S_U(U)$  of a cell  $U$  as the region of the u-disparity image delimited by the intervals  $[u - 0.5, u + 0.5[$  and  $[d - 0.5, d + 0.5[$ .

Call  $S_X(U) \subset \mathcal{P}$  the image of  $S_U(U)$  in the detection plane  $\mathcal{P}$ . To compute the occupancy grid, the occupancy probability of cell  $U$  is simply attributed to the area  $S_X(U)$  of the detection plane. For short distances from the camera, several pixels can affect the same cell  $X \subset \mathcal{P}$  of the metric grid. The occupancy is estimated according to this set of data by using a *max* estimator, which ensures a conservative estimation of the occupancy probability:

$$P(O_X) = \max \{ P(O_U) | X \in S_X(U) \subset \mathcal{P} \}. \quad (7)$$

The occupancy grid presents strong discretization effects, due to the pixel-level sampling and the disparity estimation on integer values. In order to obtain a smoother and more realistic representation, an image-like filter, based on the convolution with a Gaussian kernel, is used. We compute a Gaussian kernel for each value in the occupancy grid, considering a constant Gaussian kernel in the u-disparity plane. Thanks to the non-constant kernel size, details are preserved at short distances, while smoothness is added at longer distances.

#### C. Bayesian Occupancy Filter (BOF)

The BOF operates with a grid representing the environment. Each cell of the grid contains a probability distribution of the *cell occupancy* and a probability distribution of the *cell velocity*. Given a set of observations, the BOF algorithm updates the estimates of the occupancy and velocity for each cell in the grid. As it is shown in Fig. 4, the inference leads to a Bayesian filtering process. The BOF model is described in detail in [30] and [31].

In this context, the prediction step propagates cell occupancy and antecedent (velocity) distributions of each cell in the grid and obtains the prediction  $P(O_c^t A_c^t)$  where  $P(O_c^t)$  denotes the occupancy distribution and  $P(A_c^t)$  denotes the antecedent (velocity) distribution of a cell  $c$  at time  $t$ . In the estimation step,  $P(O_c^t A_c^t)$  is updated by taking into account the observations  $Z_i^t, i = 1, 2, \dots, S$  yielded by the sensors at time  $t$ :  $\prod_{i=1}^S P(Z_i^t | A_c^t O_c^t)$ , where  $P(Z_i^t | A_c^t O_c^t)$  is the model of sensor  $i$ , in order to obtain the a posteriori state estimate

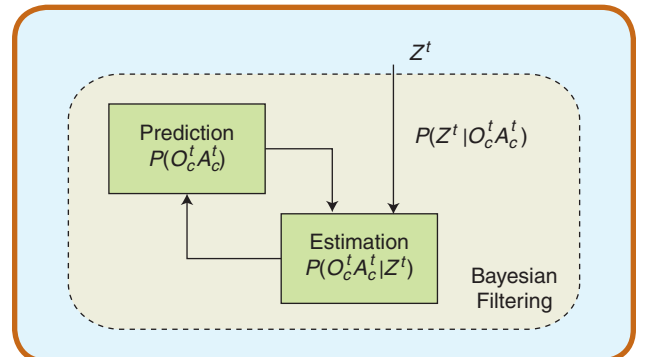
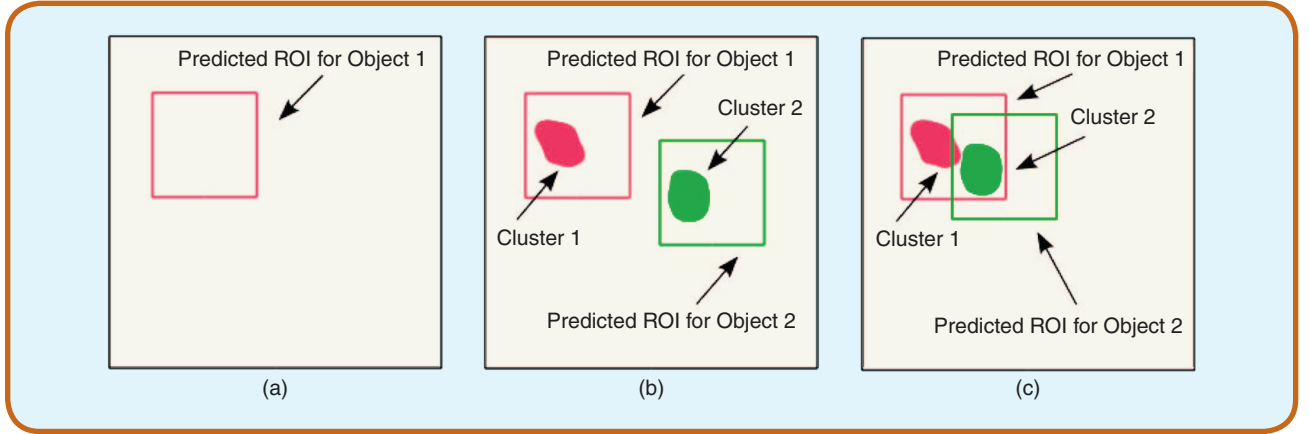


FIG 4 Bayesian filtering in the estimation of occupancy and velocity distributions in the BOF grid.



**FIG 5** Cases of the clustering result, ROIs are predicted from the previous timestep to speed-up data association: (a) no observation, (b) no ambiguity, (c) ambiguous association.

$P(O_c^t A_c^t | [Z_1^t \cdots Z_s^t])$ . This allows us to compute by marginalization  $P(O_c^t | [Z_1^t \cdots Z_s^t])$  and  $P(A_c^t | [Z_1^t \cdots Z_s^t])$ , which will be used for prediction in the next iteration.

#### D. Fast Clustering and Tracking Algorithm (FCTA)

The FCTA works at the level of object representation to track objects [33] and it can be roughly divided into three modules: a clustering module, a data association module, and a tracking and tracks management module. The clustering module takes two inputs: the occupancy/velocity grids estimated by the BOF, and the prediction of the tracker which provides a region of interest (ROI) for each object being tracked. We then try to extract a cluster in each ROI and associate it with the corresponding object. This ROI based approach is designed to improve the computational efficiency of the algorithm.

There could be a variety of cluster extracting algorithms, however, we have found that a simple neighborhood-based algorithm provides satisfactory result. The eight-neighbor cells are connected according to an occupancy threshold and additionally a threshold of the Mahalanobis distance between the velocity distributions is employed to distinguish the objects that are close to each other but move at different velocities.

The output of this clustering module leads to three possible cases, as shown in Fig. 5: (a) no observation, where the object is not observed in the ROI, (b) ambiguity free, where one and only one cluster is extracted and is implicitly associated with the given object, and (c) ambiguity, where the extracted cluster is associated with multiple objects. The data association module is designed to remove the ambiguity. Let  $N_k$  be the number of objects associated with a single cluster. The causes of the ambiguity are further analyzed as twofold: (a) objects are too close to each other and the observed cluster is the union of more than one observations generated by  $N_k$  different real objects, (b)  $N_k$  different objects correspond to a single object in the real world and they should be merged into one.

We employ a re-clustering strategy to the first situation and a cluster merging strategy to the second one. The re-clustering step divides the cluster into  $N_k$  sub-clusters and associates them with the  $N_k$  objects, respectively. Because the number  $N_k$  is known from the prediction step, we apply a K-means based algorithm [48]. The cluster merging step is based on a probabilistic approach. Whenever an ambiguous association  $F_{ij}$  between two tracks  $T_i$  and  $T_j$  is observed, a random variable  $S_{ij}$  is updated to indicate the probability of  $T_i$  and  $T_j$  to be parts of a single object in the real world.

The probability values  $P(F_{ij} | S_{ij})$  and  $P(F_{ij} | \neg S_{ij})$  are parameters of the algorithm which are constant with regard to  $i$  and  $j$ . Similarly, the probability  $P(S_{ij} | \neg F_{ij})$  is updated when no ambiguity between  $T_i$  and  $T_j$  is observed. Then, by thresholding the probability  $P(S_{ij})$ , the decision of merging the tracks  $T_i$  and  $T_j$  can be made by calculating the Mahalanobis distance between them. Now we arrive at a set of clusters which are associated with the objects being tracked without ambiguity. Then, it is straightforward to apply a general tracks management algorithm to create and delete the tracks, and use a Kalman filter to update their states [49].

### III. Collision Risk Estimation

Consider vehicle A and the ego-vehicle B traveling in the same direction on the adjacent lanes, as shown in Fig. 6. The risk of collision has to be estimated for vehicle B. From the driver's viewpoint, the road structure is described implicitly by such maneuvers as: move straight, turn left, turn right or change a lane, which define a set of possible behaviors. Each behavior is represented as a probability distribution over the possible future realizations of the vehicle's paths.

The GP samples for such behaviors as "lane change" and "moving straight" are depicted in Fig. 6, where the dotted lines represent the paths sampled from the GPs. The set of GPs for each feasible behavior and the probability of vehicle A executing a certain behavior, give a probabilistic model of the evolution of vehicle A in the scene. In contrast



to the TTC's linearity assumption about the future paths, we evaluate the collision risk of the intended path of vehicle B against all possible paths of vehicle A. The weights are assigned according to the probabilistic model of the behaviors' evolution of vehicle A, and the collision risk is a weighted sum of the paths leading to a collision.

An overall architecture of our risk estimation module is sketched in Fig. 7. It comprises three sub-modules, such as: driving behavior recognition, driving behavior realization, and collision risk estimation [12, 50].

### A. Behavior Recognition and Modeling

The behavior recognition aims at estimating the probability for a vehicle to perform one of its feasible behaviors. The behaviors are high-level representations of road structure, which contain semantics. The probability distribution over behaviors is obtained by HMM. Our current model includes the following four behaviors: move straight, turn left, turn right, and overtake. The algorithm assigns a label and a probability measure to sequential data, i.e. observations from the sensors. Examples of sensor values are: distance to lane borders, signaling light status or a proximity to an intersection. The objective is to obtain the probability values over behaviors, i.e. the behaviors are hidden variables.

The behavior modeling contains two hierarchical layers, and each layer consists of one or more HMMs. The upper layer is a single HMM where its hidden states repre-

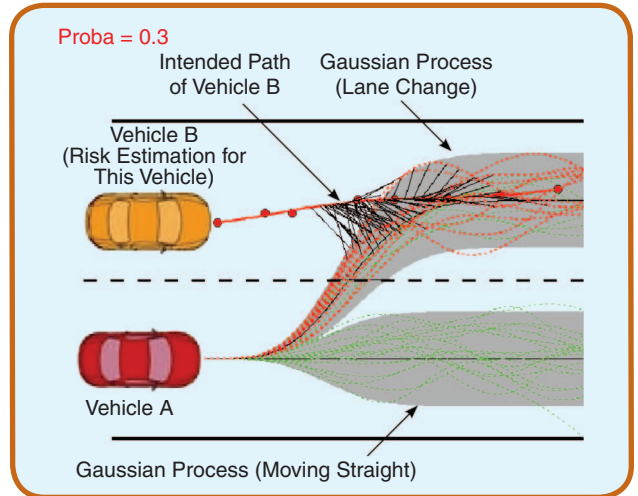


FIG 6 Example of collision risk estimation by predicting the path of vehicle A: sampling from the GPs for two possible behaviors "moving straight" and "lane change."

sent high-level behaviors, such as: move straight, turn left, turn right, and overtake. For each hidden state or behavior in the upper layer HMM, there is a corresponding HMM in the lower layer which represents the sequence of finer state transitions of a single behavior, as depicted in Fig. 8.

Define the following hidden state semantics in the lower layer HMMs for each behavior of the higher layer HMM:

- *Move straight (1 hidden state):* move forward.

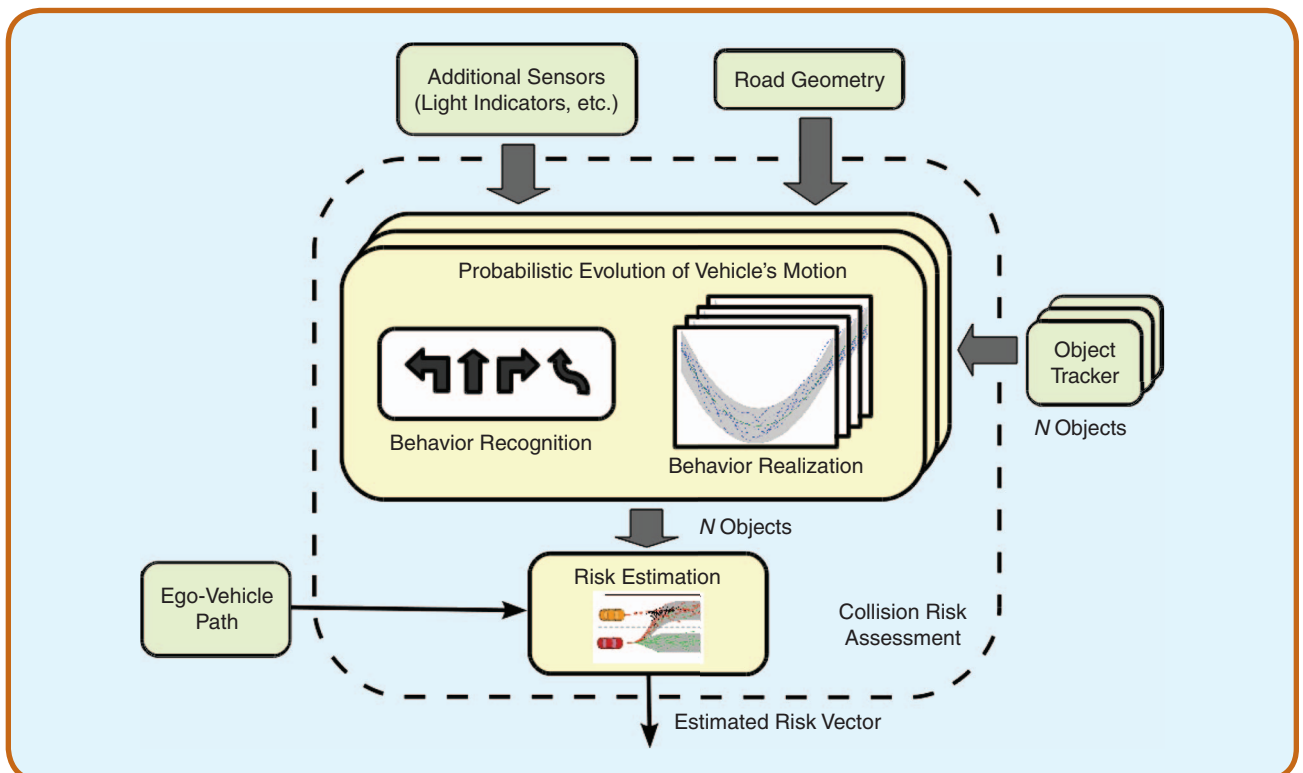
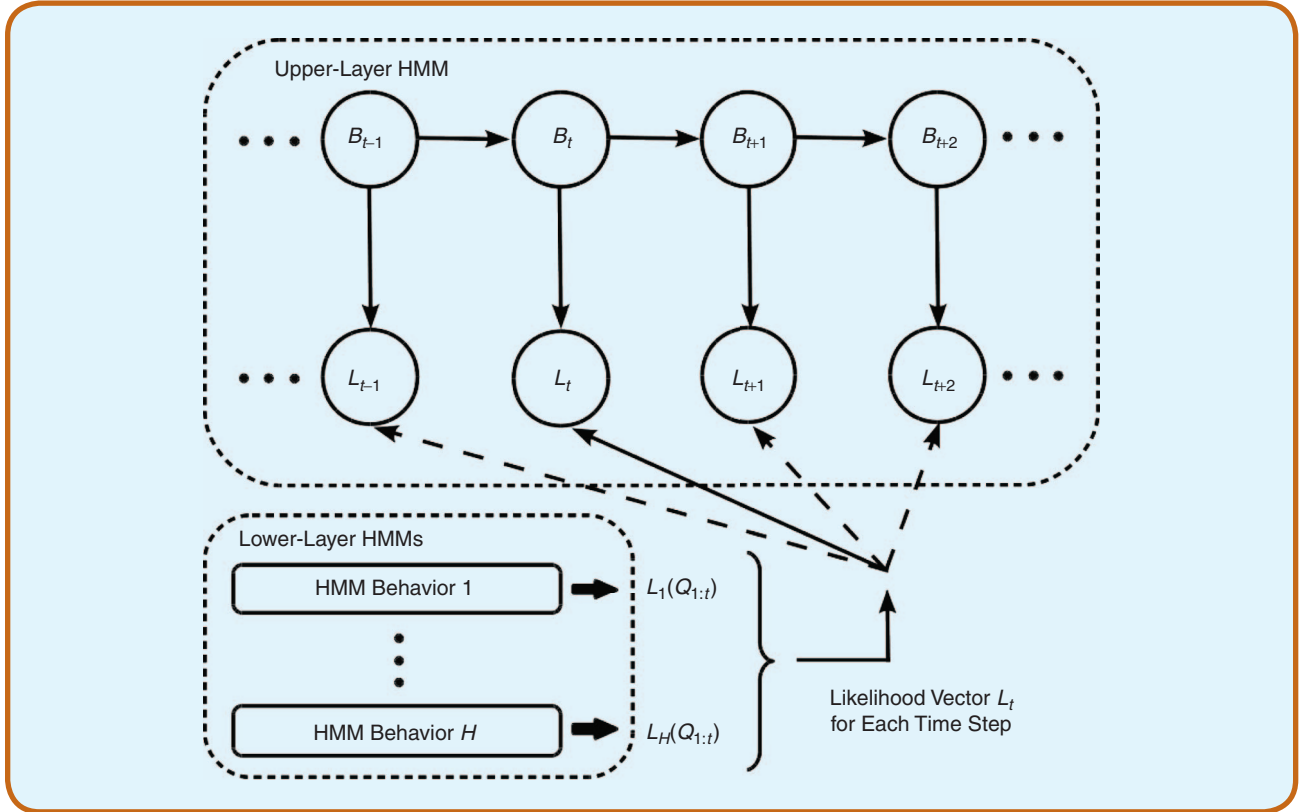


FIG 7 Architecture of the risk assessment module.



**FIG 8** Layered HMM where each lower layer HMM's observation likelihood is the upper layer HMM's observation.

- *Turn left or turn right (3 hidden states)*: Decelerate before a turn, execute a turn, and resume a cruise speed.
- *Overtake (4 hidden states)*: lane change, accelerate (while overtaking a car), lane change to return to the original lane, resume a cruise speed.

In order to infer the behaviors, we maintain a probability distribution over the behaviors represented by the hidden states of the upper layer HMM. The observations of cars (i.e. sensor data) interact with the HMM in the lower layer and the information is then propagated up to the upper layer. In the lower layer, there is a corresponding HMM for each higher level behavior description. Each HMM in the lower layer, indexed by  $h = 1, \dots, H$ , updates its current state by:

$$P(S_{t,h} | Q_{1:t}) \propto P(Q_t | S_{t,h}) \sum_{S_{t-1,h}} P(S_{t-1,h}) P(S_{t,h} | S_{t-1,h}), \quad (8)$$

where  $S_{t,h}$  is the hidden state variable of HMM  $h$  at time  $t$ ,  $Q_{1:t} = \{Q_t, Q_{t-1}, \dots, Q_1\}$  and  $Q_t$  are the observations at time  $t$ . Then, the observation likelihood for each lower level HMM is:

$$L_h(Q_{1:t}) = \sum_{S_{t,h}} P(S_{t,h} | Q_{1:t}). \quad (9)$$

The observation likelihoods  $L_h(Q_{1:t})$  are the “observations” for the upper layer HMM. The inference of the upper level behaviors takes a similar form:

$$P(B_t | Q_{1:t}) = P(Q_{1:t} | B_t) \sum_{B_{t-1}} P(B_{t-1}) P(B_t | B_{t-1}) \quad (10)$$

$$= L_{B_t}(Q_{1:t}) \sum_{B_{t-1}} P(B_{t-1}) P(B_t | B_{t-1}), \quad (11)$$

where  $B_t$  is the hidden state variable of the upper level HMM at time  $t$ , and  $P(B_t | B_{t-1})$  is the upper level behavior transition matrix. In order to distinguish whether the change of the high-level behavior occurs after the completion of the low-level behavioral sequence, two transition matrices are used:  $\mathbf{T}_{\text{final}}$  corresponds to the high-level behavior transition when the lower level behavioral sequence is completed, otherwise the transition matrix  $\mathbf{T}_{\text{not-final}}$  is used. The upper level behavior transition matrix depends on the lower level states:

$$P(B_t | B_{t-1}) = \sum_{S_{t,B_{t-1}}} P(S_{t,B_{t-1}}) P(B_t | S_{t,B_{t-1}} B_{t-1}), \quad (12)$$

where  $S_{t,B_{t-1}}$  is the state at time  $t$  of the lower level HMM, corresponding to the previous behavior  $B_{t-1}$ , and by definition:

$$P(B_t | S_{t,B_{t-1}} B_{t-1}) = \begin{cases} \mathbf{T}_{\text{final}}, & \text{if } S_{t,B_{t-1}} \text{ is a final state,} \\ \mathbf{T}_{\text{not-final}}, & \text{otherwise.} \end{cases} \quad (13)$$

The probability distributions over high-level behaviors  $P(B_i|Q_{1:t})$  are maintained iteratively, and the layered HMM is updated according to Algorithm 1.

### B. Driving Behavior Realization

A behavior is an abstract representation of the vehicle motion. Driving behavior realization is modeled as GP, i.e. a probabilistic representation of the possible evolution of the vehicle motion for a given behavior [51]. This model allows us to obtain the probability distribution over the physical realization of the vehicle motion by assuming a usual driving represented by GP, i.e. lane following without drifting too far off to the lane sides. On a straight road, this is a *canonical* GP with the mean corresponding to the lane middle.

To deal with variations of lane curvature or such behaviors as “turn left” or “turn right”, we propose an adaptation procedure, where the canonical GP serves as a basis and it is deformed according to the road geometry. The deformation method is based on the Least Squares Conformal Map (LSCM) [52]. Its advantage is a compact and flexible representation of the road geometry. The canonical GP can be calculated once and then can be reused for different lane configurations, thus, resulting in a better computational efficiency. An example is shown in Fig. 9 for a curved road.

### C. Estimation of Risk

A complete probabilistic model of the possible future motion of the vehicle is given by the probability distribution over behaviors from driving behavior *recognition* and driving behavior *realization*. The layered HMM approach assigns a probability distribution over behaviors at each time instance, and a GP gives the probability distribution over the physical realization for each behavior. Because the behavioral semantics are propagated from the layered HMM down to the physical level, it is now possible to assign semantics to risk values. Note that the definition of risk can take a variety of forms, depending on how the risk output is going to be used. A risk scalar value might be sufficient for a crash warning system, or an application might require the risk values against each vehicle in the traffic scene.

The risk calculation is performed by first sampling of the paths from the GP. The fraction of the samples in collision gives the risk of collision, which corresponds to the behavior represented by the GP. A general risk value is obtained by marginalizing over behaviors based on the probability distribution over behaviors obtained from the layered HMM. It is possible to calculate risk of taking a certain path, a certain behavior, or a general risk value of a certain vehicle against another vehicle. The flexibility of this estimation is due to HMMs in identifying behaviors and the use of GPs for behavior realization, while taking into account the road geometry and topology. Intuitively, the result of our risk estimation can be explained as “col-

#### Algorithm 1. Layered HMM Updates.

```

Input: Current observation  $Q_t$ 
Output:  $P(B_i|Q_{1:t})$ 
foreach Lower layer HMM  $h$  do
  Update  $P(S_{t,h}|Q_{1:t})$  (Eqn. 8);
  Calculate log-likelihood  $L_h(Q_{1:t})$  (Eqn. 9);
end
Update upper layer HMM  $P(B_i|Q_{1:t})$  (Eqn. 11).

```

lision risk for a few seconds ahead.” A systematic framework for evaluation of different types of collision risk can be found in [12].

## IV. Experimental Results

The relevance and feasibility of the two main functionalities have been evaluated experimentally. For environment modeling, early experiments have been performed on real urban data obtained with our experimental vehicle. The collision risk assessment has been evaluated on a driving simulator, and behavior estimation has also been tested during a highway driving.

### A. Experimental Setup

Our experimental platform is a Lexus LS600h car shown in Fig. 10. The vehicle is equipped with a variety of sensors including two IBEO Lux lidars placed toward the edges of the front bumper, a TYZX stereo camera situated behind the windshield, and an Xsens MTi-G inertial sensor with GPS. Extrinsic calibration of these sensors is done manually for this work. Note that, thanks to the grid-based approach and considering the resolution of the grid, a slight calibration error has very little impact on the final results.

The stereo camera baseline is 22 cm, with a field of view of 62°. Camera resolution is 512 × 320 pixels with a focal length of 410 pixels. Each lidar provides four layers of up to 200 impacts with a sampling period of 20 ms. The angular range is 100°, and the angular resolution is 0.5°. The

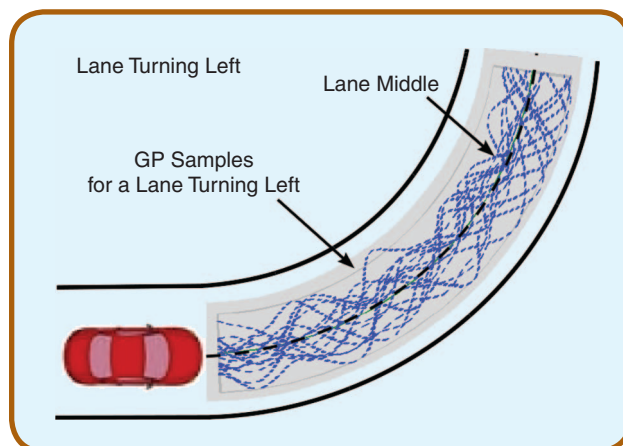


FIG 9 Deformed GP model example for a lane turning left.



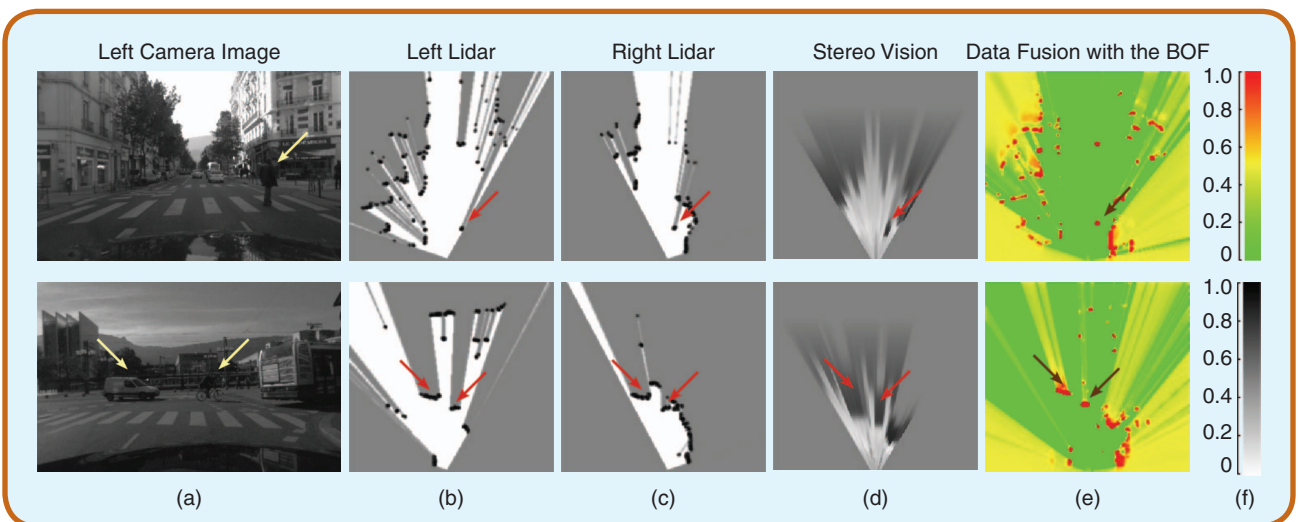
**FIG 10** (a) Lexus LS600h car equipped with two IBEO Lux lidars, (b) a TYZX stereo camera, and (c) an Xsens MTi-G inertial sensor with GPS.

on-board computer is equipped with 8 GB of RAM, an Intel Xeon 3.4 GHz processor and an NVIDIA GeForce GTX 480 for GPU. The observed region is 40 m long by 40 m wide, with a maximum height of 2 m. Cell size of the occupancy grids is  $0.2 \times 0.2$  m. For stereo-vision, the correlation window measures 11 pixels in width and 21 pixels in height.

### B. Occupancy Grid Mapping

Fig. 11 shows examples of occupancy grid mapping with the proposed approach. The arrows indicate the pedestrian, the car, and the bicycle, which appear in the camera images and the occupancy grids. Because the accuracy

of stereo-vision tends to become poor at large distance, the corresponding grid has been attenuated beyond 20 m and the system is tuned to give more confidence to the lidars than to the stereo-vision. One of advantages of sensor fusion is a larger viewfield so that the vehicles overtaking the ego-vehicle (they are not seen in the camera images) are correctly mapped on the resulting BOF grid. Moreover, the sensor fusion as well as the Bayesian estimation provide to filter out the laser impacts with the road surface, e.g. right lidar in Fig. 11. Note that a large number of dynamic objects in the traffic scenes may lead to a failure of object-based fusion because of a



**FIG 11** Examples (a)–(f) of occupancy grid mapping in typical urban traffic scenes: (a) left image from the stereo pair, (b) an occupancy grid from the left lidar, (c) an occupancy grid from the right lidar, (d) an occupancy grid from stereo-vision, (e) an occupancy grid estimated by data fusion with the BOF, and (f) a probability scale.



large number of association hypotheses. The grid-based approach allows us to avoid the object association problem for sensor fusion.

### C. Object Detection and Tracking

The object level representation is obtained from the BOF, by clustering the occupancy and velocity grids by means of the FCTA. Examples of detections in typical urban scenes are shown in Fig. 12. The output of FCTA being a set of ellipses in the detection plane, the ROIs in the images are obtained by using a ground plane hypothesis. The height of ROI is set empirically to 1.8 m, and the width is double of the lateral standard deviation of the detected object. As it is shown in Fig. 12-a, both vehicles and a pedestrian are correctly detected and tracked. An advantage of the BOF over other occupancy grid approaches is illustrated by Fig. 12-b: the estimated velocity of the white vehicle and of the bicycle being very different (here the bicycle is faster), they are correctly detected as two different objects, even if they are very close (adjacent cells in the grid).

### D. Computation Time

Two critical stages of the algorithm have been implemented on GPU: the BOF and the stereo processing, including matching and occupancy grid computation. FCTA has not, since it has been shown in [31] and [33] that its computational cost can be neglected, compared with the computational cost of the BOF. The BOF being designed to be highly parallelizable, it runs on GPU in 20 ms, without specific optimization. Concerning stereo-vision, the matching process is performed in 6 ms and the occupancy grid computation in 0.1 ms. This level of performance is reached thanks to the u-disparity approach, which allows for highly parallel computation on GPU.

### E. Collision Risk Assessment

The simulation of crash situations is performed in a virtual environment. This environment is a 3D geometric model of a road network, where each vehicle is driven by a human driver. The simulator was developed by Toyota Motor Europe (TME). Each human driver controls his or her virtual vehicle by means of a steering wheel, the acceleration and brake pedals. Recording a scenario with multiple vehicles, which are driven concurrently, requires a large number of human drivers. An alternative is to generate the scenario iteratively, with one human-driven vehicle at a time and “adding” human drivers iteratively, with a replay of the previously recorded human-driven vehicles. The resulting virtual environment allows us to simulate crash situations safely.

The layered HMM evaluates the behavior of every vehicle in the scene for different time horizons, except the ego-vehicle. The training data are obtained by collecting sequences for a series of human-driven cases, where each



FIG 12 Example of the objects detected using BOF and FCTA: (a) cars and a pedestrian, (b) cars, a bicycle and a bus.

driver uses the steering wheel as an interface to the virtual environment of the simulator. The driving sequences are annotated manually by means of an annotation tool of ProBayes. Then, the annotated data are used to train the layered HMM.

The TME simulator provides a 3D road view for the driver and a 2D view of the road network, as shown in Fig. 13. The collision risk is calculated for a yellow vehicle, while other vehicles are shown by red rectangles (relevant area is inside a large yellow circle). The right-hand traffic rule is assumed. The trail behind the yellow vehicle in 2D view indicates the

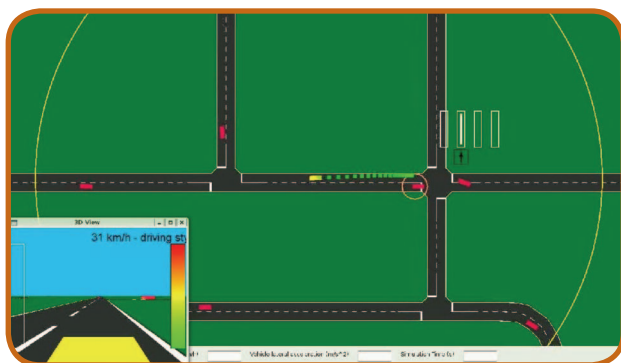


FIG 13 Virtual environment of the TME simulator.



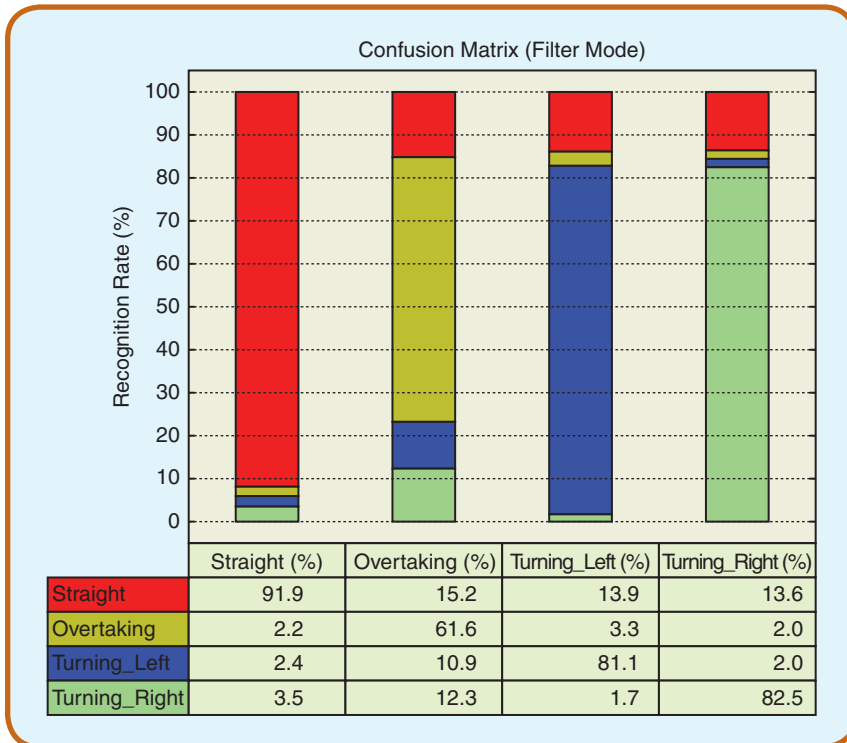


FIG 14 Performance summary of the behaviors recognition with layered HMM.

ally to this functionality, our algorithm evaluates risk at intersections, where the linearity assumption about the vehicle motion would result in underestimated values of risk. The combination of the behavior estimation by the layered HMM and the use of semantics (e.g. turn right or move straight) at the geometric level allows us to obtain the appropriate risk values.

The training data for the layered HMM were collected with ten human drivers who were asked to show different driving behaviors. The data is split by the uniform distribution into the training data and the test data (30% of the samples). The behavior recognition is trained on the training data and is evaluated against the test data.

Fig. 14 summarizes the recognition performance of the layered HMM. The results are presented as a confusion matrix, where the columns correspond to the true class and the rows correspond to the estimated class.

The diagonal values of the confusion

matrix give the correctly predicted class, while non-diagonal values show the percentage of mislabelling for each class. The highest recognition rate is for “move straight” behavior (91.9%) as well as “turn right” or “turn left” ones (82.5% and 81.1%, respectively). The “overtake” behavior has a relatively low recognition rate of 61.6%. Intuitively, this lower rate can be explained by a composite structure of the overtaking maneuver because it consists of such behaviors as: accelerating, lane changing, returning to the original lane, and resuming a cruise speed. Consequently,

risk levels estimated previously. At each instant, the probabilities of the possible behaviors of the nearest neighbor (red vehicle) are estimated by the layered HMM and are displayed by the vertical white bars. The speed of the yellow vehicle is shown in 3D view, where the right-side vertical bar shows the risk encoding by color from “low” (green) to “high” (red). The left-side vertical bar in 3D view indicates the current risk value for the yellow vehicle.

The speed warning about a potential danger of frontal collision is available in most commercial systems. Addition-

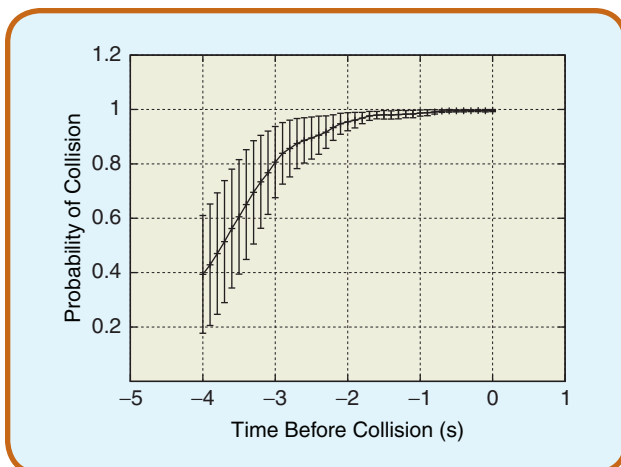


FIG 15 Example of collision risk assessment for ten human-driven scenarios and a three-second prediction horizon.

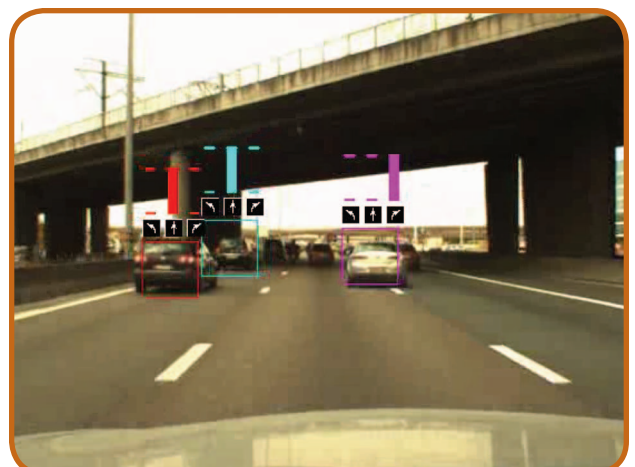


FIG 16 Example of behavior estimation on a highway where a vehicle on the middle lane performs a lane change to the right.

it also takes longer than a three-second period (current prediction horizon) to complete an overtaking maneuver.

The approach to risk assessment is illustrated by Fig. 15, where the probability of collision is estimated for a period of three seconds ahead of each collision for ten different traffic scenarios. The rapid increase in the probability of collision and its certainty are observed when the collision instant approaches.

#### F. Behavior Estimation on a Highway

The first phase is to gather sensor data when driving on a highway to estimate behaviors of other vehicles. The experiments have been conducted jointly by the TME and ProBayes on a different vehicle. The data acquisition was performed for four scenarios on a highway, with each scenario lasting for ten minutes approximately and the sensor data (stereo camera images, vehicle odometry, and GPS information) being recorded. The behaviors to be estimated are: move straight, a lane change to the left, and a lane change to the right.

The detection of vehicles is performed by clustering of the disparity points obtained from the stereo camera mounted behind the windshield. The clustering is performed in the image areas, which are indicated by the image based detection using support vector machines. The positions of vehicles are tracked on the road plane by means of the BOF.

The observation variables for behavior recognition include the vehicle's speed, the distances to the lane borders, and the information about the presence of other vehicles on the adjacent lanes. In order to obtain the observation variables in a global reference frame, a particle filter is used for localizing the vehicle on the highway map obtained from the Geographic Information System. The particle filter allows us to estimate the position and direction of the vehicle at each time instant and to employ the observations from stereo-vision (lanes detection), GPS and vehicle odometry. A similar approach is used for the training phase, when the acquired data is divided into the training and evaluation sets annotated manually to indicate the current behavior for each time instance of the data acquired.

An example of the behavior estimation on a highway is shown in Fig. 16. The positions of the tracked vehicles are projected onto the image plane and are represented by the rectangles. The probability distribution of the estimated behaviors is shown by the height of the color bars above the vehicles, e.g. the "lane change to the right" behavior of the vehicle on the middle lane and the "move straight" behavior of the two vehicles on the left lane are evaluated correctly. These results illustrate the validity of the proposed approach for behavior estimation. The different probability decomposition of the observation variables, the selection of the observation variables and the reactivity of the behavior estimation are topics of our ongoing work to generalize the approach.

## V. Conclusion

We proposed a conceptual framework to analyze and interpret the dynamic traffic scenes by means of sensor fusion with the BOF and risk evaluation for the ego-vehicle. Our concept differs from other approaches due to its underlying probabilistic methods and its computational efficiency because of the parallel implementation of data processing from stereo-vision and lidars. The experimental platform was built on a Lexus car with embedded sensors and the dedicated software modules. This system is capable of monitoring its local environment, detecting and tracking static and dynamic objects in real traffic scenarios. The analysis and interpretation of traffic scenes rely on evaluation of driving behaviors as stochastic variables to estimate and predict collision risks for the ego-vehicle for a short period ahead, in order to alert the driver and help improve the safety of car driving. The experiments and simulation have shown promising results. The discussed conceptual framework will be extended to deal with complex traffic scenarios, and the experimental system will be used to create a database to allow for benchmarking, quantitative evaluation and comparison with alternative approaches.

## VI. Acknowledgment

The authors thank Gabriel Othmezzouri and Ichiro Sakata of Toyota Motor Europe as well as Hiromichi Yanagihara of Toyota Motor Corporation for their continuous support of our experimental work on the Lexus car and collaboration on collision risk assessment. Our thanks are given to Nicolas Turro and Jean-François Cuniberto of INRIA for their technical assistance in setting our experimental platform.

## About the Authors



*Christian Laugier* is Research Director at INRIA and Scientific Leader of the e-Motion Team. He is also responsible at INRIA for scientific relations with Asia & Oceania. He received the PhD degree in computer science from Grenoble University, France in 1976. His current research interests mainly lie in the areas of

motion autonomy, intelligent vehicles and probabilistic robotics. He has co-edited several books in the field of robotics and several special issues of scientific journals such as IJRR, Advanced Robotics, JFR, or IEEE Transactions on ITS. In 1997, he was awarded the Nakamura Prize for his contributions to "Intelligent Robots and Systems." Dr. Laugier is a member of several scientific committees including the Steering/Advisory Committees of the IEEE/RSJ IROS, FSR, and ICARCV conferences. He is also co-Chair of the IEEE RAS Technical Committee on AGV & ITS. He has been General Chair or Program Chair of such conferences as: IEEE/RSJ IROS'97, IROS'02, IROS'08, or FSR'07. Additionally to his research and teaching activities, he co-founded four start-up companies in the fields

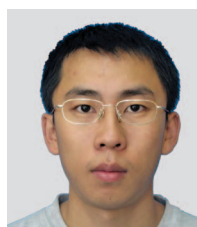
of robotics, computer vision, computer graphics, and Bayesian programming tools. He has served as scientific consultant for the ITMI, Aleph Technologies, and ProBayes companies.



**Igor E. Paromtchik** is Principal Engineer at INRIA Grenoble Rhône-Alpes. He develops and integrates technologies and software for data processing from multiple sensors for automotive applications. Previously, Igor has worked on numerous projects in robotics and ITS with teams at RIKEN, Japan in 1997–2007; INRIA, France in 1995–2002; and the University of Karlsruhe, Germany in 1992–1994. Among his accomplishments is the development of one of the world-first experimental prototypes of automated parking systems during his work at INRIA. He received the MBA degree from the University of Chicago, IL, USA in 2006, the PhD degree in systemic analysis and automatic control in 1990, and the MS degree in radio-physics in 1985 from the Belarusian State University, where he taught control engineering and robotics as an assistant professor in the early 1990s. His current research interests focus on ITS and high-frequency data processing. Dr. Paromtchik has been involved in the program committees of several conferences, including IEEE/RSJ IROS and DARS. He has authored more than 70 research and technical publications and holds seven patents.



**Mathias Perrollaz** graduated from the National Polytechnical Institute of Grenoble (INPG), France in 2003, with a major in signal and image processing. He worked on ITS at the LIS (CNRS) in Grenoble and with the perception team of LIVIC (INRETS). He received the PhD degree from the University of Paris-6 (UPMC) in 2008 for his work on multi-sensor obstacle detection. He has been developing probabilistic methods for ITS at INRIA Grenoble since 2009. He worked on perception for robotics at Ohio Northern University, OH, USA in 2011. Mathias taught in the Paris-10 and Grenoble-2 Universities in France.



**Mao Yong** received the BS degree in computer science and technology from the Tsinghua University in Beijing, China in 2001, where he also obtained his MS degree and PhD degree in 2004 and 2007, respectively. He worked on probabilistic methods for ITS during his post-doctoral fellowship at INRIA Grenoble Rhône-Alpes, France until 2010. His current research interests include autonomous vehicles, multiple sensor fusion, multi-object detection and tracking.



**John-David Yoder** graduated from the University of Notre Dame, Notre Dame, IN, USA with the BS degree in mechanical engineering in 1991. He received his MS and PhD degrees in mechanical engineering also from Notre Dame in 1996 and 1999, respectively. He is Associate Professor and Chair of the mechanical engineering department at Ohio Northern University, Ada, OH. He has previously served as Proposal Engineer and Proposal Engineering Supervisor at Grob System, Inc. and Software Engineer at Shaum Manufacturing, Inc. He has held a number of leadership and advisory positions in various entrepreneurial ventures. He is currently a KEEN (Kern Entrepreneurial Education Network) Fellow, and for the 2009-10 academic year is an invited professor at INRIA Grenoble Rhône-Alpes, France. Research interests include computer vision, mobile robotics, intelligent vehicles, entrepreneurship, and education.



**Christopher Tay** received the B.Eng degree in computer engineering from Nanyang Technological University, Singapore in 2003, and the MS and PhD degrees in computer science from the National Polytechnical Institute of Grenoble (INPG), France in 2004 and 2009, respectively. He works as Research and Development Engineer with ProBayes in France. His research interests include Bayesian statistics, machine learning and its real-world applications and implementation.



**Kamel Mekhnacha** is a co-founder and CTO of ProBayes since 2004. He received the engineering degrees in computer science from Constantine University, Algeria in 1994, and the MS (1995) and PhD (1999) degrees both in computer science from the National Polytechnical Institute of Grenoble (INPG), France. He worked as a research engineer at INRIA for four years on the most recent approaches in Bayesian modeling, inference, and learning. He is a main contributor in the design and development of the general-purposes *ProBT* Bayesian programming toolkit and the *Bayesian Occupancy Filter (BOF)* software dedicated to sensor fusion and multi-object tracking. His applications fields include robotics, heterogeneous sensor fusion, behavior and gesture recognition, and driver assistance.

**Amaury Nègre** is Research Engineer at CNRS since 2008. He graduated from the National Polytechnical Institute of Grenoble (INPG), France in 2005, with a major in image





processing. He received his PhD degree in vision and robotics from INPG in 2009 for his work on vision based obstacle detection for mobile robots. Now he is with the Laboratory of Informatics of Grenoble (LIG) and offers engineering support in experimental robotics and ambient intelligence.

## References

- [1] R. Horowitz and P. Varaiya, "Control design of an automated highway system," *IEEE Special Issue on Hybrid Syst.*, vol. 88, no. 7, 2000.
- [2] I. E. Paromtchik and C. Laugier. (2011, June 8). Automatic parallel parking and returning to traffic. *Video Proc. IEEE Int. Conf. Robotics and Automation* [Online]. Leuven, Belgium, 1998. Available: [http://www.youtube.com/watch?v=6yS\\_inHMJJI](http://www.youtube.com/watch?v=6yS_inHMJJI)
- [3] I. E. Paromtchik, "Planning control commands to assist in car maneuvers," in *Proc. IEEE Int. Conf. Advanced Robotics*, Coimbra, Portugal, June 2005, pp. 1508–1515.
- [4] R. Benenson, S. Petti, T. Fraichard, and M. Parent, "Toward urban driverless vehicles," *Int. J. Vehicle Auton. Syst. (Special Issue on Advances in Autonomous Vehicle Technologies for Urban Environment)*, vol. 1, no. 6, pp. 4–25, 2008.
- [5] C. Urmson et al., "Autonomous driving in urban environments: Boss and the urban challenge," *J. Field Robot.*, vol. 25, no. 8, 2008.
- [6] M. Montemerlo et al., "Junior: The Stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, 2008.
- [7] J. Markoff, "Google cars drive themselves, in traffic," *The New York Times*, Oct. 2010.
- [8] Ford. (2011, June 8). Safety and security. [Online]. Available: <http://www.ford.co.uk/Cars/Mondeo/Safetyandsecurity>
- [9] G. Vasilash. (2011, June 8). The Lexus LS 600H L: Not just another production car. [Online]. Available: <http://www.autofieldguide.com/articles/060705.html>
- [10] Mercedes-Benz. (2011, June 8). 2010 E-class sedans. [Online]. Available: <http://www.mbusa.com/mercedes/#/bodyStyleOverview?vc=E&bs=SDN>
- [11] D. N. Lee, "A theory of visual control of braking based on information about time-to-collision," *Perception*, vol. 5, no. 4, pp. 437–459, 1976.
- [12] C. Tay, "Analysis of dynamics scenes: Application to driving assistance," Ph.D. dissertation, INRIA, Grenoble, France, 2009.
- [13] M. Skutec, M. Mekhnacha, and G. Wanielik, "Pre-crash system based on radar for automotive applications," in *Proc. IEEE Intelligent Vehicles Symp.*, Columbus, OH, 2005.
- [14] T.-D. Vu and O. Aycard, "Laser-based detection and tracking moving objects using data-driven Markov chain Monte-Carlo," in *Proc. IEEE Int. Conf. Robotics and Automation*, Kobe, Japan, May 2009.
- [15] P. Griffiths, D. Langer, J. A. Misener, M. Siegel, and C. Thorpe, "Sensor-friendly vehicle and roadway systems," in *Proc. IEEE Instrumentation and Measurement Technology Conf., Budapest, Hungary*, 2001.
- [16] M. Enzweiler and D. Gavrilu, "Monocular pedestrian detection: Survey and experiments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 31, 2009.
- [17] R. Labayrade, D. Aubert, and J. P. Tarel, "Real time obstacle detection on non flat road geometry through 'v-disparity'," in *Proc. IEEE Intelligent Vehicles Symp.*, Versailles, France, 2002.
- [18] S. Nedeveschi, R. Danescu, D. Frentiu, T. Marita, F. T. Graf, and R. Schmidt, "High accuracy stereovision obstacle detection on non planar roads," in *Proc. IEEE Intelligent Engineering Systems*, Cluj Napoca, Romania, 2004.
- [19] A. Makris, M. Perrollaz, I. E. Paromtchik, and C. Laugier, "Integration of visual and depth information for vehicle detection," in *Proc. IEEE/RSJ IROS Workshop Perception and Navigation for Autonomous Vehicles in Human Environment*, San-Francisco, CA, Sept. 2011.
- [20] H. Hirschmüller, "Stereo processing by semi-global matching and mutual information," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 30, no. 2, 2008.
- [21] CUDA. (2011, June 8). Available: <http://www.nvidia.com>
- [22] U. Franke, C. Rabe, H. Badino, and S. Gehrig, "6D-Vision: Fusion of stereo and motion for robust environment perception," in *Proc. DAGM Symp. Pattern Recognition*, Vienna, Austria, 2005.
- [23] Velodyne. (2011, June 8). High definition lidar. [Online]. Available: <http://www.velodyne.com/lidar/>
- [24] R. Labayrade, C. Royere, D. Gruyer, and D. Aubert, "Cooperative fusion for multi-obstacles detection with the use of stereovision and laser scanner," *Auton. Robots*, vol. 19, no. 2, 2005.
- [25] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri, and H. G. Jung, "A new approach to urban pedestrian detection for automatic braking," *IEEE Trans. Intell. Transport. Syst.*, vol. 10, no. 4, 2009.
- [26] M. Munz, M. Mählich, and K. Dietmayer, "Generic centralized multi sensor data fusion based on probabilistic sensor and environment models for driving assistance systems," *IEEE Intell. Transport. Syst. Mag.*, vol. 2, no. 1, 2010.
- [27] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.*, vol. 9, no. 2, 1988.
- [28] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.
- [29] M. Yguel, O. Aycard, and C. Laugier, "Efficient GPU-based construction of occupancy grids using several laser range-finders," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Beijing, China, Oct. 2006.
- [30] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière, "Bayesian occupancy filtering for multitarget tracking: An automotive application," *Int. J. Robot. Res.*, no. 1, 2006.
- [31] M. K. Tay, K. Mekhnacha, C. Chen, M. Yguel, and C. Laugier, "An efficient formulation of the Bayesian occupation filter for target tracking in dynamic environments," *Int. J. Auton. Veh.*, vol. 6, no. 1–2, 2008.
- [32] M. Yguel, C. Laugier, and K. Mekhnacha, "Vehicle driving assistance and related processes," *European Patent 0552736*, 2005.
- [33] K. Mekhnacha, Y. Mao, D. Raulo, and C. Laugier, "Bayesian occupancy filter based 'Fast Clustering-Tracking' algorithm," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Nice, France, 2008.
- [34] A. Eidehall and L. Petersson, "Statistical threat assessment for general road scenes using Monte Carlo sampling," *IEEE Trans. Intell. Transport. Syst.*, vol. 9, no. 1, pp. 137–147, 2008.
- [35] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," *IEEE Trans. Intell. Transport. Syst.*, vol. 10, no. 4, pp. 678–687, 2009.
- [36] F. Sandblom and M. Brännström, "Probabilistic threat assessment and driver modeling in collision avoidance systems," in *Proc. IEEE Intelligent Vehicles Symp.*, Baden-Baden, Germany, June 2011, pp. 914–919.
- [37] "Vehicle-based countermeasures for signal and stop sign violation," *NHTSA, U.S. DOT*, Washington, DC, Tech. Rep. DOT HS 809 716, 2004.
- [38] K. Fuerstenberg and J. Chen, "New European approach for intersection safety—Results of the EC project INTERSAFE," in *Proc. Int. Forum Advanced Microsystems for Automotive Application*, 2007.
- [39] A. Polychronopoulos, M. Tsogas, A. J. Amditis, and L. Andreone, "Sensor fusion for predicting vehicles' path for collision avoidance systems," *IEEE Trans. Intell. Transport. Syst.*, vol. 8, no. 3, pp. 549–562, 2007.
- [40] M. Althoff, O. Stursberg, and M. Buss, "Model-based probabilistic collision detection in autonomous driving," *IEEE Trans. Intell. Transport. Syst.*, vol. 10, no. 2, pp. 299–310, 2009.
- [41] M. Eilers and C. Möbius, "Learning the human longitudinal control behavior with a modular hierarchical Bayesian mixture-of-behaviors model," in *Proc. IEEE Intell. Veh. Symp.*, Baden-Baden, Germany, June 2011, pp. 540–545.
- [42] I. E. Paromtchik, C. Laugier, M. Perrollaz, M. Yong, A. Nègre, and C. Tay, "The ArosDyn project: Robust analysis of dynamic scenes," in *Proc. Int. Conf. Control, Automation, Robotics and Vision*, Singapore, Dec. 2010.
- [43] I. E. Paromtchik, M. Perrollaz, and C. Laugier, "Fusion of telemetric and visual data from road scenes with a Lexus experimental platform," in *Proc. IEEE Intelligent Vehicles Symp.*, Baden-Baden, Germany, June 2011, pp. 746–751.
- [44] P. Burt, L. Wixson, and G. Salgian, "Electronically directed 'focal' stereo," in *Proc. IEEE Int. Conf. Computer Vision*, 1995.
- [45] M. Perrollaz, A. Spalanzani, and D. Aubert, "A probabilistic representation of the uncertainty of stereo-vision and its application for obstacle detection," in *Proc. IEEE Intelligent Vehicles Symp.*, San Diego, CA, 2010.
- [46] H. Badino, U. Franke, and R. Mester, "Free space computation using stochastic occupancy grids and dynamic programming," in *Proc. Workshop Dynamical Vision, ICCV, Rio de Janeiro*, Brazil, 2007.
- [47] M. Perrollaz, J.-D. Yoder, and C. Laugier, "Using obstacle and road pixels in the disparity space computation of stereo-vision based occupancy grids," in *Proc. IEEE Int. Conf. Intelligent Transportation Systems, Madeira, Portugal*, 2010.
- [48] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin: Springer-Verlag, 2006.
- [49] G. Welch and G. Bishop. (2011, June 8). An introduction to the Kalman filter. [Online]. Available: <http://www.cs.unc.edu/~welch/kalman/kalmanIntro.html>
- [50] C. Laugier et al., "Vehicle or traffic control method and system," *European Patent Application Number 09169060.2-1264*, Aug. 2009.
- [51] C. Tay and C. Laugier, "Modelling smooth paths using Gaussian processes," in *Proc. Int. Conf. Field and Service Robotics*, 2007.
- [52] B. Lévy, S. Petitjean, N. Ray, and J. Mailliot, "Least squares conformal maps for automatic texture atlas generation," in *Proc. ACM SIGGRAPH Conf.*, 2002.