

# Extended occupation grids for non-rigid moving objects tracking

Benjamin Lefaudeux, Fawzi Nashashibi, Gwennaël Gate

► **To cite this version:**

Benjamin Lefaudeux, Fawzi Nashashibi, Gwennaël Gate. Extended occupation grids for non-rigid moving objects tracking. Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on, IEEE, 2011. <hal-00662175>

**HAL Id: hal-00662175**

**<https://hal.inria.fr/hal-00662175>**

Submitted on 23 Jan 2012

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

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

# Extended occupation grids for non-rigid moving objects tracking

Benjamin Lefaudeux  
IMARA - INRIA Rocquencourt  
benjamin.lefaudeux@inria.fr

Gwennael Gate  
Aldebaran Robotics  
Email: gwennael.gate@m4x.org

Fawzi Nashashibi  
IMARA - INRIA Rocquencourt  
CAOR - Mines Paristech  
Email: fawzi.nashashibi@inria.fr

**Abstract**—We present an evolution of traditional occupancy grid algorithm, based on an extensive probabilistic calculus of the evolution of several variables on a cell neighbourhood. Occupancy, speed and classification are taken into account, the aim being to improve overall perception of an highly changing unstructured environment. Contrary to classical SLAM algorithms, no requisite is made on the amount of rigidity of the scene, and tracking do not rely on geometrical characteristics. We believe that this could have important applications in the automotive field, both from autonomous vehicle and driver assistance, in some areas difficult to address with current algorithms. This article begins with a general presentation of what we aim to do, along with considerations over traditional occupancy grids limits and their reasons. We will then present our proposition, and detail some of its key aspects, namely update rules and performance consequences. A second part will be more practical, and will begin with a brief presentation of the GPU implementation of the algorithm, before turning to sensor models and some results.

## I. OCCUPANCY GRIDS - EXTENDED

To begin with, we present in this part our motivations for this work, and some limits of current algorithms for the specific task which we aim to deal with. A comeback to the occupancy grid paradigm is our next point, before some thoughts about what could be an optimal occupancy grid update rule, and why its practical use is limited. A second part is devoted to our proposition, trying to tackle some of the observations previously made while dealing with some approximations and limits inherent to practical realisation. A last small part presents some performance considerations, which are important for scalability issues in case of collaborative work, and expected evolution with new computing hardware.

### A. General considerations

1) *Perception of non-structured moving objects*: Amid a vast set of localisation and perception algorithms currently developed, among which various SLAM have taken the greater share for the last years, some challenges are still untackled in the perception field. Detection and tracking of moving objects, without pre-requisites on rigidity, structured environment or immobility remains a tough question, and is not yet addressed by many approaches. Different SLAM-based approaches have been used for the last years, some of them being very successful in building a map of the environment from a non-associative sensor while dealing with the identification of some moving objects amid the map.

FastSLAM, using particle filtering and initially presented by Thrun et. al ([1]) or other iterative optimisation techniques (ICP, [2] [3]) are very powerful algorithms which can deal with some level of movement amid tracked points, as long as fixed points of the map are enough to estimate robot pose and environment mapping, and to isolate non conforming elements. This proved effective, for example by Wang et. al ([4]), in constrained cases where the number of moving parts remains low. Overall scene requisites are indeed strong using these techniques, and would probably not be enough for reliable moving object perception and tracking in a highly unstructured environment. Contrary to SLAM, our aim is thus not primarily to localise oneself in the environment along with building a map of its static features, but instead to be able to reliably detect and monitor position and dynamics of surrounding moving objects, possibly outdoor and without much visible solid infrastructure.

2) *Interests and limits of the simple grid-based approach*: Occupancy grids are very common in robotics, since their first introduction by Moravec and Elfes ([5]), initially related to sonar based mapping. Principles are simple and effective, relying on the information storage and information source spatial localisation similarities, thus allowing to keep any spatial relations between cells at a minimal cost. Occupancy grids were thus at the heart of most of the latests SLAM propositions (based on filtering or optimisation), and in many of broadly speaking perception systems (Badino et al [6] showed for example a promising free-space perception system based on vision and occupancy grids in 2007). Additional processing of information is present in those examples, due to the very limited nature of conventional grids data integration, usually relying on *large number law* for any kind of temporal filtering. This is one of the major approximations chosen in most occupancy grids to limit computational complexity, at the expense of information loss.

Another major approximation historically in the heart of occupancy grids is the cell independence pre-requisite, which greatly simplifies eventual probabilities calculus. Obviously a big step from accurate world description, this is basically of little consequences in case the aim is, as it was in the original Moravec article [5], the cartography of still environment. In case we chose the occupancy grid formalism to track moving objects, this is however a major limitation, as one could

expect the displacement of physically related points to be correlated. Those limitations are intricate, and could be seen as two limits stemming from the same cell independence (in time and space) from one iteration to another, on conventional grid-based processing and information storage.

3) *Optimal occupancy grids*: Considering a hypothetical occupancy grid propagation, one could wonder why an optimal Bayesian propagation would not be possible. The probability of any combined cell displacement could be weighted, thus computing the most-probable map prediction from a given set of measurement, allowing a temporal and spatial filtering, and a link between new measurements and known information. Trouble is, computing the formal probability is extremely demanding, if one is to consider every possible move. A complete probability calculus, considering a set of  $S$  possible state in any cell of a  $N * M$  grid, would imply the consideration of each and every possible map, which is  $S^{N*M}$ . Taking the smallest possible state sampling (2 states), and a ridiculously small map of 10\*10 cells, and the amount of maps to compute to get to the full solution is already beyond any reasonable range ( $2^{100}$ ).

The key to this complexity surely in a lot of useless probabilities being taken into account, even possibly beyond the causality principle. Anything at a given place have little to no chance of influencing the very next future of a very remote location, and this is the idea behind dealing with neighbourhood-restricted probabilities.

### B. Tackling inter-dependence within computing boundaries

We propose the use of extended occupancy grids, able to deal with part of the interactions between cells, in a common prediction/measurement Bayesian cycle. Our algorithm aims at taking into account both temporal and spatial relations between measurements, while keeping computing costs low enough to conceive a real-time use and concurrent use of other more specialised algorithms.

Firstly, we introduce the probability for every cell to move to its neighbourhood, given previous knowledge of the scene (occupation, speed, classification) and specific heuristics (separate cells cannot converge, nor can cells from the same object diverge). Secondly, we compare this prediction to a new measurement, and compute the most probable estimate given prediction and latest measurement. Thirdly, we update associated knowledge used in the prediction step, namely occupancy of every cell of the grid, speed, relation between cells (in a neighbourhood) or object classification used for different sensor models. Those principles were presented by Gate in [7], initially on a standard CPU implementation, and showed very promising results despite a high computing cost making it prohibitive for any real-time application.

1) *Bayesian filtering*: Bayesian filtering is at the heart of numerous recent publications, a lot of them being aimed at moving object perception and uncertainty handling. One

of the reference article in this domain would be from Coue et. al ([8]), in which a *Bayesian Occupancy Filter* is first presented. In this paper, every cell of the occupancy grid is part of a Bayesian network, whose transition probabilities will be estimated by the filter. The end result is an estimation of occupancy and velocity of every cell of the network, given past measures. This article has been followed by [9], which refines some of the concepts, but keeps most of the initial BOF computations. One difference with the algorithm we propose may be that we iteratively try to estimate the most probable association, or that we explicitly consider relations between close cells through rigidity and non-penetration constraints, thus giving a higher priority on spatial filtering. Another possibility to remove impossible moves from the filter estimation was proposed by Gindele in [10], using prior knowledge of the environment to exclude specific moves which could trick a Bayesian occupancy filter.

Another very interesting work closely related to our proposition would be the algorithm demonstrated by Danescu et. al in two recent papers ([11], [12]), in which cell speed is iteratively estimated by a particle filter. Similarly to BOF, this proposition is not based on the simple motion models and combination rules which we rely on, which could make some difference in dense scenes with occultation.

2) *Initial definitions*: Initial definitions of the used probability mass functions (*pmf*) are as follows :

- Mapping (occupancy) probability  $M_k(x_i) \in [0, 1]$  of the cell  $x_i$  in the spatial environment  $E$  at the time  $k$ , provided the measurements  $Z_{0:k} = \{z_0, ..z_k\}$  :

$$P(M_k(x_i) = 1 | Z_{0:k}) \quad \forall x_i \in E \quad (1)$$

- Vehicle localisation (including position and speed in  $E \times V$ ), at the iteration  $k$ . This is not yet addressed by the algorithm, and external sensors (odometers, wheel angle sensor) are used.

$$P(L_k = l_j | Z_{0:k}) \quad \forall l_j \in (E \times V) \quad (2)$$

- Association, *ie* the probability for a given cell from the iteration  $k-1$  to be associated with another given cell at the iteration  $k$ . In our case, only associations coming from a restricted neighbourhood are taken into account, which cuts the number evaluated map candidates from an exponential dependence on the number of cells to a more reasonable dependence on neighbourhood scale.

$$P(X_{k-1}^{next}(x_i) = x_j | M_{k-1}(x_i) = 1, Z_{0:k}) \quad \forall (x_i, x_j) \in E^2 \quad (3)$$

- Velocity probability of a cell, given its occupancy and previous measurements :

$$P(V_k(x_i) = v | M_k(x_i) = 1, Z_{0:k}) \quad \forall (x_i, v) \in E \times V \quad (4)$$

- Detection probability, to handle the probability that two given

cells  $x_i$  and  $x_j$  are part of the same object. The neighbourhood constraint limiting interactions to a finite part of the map is used once more, to limit the intricateness and heavy computing cost. Affiliation to a given object can however be “propagated” further than one cell’s neighbourhood, although our span is limited and this could prove to be a problem. Detection probability,  $D_k(x_i, x_j) \in [0, 1]$  is 1 if  $x_i$   $x_j$  are from the same object.

$$P(D_k(x_i, x_j) = 1 | M_k(x_i) = 1, M_k(x_j) = 1, Z_{0:k}) \\ \forall (x_i, x_j) \in E^2 \quad (5)$$

- Classification designs the probability of this set of cells to be part of a given class of objects (car, pedestrian, still object,...). We attempt to model this by matching extended characteristics of a set of cells (beyond geometrical characteristics for example) to a model. This conditions the update rule of association calculus, envisaged future positions of a cell being for example adapted from its class motion model. Classification probability are simultaneously kept from a different set of classes, a cell being capable of a partial fit with different classes. This ensures a more robust classification, initially prone to errors.

$$P(C_k(x_i) = c_j | M_k(x_i) = 1, Z_{0:k}) \quad \forall (C \times E) \quad (6)$$

3) *Update rules:* Having set these definitions, the proposed update rules make a number of approximations, in order to shorten computing time and adapt to existing sensors information :

- Associations are updated in a two-pass mechanism, making an extra initial assumption of independence between cells behaviour, which we attempt to correct in a second part. This behaviour was already present in the [7] proposition, and is a key to the possible use of massively parallel computing. Approximations are obviously primordial in our attempt to make the calculus feasible in real-time, but we believe most of the interactions between cells are mostly taken into account with this proposition.

First we then compute the local associations prediction, which is to say that macroscopic interactions are not yet taken into account :

$$P_{local}(X_{k-1}^{next}(x_j) | M_{k-1}(x_j) = 1, Z_{0:k}) = \\ \eta \cdot \underbrace{P_{local}(z_k | X_{k-1}^{next}(x_j), M_{k-1}(x_j) = 1, Z_{0:k-1})}_{Correction} \\ \cdot \underbrace{P_{local}(X_{k-1}^{next}(x_j) | M_{k-1}(x_j) = 1, Z_{0:k-1})}_{Prediction} \quad (7)$$

$\eta$  is here a normalisation constraint, to ensure that possible moves sum up to one for any given cell. Object previous speed and class are used to predict the asserted new positions. This could be seen, similarly to SLAM particle filters ([1]), as a new set of particles generated for every cell of the grid

iteratively, representing this cell’s occupancy possible next moves, depending on previous knowledge and motion model.

$$P_{local}(X_{k-1}^{next}(x_i) = x_j | M_{k-1}(x_j) = 1, Z_{0:k-1}) = \\ \Psi(x_j, x_i, V_{k-1}(x_j), C_{k-1}(x_j)) \quad (8)$$

The last measurement is then taken into account to produce an estimated local association, still without macroscopic constraints. Predictions are weighted according to the sensor model occupancy output, simply emphasising association estimation matching the measurement on a linear basis. Simple rule from conditional probability is thus applied, making the supposition that the result from our prediction process best represents the knowledge from the past.

Initial association estimation is then altered according to additional constraints : unlikely moves are penalised according to different heuristics (different cells cannot converge to the same place, cells from the same rigid object cannot diverge). Rigidity and non inter-penetration constraints are modelled by the potential function  $\Phi_{association}$ , which is currently based on two Gaussian weight function whose optimal width is still under investigation on real data.

$$P(X_{k-1}^{next}(x_j) = \hat{x} | M_{k-1}(x_j) = 1, Z_{0:k-1}) \\ \simeq \sum_{a_k \in A} \left\{ \prod_{1 \leq j \leq N} P_{local}(X_{k-1}^{next}(x_j) | M_{k-1}(x_i) = 0, Z_{0:k}) \right\} \\ \cdot \Phi_{association}(a_k, E, Z_{0:k}) \quad (9)$$

where  $A$  is the set of all possible associations. The process here described can be differently factorised, but was split into several summations in an attempt to increase its readability.

- Mapping is computed taking into account the two cases : in the cell is a newly observed presence, or the displacement (possibly null) of a previously seen cell. In the first case, the sensor model is the only input taken into account, in the second case contributions from all the possible associations are summed up to compute the predicted occupancy. Cell interactions have in this case already been taken into account in the association computation. The two cases are dissociated by a random variable  $S_k$ , which can take two values : 0 if the cell has never been seen, 1 if the cell has already been seen. Its probability is computed with association computation results : to sum up, if the considered cell corresponds to a local association maximum,  $P(S_k(x_i) = 1 | Z_{0:k})$  takes the value  $\gamma \in [0, 1]$ , else it takes the value  $1 - \gamma$ . The value of  $\gamma$  is chosen depending on the “renewal” rate of the map, that is to say “how often do we think a new object can appear from nowhere” ? A value around 0.5 has proven to work well in practice.

$$\begin{aligned}
P(M_k(x_i)|Z_{0:k}) = & \\
& P_{seen}(M_k(x_i)|S_k x_i = 1, Z_{0:k}) \cdot P(S_k(x_i) = 1|Z_{0:k}) \\
& + P_{unseen}(M_k(x_i)|S_k x_i = 0, Z_{0:k}) \\
& \cdot P(S_k(x_i) = 0|Z_{0:k}) \quad (10)
\end{aligned}$$

with the corresponding calculus (A being the cell neighbourhood) :

$$\begin{aligned}
P_{seen}(M_k(x_i)|Z_{0:k}) = & \\
\sum_{j \in A} \{ & P(X_{k-1}^{next}(x_j) = x_i | M_{k-1}(x_j) = 1, Z_{0:k-1}) \\
& \cdot P(M_{k-1}(x_j) = 1 | Z_{0:k-1}) \} \quad (11)
\end{aligned}$$

$P_{unseen}$  is in this case typically related to the sensor occupancy model.

- Velocities are computed taking into account the same two possibilities, depending if the observed cell is considered a new one, or the association of an already-observed cell to a new position :

- considering the velocity of already-observed cells, velocity is simply computed from the associations, summing up speed values stemming from all the possible contributors.
- considering appearing cells, the probability distribution of velocities had been proposed by Gate in [7] as follows, and kept in this proposition :

$$P(V_k(x_i)|S_k(x_i) = 0, Z_{0:k}) = \frac{1}{card(V)} \quad (12)$$

Merging of the two possibilities is done similarly to eq. 10.

- Detection update on the GPU-based algorithm is not yet on par with the initial proposition of G. Gate ([7]), which rightfully took into account updated velocity, mapping and associations estimates, as well as previous detection and classification values. For the sake of simplicity in this initial implementation, detection map update concerning *already seen observations* is currently done via cross-correlation of updated local associations *pmf*, thus focusing on rigidity links. This calculus, although a partial exploitation of available information, is very fast (a matter of milliseconds for a map of thousands of cells), albeit not precise enough to handle structured objects behaviour, and is to be refined in a future work.

- Classification updates can similarly be done using every gathered information (mapping, velocity, rigidity links,..) correspondence to a given sensor model, which would on the other hand improve prediction steps of the algorithm. This is not yet present in the GPU-based algorithm, and presented results can thus be seen as perfectible. There is however no theoretical constraints on this calculus, which should be in

place in our implementation algorithm in a short time for several classes (pedestrian, cars,..)

4) *Performance considerations*: As stated in eq. I-A3, complexity constraints on calculus are not to be neglected in occupancy grid update rules, many thinkable algorithms being simply not realistic. The complexity of the mechanism we propose can be computed as follows :

- for the sake of simplicity, we state a  $N * N$  grid, every cell being able to move in a  $M * M$  neighbourhood.

- considering the propagation of one cell, every possible move of every of its neighbours ( $O(M^4)$ ) are to be investigated for each individual envisaged propagation ( $O(M^2)$ ), which translates in  $O(M^6)$  complexity.

- cells updates being independent except for conformation rules already taken into account in the previous step, overall cell considerations is finally of  $O(N^2 \cdot M^6)$  complexity for the full update step.

It is here important to state that heavy approximations are still present when compared to an optimal calculus, namely that each and every cell moves are initially computed individually, although being later filtered to take some interactions into account. Computation is thus still not intricate, which keeps the complexity “low”, although  $O(N^2 \cdot M^6)$  remains a heavy burden for any realistic dimensions. Several points can be emphasised from this simple complexity calculus :

- although the computation remains heavy by all means, key approximations (namely cell propagation initial independence) have cut computing needs, and are moreover a key aspect for practical implementation. As such, the algorithm can easily be thought for massively parallel computing, every probability summation related to cell propagation being doable in a parallel and synchronised way. Our early tests using CUDA prove that real-time computing is possible on current hardware within reasonable grid dimensions. Faster implementation on FPGA should be possible on the same basis, the algorithm having proved to sustain heavy multi-threading

- digging into specific complexity aspects, our algorithm is also linear in complexity as regards the number of cells for a given neighbourhood, the size of which being responsible for most of the computational burden. This relative lightness in the grid size complexity is a key benefit for sharing applications : extending the size of the grid is not necessarily a problem as regards computational requirements, especially taking the massively parallel implementation which we chose (several GPU could be used to extend possible grid propagation). Extending the domain of tracked speed is however more complicated : tracked speed depends on the width of the move taken into account at every iteration, literally  $V_{max} = \frac{M/2 * \Delta}{T_{iteration}}$  with  $\Delta$  the spatial extension relative to one grid cell. Considering a given iteration maximum computing time (limited for example by the Lidar frequency for real-time operations), the maximum tracked speed is rapidly capped by the computing power at disposal relatively to the maximum computing time. We’ll see with our preliminary results that this translates to very

acceptable maximum speeds for our initial implementation on current hardware.

## II. PRACTICAL IMPLEMENTATION AND RESULTS

We present in this part some elements of the implementation we are currently testing and developing, which notably differs from the initial Gate publication ([7]) by its implementation, taking advantage of the CUDA API from nVidia Corp. A few remarks are initially made on this evolution, after which we explain the sensor model used to feed the algorithm, and expected sensor models to develop. We finally present some results obtained on actual data.

### A. GPU-based algorithm

The initial proposition by Gate and Nashashibi ([7]) being too much of a burden for current CPU processing despite its performances in tracking moving objects, we investigated different possibilities to make this algorithm a plausible real-time candidate. Due to the very parallel nature of its different information representations, a GPU-based implementation was started, using the nVidia API Cuda. The algorithm presented in this paper is thus partly different from the 2009 work, due to the fundamental differences between Cpu and Gpu processing abilities. Massively-parallel processing indeed comes with intrinsic limitations, being notably negatively affected by branching and conditional calculus. The main differences could be summarised as follows :

- cells differentiation into "key-cells" representing objects geometrical key points and standard cell (whose propagation is not investigated) could not be used within this non-branching restrictions, and the algorithm is now possibly more generic, considering displacements and constraints for each and every cell of the grid.
- cells neighbourhood, previously considered on-the-fly depending on their dynamic characteristics is now fixed, each and every cell keeping a given *influence area* on which interactions are considered. At stake is the capability of computing map updates in real-time with a neighbourhood far enough so that appropriate speeds are tracked. This reliability limit can easily be computed given the map pitch, sensor frequency, time between updates and corresponding neighbourhood.

### B. Some results

As usual when dealing with grid-based algorithms, sensor occupancy models are a key factor in our proposition. A standard simple Lidar occupancy model is used in our initial implementation, computed on GPU. Occupancy of areas in the shadow of Lidar impacts are chosen to 0.5 out of 1, neutral in our occupancy ratio. Although not entirely implemented yet, this algorithm is currently tested on real data involving Lidar measurements in an urban environment. Logging and replay framework is RTMaps software from Intempora, for every considered data. It is important to remember the exhaustiveness and generic nature of the algorithm : every mapped cell of the environment is considered equal, and no *a priori* is ever made on geometrical bounding, preferred

positions, structures, stationary or moving parts. Although still quite demanding on computing resources, the algorithm works in near real-time on current state-of-the-art hardware. The first scene computed below need 120ms to compute on a early 2010 GF100 GPU from nVidia (counting 448 CUDA cores), which should translate to well below 100 ms on current higher-end offerings. As regards performance, this algorithm is also an initial port, and could certainly be greatly improved. The second scene computed below used a larger neighbourhood, which drives to 1s calculus on the same hardware, but is not compulsory for this task. This can be seen as "best case" reasonably feasible if the algorithm was ported onto a dedicated FPGA for example.

1) *Algorithm memory*: In this example, we emphasise a temporary occlusion situation, where a pedestrian shadow hides another previously seen pedestrian (figures 1). Only the Lidar sensor is used in this case, camera captures being presented for illustrative purpose, along with bounding boxes. Lidar shadow occupancy is set to 0.3. On the *pmf* representations (figures 2 and 3), the point of view is from above, in a common "bird-view" perspective. All the boxes are drawn for illustrative purpose, we don't present here the output of a detection algorithm. The resolution of grid mapping is 15cm, speeds up to 4.5m/s being theoretically tracked. This last value can be improved without any computing cost by simultaneously degrading the spatial resolution and increasing the range of the measurements, which could be a dynamic trade-off depending on the vehicle speed.



Fig. 1. Successive camera views

The shadowed pedestrian is still clearly visible on the occupancy map, although the Lidar cannot get through the first occluding person. Its position spread over time, which shows that our knowledge decreases with the age of the data. We thus stress the importance of sensor filtering, taking into account possible spatial and temporal correlation. In this case, consequences of the pedestrian not being visible without filtering has no practical consequences, but this is not always the case and we believe that such filtering would be compulsory for autonomous vehicle navigation in an urban area.

2) *Dynamics estimation*: This example plans to emphasise dynamics estimation capabilities of the algorithm, along with segmentation of the scene. Although this is already possible via bounding boxes in the case of clearly separated persons, those matchings often miss when persons are too close to

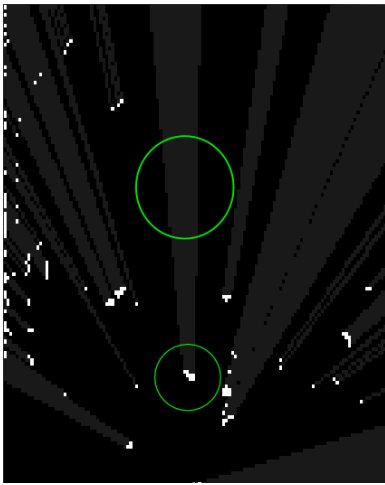


Fig. 2. Lidar output when occluded

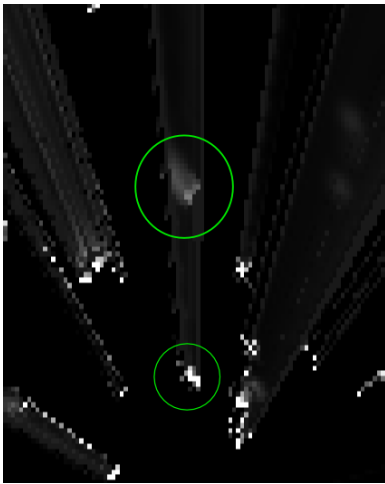


Fig. 3. Occupancy evaluation from our algorithm

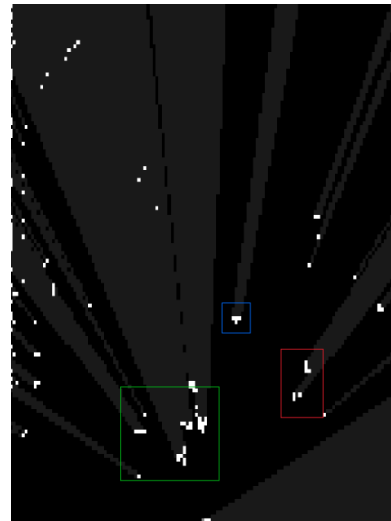


Fig. 5. Lidar sensor output

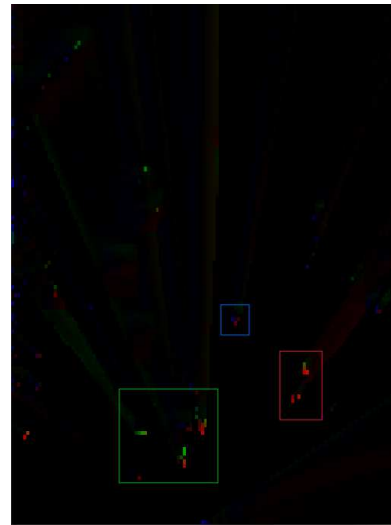


Fig. 6. Output of the algorithm - speed map

each other, or when groups change in size due to some people joining and leaving. The exhaustive approach that we carry on a per-cell basis insures a reliable estimation of the probable speeds. All boxes presented on figures 4 5 6 are here on illustrative purpose, and do not come from a detection algorithm.



Fig. 4. Camera capture

Figure 6 shows the speed map maintained by the algorithm,

speed orientation being coded by colour, while speed value (in the car referential) is represented by the brightness. Three boxes have been overlaid by hand on the figure, to emphasise specific cases. In green are the crossing pedestrians, which may have been difficult to track on sensor data alone (cf figure 5), due to the pushchair and their proximity. In blue is the road sign, which is obviously standing still, but which shows the residual speed of the vehicle (end of the braking sequence) and noise on Lidar data. It is barely visible on the speed map, due to the very low residual speed of the car. Segmentation of the scene after the algorithm process between moving and still parts proves effective. In red is the car coming on the other way, which is also going slow due to the crossing pedestrians, and have very few Lidar impacts (laser beam was oriented upwards, maybe too much). Tracking on geometrical grounds on Lidar data alone may have been difficult in this case.

### C. Future works

Alternative source of occupancy, velocity or classification are planned, mainly based on vision processing. Stereo-vision is for example a proven source of free-space measurement, as proven by *Moravec* once again in [13], or more recently *Badino et al.* using dense disparity calculus and occupancy grids ([6] and articles following). Motion detection and evaluation have also be proven to be a valuable output from stereo-vision capture, *Argrawal et al.* proving in [14] that platform ego-motion could also be removed from the initial optical flux in order to track moving objects. Initial theoretical work on this use would be from *Adiv et. al* ([15]), although this field has received a lot of attention in the past years, notably since dense stereo-vision processing is now possible in real-time. To finish with, the state of the art as regards vision-based SLAM (notably *Davison* [16] [17])) leads us to believe that laser-based sensors could possibly be replaced in the near future for most perception tasks, this being a strong incentive for us to develop visual inputs. Another source of possible improvements would be an evolution to handle collaborative perception, by means of merging grid-based beliefs in our algorithm.

### III. CONCLUSION

Perception of moving objects, such as pedestrians, with minimal requirements on their size, moves or behaviour is a difficult task ; which will nevertheless be a key to enable autonomous navigation in urban environments, or even comprehensive assistance on current automotive devices. We proposed a novel technique based on an extension of the traditional occupancy grids registration, using probabilist propagation and extensive consideration of displacements and interactions over a restricted neighbourhood.

An evolution from an initial ambitious proposition, our work shows promising results while being capable of real-time execution, although still being a work-in-progress. An interesting evolution would be its extension towards collaborative perception, which should be easier than for some other approaches due to the grid-based principle being kept.

This proposition alone would be far from cooperative perception of moving objects. Key problems are indeed to be considered for this development to occur, among which relative localisation inaccuracies, or synchronisation problems. However, the grid-based framework we propose is naturally simpler to use in information-sharing applications than some other techniques, among which common SLAMs based on Kalman-related filters. We finally believe this application could be an interesting development from current work.

### REFERENCES

- [1] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *Journal of Machine Learning Research*, vol. 4, no. 3, pp. 380–407, 2004. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.74.6632&rep=rep1&type=pdf>
- [2] C.-c. Wang and C. Thorpe, "A Hierarchical Object Based Representation for Simultaneous Localization and Mapping A Hierarchical Object Based Representation for Simultaneous Localization and Mapping," *Robotics*, 2004.
- [3] B. Steux and O. El Hamzaoui, "CoreSLAM: a SLAM Algorithm in less than 200 lines of C code," *Submission ICARCV*, 2010. [Online]. Available: [http://www-roc.inria.fr/imara/dw/\\_media/users/oussamaelhamzaoui/coreslam.pdf](http://www-roc.inria.fr/imara/dw/_media/users/oussamaelhamzaoui/coreslam.pdf)
- [4] C. C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous Localization, Mapping and Moving Object Tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, 2007. [Online]. Available: <http://ijr.sagepub.com/cgi/doi/10.1177/0278364907081229>
- [5] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2. IEEE, 1985, pp. 116–121. [Online]. Available: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1087316](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1087316)
- [6] H. Badino, U. Franke, and R. Mester, "Free space computation using stochastic occupancy grids and dynamic programming," in *Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil*. Rio de Janeiro: Citeseer, 2007, pp. 1–12. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.4161&rep=rep1&type=pdf>
- [7] G. Gate, a. Breheret, and F. Nashashibi, "Fast Pedestrian Detection in Dense Environment with a Laser Scanner and a Camera," *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, pp. 1–6, Apr. 2009. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5073555>
- [8] C. Coue, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessiere, "Bayesian Occupancy Filtering for Multitarget Tracking: An Automotive Application," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, Jan. 2006. [Online]. Available: <http://ijr.sagepub.com/cgi/doi/10.1177/0278364906061158>
- [9] C. Chen, C. Tay, C. Laugier, and K. Mekhnacha, "Dynamic Environment Modeling with Gridmap: A Multiple-Object Tracking Application," *2006 9th International Conference on Control, Automation, Robotics and Vision*, pp. 1–6, 2006. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4150365>
- [10] T. Gindele, S. Brechtel, J. Schroder, and R. Dillmann, "Bayesian Occupancy grid Filter for dynamic environments using prior map knowledge," *2009 IEEE Intelligent Vehicles Symposium*, pp. 669–676, Jun. 2009. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5164357>
- [11] R. Danescu and F. Oniga, "Particle grid tracking system for stereovision based environment perception," *Intelligent Vehicles Symposium* (. [Online]. Available: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=5548076](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5548076)
- [12] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and Tracking the Driving Environment with a Particle Based Occupancy Grid," *users.utcluj.ro*, 2011. [Online]. Available: <http://users.utcluj.ro/~rdanescu/final-danescu-its2011.pdf>
- [13] H. Moravec, "Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids," *Perception*, no. September, 1996. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.15.9.9045&rep=rep1&type=pdf>
- [14] M. Agrawal, K. Konolige, and L. Iocchi, "Real-time detection of independent motion using stereo," in *Motion and Video Computing, 2005. WACV/MOTIONS'05 Volume 2. IEEE Workshop on*, vol. 2. IEEE, 2007, pp. 207–214. [Online]. Available: [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=4129607](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4129607)
- [15] G. Adiv, "Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 4, pp. 384–401, Jul. 1985. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4767678>
- [16] A. J. Davison and N. Kita, "3D Simultaneous Localisation and Map-Building Using Active Vision for a Robot Moving on Undulating Terrain (PDF)," *Measurement*, vol. 00, no. C, 2001. [Online]. Available: <http://www.computer.org/portal/web/csdl/doi/10.1109/CVPR.2001.990501>
- [17] A. J. Davison, "Real-Time Simultaneous Localisation and Mapping with a Single Camera," London, pp. 0–7, 2003.