

The ArosDyn Project: Robust Analysis of Dynamic Scenes

Igor E. Paromtchik*, Christian Laugier*, Mathias Perrollaz*,
Mao Yong*, Amaury Nègre*, Christopher Tay†

*INRIA Grenoble Rhône-Alpes, 38334 Saint Ismier, France

†ProBayes, 38334 Saint Ismier, France

Abstract—The ArosDyn project aims to develop embedded software for robust analysis of dynamic scenes in urban traffic environments, in order to estimate and predict collision risks during car driving. The on-board telemetric sensors (lidars) and visual sensors (stereo camera) are used to monitor the environment around the car. The algorithms make use of Bayesian fusion of heterogenous sensor data. The key objective is to process sensor data for robust detection and tracking of multiple moving objects for estimating and predicting collision risks in real time, in order to help avoid potentially dangerous situations.

Index Terms—Mobile robot, sensor fusion, Bayesian filter, stereo vision, lidar, collision risk, traffic environment

I. INTRODUCTION

The urban traffic environment with multiple participants contains risks of potential collision and damage. The car safety technologies (e.g. seat belts, airbags, safety glass, energy-absorbing frames) mitigate the effects of accidents. The advanced technologies will be capable of monitoring the environment to estimate and predict collision risks during car driving, in order to help reduce the likelihood of accidents occurring. The risk management by traffic participants is an efficient way to improve traffic safety toward *zero-collision* driving. The key problem is to correctly interpret the traffic scene by means of processing information from a variety of sensors. In this context, robust analysis of traffic scenes by means of data processing from on-board sensors is the objective of our ArosDyn project.

The relevant sensors include stereo vision, lidars, an inertial measurement unit (IMU) combined with a global positioning system (GPS), and odometry. The local environment is represented by a grid. The fusion of sensor data is accomplished by means of the Bayesian Occupancy Filter (BOF) [1], [2], that provides to assign probabilities of *cell occupancy* and *cell velocity* for each cell in the grid. 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 behaviours of multiple dynamic agents in road scenes.

Various approaches have been proposed to represent the environment and interpret traffic scenes. They use such sensors as: a telemeter like radar [3], a laser scanner [4], cooperative detection systems [5], or monocular vision for detecting specific features like edges, symmetry [6], colour [7], or movement [8]. Most monocular approaches are capable of recognizing vehicles and pedestrians. Stereo vision provides a three-dimensional scene representation and it is particularly suitable for generic obstacle detection [9], [10], [11].

The long-term solution for improving traffic safety may be automated cars. The promising accomplishments range from such early projects as PATH [12] to more recent ones as CityCars [13] and the DARPA Urban Challenge [14]. In the short- to medium-term, traffic accidents can be reduced by recognizing high-risk situations, which can be evaluated by means of sensor data processing about the local environment, i.e. obstacle detection and alerting the driver (passive safety), or modifying the driving parameters (active safety) if a collision becomes imminent.

We represent the environment by a grid [15] and use the BOF for sensor fusion, as explained in section II. In order to identify objects, our Fast Clustering-Tracking (FCT) algorithm takes the BOF result as input, and it outputs the position and velocity of the detected objects and the associated uncertainties [16]. The subsequent risk assessment produces estimates of the possible behaviours (continue straight, turn right, turn left, or stop) in our probabilistic model of the future [17]. This model contains HMMs and GPs for predicting the likelihood of drivers' actions and combines information about the car position and velocity with the observations of other cars.

II. BAYESIAN OCCUPANCY FILTER (BOF)

The BOF serves for data fusion from stereo vision and lidars. It operates with a two-dimensional grid representing the environment. Each cell of the grid contains a probability of the cell occupancy and a probability of the cell velocity. The probabilistic models of a lidar and a stereo camera are developed, in order to use the BOF.

The lidar model is beam-based [15]. It includes four layers of beams and assumes each beam being independent. We build a probabilistic model for each beam layer independently. Knowing the position of the lidar, we filter out those beams which produce impacts with the ground.

The stereo camera is assumed in a "rectified" geometrical configuration, that allows us to compute a disparity map, which is equivalent to a partial three-dimensional representation of the scene, as shown in Fig. 1. The disparity map computation is based on the double correlation method [18], which provides two major advantages: a better matching over the road surface and an instant separation between "road" and "obstacle" pixels, without using any arbitrary threshold. The computation of the occupancy grid is directly performed in the disparity space associated with the disparity map, thus, preserving the intrinsic precision of the stereo camera.

The partially occluded areas of the scene are monitored by means of our visibility estimation approach. Consider a pixel U in the u-disparity plane. The occupancy of U is expressed by a combination of the visibility of U and the occupancy confidence of U , as estimated from the disparity map. Let $P(C_U = 1)$ denote the confidence of U being occupied, and $P(V_U = 1)$ be the probability of U being visible. Then, the occupancy probability of U is

$$P(O_U) = P(V_U = 1) \cdot P(C_U = 1) \cdot (1 - P_{FP}) + P(V_U = 1) \cdot (1 - P(C_U = 1)) \cdot P_{FN} + (1 - P(V_U = 1)) \cdot 0.5, \quad (1)$$

where P_{FP} and P_{FN} are the false positive and false negative probabilities of the stereo camera. Then, the u-disparity occupancy grid is transformed into a metric grid for its use in the BOF. This probabilistic model of the stereo camera is described in detail in [19].



Figure 1: Left-side image from a stereo camera (a) and a corresponding disparity map (b)

At each time step, the probabilities of cell occupancy and cell velocity are estimated by means of Bayesian inference with our models of the sensors. The inference leads to a Bayesian filtering process, as shown in Fig. 2. Given a set of observations, the BOF algorithm updates the probability estimates for each cell in the grid [1], [2].

In this context, the prediction step propagates the cell occupancy and antecedent (velocity) probability distributions of each cell and obtains the prediction $P(O_c^t A_c^t)$, where $P(O_c^t)$ is the occupancy probability and $P(A_c^t)$ is the antecedent (velocity) probability 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

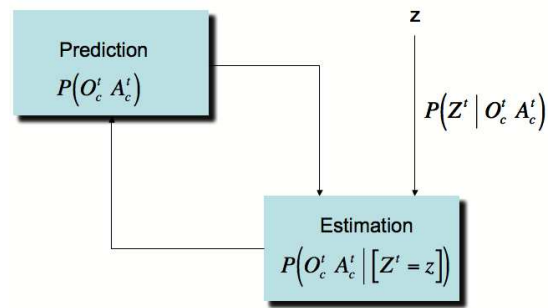


Figure 2: Bayesian filtering for estimation of the probability distribution of the cell occupancy and the cell velocity

observations yielded by the sensors $\prod_{i=1}^S P(Z_i^t | O_c^t A_c^t)$ to obtain the a posteriori state estimate $P(O_c^t A_c^t | [Z_1^t \dots Z_S^t])$, where Z_i^t denotes the observation of a sensor i at time t . This allows us to compute by marginalization $P(O_c^t | [Z_1^t \dots Z_S^t])$ and $P(A_c^t | [Z_1^t \dots Z_S^t])$ used for prediction in the next iteration.

Our FCT algorithm provides to track the objects' trajectories [16]. It operates at an object representation level and contains three modules: a clustering module, a data association module, and a tracking and tracks management module.

The clustering module combines the probabilities of the cell occupancy/velocity estimated by the BOF with the prediction for each object being tracked by the tracker, i.e. a region of interest (ROI). We then try to extract a cluster in each ROI and associate it with the corresponding object. There could be a variety of cluster extracting algorithms, however, we have found that a simple neighbourhood-based algorithm provides satisfactory results. The output of this module leads to three possible cases, as shown in Fig. 3: (i) no object is observed in the ROI, (ii) unambiguous observation with one and only one cluster extracted and implicitly associated with the given object, and (iii) ambiguous observation, where the extracted cluster is associated with multiple objects.

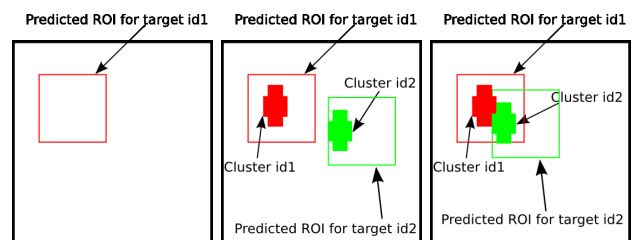


Figure 3: The possible cases of clustering result: no object observed, unambiguous observation, and ambiguous observation

The data association module aims to solve the problem of ambiguous observation (multiple tracked objects, overlapped ROIs) in the clustering module. Assuming N objects associated with a single cluster with a number N known exactly, the cause of the ambiguity is twofold: (i) numerous objects are

very close to each other and the observed cluster is the union of observations generated by N different objects, and (ii) N different objects correspond to a single real object and the observations must be merged into one. We employ a re-clustering strategy to deal with the first situation and a cluster merging strategy for the second one. The re-clustering aims to divide the cluster into N sub-clusters and associate them with the N objects, respectively. Because the number N is known, a K-means based algorithm can be applied [20].

The cluster merging 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 being parts of a single object. The probability values $P(F_{ij} | S_{ij})$ and $P(F_{ij} | \neg S_{ij})$ are the algorithm parameters which are constant with regard to i and j . Similarly, the probability $P^t(S_{ij} | \neg F_{ij})$ is updated when no ambiguity between T_i and T_j is observed. Then, by thresholding the probability $P^t(S_{ij})$, 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 reports which are associated with the objects being tracked without ambiguity. Then, the tracking and tracks management module uses a general tracks management algorithm to create and delete the tracks, and use a Kalman filter to update their states.

III. COLLISION RISK ESTIMATION

An overall architecture of our risk estimation module is sketched in Fig. 4.¹ The problem that we are interested in is associated with the following sub-modules.

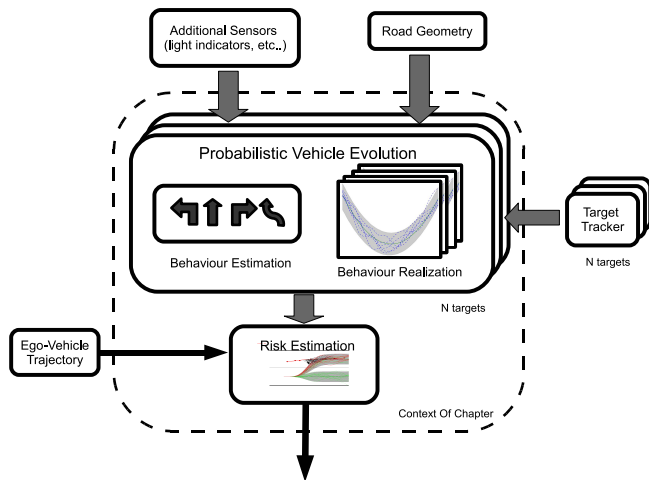


Figure 4: Architecture of the risk estimation module

Driving behaviour recognition. The behaviour recognition aims at estimating the probability distribution when executing one of the feasible behaviours, e.g. $P(\text{turn_left})$ represents the probability of turning left by the car. The behaviours give implicitly high-level representations of a road structure, which

¹This risk estimation method has been awarded a European Patent to INRIA - Probayes - Toyota Motor Europe.

contain semantics. The probability distribution over behaviours is obtained by HMM.

Driving behaviour realization. The collision risk evaluation requires the motion geometry. Driving behaviour realization takes the form of GPs, i.e. a probabilistic representation of a possible evolution of the car motion for a given behaviour. The adaptation of GP according to the behaviour is based on the geometrical transformation known as the Least Squares Conformal Map (LSCM) [21].

Collision risk estimation. A complete probabilistic model of the possible future motion of the car is given by the probability distribution over behaviours from driving behaviour recognition and driving behaviour realization. The collision risk can be calculated from this model. Intuitively, the result of our risk estimation module can be summarized under a notion of “collision risk for a few seconds ahead”. However, its precise mathematical definition depends on the meaning and interpretation of estimated risks, as discussed in [17].

Behaviour recognition and modelling

The aim of behaviour recognition is to assign a label and a probability measure to sequential data. In this context, the sequential data are the observations received from the sensors. Examples of sensor values are: distance to lane borders, signaling lights, or a proximity to an intersection. However, the output we wish to obtain are the probability values over behaviours, i.e. the behaviours are hidden variables.

The behaviour modelling contains two layers, where each layer consists of one or more HMMs. The upper layer is a single HMM, where its hidden states represent high-level behaviours, such as overtaking, turning left, turning right, or moving straight. For each hidden state or behaviour in the upper layer HMM, there is a corresponding HMM in the lower layer to represent the sequence of the finer state transitions of a single behaviour, as depicted in Fig. 5.

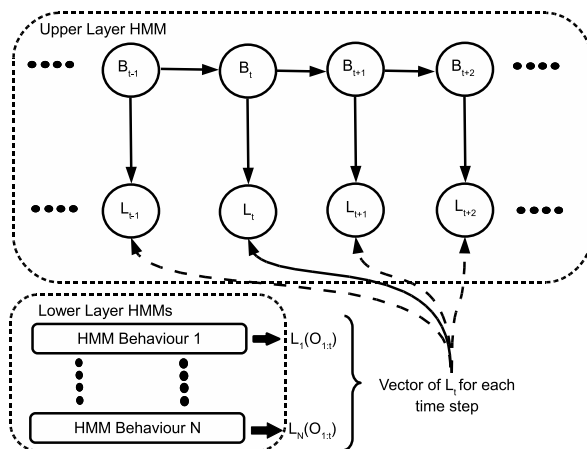


Figure 5: Layered HMM, where each lower layer HMM’s likelihood is computed and used as the upper layer HMM’s observation

Let us define the following hidden state semantics in the lower layer HMMs for each of the following behaviours of the higher layer HMM:

- *Move straight (1 hidden state)*: move forward.
- *Overtake (4 hidden states)*: lane change, accelerate (while overtaking a car), lane change to catch up the original lane, resume a cruise speed.
- *Turn left or right (3 hidden states)*: Decelerate before a turn, execute a turn, resume a cruise speed.

In order to infer the behaviours in our context, we wish to maintain a probability distribution over the behaviours represented by the hidden states of the upper layer HMM. Sensor-based observations of cars interact with the HMM in the lower layer and the information is then propagated to the upper layer.

Driving behaviour realization

A behaviour is an abstract representation of the car motion. For a given behaviour, a probability distribution over the physical realization of the car motion is indispensable for risk estimation. The GP provides to obtain this probability distribution by assuming that usual driving is represented by the GP, i.e. lane following without drifting too far off to the lane sides. On a straight road, this would be a *canonical GP* with the mean corresponding to the lane median.

To deal with the variations of lane curvature or such behaviours as “turning left” or “turning 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 LSCM. Its advantage is a compact and flexible representation of the lane geometry. The canonical GP can be calculated once and, then, be reused for different situations, thus resulting in a better computational efficiency. An example is shown in Fig. 6 for a non-zero curvature lane.

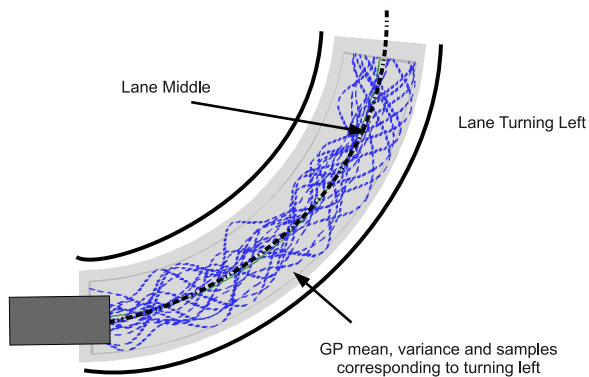


Figure 6: Deformation of a canonical GP for a left-turning lane

Collision risk estimation

The layered HMM approach assigns a probability distribution over behaviours at each time instance, and a GP gives the probability distribution over its physical realization for each

behaviour. Because the behavioural semantics are propagated from the layered HMM down to the physical level, it is now possible to assign semantics to risk values.

One should note that the definition of risk can take a variety of forms, which is largely dependent 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 car in the traffic scene.

The risk calculation is performed by first sampling of the trajectories from the GP. The fraction of samples in collision gives the risk of collision, which corresponds to the behaviour represented by the GP. A general risk value is obtained by marginalizing over behaviours based on the probability distribution over behaviours obtained from the layered HMM. It is possible to calculate risk of taking a certain path, a certain behaviour, or a general risk value of a certain car against another car. A systematic framework for evaluation of different types of risk can be found in [17].

IV. EXPERIMENTAL RESULTS

Our experimental platform is built on a Lexus LS600HL. The car is equipped with a TYZX stereo camera situated behind the windshield, two IBEO Lux lidars placed inside the frontal bumper, and an Xsens IMU combined with GPS. The stereo camera and the left lidar are shown in Fig. 7. The on-board DELL computer with an NVidia graphics processing unit (GPU) is used for collecting and processing of the sensor data and the risk assessment. The visual and telemetric data are used concurrently for a preliminary qualitative evaluation.



Figure 7: The TYZX stereo camera and IBEO Lux lidar

The TYZX stereo camera has a baseline of 22 cm, a resolution of 512x320 pixels, and a focal length of 410 pixels. The IBEO Lux lidar provides four layers of upto 200 impacts at a sampling period of 20 ms. The lidar The maximum detection range is about 80 m, the angular range is 100°, and the angular resolution is 0.5°. We use two lidars to monitor the area in front of the car. The observed region is 40 m in length and 40 m in width, a maximum height is 2 m, and the cell size of the grid is 0.2x0.2 m. The user interface is based on Qt library and it provides access to several parameters of the system, e.g. filtering, disparity computation, BOF. The Hmgr middleware [22] provides to record and synchronize the data from different sensors as well as the replay capability. One should note that the sensor data fusion with the BOF requires calibration of the extrinsic parameters of the sensors in the common coordinate system.

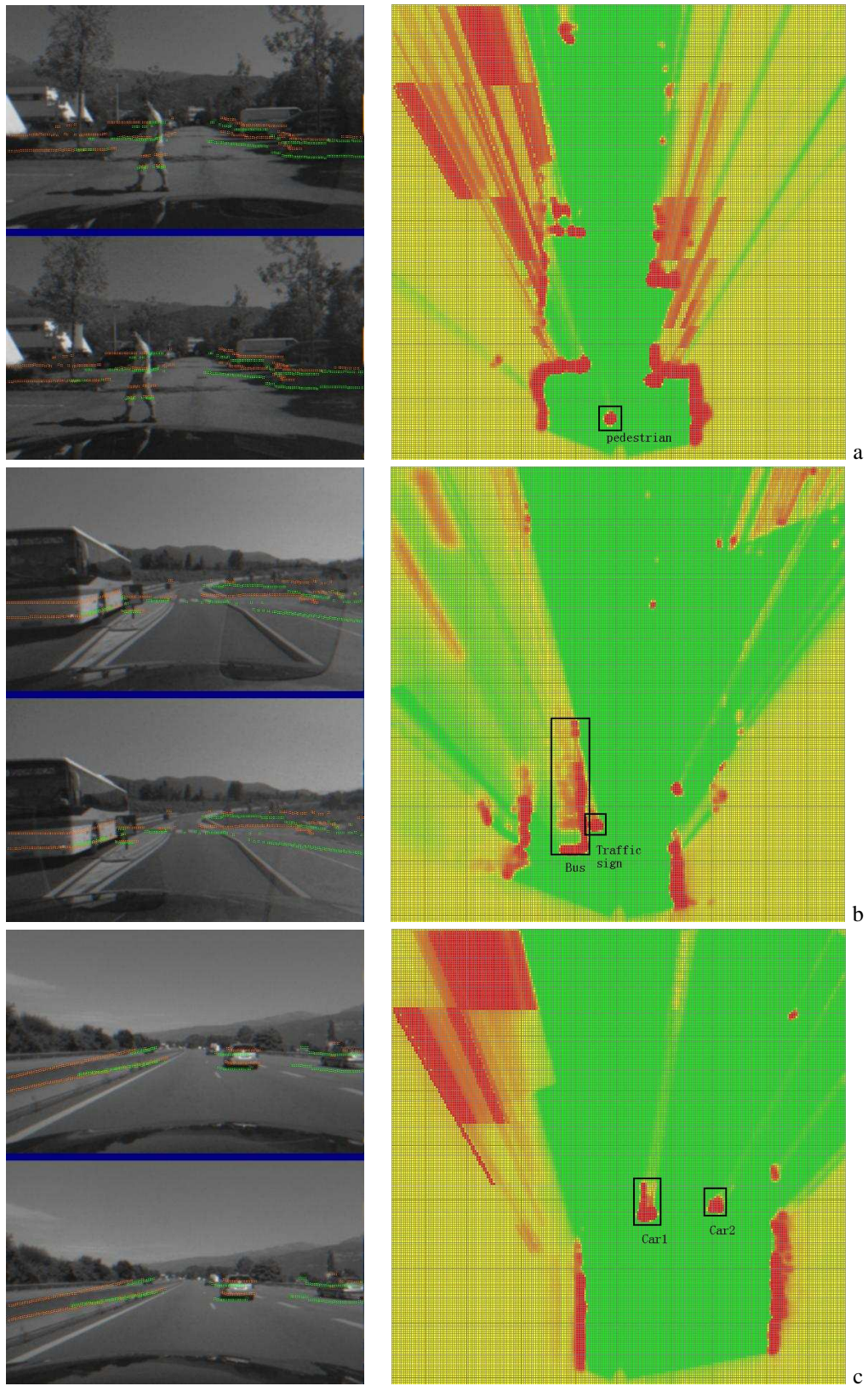


Figure 8: Fusion of visual and telemetric data by means of BOF (stereo images are at the left side, and the corresponding grid representations are at the right side): (a) a pedestrian walking in the parking area, (b) a bus approaching on a low speed road, (c) two cars on a highway

Occupancy grid and data fusion

We selected three typical road scenes (with a pedestrian, a bus, and two cars) from our large dataset, as shown in Fig. 8. The stereo images are displayed at the left-hand side (upper image is from the left camera). The laser impacts are plotted onto the images (coloured dots). The occupancy grid is estimated by the BOF and is shown at the right-hand side in Fig. 8, where the occupancy probability is represented by red colour (high probability) or green colour (low probability), and the yellow-coloured cells correspond to the areas, where the occupancy probability is unknown.

The scene with a pedestrian walking in a parking area is shown in Fig. 8a. The corresponding occupancy grid provides to correctly discriminate the pedestrian and the surrounding cars from the unoccupied area. This example illustrates an advantage of stereo vision over lidars because of its capability of perceiving partially occluded objects, such as the white car at the right-hand side in Fig. 8a. Yet the accuracy of stereo vision decreases with the distance and becomes weak at long range, while the high accuracy of lidars remains constant over the distance. The road scene with a bus approaching the car is shown in Fig. 8b. The bus is correctly detected by visual and telemetric sensors. The accuracy appears to be sufficient for distinguishing the road sign and the bus at a short distance range. The highway example is shown in Fig. 8c. The both cars are correctly detected. It appears that almost all road surface in the front is estimated as unoccupied, including the areas occluded by the cars. The fusion of data from the two lidars, and the time filtering capability of the BOF allow us to attain this performance. When analyzing the image sequences, the objects are tracked correctly, in general. The estimation of velocities of cells in the BOF results in distinguishing between two adjacent objects moving at different speeds.

Processing time

In comparison to high computational cost of the BOF, the costs of the FCT algorithm can be neglected [2], [16]. Our efforts to improve the computational efficiency focus on the BOF implementation. This grid-based algorithm provides a way to parallelize the computation on the GPU. In order to compare the computational efficiency of the BOF on two different GPUs, we implemented the BOF in C++ language without optimization on a GPU with 4 processors (NVIDIA Quadro FX 1700) and on a GPU with 30 processors (NVIDIA GeForce GTX 280). For example, the complete processing chain for a lidar (including the BOF and the FCT algorithm) is capable of running at 20 Hz on a GPU with 30 processors. The implementation of our stereo image processing on the GPU allows us to run both the matching stage and the occupancy grid computation in real time at upto 30 fps.

Validation of risk estimation

We tested our risk estimation approach on a driving simulator in a virtual environment, where a human drives a virtual car by using a usual steering wheel. The virtual environment allows us to deal with various collision situations, which are

difficult to reproduce in real world. The estimated risk values are recorded for a period of several seconds ahead of each collision. The experiments were jointly conducted with Toyota Motor Europe (TME) to evaluate the reliability of generated trajectories by the GP and the estimated behaviours.

Fig. 9 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 “moving straight” behaviour (91.9%) as well as “turning right” or “turning left” behaviours (82.5% and 81.1%, respectively). The “overtaking” behaviour has a relatively low recognition rate of 61.6%. Intuitively, this lower rate can be explained by a composite structure of the overtaking behaviour because it consists of such behaviours as: accelerating, lane changing, catching up the original lane, and resuming the cruise speed.

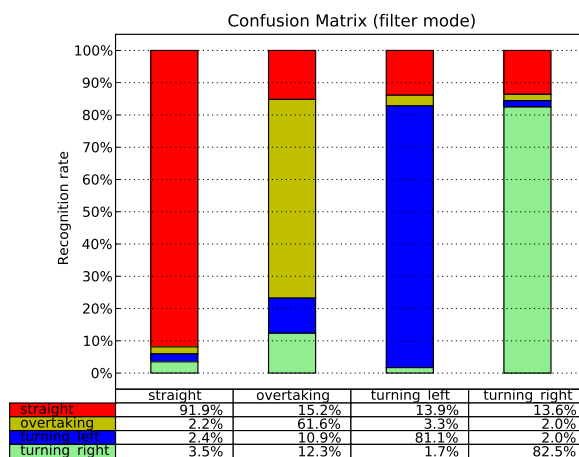


Figure 9: Performance summary of the behaviours detection with layered HMM

The efficiency of our approach to risk estimation is illustrated by Fig. 10, where one can see the estimated risk values (means and variances) for a period of 3 seconds ahead of each collision for ten different traffic scenarios. When the collision instant approaches, the probability of collision increases rapidly while its variance diminishes.

V. CONCLUSION

Collision risk estimation and prediction will be mandatory for future cars. A fraction of a second of the driver’s reaction time can help save human lives. Our data processing approach, sensor models and software modules allow us to monitor the urban traffic environment and perform data fusion from stereo vision and lidars, as well as to detect and track stationary and dynamic objects in real traffic scenarios. The analysis and interpretation of traffic scenes rely on evaluation of driving

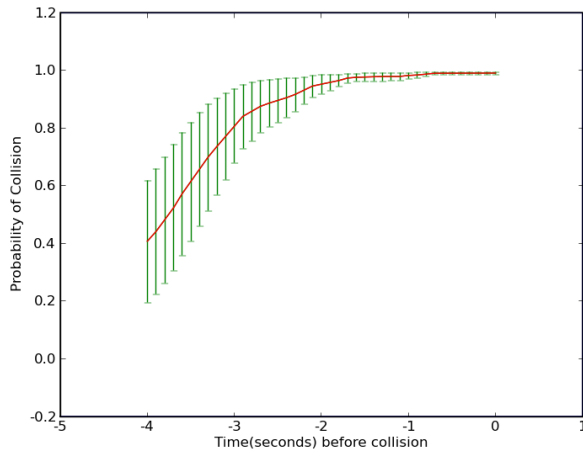


Figure 10: Aggregate mean and variance values of collision risk for ten human-driven scenarios and a three-second prediction horizon

behaviours as stochastic variables to estimate and predict collision risks for a short period ahead. Our future work will deal with the complete integration of the described approaches and their in-depth evaluation with the Lexus car.

ACKNOWLEDGEMENT

The authors thank to Toyota Motor Europe 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 (INRIA) for their technical assistance in setting our experimental platform, and to John-David Yoder (Ohio Northern University) for valuable discussions.

REFERENCES

[1] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, P. Bessière, "Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application," *Int. J. Robotics Research*, No. 1, 2006.
 [2] M.K. Tay, K. Mekhnacha, C. Chen, M. Yguel, C. Laugier, "An Efficient Formulation of the Bayesian Occupation Filter for Target Tracking in Dynamic Environments," *Int. J. Autonomous Vehicles*, 6(1-2):155-171, 2008.

[3] M. Skutek, M. Mekhaïel, G. Wanielik, "Pre-crash System based on Radar for Automotive Applications," *Proc. of the IEEE Intelligent Vehicles Symp.*, Columbus, USA, 2003.
 [4] A. Mendes, L. Conde Bento, U. Nunes, "Multi-target Detection and Tracking with a Laserscanner," *Proc. of the IEEE Intelligent Vehicles Symp.*, University of Parma, Italy, 2004.
 [5] P. Griffiths, D. Langer, J. A. Misener, M. Siegel, C. Thorpe, "Sensor-friendly Vehicle and Roadway Systems," *Proc. of the IEEE Instrumentation and Measurement Technology Conf.*, Budapest, Hungary, 2001.
 [6] M. Bertozzi, A. Broggi, A. Fascioli, S. Nichele, "Stereo Vision-based Vehicle Detection," *Proc. of the IEEE Intelligent Vehicles Symp.*, Detroit, USA, Oct. 2000.
 [7] M. Betke, H. Nguyen, "Highway Scene Analysis from a Moving Vehicle under Reduced Visibility Conditions," *Proc. of the IEEE Int. Conf. on Intelligent Vehicles*, Stuttgart, Germany, Oct. 1998.
 [8] K. Yamaguchi, T. Kato, Y. Ninomiya, "Moving Obstacle Detection using Monocular Vision," *Proc. of the IEEE Intelligent Vehicles Symp.*, Tokyo, Japan, June 2006.
 [9] M. Bertozzi, A. Broggi, "Gold : A Parallel Real-time Stereo Vision System for Generic Obstacle and Lane Detection," *IEEE Trans. on Image Processing*, 7(1), Jan. 1998.
 [10] R. Labayrade, D. Aubert, J.-P. Tarel, "Real Time Obstacle Detection on non Flat Road Geometry through 'v-disparity'," *Proc. of the IEEE Intelligent Vehicles Symp.*, Versailles, France, 2002.
 [11] S. Nedeveschi, R. Danescu, D. Frentiu, T. Marita, F. T. Graf, R. Schmidt, "High Accuracy Stereovision Obstacle Detection on non Planar Roads," *Proc. of the IEEE Intelligent Engineering Systems*, Cluj Napoca, Romania, Sept. 2004.
 [12] R. Horowitz, P. Varaiya, "Control Design of an Automated Highway System," *Proc. of the IEEE*, 88(7):913-925, July 2000.
 [13] R. Benenson, S. Petti, T. Fraichard, M. Parent, "Toward Urban Driverless Vehicles," *Int. J. of Vehicle Autonomous Systems*, Special Issue on Advances in Autonomous Vehicle Technologies for Urban Environment, 1(6):4 - 23, 2008.
 [14] C. Urmsen *et al.*, "Autonomous Driving in Urban Environments: Boss and the Urban Challenge," *J. of Field Robotics*, vol. 25(8), 2008.
 [15] S. Thrun, W. Burgard, D. Fox, "Probabilistic Robotics," *MIT Press*, 2005.
 [16] K. Mekhnacha, Y. Mao, D. Raulo, C. Laugier, "Bayesian Occupancy Filter based "Fast Clustering-Tracking" Algorithm," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Nice, 2008.
 [17] C. Tay, "Analysis of Dynamics Scenes: Application to Driving Assistance," *PhD Thesis*, INRIA, France, 2009.
 [18] M. Perrollaz, R. Labayrade, R. Gallen, D. Aubert, "A Three Resolution Framework for Reliable Road Obstacle Detection Using Stereovision," *Proc. of the IAPR MVA Conf.*, 2007.
 [19] M. Perrollaz, J.-D. Yoder, C. Laugier, "Using Obstacle and Road Pixels in the Disparity Space Computation of Stereo-vision based Occupancy Grids," *Proc. of the IEEE Int. Conf. on Intelligent Transportation Systems*, Madeira, Portugal, Sept. 19-22, 2010.
 [20] C. M. Bishop, "Pattern Recognition and Machine Learning," *Springer*, 2006.
 [21] B. Lévy, S. Petitjean, N. Ray, J. Maillot, "Least Squares Conformal Maps for Automatic Texture Atlas Generation," *Proc. ACM SIGGRAPH Conf.*, 2002.
 [22] CyCab Toolkit, <http://cycabtk.gforge.inria.fr>