Perception and Automation for Intelligent Mobility in Dynamic Environments

Lukas Rummelhard[1,3], Jérôme Lussereau[1], Jean-Alix David[1], Christian Laugier[1], Salvador Dominguez[2], Gaetan Garcia[2] and Philippe Martinet[2]

I. Introduction

Along the progressive equipment of perception capabilities on vehicles, from ultrasonic sensors, radars, and low-cost cameras, to more powerful LiDAR technologies in a likely near future, the digital connection and control over the different vehicle functionalities have been developed to such an extent that the question of automation of certain key operations, even in complex environments, can be addressed. The rise in importance of ADAS (Advanced Driver Assistance Systems) technologies is a key component of the evolution of the automotive industry, in which many approaches and systems are competing in order to present the most efficient and safest responses to complex situations. A first challenge is to extract from the sensor data not only the meaningful pieces of information, but also to match them over space and time, in order to generate a proper representation of the environment, allowing clear situation awareness, necessary for any reasonable response. Another key aspect is the ability to perform the decisions proposed by the ADAS system, effectively affecting the vehicle commands and trajectory, while still taking into account the presence of the driver. In this paper is presented the development of an ADAS system architecture, focusing on the two previously mentioned features. After a presentation of the perception system, based on dynamic occupancy grid generation, fusion, filtering and projection, leading to collision risk assessment, the control system, consisting in a specific vehicle command system modification, will be described. Finally, experimental results will be presented and discussed, in the case of an advanced emergency braking system, implemented, embedded and tested on an actual vehicle, in different collision risk situations.

II. Perception System Overview

The original perception system developed and deployed on the experimental platforms is based on probabilistic occupancy grid generation and filtering, using a formalism corresponding to a Bayesian programming framework [1]. The use of such a Bayesian formalism allows proper confidence estimation and combination, particularly important features when confronted with incomplete or even contradictory data coming from different sensors. Another major feature of the system is its highly-parallelized design : from the data fusion, to the grid filtering, velocity inference and collision risk assessment, the methods have been designed to allow massive parallelization of computations, and so benefit from parallel-computing devices (Nvidia GPU at first, then other GPUs and many-core technologies), allowing real-time performances on embedded devices.

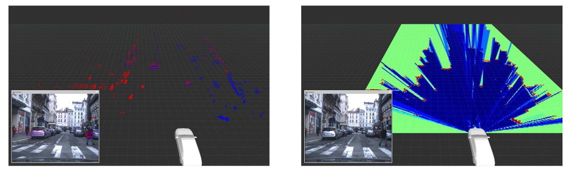

Fig. 1. Data fusion in an occupancy grid. Data from each of the 2 LiDARs are used to generate occupancy grids using sensor models, which are then fused by Bayesian fusion.

A. Data Fusion and Temporal Filtering : CMCDOT algorithm

In the presented approach, the environment is represented through probabilistic occupancy grids, a dense and generic representation especially adapted to parallel computing and clear management of uncertainty [2], [3]. Sensor data is converted to occupancy estimation using specific sensor model, sensor occupancy estimates are then combined by Bayesian fusion in every grid cell (Fig. 1). The Conditional Monte Carlo Dense Occupancy Tracker (CMCDOT) [4], a generic spatial occupancy tracker, then infers dynamics of the scene through a hybrid representation of the environment consisting of static and dynamic occupancy, empty spaces and unknown areas(Fig. 2, 3). This differentiation enables the use of state-specific models (classic occupancy grids for motionless components and sets of moving particles for dynamic occupancy), as well as relevant confidence estimation and management of data-less areas. The approach leads to a compact model that dramatically improves the accuracy of the results and the global efficiency in comparison to previous models.

This method is particularly suitable for heterogeneous sensor data fusion (camera, lidars, radars etc), both in term of localization and temporality. The occupancy of each cell over time can be estimated from various sensors data whose specific uncertainty (noise, measurement errors) are taken into consideration. Filtered cell estimates are thus much more robust, leading to a very reliable global occupancy of the environment, reducing false detections.

[1]Chroma, Inria Grenoble Rhône Alpes, Grenoble, France. name.surname@inria.fr

[2]IRCCyN, Ecole Centrale de Nantes, France. name.surname@irccyn.ec-nantes.fr

[3]currently CEA, name.surname@cea.fr

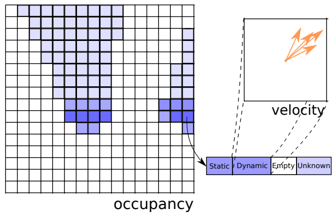

Fig. 2. Data representation in the CMCDOT formulation. The environment is divided into cells, to which are associated static, dynamic, empty and unknown coefficients. The dynamic part is allotted to weighted particles which sample the velocity space

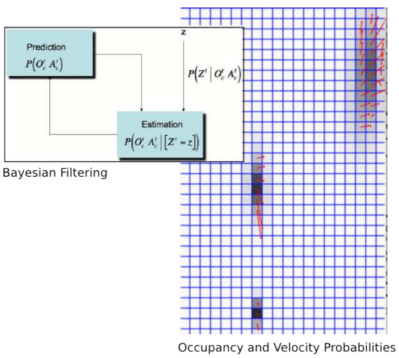

Fig. 3. Occupancy grid filtering and motion inference in every cell. At every time step, every cell updates its estimates of hidden state variables, representing its content, by Bayesian filtering. Velocity is inferred by particle generation and displacement over the grid of dynamic components.

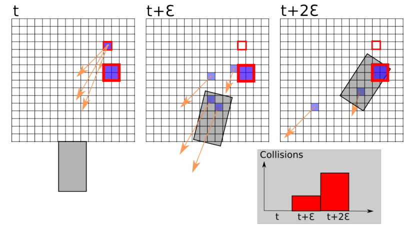

Fig. 4. Collision risk estimation over time for a specific cell. The cell position is predicted according to its velocity, along with the mobile robot. This risk profile is computed for every cell, and then used to integrate over time the global collision risk.

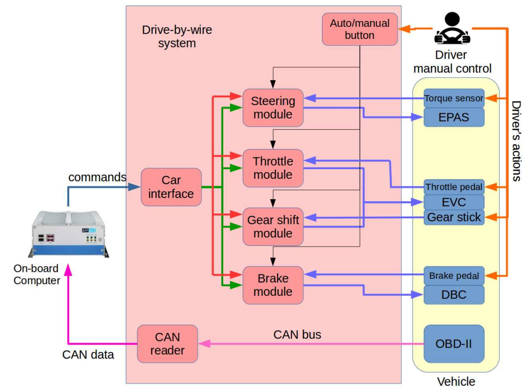

Fig. 5. Drive-by-wire hardware architecture

B. Risk assessment

Most of risk estimation methods consist in detecting and tracking dynamic objects in the scene [5], [6], the risk being then estimated through a Time to Collision (TTC) approach by projecting object trajectories to the future [7], [8]. The grid-based approach using the CMCDOT framework[4] instead directly computes estimations of the position in the near future of every static and dynamic part of the grid, as well as the trajectory of the vehicle. These estimations are iteratively computed over short time periods, until a potential collision is detected, in which case a TTC is associated to the cell from which the colliding element came from (Fig. 4). In every cell, the associated TTCs are cumulated over different time periods (1, 2, 3 seconds for example) to estimate a cell-specific collision risk profile. Risk grids, and global aggregated risks, are thus generated, and later used to generate response impulses for the control system. This strategy[9] avoids solving the complex problem of multi-object detection and tracking, while integrating the totality of the available information. It provides a probabilistic estimation of the risk associated to each part of the scene.

III. Vehicle control

The drive-by-wire system is composed of the following main parts: The car interface is the module that converts the commands sent by the control software into valid signals for the corresponding actuator module. The steering module converts the signals generated in the car interface into valid torque signals for the EPAS(Electric Power Assisted System). The throttle module converts the signals generated in the car interface into valid throttle signals for the EVC(Electric Vehicle Calculator). The brake module converts the signals generated in the car interface into valid brake signals for the DBC(Decoupled Brake Calculator). The gear shift module allows to set gear shift position R, N or D. In auto mode the real position of the gear stick is ignored except for the Parking position (P), as it involves a mechanical blocking.

The auto/manual button allows the user to choose the mode in which the car is used. In manual mode, the car behaves as a normal car, fully controlled by the driver. The computer can not affect any of the car actuators (steering, throttle, brake, gear shift). In automatic mode, the computer takes control of the car while still allowing the user to get it back, that is, the user can move the steering, accelerate or brake if necessary. The on-board computer accesses to the car internal data through a CAN reader, decoding it and updating the car status. The whole control kit has been used in a complete autonomous mode study ([10], [11]) where different lateral control laws have been compared.

IV. Experimentations

A. Experimental Platform

For the experiments, a Renault Zoe car (Fig. 6) has been equipped with a Velodyne HDL64 on the top, 3 Ibeo Lux LiDARs on the front and 1 on the back, Xsens GPS and IMU providing vehicle velocity and orientation, a stereo

Fig. 6. Experimental Platform: Renault Zoe car equipped with Velodyne HDL64, 4 Ibeo Lux LiDARs, Xsens GPS and IMU and cameras, and a crash test dummy crossing the dedicated street for the experiments.

camera and 2 IDS cameras. Data from LiDARS are fused and synchronized using the IBEO fusion box. The perception system described earlier has been implemented on a PC in the trunk of the car, equipped with a Nvidia Titan X GPU, while the previously described automation process has been integrated in the vehicle.

For the sake of the experiments, a dedicated portion of road has been designed and equipped with sensors and security requirements (PTL platform of IRT Nanoelec). A hand-made crash test dummy is used to simulate a pedestrian crossing a street (Fig. 6).

B. Short-term Risk Estimation and Automatic Emergency Braking

A first application of the system has been implemented and tested on real road data, consisting in a short-term risk estimation and automatic emergency braking process.

1) Risk Estimation results: On Fig. 7 can be seen some results of the perception system, allowing to localize in space and time potential collisions with the vehicle, even before the potential threat is even on its trajectory.

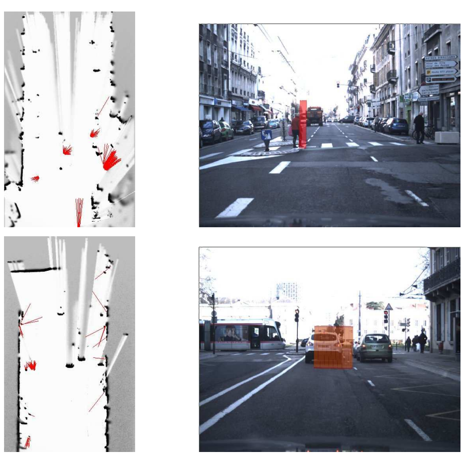

Fig. 7. CMCDOT and risk estimation results. Left image: CMCDOT filtered occupancy grid (white for empty space, black for occupied space) velocity field (from each moving cell are drawn red rays representing the velocities). Right image: dangerous cells are reprojected in the camera image, color indicating the timing of the risk.

2) Automatic Emergency results: On Fig. 8, the risk estimation was used to generate automatic braking, according to the potential collision risks.

Fig. 9 shows the response of the vehicle after sending an Emergency brake command when traveling at 25 Km/h. Since the command is sent till the car starts decelerating, there is a delay of about 120 ms, due mainly to the activation of the hydraulic brake system. After applying the maximum braking force, the vehicle decelerates 15Km/h in about 100ms. A little slippage was observed at the end of the first deceleration ramp that produces that the ABS system of the vehicle was activated, causing the speed to increase again to avoid sliding. The total stop time was about 1.02 s, producing an average deceleration of -8.73 m/s^2.

V. Conclusion

In this paper were presented the design and integration of a perception and automation system on a experimental vehicle, and first experimentations on real data.

The perception is based on the CMCDOT [4], a generic spatial occupancy tracker, which infers dynamics in the scene through a hybrid representation of the environment consisting of static and dynamic occupancy, empty spaces and unknown areas, using state-specific models to achieve proper state and confidence estimation. The collision risk is then assessed, by cell-level projection of the scene and the vehicle over time, describing a full and dense risk profile over time.

The vehicle has been modified in order to be able to control the steering, throttle, brake and gearshift by computer in a simple manner. A modular hardware architecture that minimizes the length of the wires reducing the risk of electrical noise by installing the signal modules close to the corresponding actuators has been used in our approach. A car interface module allows controlling the individual modules by computer at a high rate (up to 1KHz). To close the low-level control loop, the status of the car is obtained by reading directly the CAN bus of the car.

A first set of experiments of the system has consisted in an automatic collision risk assessment and emergency braking. The system showed promising results, opening the opportunity of further experiments (automatic trajectory diversion for risk avoidance,..).

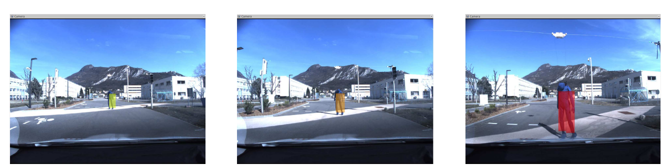

Fig. 8. Risk detection combined with automatic braking sequence. The evolution of the color of the reprojected collision risk corresponds to the time window before impact, activating accordingly the automatic braking.

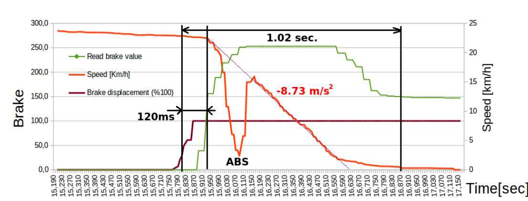

Fig. 9. Emergency brake response after sending a maximum brake limit from the on-board computer in automatic control mode at 25Km/h.

Acknowledgement

This work has been developed within the scope of the IRT Nanoelec, founded by the French program Investissement dAvenir ANR-10-AIRT-05. It has also been supported by Inria, in the scope of cooperations with Renault and with Toyota Motor Europe. This work was supported by French ANR project EquipEx RobotEx (ANR-10-EQPC-44) and by the European Union Program ”Compétitivité Régionale et Emploi” (FEDER).

References

[1] P. Bessière, E. Mazer, J. Ahuactzin-Larios, and K. Mekhnacha, *Bayesian Programming*. CRC Press, Dec. 2013.

[2] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[3] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI magazine*, vol. 9, no. 2, p. 61, 1988.

[4] L. Rummelhard, A. Négre, and C. Laugier, "Conditional monte carlo dense occupancy tracker," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sept 2015, pp. 2485–2490.

[5] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Multi-target tracking using joint probabilistic data association," in *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*, vol. 19. IEEE, 1980, pp. 807–812.

[6] Z. Khan, T. Balch, and F. Dellaert, "An mcmc-based particle filter for tracking multiple interacting targets," in *Computer Vision-ECCV 2004*. Springer, 2004, pp. 279–290.

[7] R. Labayrade, C. Royere, and D. Aubert, "Experimental assessment of the rescue collision-mitigation system," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 1, pp. 89–102, Jan 2007.

[8] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, Jan 2009.

[9] L. Rummelhard, A. Nègre, M. Perrollaz, and C. Laugier, "Probabilistic grid-based collision risk prediction for driving application," in *International Synposium on Experimental Robotics*, Springer, Ed., Marrakech, Marocco, 2014.

[10] S. Dominguez Quijada, A. Ali, G. Garcia, and P. Martinet, "Comparison of lateral controllers for autonomous vehicle : experimental results," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems*, Nov 2016, pp. 1418–1423.

[11] S. Dominguez Quijada, B. Khomutenko, G. Garcia, and P. Martinet, "An optimization technique for positioning multiple maps for self-driving car's autonomous navigation," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, Sept 2015, pp. 2694–2699.