Object-based Velocity Feedback for Dynamic Occupancy Grids

Víctor Jiménez[1], Jorge Godoy[1], Antonio Artuñedo[1] and Jorge Villagra[1]

***Abstract*— Dynamic occupancy grids (DOGs) have raised interest in the last years due to their ability to fuse information without explicit data association, to represent free space and arbitrary-shape objects and to estimate obstacles' dynamics. Different works have presented strategies with demonstrated good performance. Most of them rely on LiDAR sensors, and some have shown that including additional velocity measurements enhance the estimation. This work aims at showing that velocity information can be directly inferred from objects displacement. Thus, a strategy using velocity feedback and its inclusion in the DOG is presented. The qualitative and quantitative analysis of results obtained from real data experimentation show a very good performance, specially in dynamic changing situations.**

I. INTRODUCTION

Perception of the environment is a challenging task, essential for autonomous driving. Occupancy grid [1] is a widely used strategy to represent the environment that divides the surroundings in cells containing an estimation about the presence of objects in this location, i.e. the probability of being occupied by an object. This strategy presents three main advantages: (i) arbitrary-shape objects can be easily represented, (ii) the areas of free or unknown space can also be modeled and (iii) the information of different sensors can be easily fused at a cell level. Nevertheless, a classical static occupancy grid map does not permit to estimate the evolution of dynamic environments.

The Bayesian Occupancy Filter (BOF), introduced in [2] and reviewed in [3], addressed the aforementioned problem relying on a generic Bayesian framework able to intrinsically manage uncertainty by updating a dynamic occupancy grid, where occupancy and velocity probability distribution functions (pdf) for each cell where estimated at each time step. However, it presented high computational loads. Works such as [4], [5], [6] tackled this problem by representing the dynamic state of the cells with particles and differentiating between occupied or unknown cells. The authors of [7] used this particle realization, but modeled the dynamic estimation problem in the random finite set domain, thus allowing a generic and rigorous filter design.

*This work was supported in part by Spanish Ministry of Science and Innovation with the National Project NEWCONTROL under Grant PCI2019-103791, in part by the Community of Madrid through SEGVAUTO 4.0-CM Programme under Grant S2018-EMT-4362, and in part by European Commission and ECSEL Joint Undertaking through the Project NEWCONTROL under Grant 826653.

[1]V. Jiménez, J. Godoy, A. Artuñedo and J. Villagra are with the Centre for Automation and Robotics (CAR), CSIC-UPM, Ctra. de Campo Real, km. 0,200, 28500 Arganda del Rey, Spain {victor.jimenez, jorge.godoy, antonio.artunedo, jorge.villagra }@csic.es

This work follows the paradigm proposed by [7], where it was shown that an accurate DOG can be achieved from laser data alone, but a faster convergence is achieved if radar data is included. As already pointed out in [8], where the dynamic state is estimated using recurrent neural networks instead of particles, the assumption of independent cells in the update step of the particles and the constant velocity model usually assumed, leads to decreased performance in dynamic changing scenarios, e.g. significant turn or acceleration/deceleration maneuvers. As discussed in [6] the use of velocity information, typically provided by radars, improves the velocity estimation in these situations by guiding the born of new particles and updating the weights.

Dynamic changes are widely found in autonomous driving scenarios, but every autonomous prototype is neither provided with radar sensors nor with the resources to train neural networks. To deal with these limitations, this work proposes to compute velocity feedback measurements from objects clustered in the DOG. Despite this calculation may be less accurate than radar information, it will be shown that it can approximately model the dynamic changes, improving thus the convergence of the DOG.

Therefore, the main contributions of this work are:

- A strategy to compute velocity feedback from DOGs.
- The application of object-based velocity feedback as velocity measurements.

The remainder of this paper is organized as follows. Section II provides an overview of the DOG framework. Section III explains the velocity feedback calculation based on clustered objects displacement. In Section IV, the measurement model defined for the computed velocity feedback is presented. Qualitative and quantitative evaluations are shown in Section V. Finally, Section VI summarizes the work presented and provides and outlook for future work.

II. DYNAMIC OCCUPANCY GRID OVERVIEW

This work is based on the DOG presented in [6], [7], but it is focused on velocity measurements calculation from clustered objects location in different time steps. Therefore, the implementation of the DOG exceeds the scope of this paper. Hereunder, a brief introduction to the main workflow is presented; the reader may refer to the aforementioned articles for further details.

A. State representation

The state of grid cells is divided in two: occupancy state and dynamic state. The occupancy state is represented in terms of the Dempster-Shafer Theory of Evidence (DST) [9], being the frame of discernment the occupied and free

events ($= \{O, F\}$). Therefore, each cell contains a mass for occupied $m(O)$ and a mass for free $m(F)$. The dynamic state is modeled by the particles within the cell. In order to reduce computational cost, only occupied cells are populated with particles. Hence, the posterior state of an individual grid cell c at the discrete time step k is represented by the particle set $\{\boldsymbol{x}^{(i;c)}, w^{(i;c)}\}_{i=1}^{c}$ – where ^{c} denotes the number of particles inside the cell, i the i-th particle and $\boldsymbol{x}$ and w define the particle's state and weight respectively – and the mass for free $m_k^c(F_k)$. The relation between the occupied mass and the set of particles is defined as:

$$m_k^c(O_k) = \sum_{i=1}^{\hat{} } w_k^{i;c} \tag{1}$$

B. Measurements

Associated with every cell, two types of measurements $z(k+1)$ are expected:

- Observed occupancy measurement modeled in terms of DST ($m^c_{z_{(k+1)}} : 2^{\{O;F\}} \to [0;1]$). Laser data is commonly used to accomplish this task; in this article the approach presented in [10] is used.
- Spatial measurements containing velocity information ($\#^c_{z_{(k+1)}}$). An association probability (p^c_A) is also attached to each measurement in order to model the probability of corresponding to the grid cell c. These measurements can be obtained from different sources: [6] used Doppler measurements, [11] used V2X communications and this article proposes the use of velocity feedback.

C. Prediction

Occupied mass prediction is carried out by the particles. The state of predicted particles is calculated applying a constant velocity model and their weight is multiplied by a persistence probability. Then, predicted occupied mass is obtained with (1). In turn, the predicted mass for free is modeled as in static grid maps, where its reliability decreases as time passes. Since the sum of masses inside a cell cannot exceed 1, the weights of particles are truncated and then the free mass is limited accordingly.

D. Update

The update step is performed according to occupancy and spatial measurements. The occupancy update is computed in terms of DST using the Dempster-Shafer rule of combination, see [7]. The updated occupied mass is splitted in two parts, one portion is used to update the weight of persistent particles with (1) and the other is used to initialize new particles' weights.

The spatial update is performed afterwards on the persistent particles. Each particle is weighted according to the spatial likelihood function $g^c_{(k+1)}(\#_{z_{(k+1)}} | x_{(k+1|k)})$ and the association probability $p^c_{A;(k+1)}$

$$w^{i;c}_{p;(k+1)} = p^c_{A;(k+1)} \; {}^{c}_{A} \; w^{i;c}_{p;(k+1)} + 1 \quad p^c_{A;(k+1)} \; {}^{c}_{A} \; w^{i;c}_{p;(k+1|k)} \tag{2}$$

with

$$w^{i;c}_{p;(k+1)} = g^c_{(k+1)}(\#^c_{z_{(k+1)}} | x^{i;c}_{p;(k+1|k)}) \; w^{i;c}_{p;(k+1|k)} \tag{3}$$

and being ${}^{c}_{A}$ and ${}^{c}_{A}$ normalization factors. The super-index i denotes the i-th particle and the sub-index p denotes persistent particle.

Given the updated persistent particles, the statistical moments of grid cells are calculated as the weighted mean and variance.

E. New particles and resampling

New particles are created in grid cells with updated occupied mass reserved for this purpose. If spatial measurements are associated with the cell, a portion of these new particles – defined by the association probability – can be initialized with respect to them. Otherwise, new states are chosen from a Gaussian distribution with zero mean and standard deviation dependant on the scenario.

Lastly, particles are resampled with a probability of being drawn proportional to its weight. The weights of resampled particles are set all equal and defined according to (1).

III. VELOCITY FEEDBACK

After the LiDAR-based update, the velocity feedback is calculated at object-level. This step consists of two stages: (i) object clustering and association between frames and (ii) velocity computation based on object's displacement.

A. Clustering and data association

Cells with an occupied mass higher than a threshold ${}_{o}$ are clusterized using a modified Connect Component Clustering [12] in order to include the velocity information. In this way cells are considered to belong to the same object if they are neighbours inside a mask of certain size and their velocity vector differs less than a threshold ${}_{v}$. These clusters are then described at an object-level with a state including position and velocity.

The velocity feedback is computed from the displacement of objects from one time step to the next one. Therefore, objects from previous and current time steps have to be associated. To that end, an approach based on Global Nearest Neighbour (GNN) [13] is used. The objects of the previous time step are predicted to the current instant using a constant speed model. Then a cost matrix between predicted and observed objects is calculated using the Mahalanobis distance [14]. Lastly the association is solved using the Munkres algorithm.

B. Velocity calculation

For each object o associated with a previous object p, the velocity feedback is computed as the two-dimensional vector $\boldsymbol{v}^o$:

$$\boldsymbol{v}^o = \frac{\boldsymbol{x}_l(o;p)}{t} \tag{4}$$

where $\boldsymbol{x}_l$ defines the location of the object.

In this work three methods are suggested in order to compute this displacement: (i) centroid, (ii) cross-correlation

(CC) and (iii) vehicle's shape assumption (VSA). In the following the three methods are introduced, followed by a discussion about how they should be used.

1) Displacement by centroids: The centroid was suggested in [5] as cluster state descriptor. Calculating the displacement with centroids is simple, but as denoted in [15] it is prone to errors due to the detected footprint change. This influence may be reduced when using recursively estimated occupancy since the change is slower, but it is still present.

2) Displacement by cross-correlation: In order to solve the aforementioned centroid's problem, [15] proposed to compute the displacement between time steps by image matching. The optimum location of an image template within other reference image is computed based on the cross-correlation coefficient metric. Following this proposal, two binary images are created containing the cells of the associated objects. Cells for which the LiDAR has modeled occupancy are set to 1 and otherwise to 0. The current time step image is used as template and the previous time step image as reference. Through an iterative process, the origin of template image is located over each pixel of the reference image and a similarity score is computed using cross-correlation:

$$S(k;l) = \frac{\sum_{i=1}^{M_1}\sum_{j=1}^{N_1} T(i;j)R(k+i-1;l+j-1)}{\left[\sum_{i=1}^{M_1}\sum_{j=1}^{N_1} T(i;j)\right]^{1=2}\left[\sum_{i=1}^{M_1}\sum_{j=1}^{N_1} R(k+i-1;l+j-1)\right]^{1=2}} \quad (5)$$

where T is the template image with size $M_1 \quad N_1$, R is the reference image with size $M_2 \quad N_2$ and the pixel $k;l$ defines the moving origin of the template over the reference image.

This process results in a score-map S from which the pixel with higher value is defined as the location of optimum matching and, therefore, the displacement is obtained.

3) Displacement by vehicle's shape assumption: It is a common assumption to represent vehicles' shape with rectangle bounding boxes. Moreover, if an object classification is performed, an approximated box size can also be assumed. This allows the estimation of the real-center of a vehicle without being influenced by the size of the footprint detected [16], [17].

In order to demonstrate the feasibility of this concept, a simplified implementation of the strategy presented in [16] is used to detect vehicles and assign boxes fixed size, but others methods could be used (e.g. vision-based methods). First, vehicles are detected among objects as those that have presented a certain size and velocity over time using a multi-object tracking. Then, according to the historical length and width, a box of fixed size is assigned.

Given the vehicles' classification and their corresponding fixed size boxes, the center of the vehicle is estimated fitting an oriented bounding box. This process is divided in two steps: (i) orientation estimation and (ii) most visible side based fitting.

The fixed box orientation is calculated as a weighted average between the orientation of the velocity vector and the orientation estimated from the detected footprint – in this article a minimum bounding box fitting method based on [18] is used. This combination provides a more robust estimation than considering only one isolated orientation estimation strategy. Indeed, on the one hand, the velocity vector is accurate for dynamic objects moving straight, but may have delay on turning vehicles and cannot be used on static objects. On the other hand, the orientation computed with the footprint is accurate when L-shapes are detected, commonly seen in turning vehicles, but, with small footprints its precision decreases and may fail in classifying the sides corresponding to the length and width. An illustrative example of the advantages of this fusion is given in Figure 1a.

In this way, the orientation of the fixed box is computed as:

$$_b = \frac{_s \; s + \quad _v \; v}{s + \quad v} \quad (6)$$

where and denote the orientations and weights computed from the velocity vector v and the shape of the detected footprint s, respectively. The weights are assigned as:

$$_s = \begin{cases} ^{max}_s; & L_s = 1 \\ ^{min}_s; & L_s = 0 \end{cases} \quad (7)$$

$$_v = \begin{cases} ^{max}_v; & jvj \quad v_d \\ ^{max}_v \, jvj=v_d; & jvj < v_d \end{cases} \quad (8)$$

where L_s is a binary variable defining whether an L-shape has being detected or not, v_d is a threshold that defines the velocity of dynamic objects and super-indexes refer to the maximum and minimum weight set as design parameters.

Then, the fixed box is fitted on the footprint searching for the most likely reference point. This point is computed in base of the theoretically visible sides of a minimum bounding box oriented with $_b$ (see Figure 1b).

The theoretical visibility of a side i is defined by the angle ($^i_{visible}$) formed by the side's normal vector ($\boldsymbol{n}^i$) and the vector between the center of this side and the origin of the perception reference system ($\boldsymbol{l}^i$).

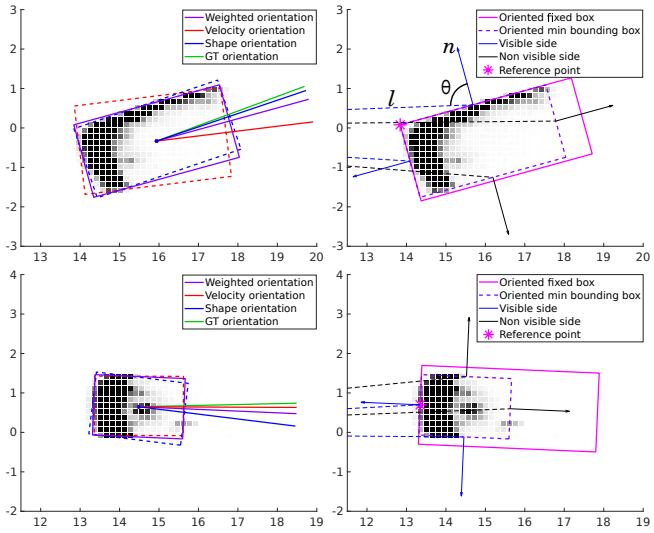
$$^i_{visible} = \arccos \frac{\boldsymbol{n}^i \quad \boldsymbol{l}^i}{j\boldsymbol{n}^i j \; j\boldsymbol{l}^i j} \quad (9)$$

A side is considered as theoretically visible if $_{visible} < 90^o$. The reference point is inferred with respect to the number of visible sides as follows:

- If only one side is visible, the reference point is defined at the center of this side.
- If two sides are visible, the reference point is defined at the corner.

Finally, the fixed box is located at the reference point with the orientation $_b$. The displacement is obtained comparing the center of the box at each time step.

4) Displacement method discussion: Table I shows an illustrative comparison of the advantages and disadvantages of the three methods. The centroid method is simple but, as already mentioned, suffers when the detected footprint changes of size. The CC method performs better in relation to this problem, but it does not model orientation variations.



(a) Orientation estimation. (b) Oriented fixed box fitting

Fig. 1: Estimation of the fixed box location.

The VSA method handles both the footprint change and turns; nevertheless a vehicle classification strategy is needed and its performance may be poor for static small objects.

Therefore, the strategy suggested is a combination of the cross-correlation and vehicle's shape assumption methods. The CC method is applied generally and, whenever a dynamic vehicle is detected, it is substituted by the VSA approach.

TABLE I: Simplified comparison of the displacement

Method	Centroid	CC	VSA
Footprint change	0	++	++
Turning objects	+	0	++
Static objects	+	++	+
Simplicity	++	+	0

IV. INCORPORATING VELOCITY FEEDBACK MEASUREMENTS INTO DOG

Given an object o for which a feedback velocity vector has been estimated, a spatial measurement $\#^{o}_{z_{k+1}}$ and probability of association p^{o}_{A} can be defined and assigned to the corresponding clusterized grid cells.

Following the strategy presented in [11], the likelihood of each particle i with respect the measurement $\#^{o}_{z_{k+1}}$ in the cell c is modeled with the bivariate Gaussian distribution:

$$g^{c}_{(k+1)}(\#^{o;c}_{z_{(k+1)}} | x^{i;c}_{p;(k+1|k)}) = \frac{1}{2\,\sigma^{o;c}_{v_x}\,\sigma^{o;c}_{v_y}\sqrt{1-(\rho^{o;c})^2}} \exp\left[\frac{1}{2(1-(\rho^{o;c})^2)}\left(\frac{(C^{o;i;c}_{v_x})^2}{\sigma^{o;c}_{v_x}} + \frac{(C^{o;i;c}_{v_y})^2}{\sigma^{o;c}_{v_y}} - 2\rho^{o;c}\frac{C^{o;i;c}_{v_x} C^{o;i;c}_{v_y}}{\sigma^{o;c}_{v_x}\sigma^{o;c}_{v_y}}\right)\right] \quad (10)$$

with

$$\rho^{o;c} = \frac{(\sigma^{o;c}_{v_{xy}})^2}{\sigma^{o;c}_{v_x}\,\sigma^{o;c}_{v_y}} \quad (11)$$

$$C^{o;i;c}_{v_x} = \frac{v^{i;c}_{x} - v^{o;c}_{x}}{\sigma^{o;c}_{v_x}}; \quad C^{o;i;c}_{v_y} = \frac{v^{i;c}_{y} - v^{o;c}_{y}}{\sigma^{o;c}_{v_y}} \quad (12)$$

where v^{i}_{x} and v^{i}_{y} denote the particle's velocity vector and $\#^{o}_{z_{k+1}}$ is assumed as gaussian, being v^{o}_{x} and v^{o}_{y} the mean velocity in x and y, defined by the calculated velocity feedback, and $\sigma^{o}_{v_x}$, $\sigma^{o}_{v_y}$ and $\sigma^{o}_{v_{xy}}$ the standard deviation and covariance, defined as design parameters.

Particle initialization can be performed likewise, the dynamic state of associated new particles can be drawn from a Gaussian distribution $N(v^{o}; \sigma^{o})$.

The probability of association p^{o}_{A} is the same for all object's cells and is defined by the quality of the matching between objects.

$$p^{o;c}_{A} = \min\left(p^{max}_{A}; 1 - \frac{C(o;q)}{C_{max}}\right) \quad (13)$$

where C denotes the cost of the matching (as explained in Section III-A), C_{max} is the maximum cost allowed for matching and p^{max}_{A} is a design parameter included to constrain the maximum association, modeling the possibility that the matching is incorrect. o and q refer to the associated objects from consecutive time steps.

V. EXPERIMENTAL RESULTS

The work proposed in this paper has been validated on tests with two automated vehicle prototypes. These vehicles have proprioceptive and exteroceptive sensors for localization and environment perception. In order to correlate data among vehicles, Global Navigation Satellite System receivers (GNSS) are used for time synchronization

For all tests the vehicles were manually driven, with the perceiving vehicle following the sensed one along its path. The perceiving vehicle combines one IBEO-Lux and two VLP-16 LiDAR sensors for perception. The leading vehicle records its own state, which is then used as ground truth. As already explained, the DOG performs properly in most of the cases, being the dynamic changing situations where velocity estimation complexity arises. Thus, the recorded datasets focus in these situations, e.g. roundabouts. An example of a test route is shown in Figure 2.

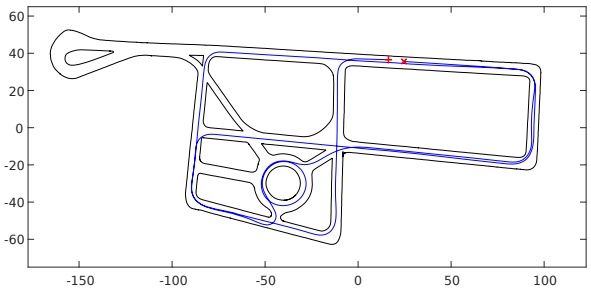


Fig. 2: Example route for dataset. The start and the end are denoted with + and respectively.

The parametrization of the perception framework is set as follows:

- **DOG:** the size of the grid map is 512 512, with a cell size of 0:15m and approximately 100 000 particles. A

road map is used to filter every obstacle outside the road, therefore particles population is mainly used to model the detected vehicle.

- **Clustering:** the clustering parameters are defined as: $_{o}$ = 0:2 and $_{v}$ = 2 m/s. The mask size is set to 5 pixeles.
- **VSA:** the velocity threshold for dynamic objects is set as v_d = 1 m/s. An L-shape is detected when both sides are larger than 1 m. The boundary weights are set as $^{max}_{v}$ = 0:5, $^{max}_{s}$ = 1 and $^{min}_{s}$ = 0:2. The fixed size for vehicles is defined as 4:5 m length and 2 m width.
- **Spatial measurements:** the standard deviation velocity defined for the feedback velocity is $_{v}$ = 1 m/s. Covariance is assumed to be zero. The maximum probability association is set as p_A^{max} = 0:5.

A. *Qualitative results*

In this section, two complex cases are analyzed, showing how the velocity information improves the estimation.

Figure 3 shows an example where the detected vehicle changes its state from static to dynamic. Free space is drawn in white, unknown space in gray and occupied space in black or in color. Occupied cells vary from black to coloured depending on their speed; the brighter the cell the higher the speed, while the colour depends on the orientation. Lastly, the green vector denotes the ground truth velocity and the red vector the estimated velocity at an object-level.

This case is considered as complex due to the large footprint detected. If the DOG is fed only with observed occupancy measurements, static particles remain with high weights until no occupied mass is modeled at their position. On the contrary, if spatial measurements, in addition to the occupancy measurements, are used to to feed particles' birth and update, a faster convergence over the real velocity is obtained. This can be clearly noticed at instant b), where the DOG without velocity information (left) estimates most of the cells as static, while the DOG with velocity information (right) has modeled most of them as dynamic. In this case, the feedback velocity is computed with the CC method, since the object is not dynamic yet.

Figure 4 shows a 90° turn. Cells' occupancy state is coloured in gray scale and dynamic state is denoted with blue vectors. The green vector depicts the ground truth velocity and the red vector the velocity estimated at an object-level. In the DOG with velocity information, the magenta box and dashed vector represent the feedback estimation. The complexity of this situation comes from two factors: (i) particles predict their state based on a constant velocity model, thus this dynamic change has to be mostly estimated by new particles with the new dynamic state and (ii) the detected vehicle occludes the straight path, therefore old particles moving straight are not suppressed as fast as if free space were detected. These facts can be clearly noticed in instants b) and c). When no velocity information is used, the turn is mainly modeled by the new cells detected at the left of the vehicle, while the vector of old cells remain orientated towards the occluded straight path. In this case, the feedback

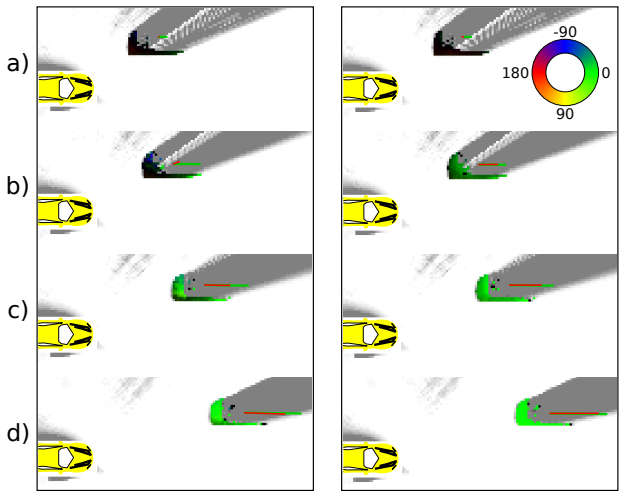


Fig. 3: Start moving situation, comparison with and without velocity information, right and left columns respectively.

velocity is computed with the VSA method, which is able to approximately model the vehicle behaviour. Indeed, results shows a smaller delay in the velocity estimation at an object-level and a smaller variance in the velocity vectors associated with the cells.

B. *Quantitative results*

A quantitative evaluation has been conducted using 5 different datasets which sum up to ten minutes approximately. Sensors data is collected at a rate of 12:5Hz. These datasets contain routes as the one shown in Figure 2 and situations similar to the qualitative tests of Section V-A. The results of the DOG using only occupancy measurements (Oc) are compared to the results obtained using occupancy and spatial measurements (Oc&Vel). Three methods for velocity feedback calculation have been evaluated: (i) computed with centroids, (ii) computed using CC and (iii) computed with the CC/VSA combination proposed in section III-B.4.

In order handle the randomness of the particle filter, the same random numbers are used for all methods and each dataset has been computed twice using a different set of random numbers.

Figure 5 and Table II show the error obtained as boxplots and summarized in terms of mean absolute error (MAE) and root mean square error (RMSE), respectively. The results are evaluated with respect the velocity module and orientation, both computed at an object-level as the weighted mean of all particles gathered inside the object's cells. Since orientation is computed from the velocity vector, its estimation for static objects is meaningless, thus the orientation error is computed only for those iterations where the sensed vehicle drives faster than 1 m/s.

As already explained, the DOG fed only with occupancy measurements is able to converge over the real object's speed. However the convergence time can be reduced if velocity information is included. Therefore, results show that the general error is low, but the use of velocity feedback

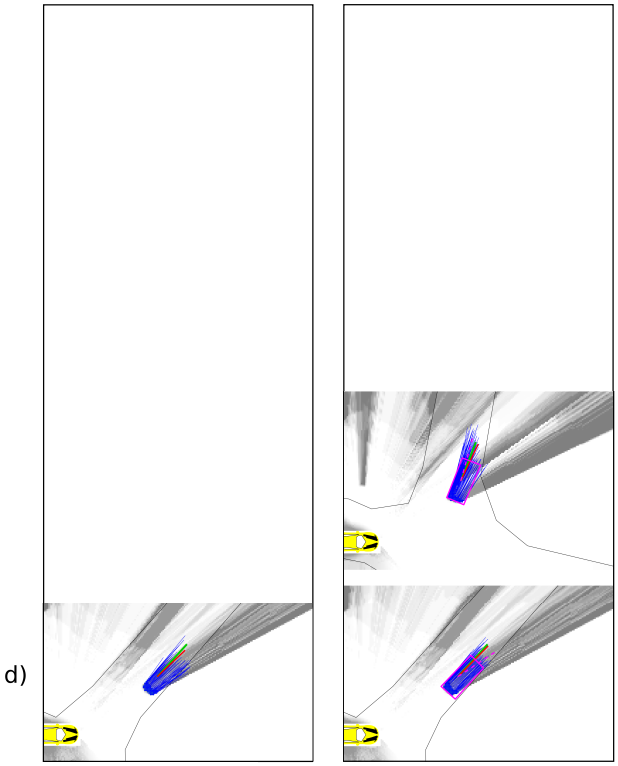


Fig. 4: Turning situation, comparison with and without velocity information, right and left columns respectively.

further reduces this error. Regarding the displacement calculation methods, the CC/VSA method obtains the best results, as expected, while the CC performs similar to the centroid method in terms of velocity estimation, but better in terms of orientation.

TABLE II: Error quantitative evaluation.

Measurements	**Oc**	**Oc & Vel**		
		Centroid	CC	CC/VSA
Velocity MAE (m/s)	0.490	0.256	0.262	**0.209**
Velocity RMSE (m/s)	0.638	0.340	0.349	**0.293**
Orientation MAE (^{o})	4.760	3.568	3.260	**2.697**
Orientation RMSE (^{o})	7.330	5.337	5.028	**3.963**

VI. CONCLUSIONS

This work presents a velocity feedback strategy in order to enhance the velocity estimation of dynamic occupancy grids. Previous works demonstrated that the use of radar data improves the estimation convergence; this paper proposes to compute velocity measurements from objects displacement when no velocity sensor is available. Three methods to calculate the displacement are discussed and evaluated using real data. Qualitative and quantitative results validate the feasibility of this strategy, showing a faster convergence in dynamic changing situations.

Future work will address the improvement of data association and clustering by using multi-object tracking strategies. Additionally, experimentation will be extended to more populated scenarios.

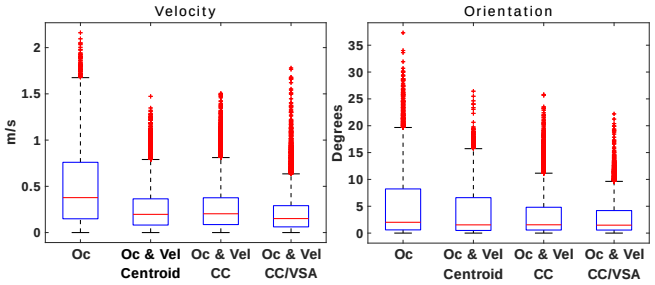


Fig. 5: Error quantitative evaluation.

REFERENCES

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

[2] 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.

[3] M. Saval-Calvo, L. Medina-Valdés, J. M. Castillo-Secilla, S. Cuenca-Asensi, A. Martínez-Álvarez, and J. Villagrá, "A review of the bayesian occupancy filter," *Sensors*, vol. 17, no. 2, 2017.

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

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

[6] D. Nuss, T. Yuan, G. Krehl, M. Stuebler, S. Reuter, and K. Dietmayer, "Fusion of laser and radar sensor data with a sequential monte carlo bayesian occupancy filter," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1074–1081, 2015.

[7] D. Nuss, S. Reuter, M. Thom, T. Yuan, G. Krehl, M. Maile, A. Gern, and K. Dietmayer, "A random finite set approach for dynamic occupancy grid maps with real-time application," *The International Journal of Robotics Research*, vol. 37, no. 8, pp. 841–866, 2018.

[8] M. Schreiber, V. Belagiannis, C. Gläser, and K. Dietmayer, "Motion estimation in occupancy grid maps in stationary settings using recurrent neural networks," *CoRR*, vol. abs/1909.11387, 2019.

[9] G. Shafer, *A Mathematical Theory of Evidence*. 1976.

[10] J. D. Adarve, M. Perrollaz, A. Makris, and C. Laugier, "Computing occupancy grids from multiple sensors using linear opinion pools," in *2012 IEEE International Conference on Robotics and Automation*, pp. 4074–4079, 2012.

[11] J. Godoy, V. Jiménez, A. Artuñedo, and J. Villagra, "A grid-based framework for collective perception in autonomous vehicles," *Sensors*, vol. 21, no. 3, 2021.

[12] A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," *J. ACM*, vol. 13, p. 471–494, oct 1966.

[13] P. Konstantinova, A. Udvarev, and T. Semerdjiev, "A study of a target tracking algorithm using global nearest neighbor approach," 01 2003.

[14] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.

[15] J. Zhang, W. Xiao, B. Coifman, and J. P. Mills, "Vehicle tracking and speed estimation from roadside lidar," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 5597–5608, 2020.

[16] S. Steyer, C. Lenk, D. Kellner, G. Tanzmeister, and D. Wollherr, "Grid-based object tracking with nonlinear dynamic state and shape estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 2874–2893, 2020.

[17] K. Liu and J. Wang, "Fast dynamic vehicle detection in road scenarios based on pose estimation with convex-hull model," *Sensors*, vol. 19, no. 14, 2019.

[18] X. Zhang, W. Xu, C. Dong, and J. M. Dolan, "Efficient l-shape fitting for vehicle detection using laser scanners," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 54–59, 2017.