Fast classification of static and dynamic environment for Bayesian Occupancy Filter (BOF)

Qadeer Baig, Mathias Perrollaz, Jander Botelho Do Nascimento, Christian Laugier
The authors are with the E-Motion team, Inria Rhone-Alpes,
38334 Montbonnot Saint Martin, France
Email: First.Last@inria.fr

***Abstract*—In this paper we present a fast motion detection technique based on laser data and odometry/imu information. This technique instead of performing a complete SLAM (Simultaneous Localization and Mapping) solution, is based on transferring occupancy information between two consecutive data grids. We plan to use the output of this work for Bayesian Occupancy Filter (BOF) framework to reduce processing time and improve the results of subsequent clustering and tracking algorithm, based on BOF. Experimental results obtained from a real demonstrator vehicle show the effectiveness of our technique.**

I. Introduction

In the field of Advanced Driver Assistance Systems (ADAS), many current approaches rely on the perception of the road scene. Particularly, the detection and tracking of the objects in the scene is essential for prediction of risky driving situations. Among the recent approaches for risk estimation, the authors in [1] propose to model and recognize the behavior of road scene participants. This approach is very promising for long term prediction of the risk, but not applicable for short term evaluation of the scene where we need to separate environment into static and dynamic parts.

A huge amount of work has been done to detect moving objects, especially by vision community [2]–[5]. These techniques have primarily been based on background subtraction or motion cues. Similar to background subtraction techniques have also been used in occupancy grids to detect moving objects [6]–[8]. These techniques are based on inconsistencies observed for new data by comparing them with maps constructed by SLAM. [9] and [10] have also proposed model based techniques to detect moving objects on the roads in the context of autonomous vehicle driving. Recently Dempster-Shafer theory based grids (evidential grids) have been used to detect moving objects using conflict analysis techniques [11], [12].

There exist various approaches for such classification of the environment. Having a very precise map of the environment, coupled with a very accurate localization algorithm, can be costly. Thus to perform a SLAM-based localization would be more realistic for commercial vehicles. The computed occupancy grid provides a description of the static environment [13]. More elaborated techniques like [8] or [9] improve this approach by performing detection and classification simultaneously, in a model-based-tracking framework.

In this paper, we propose a fast and efficient method for static/dynamic environment classification, which can be plugged in Bayesian Occupancy Filter [14], [15], to provide a more accurate representation of cell velocities. This will result in an improved results of Fast Clustering and Tracking Algorithm (FCTA) [16]. Our approach is different from other grid based methods in the sense that usually a complete SLAM solution is implemented to separate the moving parts from the static parts (as in [10]). However in current work we have developed a technique that deals with only two consecutive frames to detect moving parts rapidly. Our claim is that, by removing all static objects from the grid, the performance of the FCTA algorithm would increase effectively.

The paper is organized as follow: In next section we describe our demonstrator vehicle used to get data sets for this work with sensors installed on it. Section III describes the Bayesian occupancy filter framework for which we plan to use the results of current work. Next in section IV we detail our technique to detect moving objects from the sensor data. We present some results in section V and conclude this work with future perspectives in section VI.

II. Demonstrator

Our experimental platform is a Lexus LS600h car shown in Fig. 1. The car is equipped with such sensors as: two IBEO Lux lidars placed in the front bumper, one on left and other on the right of the vehicle, a TYZX stereo camera situated behind the windshield, and an Xsens IMU with GPS. Extrinsic calibration of the 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 hardware specification are the following: IBEO Lux LIDAR laser scanner provides four layers of up to 200 beams with a sampling period of 20 ms. The angular range is 100°, and the angular resolution is 0.5°. The on-board computer is equipped with 8GB of RAM, an Intel Xeon 3.4GHz processor and a NVIDIA GeForce GTX 480 for GPU. The observed region is 60 m long by 20 m wide, with a maximum height of 2 m (due to different verticle angle of the 4 layers of each laser scanner). Cell size for the occupancy grids is 0.2x0.2 m. The car is equipped with an IMU sensor: MTi-G XSens which is responsible for collecting the innertial data, it is deployed in the middle of the rear wheel axis, In this work we do not use the stereo vision cameras.

Different sensors installed on the demonstrator vehicle and other hardware setup are shown in figures 1 , 2 and 3.

Fig. 1. Lexus LS600h car equipped with two IBEO Lux lidars and a TYZX stereo camera

Fig. 2. MTi-G XSens IMU unit

Fig. 3. Intel Xeon 3.4GHz linux box

III. Bayesian Occupancy Filter (BOF)

In this section we briefly introduce the BOF framework with FCTA module to detect and track the moving objects. In BOF framework the idea is to perform sensor fusion and environment monitoring at low level, hence the fast and efficient grid based representation during all the processing. Objects are only retrieved at the end of the processing through clustering of the dynamic parts of the grid. So the complete processing BOF framework is divided into three stages: i) Multi sensor fusion using occupancy grid representation ii) filtering and estimation of dynamic grid and finally iii) Clustering of the objects and tracking. These three parts are elaborated below.

A. Sensor fusion from the multiple lidar layers

Each of the two lidar sensors installed on the vehicle provides 4 layers of scanning points. Each layer is used to compute an occupancy grid using the classical approach described in [17]. In order to retrieve a single grid for representation of the environment, the data from all these layers are merged using the approach described in [18]. This approach fuses the sensory information by using Linear Opinion Pools [19]. It has the advantage of reducing the errors due to conflicting information from the multiple layers.

The principle is to generate a posterior distribution over the occupancy of a cell C of the grid given the opinion of m sensors $\{Y_1 \dots Y_m\}$. Each sensor gives two quantities: its estimation for the occupancy of the cell $P(C|Y_i)$ and $w_i(C)$, a measure of the confidence for such estimations. The idea is to shut-down those sensors that do not give relevant information to the process by assigning a low weight to them. The fusion of all sensory information will be as follows:

$$P(C|Y_1 \dots Y_m) = \alpha \sum_{i=1}^{m} w_i(C) P(C|Y_i) \tag{1}$$

where $\alpha = \left[\sum_{i=1}^{m} w_i(C)\right]^{-1}$ is a normalization factor for the weights. Equation (1) is used to generate 2D-occupancy grids. For each sensor Y_i we must define $P(C|Y_i)$, the probability of a cell being occupied given the sensor information; and $w_i(C)$, the confidence on the opinion. Note that we assume independence among cells. This assumption, though it is very strong, is necessary to be efficient in computing equation (1), for each cell in parallel.

B. Filtering the grid using the Bayesian Occupancy Filter

The Bayesian Occupancy Filter (BOF) framework provides filtering capability, as well as the ability to estimate a velocity distribution for each cell of the grid. The BOF [14], [15] provides an adaptation of the Bayesian filtering methodology to the occupancy grid framework. It is based on a prediction-estimation paradigm. As an input, it uses an observed occupancy grid. On its output, it provides an estimated occupancy grid and a velocity grid, representing the probability distribution over possible velocities for each cell. An efficient formulation of this filter can be found in [20].

The BOF operates with a four-dimensional 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. The inference leads to a Bayesian filtering process, as shown in Fig. 4.

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

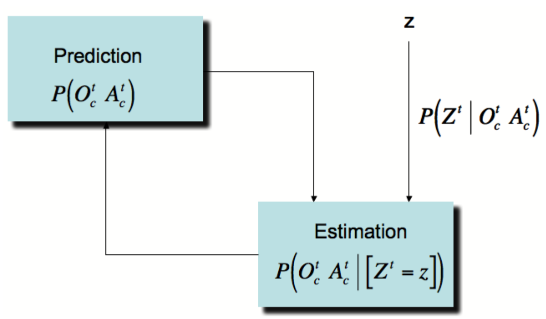

Fig. 4. Bayesian filtering in the estimation of occupancy and velocity distributions in the BOF grid

C. Detecting objects with the Fast Clustering and Tracking Algorithm

Obstacle detection requires to retrieve an object level representation of the scene. This can not be directly reached from the occupancy grid, and therefore a clustering algorithm is necessary. An algorithm adapted to the BOF framework is the "Fast Clustering Tracking Algorithm" described in [16]. It has the major interest to create clusters considering not only the connectivity in the occupancy grid, but also the Mahalanobis distance between cells in the estimated velocity grids. Thus two connected cells with different velocities are not merged during the clustering process.
FCTA includes a Kalman filter for target tracking and a ROI prediction approach that allows computation to be performed in real time. The output of the algorithm is a set of tracked objects, with position, velocity and associated uncertainties.

IV. MOTION DETECTION

In this section we detail the technique that we have developed to find moving parts of the evironment. The input to this motion detection module consists of an occupancy grid generated by the fusion module described in previous section and that fuses data from eight layers of two laser scanners installed on the demonstrator vehicle. Let us represent this occupancy grid at time t as $OG_t[i]$ where $0 \leq i < N$ with N being the total cells of this occupancy grid. The value of each cell of this grid is between 0 and 1 i.e. $0 \leq OG_t[i] \leq 1$ and represents internal belief of the ego vehicle about the occupancy state of each cell with 0 means empty and 1 means occupied.

The output of the *XSens MTi-G* motion sensor installed on the demonstrator, at time instant t, consists of (along with other information) two components of velocity $v_t = (v_x, v_y)$ and values of quaternion components for orientation $Q_t = (q_0, q_1, q_2, q_3)$. From these information we calculate the translational and rotational velocities $u_t = (\nu_t, \omega_t)$ of the demonstrator vehicle as follows.

$$\nu_t = \sqrt{v_x^2 + v_y^2} \quad (2)$$

To calculate rotational velocity of the vehicle we calculate yaw angle of the vehicle from the quaternion as follows

$$\mathcal{Y} = atan2(2*(q0*q3+q1*q2), 1-2*(q2*q2+q3*q3)) \quad (3)$$

And if dt is the time difference between two successive data frames at time t and $t-1$ then rotational speed ω at time t is equal to the yaw rate given as:

$$\omega_t = \frac{\mathcal{Y}_t - \mathcal{Y}_{t-1}}{dt} \quad (4)$$

At each time instant t these OG_t and u_t are input to the algorithm that consists of following steps.

I) Free and occupied counts arrays: for each new input occupancy grid OG_t we create two count arrays, the first one called $FreeCount_t$ and the other called $OccupiedCount_t$ to keep count of the number of times a cell has been observed free and number of times it has been observed occupied respectively. These arrays are initialized from OG_t as follows:

$$OccupiedCount_t[i] = \begin{cases} 1, & \text{if } OG_t[i] > 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

and

$$FreeCount_t[i] = \begin{cases} 1, & \text{if } OG_t[i] < 0.5 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

II) Counts update from previous step: Suppose $FreeCount_{t-1}$ and $OccupiedCount_{t-1}$ are the updated counts arrays at time $t-1$ and we want to update new counts $FreeCount_t$ and $OccupiedCount_t$ from these old counts. Since vehicle has undergone a position change determined by $u_t = (\nu_t, \omega_t)$, so there is no direct correspondence between cells of two occupancy grids OG_t and OG_{t-1}. We must find transformations that map a cell in OG_{t-1} to a cell in OG_t using u_t. This situation is shown in figure 5, OG_{t-1} has origin at O_{t-1} and OG_t has origin at O_t. To find this transformation suppose $O_{t-1} = (x_{t-1}, y_{t-1}, \theta_{t-1}) = (0, 0, 0)$ is the pose (position and orientation) of the occupancy grid at time instant $t-1$ (i.e of OG_{t-1}) and we want to find $O_t = (x_t, y_t, \theta_t)$, the pose of OG_t under u_t. Considering a circular motion trajectory, the pose of O_t w.r.t O_{t-1} is given as:

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} \nu_t/\omega_t * sin(\omega_t * dt) \\ \nu_t/\omega_t - \nu_t/\omega_t * cos(\omega_t * dt) \\ \omega_t * dt \end{bmatrix} \quad (7)$$

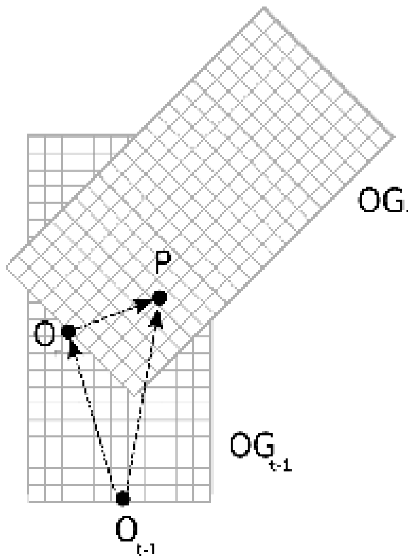

Fig. 5. Positon of the grid at time instants $t-1$ and t. Vehicle undergoes a motion of $u_t = (\nu_t, \omega_t)$ to move from O_{t-1} to O_t. We need to find the position of point P of grid OG_{t-1} in grid OG_t.

An important thing to note here is that we are concerned with the localization of two consecutive frames only. We do not solve the complete SLAM problem, making the technique very fast and avoiding the error accumulation over time. Moreover empirically observed odometry error between two consective frames is less than 10 cm whereas the cell size is 20cm x 20cm enabling us to assume that cell mappping (explained next) from grid at $t-1$ to grid at t is exact.

To map a cell of grid OG_{t-1} to grid OG_t we proceed as follows. Suppose point P (shown in figure 5) is the center of a cell in grid OG_{t-1} and we want to find its corresponding cell in grid OG_t. We define following two pose manipulation operations:

If P_{ij} is the pose of origin j w.r.t origin i and $P_{jk} = [x_{jk}, y_{jk}, \theta_{jk}]^T$ is the pose of origin k w.r.t j then the pose of k w.r.t i denoted as $P_{ik} = [x_{ik}, y_{ik}, \theta_{ik}]^T$ is given as:

$$P_{ik} \equiv \oplus(P_{ij}, P_{jk}) = \begin{bmatrix} x_{jk}\cos(\theta_{ij}) - y_{jk}\sin(\theta_{ij}) + x_{ij} \\ x_{jk}\sin(\theta_{ij}) + y_{jk}\cos(\theta_{ij}) + y_{ij} \\ \theta_{ij} + \theta_{jk} \end{bmatrix} \quad (8)$$

For the pose P_{ij} the reverse pose relationship $P_{ji} = [x_{ji}, y_{ji}, \theta_{ji}]^T$ (pose of i w.r.t j) is defined as:

$$P_{ji} \equiv \ominus(P_{ij}) = \begin{bmatrix} -x_{ij}\cos(\theta_{ij}) - y_{ij}\sin(\theta_{ij}) \\ x_{ij}\sin(\theta_{ij}) - y_{ij}\cos(\theta_{ij}) \\ -\theta_{ij} \end{bmatrix} \quad (9)$$

Since the pose of O_t w.r.t O_{t-1} is $P_{O_{t-1}O_t} = [x_t, y_t, \theta_t]^T$ and point P has pose $P_{O_{t-1}P} = [x, y, 0]^T$ w.r.t O_{t-1}. The pose of this point P w.r.t O_t is calculated as.

$$P_{O_tP} = \oplus(P_{O_tO_{t-1}}, P_{O_{t-1}P}) \quad (10)$$

or

$$P_{O_tP} = \oplus(\ominus(P_{O_{t-1}O_t}), P_{O_{t-1}P}) \quad (11)$$

First two components of P_{O_tP} give x and y positon of point P w.r.t origin O_t. From these x and y values we can easily calculate the index of the cell where point P lies in grid OG_t. These transformations will map a cell having index i in OG_{t-1} to a cell having index j in grid OG_t. If this cell j is visible in OG_t i.e $0 \leq j < N$ then we can update new count values for this cell as follows:

$$FreeCount_t[j] = FreeCount_t[j] + FreeCount_{t-1}[i] \quad (12)$$

and

$$OccupiedCount_t[j] = OccupiedCount_t[j] + OccupiedCount_{t-1}[i] \quad (13)$$

We repeat this process for all cells of grid OG_{t-1} to update counts values in grid OG_t.

III) Motion detection: After the counts arrays have been updated as explained above the motion grid can be calculated from the new data using following heuristic:

$$MotionGrid_t[i] = \begin{cases} 1, & OG_t[i] > 0.5 \text{ and} \\ & FreeCount_t[i] > 2 * OccupiedCount_t[i] \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

After this processing $MotionGrid_t$ has 1s in the cells which are detected as belonging to moving objects.

V. RESULTS

Some qualitative results are shown in figures 6, 7, 8 and 9 (rectangles around the objects are drawn manually to highlight them). Figure 6 shows the motion detection scenario of two cars. A car moving around a round point has been successfully detected in figure 7. Detection of two moving cars on a highway is shown in figure 8. We see some false positives as well but we believe that this noise can be easilyl removed by the later steps. Finally figure 9 shows the case when there is no moving object in the view, we see that no significant object is detected as moving. A video showing some more results of this work can be found here[1].

VI. CONCLUSION AND FUTURE WORK

In this paper we have presented a fast technique to find moving objects from laser data. The presented technique does not require to perform complete SLAM to detected moving objects but uses laser data along with odometry/IMU information to transfer occupancy information between two consecutive grids.

We plan to use this fast motion detection technique for the BOF framework to provide a priori motion information for the

[1]https://sites.google.com/site/qadeerbaig/motion-detection

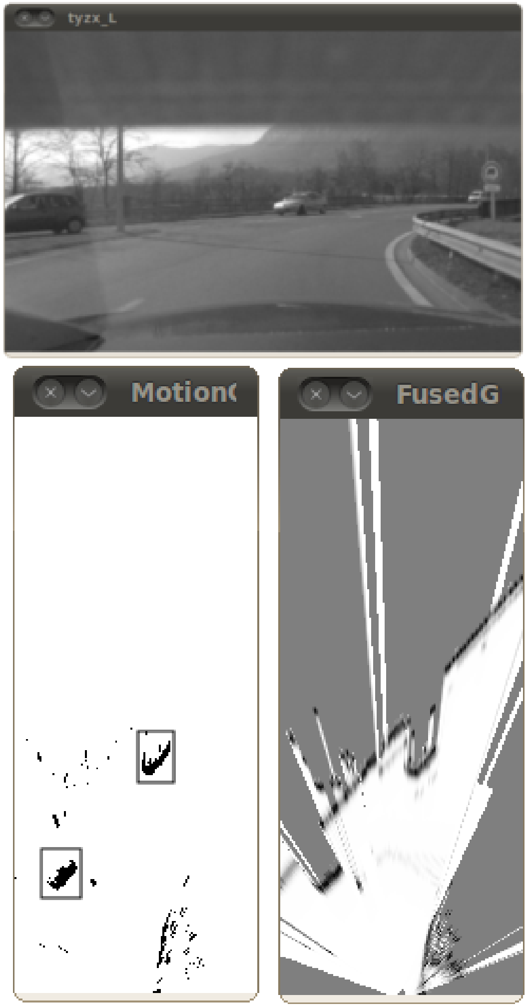

Fig. 6. Motion detection results of two cars. Top, scenario, bottom right input fused grid, bottom left resulting motion grid.

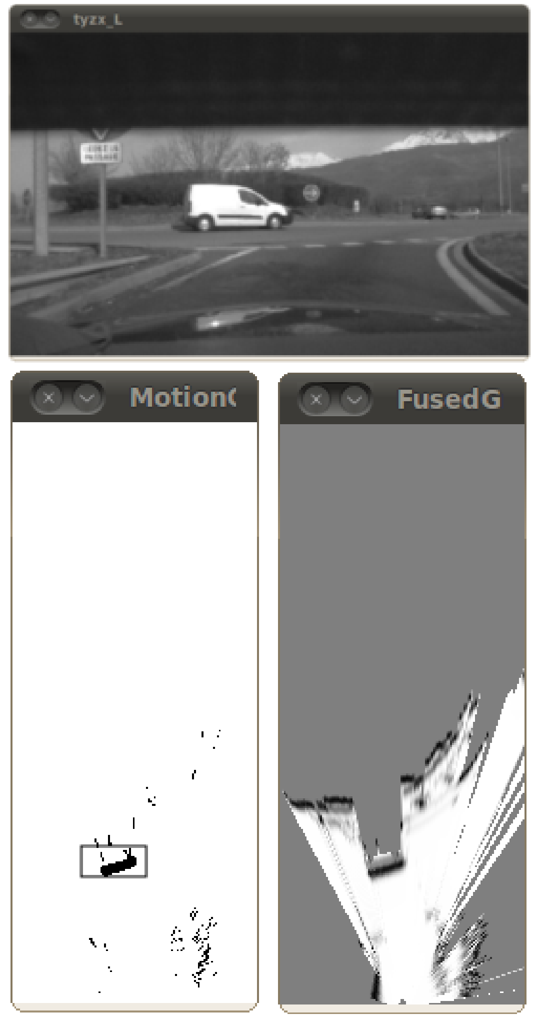

Fig. 7. Motion detection results of a car on a round point. Some noise due to sensor uncertainty is also visible

calculation of cell velocities. Currently BOF, in the absence of a motion sensor, uses a bayesian inference to calculate a probability distribution on a range of velocities for each cell requiring to perform calculations for cells which belong to actually static parts of the environment. With current work we will be able to reduce this processing time by limiting to those cells which have been detected as belonging to moving objects only. This will, in turn, result into an improved performance of FCTA algorithm which is based on BOF output.

REFERENCES

[1] C. Laugier, I. Paromtchik, M. Perrollaz, M. Yong, J. Yoder, C. Tay, K. Mekhnacha, and A. Negre, "Probabilistic analysis of dynamic scenes and collision risks assessment to improve driving safety," *Intelligent Transportation Systems Magazine, IEEE*, vol. 3, no. 4, pp. 4 –19, winter 2011.

[2] R. Jain, W. N. Martin, and J. K. Aggarwal, "Segmentation through the detection of changes due to motion," *Computer Graphics and Image Processing*, vol. 11, no. 1, pp. 13–34, September 1979.

[3] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119–152, 1994.

[4] D. Li, "Moving objects detection by block comparison," in *Electronics, Circuits and Systems, 2000. ICECS 2000. The 7th IEEE International Conference on*, vol. 1, 2000, pp. 341–344.

[5] S. Taleghani, S. Aslani, and S. Shiry, *Robust Moving Object Detection from a Moving Video Camera Using Neural Network and Kalman Filter*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 638–648. [Online]. Available: http://dl.acm.org/citation.cfm?id=1575210.1575268

[6] C.-C. Wang, C. Thorpe, and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas," vol. 1, 2003.

[7] J. Burlet, T. Vu, and O. Aycard, "Grid-based localization and online mapping with moving objects detection and tracking," INRIA-UJF, Tech. Rep., 2007.

[8] T. Vu, J. Burlet, and O. Aycard, "Grid-based localization and local mapping with moving objects detection and tracking," *International Journal on Information Fusion, Elsevier*, 2009, to appear.

[9] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2, pp. 123–139, 2009.

[10] T.-D. Vu and O. Aycard, "Lased-based detection and tracking moving object using data-driven markov chain monte carlo," in *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009.

[11] J. Moras, V. Cherfaoui, and P. Bonnifait, "Moving objects detection by conflict analysis in evidential grids," in *2011 Intelligent Vehicles*

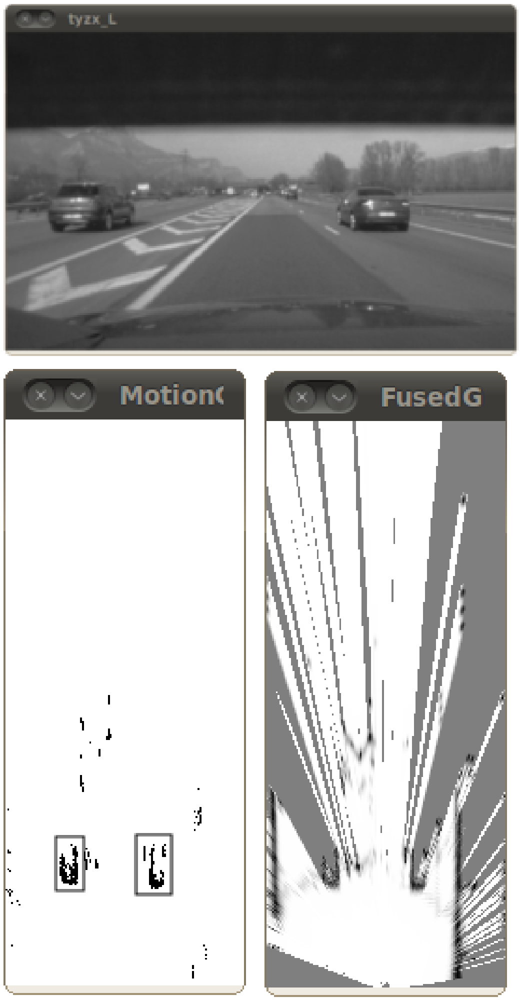

Fig. 8. Motion detection results of two cars on a highway.

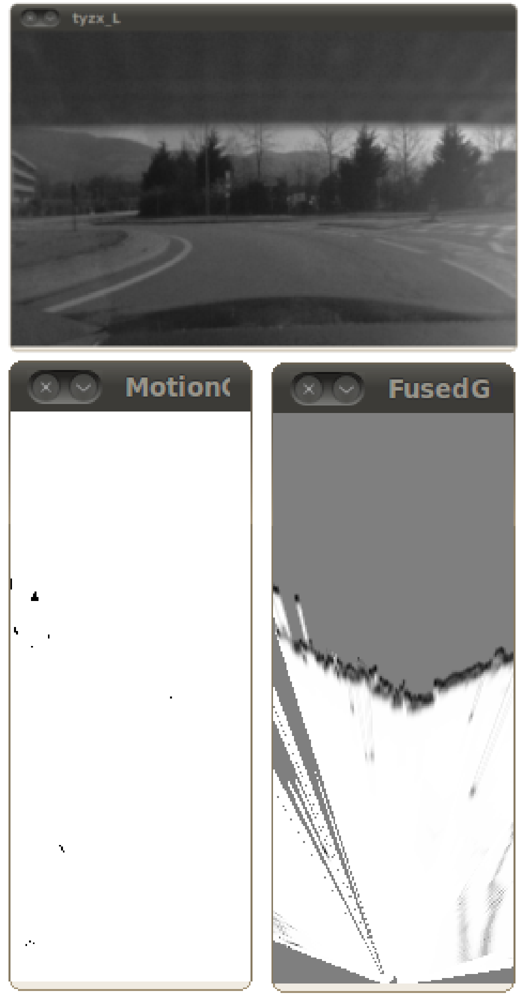

Fig. 9. Motion detection results with no moving object in the view.

Symposium, Baden-Baden, Allemagne, June 5-9 2011, pp. 1120–1125. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00615304/en/

[12] ——, "Credibilist occupancy grids for vehicle perception in dynamic environments," in *2011 IEEE International Conference on Robotics and Automation, Shanghai International Conference Center*, Shanghai, Chine, May 9-13 2011, pp. 84–89. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00615303/en/

[13] A. Elfes, "Occupancy grids: A stochastic spatial representation for active robot perception," in *Proceedings of the Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-90)*. New York, NY: Elsevier Science, 1990, pp. 136–146.

[14] C. Tay, K. Mekhnacha, C. Chen, M. Yguel, and C. Laugier, "An efficient formulation of the bayesian occupation filter for target tracking in dynamic environments," 2007, (Accepted) To be published. [Online]. Available: http://emotion.inrialpes.fr/bibemotion/2007/TMCYL07

[15] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessiere, "Bayesian Occupancy Filtering for Multitarget Tracking: an Automotive Application," *International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, Jan. 2006, voir basilic : http://emotion.inrialpes.fr/bibemotion/2006/CPLFB06/.

[16] K. Mekhnacha, Y. Mao, D. Raulo, and C. Laugier, "Bayesian occupancy filter based "Fast Clustering-Tracking" algorithm," in *IROS 2008*, Nice, France, 2008.

[17] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents series)*, ser. Intelligent robotics and autonomous agents. The MIT Press, Aug. 2005. [Online]. Available: http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0262201623

[18] J. D. Adarve, M. Perrollaz, A. Makris, and C. Laugier, "Computing Occupancy Grids from Multiple Sensors using Linear Opinion Pools," in *IEEE International Conference on Robotics and Automation*, St Paul, Minnesota, États-Unis, May 2012.

[19] M. H. DeGroot, "Reaching a consensus," *Journal of the American Statistical Association*, vol. 69, no. 345, pp. pp. 118–121, 1974. [Online]. Available: http://www.jstor.org/stable/2285509

[20] C. Tay, K. Mekhnacha, C. Chen, M. Yguel, and C. Laugier, "An Efficient Formulation of the Bayesian Occupation Filter for Target Tracking in Dynamic Environments," *International Journal Of Autonomous Vehicles*, 2007, voir basilic : http://emotion.inrialpes.fr/bibemotion/2007/TMCYL07/.