Computing Occupancy Grids from Multiple Sensors using Linear Opinion Pools

Juan David Adarve, Mathias Perrollaz, Alexandros Makris and Christian Laugier

***Abstract*— Perception is a key component for any robotic system. In this paper we present a method to construct occupancy grids by fusing sensory information using Linear Opinion Pools. We used lidar sensors and a stereo-vision system mounted on a vehicle to make the experiments. To perform the validation, we compared the proposed method with the fusion method previously used in the Bayesian Occupancy Filter framework, using real data taken from highway and urban scenarios. The results show that our method is better at dealing with conflicting information coming from the sensors. We propose an implementation on parallel hardware which allows real-time execution.**

I. Introduction

The task of perceiving the environment and building a representation of it is crucial for any robotic system. In the case of a robot equipped with several sensors, it is possible to combine the information coming from every sensor and to build an unique representation, on which upper layers of the system can be constructed.

In Robotics, occupancy grids are widely used as a tool to represent the environment. Grids are most often used for large scale environment mapping, but they are also more and more used in the Intelligent Vehicles community for compact representation of the near environment. As such, the occupancy grids correspond to the observed field of view in front of the vehicle. Early works like [8] and [5] compute occupancy grids by combining the likelihood functions of each sensor by using Bayesian reasoning. More recent works like the Bayesian Occupancy Filter [3] uses the Bayesian programming framework [2] to construct the grids, adding temporal filtering capability. All these works make use of the Bayesian theory to fuse the information coming from every sensor and by that, create a single representation of the environment. The probability of a cell C in the grid to be occupied given a set of sensor observations $\{Y_1 \dots Y_m\}$ is as follows:

$$P(C|Y_1 \dots Y_m) \propto P(C) \prod_{i=1}^{m} \frac{P(Y_i|C)}{P(Y_i)} \tag{1}$$

Using this Bayesian approach may carry some problems, though, specially when conflicting information has to be fused. As an example, consider that only one out of the total number of sensors is capable of observing a given region of the environment. If the value of all the other sensors has to be considered, then the result after the fusion will be strongly influenced by the opinion of the sensors actually unable of observing that specific region. To reduce the errors in case of conflicting sources, the authors in [9] use the Superbayesian Independent Opinion Pool formula. In [11], the fusion scheme also considers the differences in accuracy of all the sensors, and in [7], the authors make use of Dempster-Shafer theory to handle conflicting information.

The opinion reinforcement [1] could be solved in several ways. It is possible to make more complex models that consider the interactions between sensors, but using such approach would mean to model these interactions for every new sensor added or removed from the system. We chose instead to use the Linear Opinion Pool [4] as the method to fuse sensory information. The idea is to perform fusion as a weighted sum of sensor observations. The weight is estimated as a confidence on every sensor's observation. Our approach intends to split the complexity of the sensor modeling in two parts. One component is dedicated to the sensor model working on normal conditions, like a beam sensor perfectly parallel to the road. The second component models the confidence by considering external factors, not considered in the sensor model. The idea is that by combining these components, the opinion of a non reliable sensor is lowered by a low confidence and hence, the result of the fusion process will depend on those reliable sensors which receive a higher weight.

The paper is divided as follows: section 2 presents the proposed fusion method with the sensor models for lidar and stereo-vision; section 3 presents experimental results, as well as information about the parallel implementation of the method on the GPU to achieve real-time processing capability. Conclusion is given in section 4.

II. Method

Our method fuses the sensory information by using Linear Opinion Pools [4] [6]. The objective 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:

Juan David Adarve (juan.adarve@inrialpes.fr), Mathias Perrollaz (mathias.perrollaz@inrialpes.fr), Alexandros Makris and Christian Laugier are with the INRIA Grenoble.

$$P(C|Y_1 \dots Y_m) = \alpha \sum_{i=1}^{m} w_i(C) P(C|Y_i) \tag{2}$$

where $\alpha = \left[\sum_{i=1}^{m} w_i(C)\right]^{-1}$ is a normalization factor for the weights. We will use equation (2) 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 (2), for each cell in parallel.

A. Lidar sensor model

The model for the lidar sensor is based on the beam sensor model described in [12]. The noise in the measurements and the presence of unexpected objects are taken into account. We assume there is a function to know the correspondence between a given cell C in the Cartesian grid and a certain beam expressed in polar coordinates. For a given beam, let z be a random variable expressing the occupancy over the distance from the sensor. For a cell C situated at a distance z from the beam's source that impacts at a distance z^*, the occupancy probability is as follows:

$$P_{lidar}(z|Y_{lidar}) = \begin{cases} g(z, z^*) = \lambda e^{\frac{(z-z^*)^2}{2\sigma^2}}, \\ \text{for } z \in [0, z^*] \\ \\ max\{0.5, g(z, z^*)\}, \\ \text{for } z \in (z^*, z_{max}] \end{cases} \tag{3}$$

where λ is a scale factor for the Gaussian bell, z_{max} the maximum range of the beam and, σ^2 the variance in the measurement. Note that beyond the hit of the laser nothing can be said about the occupancy of a cell and hence, $P_{lidar}(z|Y_{lidar}) = 0.5$.

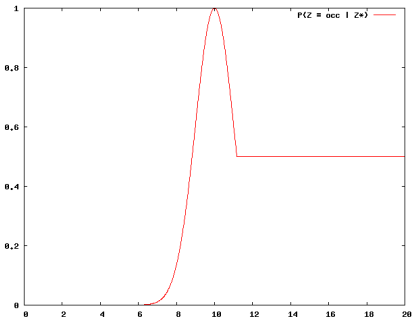

Fig. 1. $P_{lidar}(z|Y_{lidar})$ for a beam that hits an obstacle at $z = 10$

For the confidence function, we consider two different components: the first one, W^{hit}_{lidar}, considers the probability that an unexpected object is detected before the real hit of the beam. This is modeled by using an exponential distribution with parameter λ_{short}:

$$W^{hit}_{lidar}(z, z^*) = \begin{cases} 1 - \eta\lambda_{short} e^{-\lambda_{short} z}, \\ \text{for } z \in [0, z^*] \\ \\ \beta e^{\frac{-(z-z^*)^2}{2\sigma^2}}, \\ \text{for } z \in (z^*, z_{max}] \end{cases} \tag{4}$$

$\eta = \frac{1}{1-e^{-\lambda_{short} z^*}}$ being a normalization factor, and $\beta = 1 - \eta\lambda_{short} e^{\lambda_{short} z^*}$ the value of the exponential distribution at the hit position. After the hit, the confidence decreases following a Gaussian function.

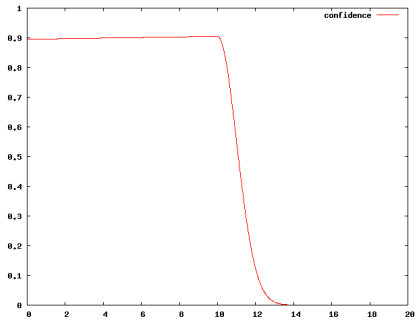

Fig. 2. $W^{hit}_{lidar}(z, z^*)$ for a beam that hits an obstacle at $z = 10$

The second component of the confidence considers the angle of inclination of the beam with respect to the road, meaning that the confidence for beams with different inclinations drops at different rate. Once a beam hits the road, any information beyond that position is not reliable. This is modeled by:

$$W^{inc}_{lidar}(z) = max\left\{1 + \frac{z \tan(\phi)}{h_0}, 0\right\} \tag{5}$$

where h_0 is the height of the beam at $z = 0$, ϕ is the angle between the beam and the road surface, considered flat.

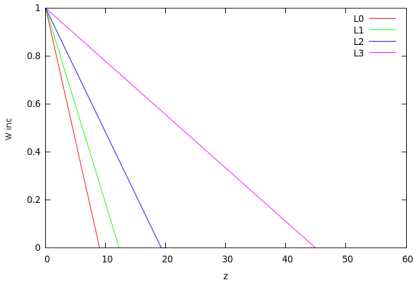

Fig. 3. $W^{inc}_{lidar}(z)$: The confidence of each lidar layer drops a different rates due to its angle with respect to the road.

The overall confidence for a given beam is:

$$W_{lidar}(z, z^*) = W^{hit}_{lidar}(z, z^*) \times W^{inc}_{lidar}(z) \tag{6}$$

B. Stereo-vision sensor model

For computation of occupancy grids with stereo-vision, our method is based on the visibility approach proposed in [10]. The idea is that an occupancy grid is computed directly in the u-disparity plane - which is a horizontal plane in the disparity space associated to the stereo camera - while taking into account the geometrical visibility of each cell. This allows to deal with partially occluded areas of the scene and to perform parallel computation. At the end of the process, the grid in u-disparity is remapped onto a Cartesian plane, in order to have an actual usable grid. The occupancy probability of a cell C in Cartesian space is given by:

$$P_{stereo}(C|Y_{stereo}) = T_{T_U \to T_X}(P(T_U)) \tag{7}$$

$T_{T_U \to T_X}$ is a transform from disparity to Cartesian space for the occupancy probability $P(T_U)$. Figure 4 shows an example of grid computed in disparity space.

(a)

(b)

Fig. 4. (a) Left image from the stereo camera. (b) corresponding occupancy grid computed in the u-disparity plane. Black regions are occupied, white regions are free and gray regions are unknown.

Similar to the method proposed for the lidar, the confidence function is splitted in two components. The first one deals with the visibility of regions, while the other component is related to the non-constant accuracy of stereo-vision:

$$W_{stereo}(C) = W^{vis}_{stereo}(C) \times W^{dist}_{stereo}(C) \tag{8}$$

W^{vis}_{stereo} depends on the visibility of the obstacles in the images. Like for the lidar, the idea is that the confidence shall be lower in regions of the space which are partially occluded. Figure 5 illustrates the concept of visibility: for a given distance/disparity value, numbers of possible (N_P) and visible (N_V) pixels can be measured in disparity space. Thus in the u-disparity plane:

$$Visibility(u,d) = \frac{N_V(u,d)}{N_P(u,d)} \tag{9}$$

W^{vis}_{stereo} is then obtained by remapping on the Cartesian plane the visibility values estimated during the disparity grid computation (see [10] for more details about this computation).

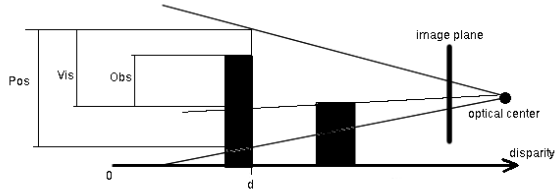

Fig. 5. Definition of the visibility used with stereo-vision.

W^{dist}_{stereo} is a confidence value related to the precision of stereo-vision. Therefore, it depends on the distance from the camera to the obstacle: it models the fact that stereo-vision works better at short distances, decreasing its accuracy when obstacles are farther. Let d_{max} be the maximum possible value for disparity in our system (i.e. the minimal perception range). W^{dist}_{stereo} is computed as:

$$W^{dist}_{stereo} = 1 - \frac{d^2_{max}}{d^2} \tag{10}$$

Figure 6 shows an example of weight maps computed for stereo-vision. On the visibility confidence map, it appears that the confidence is very low beyond the white vehicle, because it occludes the field of view. On the distance confidence map, it is clearly visible that the confidence decreases with the distance. The variation is quantified, because integer values of disparity are used for computation. This choice is made because with our approach, disparity is estimated over integer values (pixels).

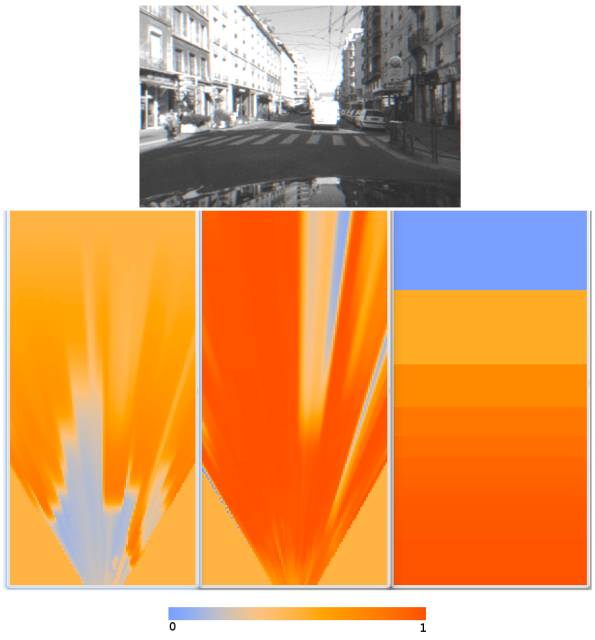

Fig. 6. Computation of the occupancy grid using stereo-vision: (up) left image from the stereo camera. (bottom, from left to right: occupancy grid, confidence based on visibility, confidence based on distance)

III. Experimental evaluation

A. Experimental setup

The approach has been tested on an experimental vehicle: a Lexus LS600h equipped with two IBEO Lux laser scanners mounted in the vehicle bumpers (figure 7). Each laser generates four layers of up to 200 beams. The angular range is $100°$, and the angular resolution is $0.5°$. For the stereo-vision system we use a TYZX stereo-camera mounted behind the wind-shield. The stereo baseline is 22 cm, with a field of view of $62°$. Camera resolution is 512×320 pixels with a focal length of 410 pixels. Real data were acquired by driving the vehicle in both road and urban environments.

Fig. 7. Experimental vehicle with the lidar sensors (bottom-left) and the stereo-vision camera (bottom-right).

The method was implemented on a Desktop computer with an Intel Core2 Duo microprocessor at 3Ghz with 3.7GB RAM and a Nvidia GeForce GTX 280 graphics card with 240 CUDA cores and 1GB of memory.

B. GPU implementation

Since real time is a critical feature in the domain of intelligent vehicles, we propose the implementation of the fusion method on GPU, using Nvidia CUDA technology. Here are some implementations choices. For the lidar sensor, only the list of laser hits is transfered from central memory to GPU memory, reducing to the minimum the memory transfer. On the GPU, the occupancy and confidence grids for each lidar layer are computed in Polar coordinate system. Since our approach is based on the independent beam model, this choice provides a convenient way to parallelize the process. Moreover, to speed up computations, look-up tables for equations (3) and (6) are created off-line. The values of these tables are indexed by (z, z^*). Once the grids in Polar coordinate system are computed, they are transformed into Cartesian grids. Such operation presents some difficulties because the number of beams that pass through a cell in Cartesian space depends on the distance from the source. If a line drawing method is used like the Bresenham algorithm, then those cells that are between two lines are not updated. Based on the results presented in [13], we use a texture mapping mechanism to transform the grids from polar to Cartesian coordinate system. This mechanism is supported by specialized hardware on the GPU which allows fast computation and its error with respect to the exact solution is not high.

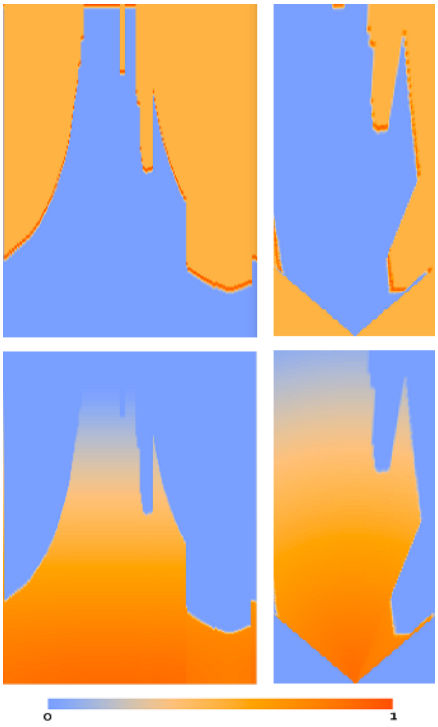

Fig. 8. On the left, the occupancy (top) and confidence (bottom) grids in Polar coordinate system. On the right, the grids in Cartesian coordinate system after applying the texture mapping.

The computation of occupancy grids from stereo-vision is also implemented on GPU. For the matching stage, disparity values are processed sequentially, by shifting the right image of the stereo pair. For each disparity value, the aggregation of the cost, using a correlation window, is done individually and parallel for each pixel. The u-disparity occupancy grid is computed by processing all the columns in parallel, since all the columns are independent.

C. Fusion results

To validate the fusion method, we made a qualitative comparison of the achieved results within the Bayesian Occupancy Filter (BOF) framework [2]. The BOF is an adaptation of Bayesian Filtering to the occupancy grid framework: the filter performs an estimation/prediction loop, takes occupancy grids as input, and outputs both filtered occupancy and velocity grids. It is therefore very useful for perception in dynamic environments.

We compared the filtered occupancy grids provided by the BOF in its original version against a modified BOF in which the estimation step is replaced by our fusion method, leaving

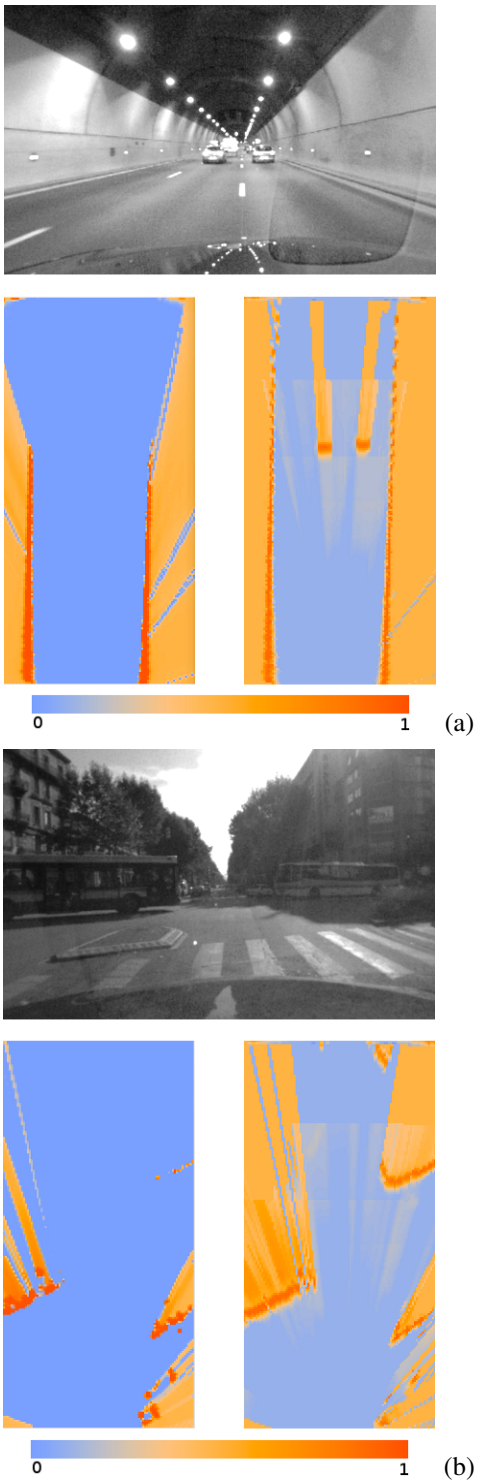

Fig. 9. (a) top: The scene viewed from one of the stereo camera. Bottom: left, occupancy grid using the original BOF; right: occupancy grid using the proposed method. Our method is able to detect the two vehicles ahead, which are seen just for a few layers of the lidars due to their inclination angle. (b) Urban scene with two buses. The BOF is able to see the bus on the left which is close to the vehicle while is able to see partially the bus on the right.

the prediction step intact. Figures 9 and 10 show examples of occupancy grids computed using both methods.

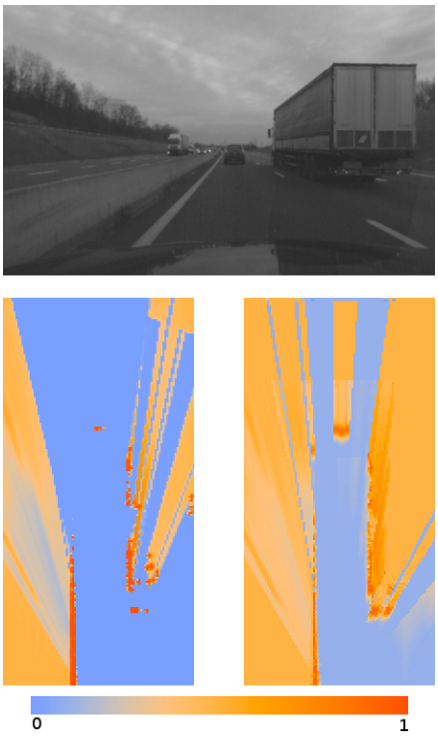

Fig. 10. On this road scene there is a vehicle in front and a truck on the right side. For the truck, some layers of the lidar hit the frame of the vehicle while others hit the tire, which makes the final result looking like that. Our method (bottom-right) makes the truck looks more like a whole.

The results achieved show that our method is better at dealing with conflicting information coming from the sensors. Particularly, in some specific situations, it allows to perceive objects that are not detected with the previous approach. On figure 9, we can see that some distant objects are not observed with the original method, because at least one layer of lidar considers the space as empty. This is fixed with the new fusion methodology. At the same time, it appears that the original version tends to give higher confidence to the lidars than to stereo-vision. Therefore, with our approach, stereo-vision adds details, but also little uncertainty (particularly at medium distance).

Figure 10 shows a typical issue with trucks and buses while using the original method: sometimes, one of the laser layers go between the road surface and the bottom of the truck. The sensor models interprets that as free-space (a beam can go through), which is in conflict with the upper layer of the laser, touching the back of the truck. The original methods tends to make the obstacle disappear from the grid, while our new approach solves the conflict and perceives the truck.

In a general way, results on many road sequences showed that our approach does better in cases of false measurements or conflicting information.

D. Execution time

An analysis about the execution time for the fusion algorithm was performed to see the behavior of our method when changing parameters such as the number of sensors to fuse or the grid resolution. For this analysis, we made use only of the lidar sensor due to the availability of several beam layers in our experimental vehicle which allows us to experiment with different number of grids to be fused. We consider the execution time as the time it takes to the system to give the results from the moment the list of hits for the lidar is transferred to the GPU until the fused grid is transferred from the GPU to main memory. The results show that real-time execution for the proposed method is possible.

In the first analysis we see the behavior of the algorithm against the number of sensors to be fused. For this, we use two different resolutions of the grid for an area of $40m \times 30m$: 200×150 with a cell size of $0.2m$, and 400×300 with cell size equal to $0.1m$. Figure 11 shows the average execution time when using varying the number of lidar layers from 1 to 8. Each layer is composed by a list of 200 impacts. The execution time grows linearly with respect to the number of sensors for both grid resolutions.

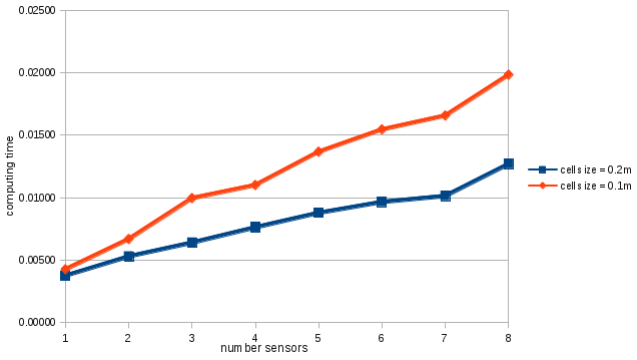

Fig. 11. Average computing time with respect to the number of sensors.

For the second analysis, we fixed the number of lidar layers to 8 and we made changes on the grid resolution. For doing so, we changed the cell size from $0.1m$ to $0.4m$ with a step size of $0.015m$. The grid resolution changed from 400×300 to 100×75 cells. Figure 12 shows the average execution time with respect to the number of cells. With small grids to be transferred to main memory, there is not a clear pattern, but after the cell number is greater than 60000 there appear a linear increase in the execution time with respect to the number of cells.

IV. Conclusion

A method to fuse sensory information on a mobile robot based on the Linear Opinion Pool has been proposed. The method is able to deal with conflicting information coming from the different sensors by assigning a low weight to unreliable or imprecise observations, making them of less importance during the fusion process. The method shows great improvements, because some not-detected parts of the scene are now observed. Moreover, the algorithm is designed for implementation on a parallel architecture and thus can it can run in real time.

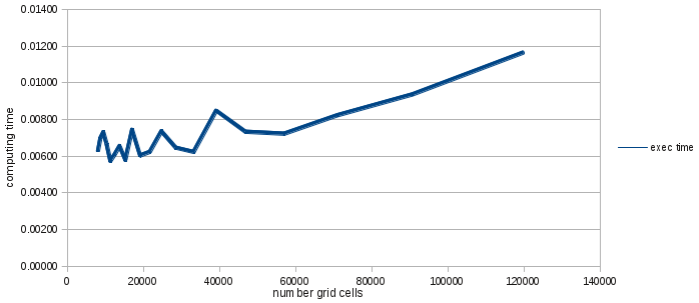

Fig. 12. Average computing time with respect to the number of cells.

Acknowledgment

This work has been done within the context of the Arosdyn project (*http://arosdyn.gforge.inria.fr/*). The authors would like to thank Toyota Motor Europe for their support of our experimental work on the vehicle.

References

[1] James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, 2010.

[2] P. Bessiere, C. Laugier, and Siegwart R. *Probabilistic Reasoning and Decision Making in Sensory-Motor Systems*. Springer, 2008.

[3] C. Coue, C. Pradalier, C. Laugier, Th. Fraichard, and P. Bessiere. Bayesian occupancy filtering for multi-target tracking: an automotive application. *International Journal of Robotic Research*, 25, January 2006.

[4] M. H DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.

[5] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46 –57, jun 1989.

[6] K. J. Mcconway. Marginalization and linear opinion pools. *Journal of the American Statistical Association*, 76(374):pp. 410–414, 1981.

[7] J. Moras, V. Cherfaoui, and P. Bonnifait. Credibilist occupancy grids for vehicel perception in dynamic environments. In *IEEE ICRA*, 2011.

[8] H. P Moravec. Sensor fusion in certainty grids for mobile robots. *AI magazine*, 9(2):61, 1988.

[9] K. Pathak, A. Birk, J. Poppinga, and S. Schwertfeger. 3d forward sensor modeling and application to occupancy grid based sensor fusion. In *IEEE/RSJ IROS*, 2007.

[10] M. Perrollaz, John-David Yoder, Anne Spalanzani, and Christian Laugier. Using the disparity space to compute occupancy grids from stereo-vision. In *IEEE IROS*, 2010.

[11] P. Stepan, M. Kulich, and L. Preucil. Robust data fusion with occupancy grid. 2005.

[12] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. Intelligent robotics and autonomous agents. The MIT Press, September 2005.

[13] Manuel Yguel, Olivier Aycard, and Christian Laugier. Efficient GPU-based Construction of Occupancy Grids Using several Laser Range-finders. *International Journal Of Autonomous Vehicles*, Volume 6, Number 1-2 / 2008:155 – 171, 2008.