HAL
open science

Conditional Monte Carlo Dense Occupancy Tracker

Lukas Rummelhard, Amaury Nègre, Christian Laugier

To cite this version:

Lukas Rummelhard, Amaury Nègre, Christian Laugier. Conditional Monte Carlo Dense Occupancy Tracker. 18th IEEE International Conference on Intelligent Transportation Systems, Sep 2015, Las Palmas, Spain. hal-01205298

HAL Id: hal-01205298

https://inria.hal.science/hal-01205298

Submitted on 25 Sep 2015

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.

Conditional Monte Carlo Dense Occupancy Tracker

Lukas Rummelhard*, Amaury Nègre†*, Christian Laugier*
*Inria Grenoble - Rhône-Alpes, surname.name@inria.fr
†CNRS, Grenoble Informatics Laboratory, surname.name@imag.fr

***Abstract*—Proper modeling of dynamic environments is a core task in the field of intelligent vehicles. The most common approaches involve the modeling of moving objects, through Detection And Tracking of Moving Objects (DATMO) methods. An alternative to a classic object model framework is the occupancy grid filtering domain. Instead of segmenting the scene into objects and track them, the environment is represented as a regular grid of occupancy, in which spatial occupancy is tracked at a sub-object level. In this paper, we present the Conditional Monte Carlo Dense Occupancy Tracker, a generic spatial occupancy tracker, which infers dynamics of the scene through an hybrid representation of the environment, consisting of static occupancy, dynamic occupancy, empty spaces and unknown areas. This differentiation enables the use of state specific models (classic occupancy grid for motion-less components, set of moving particles for dynamic occupancy) as well as proper confidence estimation and management of data-less areas. The approach leads to a compact model that drastically improves the accuracy of the results and the global efficiency in comparison to previous methods.**

I. Introduction and Related Works

For most robotics or intelligent vehicles, the ability for the autonomous agent to accurately perceive and model in a pertinent form its surrounding environment is a core task. Whether it be for navigation, collision awareness, intention planning or mapping, this perception step is a challenging requirement, in term of accuracy, complexity and uncertainty management. As for human sensitive aptitude, the quality of the environment modeling rely on sensor measurement, but in particular on interpretation schemes. Indeed, sensor errors, occlusions, data contradictions, heavily complex settings are to be dealt with. Probabilistic methods were designed to formally introduce in the models the uncertainties and prior knowledge.

When confronted to moving objects, many issues are raised, the field of dynamic environment modeling still being active [1]. A classic way to address this problem is to adopt an object-based representation, which leads to multiple targets object tracking literature [2], [3]. Another common approach is the field of occupancy grids [4], [5], which works on spatial occupancy without higher level segmentation. This approach presents significant advantages. The model is by design spatially dense, and properly represents information about free space, which is an important data in mobile robotics and intelligent vehicles. Furthermore the delicate data segmentation and recognition step required in object-based representation can be avoided.

The field of occupancy grid based interpretation of the environment is a developed study domain, overlapping various applications, such as intelligent environment management, automatic autonomous navigation or extended vehicle perception. The aim is to produce a compact, regularly subdivided, probabilistic estimation of the spatial occupancy, without requiring the concept of objects. When applied to dynamic environments, it is usually necessary to enrich this representation by estimating velocity information for each cell.

In case of methods specifically designed for specific sensors, it is possible to derive velocity information from the sensors measurements. For instance, in Pfeiffer *et al.* [6], the authors integrate both occupancy and velocity information from both stereo-vision and optical flow in a medium level representation called dynamic stixel. This approach, between grid level and object level representation, shows particular advantages for traffic application. But it is specifically designed for stereo-vision.

In many practical cases, in which are not *a priori* known the type surroundings or the used sensors, a more generic approach is necessary. The occupancy grid is well designed for this purpose, but this structure implies the inference of cell dynamics. The Bayesian Occupancy Filter [7] is a generic Bayesian framework which updates a dynamic occupancy grid by filtering the occupancy and velocity in each cell in parallel. The observation model is incorporated as an instantaneous occupancy grid generated by mapping the sensor data over a grid using a probabilistic sensor model [8]. A Bayesian filtering methodology, based on a prediction/correction loop is used to filter the grid and estimate a velocity probability distribution for each cell, in the form of a neighbourhood transition histogram. This particular motion discretization design is convenient to model the cells evolution over the grid, but it leads to high computational costs and aliasing issues.

The BOFUM [9] proposes to use prior map knowledge to accelerate the convergence of the BOF, and then potentially reduce the required data dimensions. The practical application of this algorithm is limited by the availability of such maps and the necessity of a very precise localization in them.

To reduce the motion field dimensions, the histogram in each cell being mainly empty, a sampling of the motion distribution can be used. To this end, an importance sampling method has been proposed [10]. This allows to highly reduce the dimension of the dynamics representation. However, by drawing a fixed number of discrete velocity vectors per cell the aliasing problem is not addressed, and the computation resources are not optimally distributed.

A more adaptive method can still drastically decrease the motion representation dimension. As most cells, like empty or static ones, do not necessitate elaborated representation, important processing resources can be saved by taking this information into consideration. In the process presented by Danescu *et al.* [11], the idea to use a variable number of samples per cell, according to the occupancy estimated, is

implemented. The scene is represented by a set of moving particles having non-discretized positions and values, which corrects most of aliasing problems. The number of particles per cell corresponds to the occupancy of the cell, which leads to a very compact model. Nevertheless, a huge number of particles are still stated to represent static objects. Furthermore this approach would deserve a generic Bayesian formalization.

An hybrid approach is presented in [12], which proposes an interaction between grid-based and object-based representation. In this work, the data is mapped into a grid which integrates the information of static/dynamic classification, while the velocity is estimated at the object level, taking advantage of both geometric and dynamic model of the objects. This is properly designed for on road applications but not as general as a generic grid filter.

The Hybrid-Sampling Bayesian Occupancy Filter (HSBOF)[13] modifies most of BOF structures and analyses the scene through a static-dynamic duality. The static part is an occupancy grid structure, and the dynamic field is modeled by moving particles. Both sections are jointly generated and evaluated, their distribution over cells being adjusted. It permits a way more efficient calculation, allowing to focus the velocity computation on the dynamic components of the environment. This approach leads to a way more compact model and a drastic improvement of the system accuracy, particularly in term of velocities. However, many particles are still allocated to irrelevant areas, as no specific representation models dataless areas. Moreover if the filtered low level representation can directly be used for various applications, for example mapping process, short term risk assessment [14], [15], etc. the retrospective object analysis by dynamic grid segmentation can be computationally expensive.

This paper presents the Conditional Monte Carlo Dense Occupancy Tracker (CMCDOT), an evolution of the HSBOF which answers these critics. The main idea is to introduce formal states in the filtering process, representing static occupancy, dynamic occupancy, empty space and unknown areas. This new formalism leads to a more compact presentation of the overall algorithm (in term of transitions, initialisation, etc.). The unknown state enables specific processings, and to focus even more the particle set on relevant dynamic areas. Moreover, a fast dynamic occupancy clustering algorithm is directly incorporated in the global process.

II. CMCDOT Formalisation

The objective of the Bayesian Occupancy Filter (BOF) techniques (original BOF, HSBOF, CMCDOT) is to estimate the spatial occupancy and its dynamics of an environment, observed using various types of sensors. To do so, the space is divided in spatial cells, and to each cell are associated random variables, whose probability distributions are recursively estimated and used as basis of the scene interpretation. The formal probabilistic model used in the HSBOF and the original BOF could be considered similar : despite differences in the joint distribution decomposition and parametric expressions, the main improvements were not in the model equations, but in the static/dynamic distinction, the differentiated representation and equation solving. In the CMCDOT, an important change is introduced in the formulas : instead of directly filtering the occupancy, hidden states representing what is present in a cell are added. The occupancy distribution can then be inferred from those hidden states. Besides presenting a neater distinction between static and dynamic parts, the main interest of this modification is to introduce a specific processing of dataless areas, excluding them from the velocity estimation and disabling their temporal persistence.

The formalism used in the description comes from the Bayesian programming framework [16]. Given the set of random variables, the expression of their joint probability decomposition, the wanted distribution can easily be expressed, as presented in the following.

A. Variable definition

C :	Index that identify each 2D cell.
C^{-1} :	Index that identify each antecedent of a cell.
S :	State of the cell at current time. Possible states are s for *"occupied by a static object"*, d for *"occupied by a dynamic object"*, e for *"empty"* and u for *"unknown"*
S^{-1} :	State of the antecedent of the cell at previous time.
O :	Occupancy of the cell at current time. Its possible values are $\{occ, emp\}$.
V :	Velocity of the cell at current time. Its possible values are in $\mathbb{R}^2$.
V^{-1} :	Velocity of the antecedent cell at previous time.
Z :	Sensor measurement.

B. Joint distribution

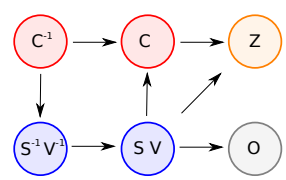

Fig. 1. Bayesian network representing the variable dependencies. The occupancy O is directly inferred from the state S.

$$\begin{aligned} P(CC^{-1}SS^{-1}VV^{-1}Z) \quad = \quad & P(C^{-1}) \\ & P(S^{-1}V^{-1}|C^{-1}) \\ & P(SV|S^{-1}V^{-1}) \\ & P(C|C^{-1}V) \\ & P(Z|S,V,C) \end{aligned} \tag{1}$$

The variable dependencies are pictured as a Bayesian network on fig 1. Each expression can be interpreted as follows:

- $P(C^{-1})$ is the distribution over all possible antecedents of the cell. It is chosen to be uniform as the cell is considered reachable from all the possible antecedents with equal probability.
- $P(S^{-1}V^{-1}|C^{-1})$ is the conditional joint distribution over the state and velocity of the antecedents. This distribution is updated at each time step.
- $P(SV|S^{-1}V^{-1})$ is the prediction model. The states and velocity are inseparable, as the state definition is directly connected to the velocity: velocity in static

parts is zero, while free and undefined parts have no velocity associated. The prediction step is further explained in part III/B.

- $P(C|C^{-1}V)$ is the distribution that explains if the cell c is reachable from the antecedent $[C^{-1} = c^{-1}]$ with the velocity $[V = v]$. This distribution is a Dirac with value equal to one if and only if $c_X^{-1} + v_X \delta t \in c$.
- $P(Z|S, V, C)$ is the distribution over the sensor measurement values. The model used is a classic sensor model for static, dynamic and free states, while the unknown state model depicts the dataless areas. Further explanations are in part III/C.

C. Problem expression

The aim of the Bayesian filtering process is to estimate the occupancy and velocity with respect to the current observation for every cell: $P(OV|ZC)$. In the system, the filtering is applied on the hidden states, the occupancy can then be deducted. By using a discretization of the antecedent cells and of the velocities, the filtering equation can be written:

$$P(SV|ZC) = \frac{\sum\limits_{C^{-1}S^{-1}V^{-1}} P(CC^{-1}SS^{-1}VV^{-1}Z)}{\sum\limits_{C^{-1}SS^{-1}VV^{-1}} P(CC^{-1}SS^{-1}VV^{-1}Z)} \quad (2)$$

In the end, the distribution can be expressed as:

$$P(SV|ZC) = \lambda \sum_{C^{-1}S^{-1}V^{-1}} P(C^{-1})P(S^{-1}V^{-1}|C^{-1}) \\ P(SV|S^{-1}V^{-1})P(C|C^{-1}V)P(Z|S, V, C) \quad (3)$$

Stating that :

$$P(O = occ|S = s) = 1 \qquad P(O = occ|S = d) = 1$$
$$P(O = occ|S = e) = 0 \qquad P(O = occ|S = u) = 0.5$$

Then the occupancy can be deducted:

$$P(OV|ZC) = \sum_S P(O|S)P(SV|ZC) \quad (4)$$

III. CMCDOT Resolution

The filtering process mainly consists in two steps, prediction and evaluation. The way those steps are handled differs according to the type of the considered state. This adapted management is at the core of the method performances.

A. State specific representation

The CMCDOT models the state distributions of cells in a grid, each cell is potentially occupied by a static object (s), occupied by a dynamic object (d), empty (e) or undefined (u). The s state refers to motionless occupancy, which includes strictly speaking static objects, but also background data (buildings, etc.). The static part is represented in a classic probabilistic grid. The d state refers to dynamic occupancy, represented by a set of weighted particles which sample the cell velocity. The e state corresponds to the free space, and is described in a classic probabilistic grid. The u state is a state used to signify the lack of information, and then the lack of confidence in the credibility of the other states. It is also represented in a classic probabilistic grid. The whole representation is summarized in fig 2.

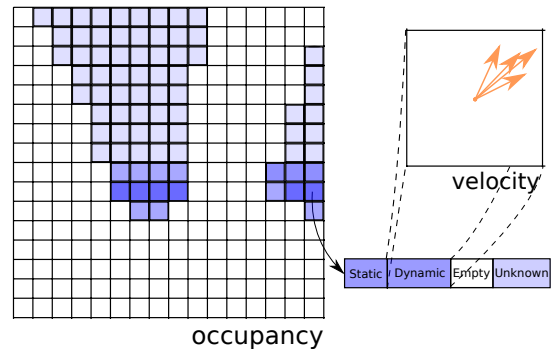

Fig. 2. Data representations in CMCDOT formulation. The environment is divided into cells, to which are associated static, dynamic, empty and unknown coefficients. The dynamic part is allotted to weighted particles which sample the velocity space

B. Prediction

The first step of the CMCDOT model prediction consists of the pure state prediction (from $P(SV|S^{-1}V^{-1})$), characterized to maintain focus on occupied areas while enabling fast changes in empty and especially in unknown ones. How the prediction is handled differs according to the considered state.

The static part model is highly conservative, a static component is assumed to remain static, or in a restricted portion to become dynamic (hypothetical set in motion of previously stationary elements). The velocity distribution of this newly generated dynamic part is supposed to be as a predefined initialization distribution (a uniform distribution in a 2D sphere, limited by a context dependant maximal velocity value, is often used). The sampling of new particles may occur in the re-sampling part, at the end of the update loop.

The dynamic state, embodied by moving particles, is also considered highly conservative, it can partially turn into static if the corresponding velocity is too weak. A Gaussian noise is applied to the velocity samples, to simulate velocity variation over time, and the location of the particles is updated in accordance with the update frequency.

While the two states representing occupied areas are durable, the empty state is seen to be way less persistent. Actually, the assumption of an empty area to remain empty is likely to be true, but with important uncertainty, as it strongly depends on the global environment evolution. This consideration is reflected on state transition coefficients, which allow a significant depreciation into the unknown state, which represents the lack of information.

Finally the unknown state is meant by definition to be volatile, its propagation in time weak. The unknown part can easily transform into free, static or dynamic state, with a initialization velocity distribution not yet sampled. An example of simplified transition matrix is given on fig 3.

Once the model is projected in time, the next step is to express it in the new reference frame. Indeed, as in most applications the perception device is mobile, the ego-motion has to be integrated. The ego-motion estimation can come from a sensor (odometry measurement, an Inertial measurement unit, GPS...) or generated by a dedicated algorithm (a SLAM-like method, using the CMCDOT outputs, has been developed, but its description exceeds the subject of this paper). When this displacement is known, the operation simply consists in transformation of the particle vectors (translation and rotation for positions, rotation for velocity) and interpolation of grids. The areas in the grid corresponding to newly discovered regions are set to undefined, while taken out areas can either be simply forgotten (in many mobile robot applications, this data storage is irrelevant) or used to generate a map based on the static part. Fig 4 summarizes the prediction process.

	$S^{-1} = s$	$S^{-1} = d$	$S^{-1} = e$	$S^{-1} = u$
$S = s$	**0.99**	**f(v)**	**0**	**0.05**
$S = d$	**0.01**	**1 - f(v)**	**0**	**0.05**
$S = e$	**0**	**0**	**0.90**	**0.10**
$S = u$	**0**	**0**	**0.10**	**0.80**

Fig. 3. Example of transition coefficients from S^{-1} to S. The transition from $S^{-1} = d$ to d or s depends on the velocity, to integrate slow motion parts in the static representation. $f(v) = 1$ when $v = 0$, and $f(v) = 0$ when v is high (f is typically a Gaussian centered on 0)

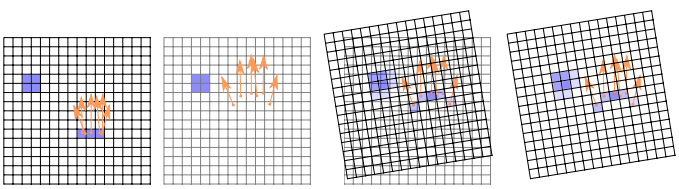

Fig. 4. Prediction step pattern. From left to right : 1. previous time step estimation. 2. state prediction and occupancy shifting according to motion models. 3. the data is expressed in the new reference frame, by interpolation of grids and particle accumulation for dynamic parts. 4. current time predicted model.

C. Evaluation

The updated distributions over states are evaluated in parallel in each cell of the current grid. In order to do so, the predicted states related to a cell are identified. The static, empty, undefined and dynamic initialization predictions are taken in the respective grids. Dynamic predictions are accumulated from moving particles located in the cell. In a given cell c, according to equation (3), those predictions are confronted with the observation. The observation model used is similar to classic probabilistic sensor model [8], the core originality is the evaluation of "unknown" state, which correlate with the relevance of sensor data. Typically, in case of laser data, the "unknown" model scores low values before and around the impact, and high values further (where no information is available). The sensor models are depicted in fig 5. While computing state distribution evaluations, each particle weight is also updated and normalized, according to its relative contribution in the dynamic part.

D. Particle re-sampling

Once the state distributions over cells are properly estimated, the remaining task is the dynamic part sampling.

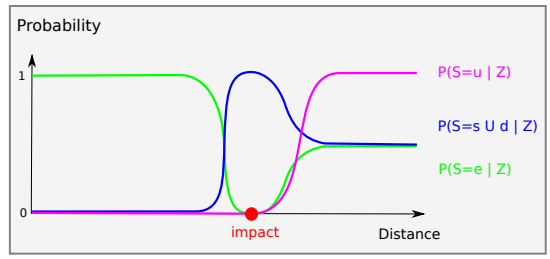

Fig. 5. Sensor model representation, in case of laser-like sensors. Occupied state likelihoods are low before the impact, high around and stabilize around 0.5 beyond. Empty state likelihoods display the symmetric profile. Unknown state likelihood is low before and around the impact, and increases beyond, i.e. where data is meaningless. The displayed profiles are not normalized.

This step purposes are to re-sample from the existing particle distribution to concentrate on its significant parts (according to their updated weights), to generate new particles for the newly dynamic parts (the "not yet sampled" part, with an initialization distribution) and to reassign the right amount of particles per cell. For each cell, the number of particles to be allocated is evaluated in accordance with its dynamic component, and in particular its newly dynamic part, which requires initialization. For parallelization practical considerations, the global amount of particles is set, these shared resources distributed among cells. For each cell, particles are then drawn from the updated previous particle distribution, and from an initialization distribution in proportion of dynamic appearance. Finally the cell dynamic coefficient is uniformly divided between the particles.

IV. Dynamic objects clustering

While the CMCDOT tracks spatial occupancy in the scene without object segmentation, detection and tracking of moving objects (DATMO) is often required for high level processing. The standard approach would then be to analyse the CMCDOT outputs, apply a clustering algorithm on the occupancy grid (enhanced by velocities) and use those clusters as potential targets. This clustering can turn out to be computationally expensive, considering the grid dimensions and most of all the dynamic particle model size and complexity. In this section we propose an extension of the CMCDOT to smoothly cluster dynamics cells and extract object properties with limited additional calculations.

The main idea is to consider that each dynamic particle belongs to an object identified by an object id (oid) and use the particle propagation within coherent moving blobs of spatial occupancy to propagate the oid. The object clustering method then consists in a simple addition in the re-sampling procedure and can be summarized as follows:

- When a particle is initialized (drawn from initialization distribution), a new oid is randomly associated to the particle.
- When a particle is duplicated (drawn from existing dynamic particle distribution) the oid associated to the particle is maintained.

Using this simple marker propagation, particles sampling dynamic occupancy which belongs to an object are quickly

associated. The *oid* convergence can be boosted if necessary, using additional sampling rules, but its speed is already satisfactory. For each *oid* can then be computed a list of properties such as the object weight (sum of particles probabilities), the gravity center, the average linear and angular speed and a covariance matrix representing the shape of the object. Significant objects can then easily be extracted, for example by setting a threshold on the weight.

V. Experimental Results

A. Experimental platform

For the experiments, a Lexus LS600h car have been equipped with two Ibeo Lux multilayer lidars under the two front lights (see fig 6). The covered horizontal field of view is almost 160 degrees. Vehicle velocity and steering data are collected from the CAN bus system. Using a merging method similar to [17], sequences of instantaneous occupancy grids are computed and used as inputs for our algorithm.

Fig. 6. Experimental platform : Lexus LS600h car equipped with two Ibeo Lux Lidars and cameras.

B. Estimation of irrelevant particle allocation ratio

In the first experiment, the objective was to evaluate the ratio of irrelevant particle allocation in different situations for the HSBOF and the CMCDOT. To calculate such a ratio, the average percentage of dynamic samples generated in unobserved areas is computed. The value itself is just indicative, as a portion of those particles transports real information, and depends of the selected grid size, but the comparison of the two methods offers a convincing evaluation of the allocation enhancement. Three scenarios have been tested : a highway scenario where some vehicles are permanently observed, a semi-urban environment where moving objects are intermittently present, and a city-center scenario where many pedestrians can be observed. The results displayed in fig 7 points out that dynamic samples are way more efficiently allocated with the CMCDOT algorithm, and so that the system is inclined to be more reactive in observed dynamic areas with the same resources.

Sequence	Grid size (m)	HSBOF	CMCDOT
Highway	20x70	76.9%	23.5%
Semi-Urban	30x60	89.3%	46.7%
City Center	30x60	93.2%	40.1%

Fig. 7. Estimation of irrelevant particle allocation ratio.

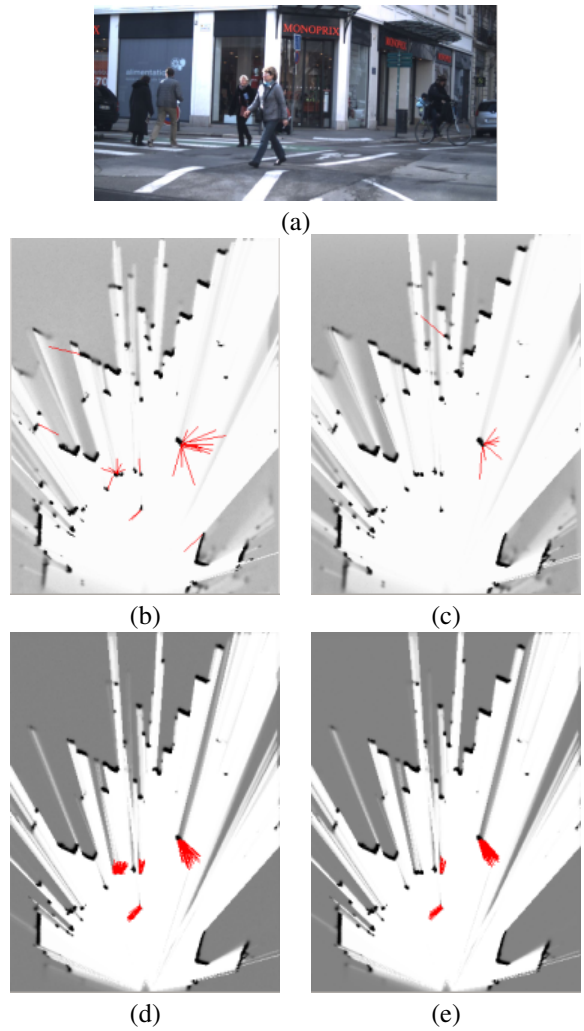

Fig. 8. Results of the HSBOF with 262144 and 32768 particles (b) and (c), and of the CMCDOT with the same number of particles (d) and (e). Red segments represent the average estimated per-cell velocity. They show that the CMCDOT is way more accurate and still manages to track most of the moving pedestrians even with a severely reduced number of samples, whereas the HSBOF loses track of almost all objects.

C. Dense tracking

In this second experiment, the HSBOF and the CMCDOT algorithm results have been compared for different numbers of samples. The size of the grid is 40x30m with a 0.1m resolution. The number of samples tested are 262144 and 32768. In fig 8 more stable and accurate velocity estimations can be observed for the new version. The results are also way less impacted by the decrease of sample number. Another noteworthy observation is that while the unknown areas turn out to be noisy in the HSBOF version, which causes incorrect dynamic cells to appear at the boundary between unknown and occupied areas, the CMCDOT provides accurate estimations.

D. Objects clustering

The last experimental set displayed in fig 9 shows an overview of the dynamic object clustering results. The interesting point is that the clustering method successfully extracts any kind of objects (pedestrians, vehicles, cyclists, etc). The

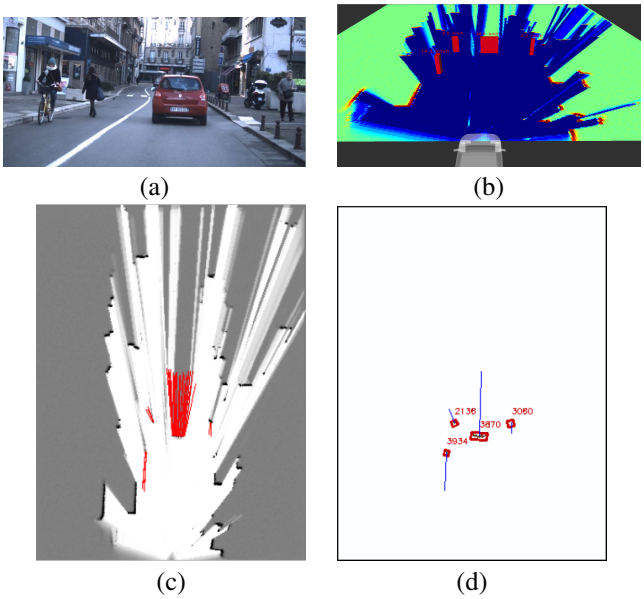

Fig. 9. Result of the dynamic objects clustering. (a) Camera image; (c) resulting occupancy grid with velocity; (d) extracted dynamic objects (red boxes) with velocity (blue segments) and id; (b) 3D view of the grid with detected objects.

algorithm has been tested within a large set of environments and we observed that almost all dynamic objects have been extracted with our method. Two main drawbacks can however be noted. The first one is that incoming large objects can be at first divided into multiple small objects, it can takes several seconds to converge to a single object. The second problem occurs when multiple dynamic objects move together (like a group of pedestrians), all objects can end up clustered in a single object, and when the different objects split up, they are still referenced in the same object. A direct extension of the object clustering would be to detect and handle the splitting of those groups (as most necessary data is already accessible).

VI. Conclusion

In this paper we presented a dense occupancy tracking method, inspired by the Bayesian Occupancy Filter framework enhanced by abstract states and a Conditional Monte Carlo approach to optimize velocity estimation and focus on relevant areas. The scene is analyzed through static, dynamic, free and unknown states, to which are associated dedicated models. Uncertainty and sensor coverage are explicitly taken into account. We also briefly presented a method to extract dynamic objects by slightly modifying the particle re-sampling step.

Experimental results showed that the insertion of an "unknown" state in the model leads to a better distribution of dynamic samples on observed areas and then allows to be way more reactive and accurate on the velocity distributions with less computing power. The intrinsic clustering approach has also been tested on real road data, showing promising results in real-time tracking of moving objects, regardless of their type. The method could be improved by managing split-and-merge events that can occur in complex urban environment.

Acknowledgment

This work has been developed within the scope of an "Institut de Recherche Technologique NanoElec" project, founded by the french program "Investissement d'Avenirs" ANR-10-AIRT-05. The authors wish to thank N. Turro, J.-F. Cuniberto and Toyota Motor Europe for their collaboration and continuous support on the experimental plateform.

References

[1] A. Petrovskaya, M. Perrollaz, L. Oliveira, L. Spinello, R. Triebel, A. Makris, J.-D. Yoder, U. Nunes, C. Laugier, and P. Bessière, "Awareness of road scene participants for autonomous driving," in *Handbook of Intelligent Vehicles*, A. Eskandarian, Ed. Springer, 2012, pp. 1383–1432.

[2] 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.

[3] 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.

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

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

[6] D. Pfeiffer. and U. Franke, "Modeling dynamic 3d environments by means of the stixel world," *Intelligent Transportation Systems Magazine, IEEE*, vol. 3, no. 3, pp. 24–36, 2011.

[7] C. Coué, C. Pradalier, C. Laugier, T. Fraichard, and P. Bessière, "Bayesian occupancy filtering for multitarget tracking: an automotive application," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 19–30, 2006.

[8] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[9] T. Gindele, S. Brechtel, J. Schroder, and R. Dillmann, "Bayesian occupancy grid filter for dynamic environments using prior map knowledge," in *Intelligent Vehicles Symposium, 2009 IEEE*. IEEE, 2009, pp. 669–676.

[10] S. Brechtel, T. Gindele, and R. Dillmann, "Recursive importance sampling for efficient grid-based occupancy filtering in dynamic environments," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3932–3938.

[11] R. Danescu, F. Oniga, and S. Nedevschi, "Modeling and tracking the driving environment with a particle-based occupancy grid," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, no. 4, pp. 1331–1342, 2011.

[12] M. E. Bouzouraa and U. Hofmann, "Fusion of occupancy grid mapping and model based object tracking for driver assistance systems using laser and radar sensors," in *IEEE Intelligent Vehicles Symposium*, San Diego, USA, 2010.

[13] A. Nègre, L. Rummelhard, and C. Laugier, "Hybrid Sampling Bayesian Occupancy Filter," in *IEEE Intelligent Vehicles Symposium (IV)*, Dearborn, United States, Jun. 2014. [Online]. Available: https://hal.inria.fr/hal-01011703

[14] C. Fulgenzi, A. Spalanzani, and C. Laugier, "Dynamic obstacle avoidance in uncertain environment combining pvos and occupancy grids," in *IEEE International Conference on Robotics and Automation*, 2007.

[15] 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.

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

[17] J. 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, United States, May 2012.