Robotics and Autonomous Systems 161 (2023) 104363

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

Object-wise comparison of LiDAR occupancy grid scan rendering methods

Víctor Jiménez *, Jorge Godoy **, Antonio Artuñedo, Jorge Villagra

Centre for Automation and Robotics (CSIC-Universidad Politécnica de Madrid), Ctra. M300 Campo Real, Km 0.200, Madrid, 28500, Spain

ARTICLE INFO

Article history:
Received 14 September 2022
Received in revised form 16 December 2022
Accepted 6 January 2023
Available online 11 January 2023

Keywords:
Occupancy grid
Evaluation method
LiDAR
Perception
Autonomous vehicles

ABSTRACT

Computing Occupancy grids with LiDAR data, is a popular strategy for environment representation. In the last two decades, several authors have proposed different methods to render the sensed information into the grids, seeking to obtain computational efficiency or accurate environment modeling. However, no comparison regarding their performance under object detection in autonomous driving applications has been found in the literature. As a result, this work compares six representative LiDAR scan rendering strategies in a quantitative manner. To that end, a novel quantitative evaluation framework for occupancy grids is proposed. It addresses the two main steps of object detection: object segmentation and features estimation, proposing a meaningful procedure, repeatable with other OG approaches. The code of this evaluation framework is available in https://git-autopia.car.upm-csic.es/open_source/occupancy_grid_object_detection_evaluation.git.

1. Introduction

Occupancy grid (OG) is a widely explored technique in mobile robotics. Under this framework, the estimated environment is divided into multiple grid cells that represent a portion of the space and contain an occupancy value representing the probability of being occupied by an obstacle. OGs provide different advantages: (i) data fusion of several sensors can be easily accomplished at a cell level, (ii) arbitrary-shape objects can be correctly represented and (iii) the occupancy state reflects not only the probability of being occupied but also of being free or unknown. In autonomous driving applications it is commonly computed aiming to obtain a low level representation of the environment from which other tasks can be launched, e.g. object detection and tracking, drivable space estimation, simultaneous localization and mapping (SLAM), etc.

This work focuses on LiDAR scan rendering into Cartesian occupancy grids and evaluates the results from an object detection perspective in the field of autonomous driving. Some examples of OG based approaches fulfilling these conditions are [1–5].

In the literature, the two most popular strategies to accomplish scan rendering are line-drawing [6–9] and polar grid based [10–13] algorithms.

Nevertheless, different authors have proposed the application of other methods or variations looking for less computational cost or a more precise representation of the information. To the authors' knowledge, there is no work in the state of the art comparing the results of these strategies using Autonomous Driving Applications criteria.

The performance of scan methods and OGs is commonly evaluated by visual inspection and based on theoretical advantages. But this type of evaluation is not acceptable for a reliable comparison. Moreover, works that evaluate the OG using quantitative results tend to strongly rely on highly controlled or simulated scenarios to obtain ground truth reference maps. Alternatively, other works focus in non-general posterior algorithms, such as path planning or environment mapping, from which measurable results can be obtained and, thus, infer the quality of the OG.

For these reasons, the main contributions of this article are:

- A comparison of six methods to render LiDAR data into OGs.
- A new evaluation method for OGs from an object detection autonomous driving perspective.

The remainder of this article is structured as follows. Section 2 presents the state of the art related with OGs scan rendering algorithms and OGs evaluation methods. In Section 3 the proposed OG evaluation method is explained. Section 4 introduces the six scan rendering methods evaluated and Section 5 provides the details related to the OG framework implementation. Experimentation results are shown in Section 6. Finally, Section 7 provides some concluding remarks and an outlook to future work.

* Corresponding author.
** Corresponding author.
E-mail addresses: victor.jimenez@csic.es (V. Jiménez), jorge.godoy@csic.es (J. Godoy).

https://doi.org/10.1016/j.robot.2023.104363

2. State of the art

2.1. Scan rendering strategies for occupancy grids

This work focuses on the comparison of algorithms that render LiDAR measurements into 2D occupancy grids. As will be shown through this section, different decisions arise when addressing this task and researches approach them differently in order to obtain more computational efficiency, a more accurate description of the environment or a balance between both.

The algorithms found in the literature can be grouped into two main strategies depending on the shape assumed for the laser beam: those modeling it as a thin line and those modeling it as an angular sector; henceforth, these will be referred as line-based and angular-based strategies, respectively.

This shape assumption is one of the main criteria when choosing a scan rendering strategy since it directly affects the number of cells covered by each beam. Line-based algorithms only update cells traversed by the beam's centerline. As a result, given the common rotational acquisition process of LiDAR sensors, farthest cells may be updated by one or none beams. The uncovered cells can be considered as a drawback of these methods as they may affect the data fusion and posterior tasks, such as matching, objects' shape estimation or path planning algorithms [10,11]. On the contrary, angular-methods are usually intended to correctly model the rotational acquisition by updating the full grid space, thus avoiding this problem [10,14]. However, as beams are modeled with an angular width wider than the real one, potential wrong-updated cells and possible object miss-detection might appear [6,13].

It is not clear from the literature whether the update of strictly traversed cells or the fulfill of the whole field of view yields better results. Moreover, different approaches based on these two shape assumptions exist, being the most representative ones described below.

2.1.1. Line-based algorithms

Probably the most known line-based method is the Bresenham's algorithm [15]. It was originally designed for rendering lines in images (line-drawing algorithms), but it has been successfully applied in the field of OGs [6,8,9], becoming one of the most popular strategies given its low computational cost.

Nevertheless, since the objective of these algorithms is to produce an acceptable visual line and not to find out all cells truly traversed, some cells can be missed. In contrast, traversal algorithms are specifically designed for this task [16]. Computational requirements are slightly higher, but the guarantee of updating every cell is considered as an advantage in some works, such as [16–18].

Also, with the intention of obtaining more accurate models, several authors have weighted the influence of the beam over each updated cell. Xiaolin Wu's algorithm [19] weights the cells straddling the line according to their distance to it. Similarly, [20] included weights that take into account both, the number of beams traversing the cell and the distance traveled across it. Other authors seek to model the uncertainty of impacts position by additionally modeling cells around it using a weighted occupancy probability accordingly to an uncertainty ellipse [7,21].

2.1.2. Angular-based algorithms

It is common to find applications which represent LiDAR data on polar grids because it leverages on the rotational acquisition process of LiDAR sensors, e.g. [22]. Indeed, in the field of OGs it is the most representative angular-based algorithm. First, the LiDAR data is renderized into an OG divided in terms of distance and angle. Then this polar OG is transformed into the desired Cartesian OG [10,11,13,23]. This approach is used by many authors not only because it is consistent with the characteristics of LiDAR data, but also because it solves the uncovered cells problem of the line-based algorithms. However, transformation from polar to Cartesian grid is a computationally expensive task.

In order to reduce the associated load, [24] assumed that beams do not overlap between them and that they cover a specific angular sector, being the cells gathered by each sector updated with respect the corresponding beam.

As for the line-based algorithms, other authors seek to introduce more accurate models. [14] also modeled the uncertainty around the impact with an uncertainty ellipse. Likewise, [25] argued that the occupancy confidence may be maximum along the beam centerline and decrease with the distance in the other directions.

As it can be seen, different methods claiming for theoretical advantages or acceptable simplifications can be found in the literature. The objective of this work is to evaluate their results from the perspective of object detection in autonomous driving applications.

2.2. Occupancy grid evaluation

The performance of OGs is typically evaluated by visual inspection. Nevertheless, this type of evaluation is highly subjective, perceived patterns may not significantly affect the real quality of the OG, while valuable differences may be hardly detected. In order to compare the results obtained from different methods, a quantitative evaluation has to be perform. Such evaluation can be carried out from two perspectives: (i) cell-wise analysis and (ii) application specific analysis.

2.2.1. Cell-wise analysis

Cell-wise analysis evaluates the estimated OG at a cell level, independently of the application. In this way, the value estimated for each cell is compared with its respective ground truth value, therefore a ground truth OG is required. Different metrics relying on this ground truth OG can be found in the literature. [26] used the Cross-correlation and Map Score methods in order to measure the similarity between two maps. [27,28] binarized the occupancy state and applied the well-known classification scores Precision and Recall. In the case of OGs including dynamic states (DOGs) similar evaluation scores can be applied by classifying occupied cells into dynamic or static [29,30].

The cell-wise analysis aims at providing a standardized accuracy measurement, since it can be employed despite of the posterior applications. However, for the same reason, the obtained conclusions can be uncorrelated with the OG usefulness. Moreover, facts such as the representation of the vegetation or the evaluation of scenes with occlusions makes the computation of ground truth OGs hardly achievable for real-world scenarios. Indeed, the above mentioned works take advantage of simulation environments or specific and controlled real-world scenarios.

2.2.2. Application specific analysis

Application specific analysis focuses on the results obtained from an specific application which is fed by the OG. In this context conclusions are less generalizable, but can be evaluated in a more meaningful way and are correlated with the application's objective.

[26,27] extended the cell-wise analysis by assessing the results of a path planning algorithm and road boundary estimation algorithm respectively. Nevertheless, the evaluation of these tasks is not easily performed in a extensive quantitative manner. Moreover, their conclusions may not be transferable to object detection. DOG based works such as [30,31] performed specific tests

where cells of a single vehicle are clusterized and the error of its speed is evaluated. [32] made a similar evaluation including additional velocity metrics and addressing multiple vehicles using the public dataset Argoverse [33]. Nevertheless, this evaluation only addresses a single specific feature of object estimation. Works including further tasks such as [3], where dynamic obstacles are detected with oriented bounding boxes, or [2], where road-users are tracked and classified at an object-level, can perform evaluations based on the already well-established object detection and tracking metrics, such as [34]. Nevertheless, these metrics require an object-level representation, which OGs do not provide without additional steps, such as object segmentation and state estimation.

As can be noted, no-standardized quantitative evaluation for OGs focused on object detection in the field of Autonomous Driving Vehicles has been found in the literature. Therefore, this article aims at providing a repeatable evaluation method that infers objects from OGs and applies scores based on the well-known object-detection metrics.

3. Occupancy grid evaluation method

Perception for autonomous driving has well-established benchmarks for object detection and tracking, e.g. [34,35]. Nevertheless, OGs represent objects as a group of independent cells without the object-level representation that these benchmarks require. Moreover, in OGs, object's location and shape are mainly modeled by the footprint captured by the sensor, being common to find objects described by one or more groups of cells which are smaller than the objects' real size. Transforming this cell-level representation into an object-level description is a complex task without a common procedure. Additionally, object-level estimation involves two key aspects: (i) object detection capability and (ii) object features quality.

Taking this into account, the proposed evaluation framework seeks to be adequately extensive and to provide conclusions transferable to other object-level strategies by: (i) evaluating multiple aspects concerning frequent clustering issues and object-level features, (ii) selecting the scores and algorithms based on widely used criteria and (iii) employing different tools to evaluate the same feature from different perspectives.

Therefore, the objective of this section is to present a strategy for object segmentation and object features estimation in order to evaluate the OG quality from an object estimation perspective. The objects estimated by this strategy are not intended to be the best estimations achievable but a repeatable process that allows to measure how easy and accurately these tasks can be performed. In this connection, the proposed scores are based on well-established object detection scores but adapted to the evaluation purpose.

3.1. Ground truth data

The proposed evaluation relies on a dataset containing LiDAR data along with a ground truth of the objects in the scene. The state of these objects, henceforth named as Ground Truth Objects (GTO), consists of (i) a 2D oriented bounding box and (ii) a velocity vector.

3.1.1. Footprint quality reference

As explained the introduction of this Section, in OGs objects are mainly modeled by the footprint captured by the sensor, with variations depending on the scan rendering method and the historical data. Given the importance of the footprint, an OG evaluation from an object perspective should take it into account. Thus, if LiDAR points corresponding to the objects are labeled by the ground truth, a reference of the expected object-estimation quality can be computed.

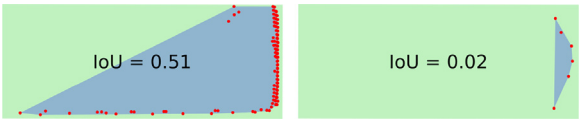


Fig. 1. Example of convex hulls computed from LiDAR points and the IoU obtained between it and the ground truth bounding box.

The quality with which a sensed footprint models the real shape and location of an object is computed as the Intersection over Union (IoU) between the bounding box of the ground truth and the convex hull of the impacts on the object. Henceforth, this reference metric is denoted as IoU(PC). The higher the IoU(PC) value, the better the object is described and the better results from object estimation are expected. Fig. 1 shows a bounding box confronted with two different footprints. Better object shape estimation is expected in the left one, where IoU(PC) is significantly higher.

3.2. Object detection capability

This evaluation aims to measure how easily objects can be segmented from the OG.

3.2.1. Object segmentation method

Object segmentation from OGs is an essential task for object detection and multiple approaches can be found in the literature. These can be divided in three groups: (i) clustering based on neighboring cells feature similarity [5], (ii) additionally guided by objects extracted in previous frames [2] and (iii) based on neural networks [36].

Given that OGs are a discretization of the space, most of object segmentation methods employ proximity criteria between occupied cells. For example, from the just cited works, [5] clusterizes cells using a Density-based Spatial Clustering of Applications with Noise (DBSCAN) [37]. [2] extracts objects using tracks information, but the obtained clusters can be additionally expanded by evaluating adjacent cells. [36] uses a connected component algorithm during training data generation.

Therefore, a simple but generalizable distance-based clustering is employed to segment objects from the OG. Henceforth, these objects are referred as Distance-based Clustered Objects (DCO).

Cells with an occupied value higher than the threshold α_{occ} are clusterized with the Connected Components Clustering algorithm [38]. An 8-connected neighborhood kernel is employed assuming that: (i) compact objects are more easily detected and better described and (ii) object merging is minimized at its most.

Since this evaluation addresses scenarios with pedestrians, which can be represented by few cells, clusters are accepted despite of their size. Nevertheless, given that in OGs it is common to find noisy occupied cells in occluded areas, only clusters which have been observed in the current frame are accepted.

For every accepted cluster, the convex hull of the cells is computed, representing, in a consistent way with the OG's model-free estimation, the estimated location and shape. In addition to that, if dynamic states are also estimated by the OG, e.g. DOGs, a label concerning its dynamic state – *dynamic* or *static* – is calculated with respect the weighted average velocity of all cells and the threshold value v_{static}.

Therefore, DCOs are defined by: (i) the number of clustered cells, (ii) the convex hull of the clustered cells and (iii) a label concerning its dynamic classification.

3.2.2. Detection function

The detection function determines which GTOs are considered as detected and by which DCO. This is an important process that must be defined with respect the evaluated framework. In OGs it is common to find objects split in groups or with small footprints. Therefore, detection functions based on distance between centers or IoU, as the ones used in [34,35], may not be suitable. On the other hand, LiDAR data is accurate and provides measurements of occupied and free. As a result, an accurate description of the environment is expected, i.e. occupied space in the location of LiDAR-sensed objects and free space where beams have passed through.

A GTO is therefore considered as detected if it overlaps with the cells of at least one DCO, i.e. if occupied space has been modeled in its position. If more than one DCO overlaps the GTO, the one with highest IoU is associated, being the IoU computed between the bounding box of the GTO and the convex hull of the cells of the DCO.

3.2.3. Detection scores

The detection scores are divided in three groups: (i) occupancy detection capability score, (ii) quality clustering score and (iii) dynamic segmentation score.

The Occupancy Detection Capability Score (*ODCS*) measures the probability of estimating occupied space in the objects location. It is the most basic indicator and it is based on the idea that if occupied space has been estimated, then object's information can be inferred. Using the detection function presented above, the number of detected GTOs ($n_{detected}$) is computed. The *ODCS* relates the number of detected GTOs with the total number of GTOs (n_{GTO}):

$$ODCS = \frac{n_{detected}}{n_{GTO}} \tag{1}$$

ODCS is a permissive score which does not take into account relevant clustering phenomenons, such as noise, split or merge, which may harm the estimation and posterior steps, e.g. filtering or data association. Hence, in order to measure the quality of the extracted objects three Quality Clustering Scores (*QCS*) are proposed:

- Quality clustering noise score (QCS_{noise}): it is a common practice in clustering tasks to define a minimum size in order to consider a cluster as reliable. Therefore, a score measuring the probability of wrongly considering a DCO as noise is defined: every detected GTO is labeled as *noise* if the number of cells of its associated DCO is less than the threshold n_{cells}. The noise score is computed as:

  $$QCS_{noise} = 1 - \frac{n_{noise}}{n_{detected}} \tag{2}$$

  where n_{noise} is the number of objects labeled as *noise*.
- Quality clustering merge score (QCS_{merge}): objects close to each other can be merged into a single one, thus obtaining a bad object segmentation of both. A detected GTO is labeled as *merged* if its associated DCO overlaps with others GTOs or if its area A is far smaller than the area of its associated DCO (seeking to control the merge with objects un-labeled by the ground truth, e.g. vegetation):

  $$\frac{A[GTO]}{A[DCO]} < \beta_{merge} \tag{3}$$

  being $\beta_{merge} \in [0, 1]$. Therefore, the merge score is computed as:

  $$QCS_{merge} = 1 - \frac{n_{merged}}{n_{detected}} \tag{4}$$

  where n_{merged} is the number of objects labeled as *merged*.
- Quality clustering split score (QCS_{split}): as explained, when working with LiDAR data it is common to find objects divided into several parts, e.g. the rear and the roof of a vehicle might be detected individually. For this reason, OGs are prone to object splitting. However, it should also be taken into account since object splitting may lead to wrong objects estimations and can provide a good indication of the objects compactness. A GTO is labeled as *split* if it is overlapped by more than one DCO. The split score is computed as:

  $$QCS_{split} = 1 - \frac{n_{split}}{n_{detected}} \tag{5}$$

  where n_{split} is the number of objects labeled as *split*.

These three *QCS* are jointly represented as the Joint Quality Clustering Score (*JQCS*), which expresses the probability that a reliable and representative object segmentation has been obtained:

$$JQCS = \frac{1}{3}\sum_{i=1}^{3} QCS_i \tag{6}$$

The *JQCS* measures the quality of the obtained clusters in terms of how good the clustering step has been performed, but it does not provide information about how representative are the obtained clusters of the actual object's size and location. Therefore, in addition to the above three metrics, the IoU between the bounding box of every detected GTO and the convex hull of its associated DCO is computed (refer to the aforementioned explanation in Section 3.1.1):

$$mIoU_{proximity} = \frac{1}{n_{detected}} \sum_{i=1}^{n_{detected}} IoU_i \tag{7}$$

Lastly, as segmentation of dynamic obstacles is a common outcome of OGs [2,3], a score measuring the ability of correctly differentiating between static and dynamic obstacles is proposed. Every detected GTO is considered as: Dynamic True Positive (DTP) if the GTO and its associated DCO are both dynamic, Dynamic False Positive (DFP) if the GTO is static and the DCO is dynamic, Dynamic True Negative (DTN) if both are static and Dynamic False Negative (DFN) if the GTO is dynamic but the DC is static. A binary dynamic label is associated with every detected GTO (see Section 3.2.1), from which the F_1 score, which measures the classification accuracy taking into account Precision and Recall, is computed:

$$F_1^{Dyn} = \frac{2DTP}{2DTP + DFP + DFN} \tag{8}$$

3.3. Object features quality

After object segmentation, each group of cells is represented at an object-level by set of features, usually: position, size, heading and velocity. This evaluation aims at measuring how easy is to obtain good object features from the OG given an ideal clustering.

3.3.1. Ideal object segmentation

Seeking to decouple from the segmentation task and avoid inherent clustering problems – merge and split – an ideal clustering is applied by taking advantage of the ground truth's bounding boxes. Thus, occupied cells gathered by the GTOs bounding boxes are clusterized. Then, these clusters are expanded to neighboring unclusterized cells with a 8-connected neighborhood kernel, being this process performed i_t times. Henceforth these extracted objects are referred as Ideally Clustered Objects (ICO). Again, only objects which have been observed in the current frame are accepted and each GTO overlapping with the cells of an ICO is considered as detected.

For each ICO, an object-level representation is generated. Shape and location are represented by a minimum oriented bounding box. This bounding box is computed applying a search-based algorithm that selects the best-fitting box based on a variance minimization criteria — a minimum rectangle gathering all cells is computed for every angle and the one that minimizes the variance on the distance between the cells and the rectangle's borders is selected [39]. The dynamic state is expressed by the module and the angle and it is estimated as the average of all the cells' velocity weighted with respect the occupied value. Lastly, the oriented bounding box is the common object-level representation, but, as explained in Section 3.2.1, the convex-hull is consistent with the OG model-free representation. Therefore, despite it is not a typical object-level feature, the convex hull of the clustered cells is obtained in order to calculate the IoU metric (7) with ideal clustering.

Therefore, ICOs are defined by: (i) a bounding box with center, orientation and size, (ii) a velocity vector defined by module and angle and (ii) the convex hull formed by its cells.

3.3.2. Object features scores

Similarly to the scores of the public dataset nuScenes [34], the following object-level metrics are computed for the detected GTOs:

- Mean Absolute Translation Error (MATE) and Mean Square Translation Error (MSTE): the location error is defined by the Euclidean distance between the bounding boxes centers.
- Mean Absolute Scale Error (MASE) and Mean Square Scale Error (MSSE): the error in the estimated scale is computed as $(1 - IoU_{boxes})$, being IoU_{boxes} the IoU between the GTOs' bounding boxes and the ICOs' bounding boxes, after location and orientation alignment.
- Mean Absolute Velocity Error (MAVE) and Mean Square Velocity Error (MSVE): the error on the velocity module is defined as the absolute velocity module difference.
- Mean Absolute Velocity Orientation Error (MAVOE) and Mean Square Velocity Orientation Error (MSVOE): the error on the velocity orientation is calculated as the absolute yaw angle difference between the velocity vectors. It is only computed for dynamic objects $(v_{GTO} > v_{static})$
- Mean Absolute Box Orientation Error (MABOE) and Mean Square Box Orientation Error (MSBOE): the error on the orientation of the estimated bounding box is defined as the minimum yaw absolute angle difference between the four possible headings of the bounding boxes.

The five mean absolute errors (MAE) are jointly represented as the Joint Feature Mean Score (JFMS):

$$JFMS = \frac{1}{5}\sum_{i=1}^{5}\left(1 - \min\left(1,\ \frac{MAE_i}{E_i^{max}}\right)\right) \tag{9}$$

where E_i^{max} refers to the maximum acceptable error for each object-level metric.

And the five mean square errors (MSE) are jointly represented as the Joint Feature Mean Square Score (JFMSS):

$$JFMSS = \frac{1}{5}\sum_{i=1}^{5}\left(1 - \min\left(1,\ \frac{MSE_i}{(E_i^{max})^2}\right)\right) \tag{10}$$

Additionally, as in Section 3.2, the IoU between the GTOs bounding box and the convex hull of the objects ideally clusterized is computed and summarized as the average $mIoU_{ideal}$.

3.4. Overall score

Five different scores have been proposed in order to measure the quality of the OG from an object estimation perspective: (i) number of detections (*ODCS*), (ii) clustering easiness (*JQCS*), (iii) accuracy of the dynamic segmentation (F_1^{Dyn}), (iv) object features estimation easiness (*JFMS*) and (v) quality of the modeled footprint (*mIoU*). Depending on the developer, each task may have higher or lower relevance. Nevertheless, in order to express the overall result with a single understandable score, the following Object Estimation Score (OES) is proposed:

$$OES = ODCS\frac{1}{4}\left(JQCS + F_1^{Dyn} + JFMS + mIoU\right) \tag{11}$$

being:

$$mIoU = \frac{1}{2}\left(mIoU_{proximity} + mIoU_{ideal}\right) \tag{12}$$

4. Scan rendering methods evaluated

From the scan rendering methods introduced in Section 2.1, three line-based and three angular-based methods have been chosen for evaluation according to three main motivations: (i) fast computation, (ii) model accuracy and (iii) balance between computation speed and model accuracy. Table 1 summarizes their main advantages and disadvantages of the selected methods and Fig. 2 provides an illustrative example of their implementation .

4.1. Line-based methods

4.1.1. Line-drawing based method

Bresenham's line-drawing algorithm [15] is selected as representative of computationally efficient line-based methods — an example of a fast OG implementation can be found in [6]. When applying this method, the space traveled by the beam is approximated as a rasterized line with width equal to the cell's size (see Fig. 2(a)). Notice that some cells truly traversed by the center of the beam are missed. All cells are considered equally, thus equally weighted.

4.1.2. Traversal based method

Traversal algorithms are considered as computationally affordable but accurate, as all the cells traversed by the beam are considered, compare Figs. 2(a) and 2(b). Again, all cells are equally weighted. In this work cells are selected if the beams' angle is within the minimum and maximum angles of the cell. A variant with lower computational cost can be found in [16].

4.1.3. Weighted-line based method

Xiaolin Wu's algorithm [19] is chosen for line-based accurate model — an OG application of this method can be found in [18]. This algorithm computes all the beam's straddling cells and assigns to each one a weight depending on its proximity to the beam's centerline. As a result, information loss is reduced and the certainty with which the beam influences the cell is modeled.

4.2. Angular-based methods

4.2.1. Beam-by-beam method

The work of [24] named as *Beam-by-beam* is selected as representative of fast angular-based algorithms. This approach makes two assumptions: (i) beams do not overlap and (ii) each cell is modeled only by one beam, the one whose field-of-view (FOV) gathers its center. Notice that these assumptions do not stand for beams of different sensors or, in the case of multi-layer sensors, for beams of different layers. The FOV of a beam is delimited

Table 1
Scan rendering methods selected for evaluation and their main motivation.

Methods	Motivation	Advantages	Disadvantages
Line-drawing	Fast computation	High computational efficiency.	Some cells truly traversed can be missed. All cells are equally weighted.
Traversal algorithm	Balance	Computationally efficient. All traversed cells are selected.	All cells are equally weighted.
Weighted line	Model accuracy	All cells straddling the beam are selected. Cells are weighted depending on its proximity to the beam.	Low computational efficiency.
Beam-by-beam	Fast computation	High computational efficiency.	Each cell is assumed to be influenced by a single beam. All cells are equally weighted.
Polar grid	Balance	Fast computation capable. Consistent with LiDAR acquisition process.	Double discretization, first into the polar grid and then into Cartesian grid. All cells are equally weighted.
Weighted angular sector	Model accuracy	Cells inside the angular sector are weighted depending on its proximity to the center of the beam.	Lowest computational efficiency.

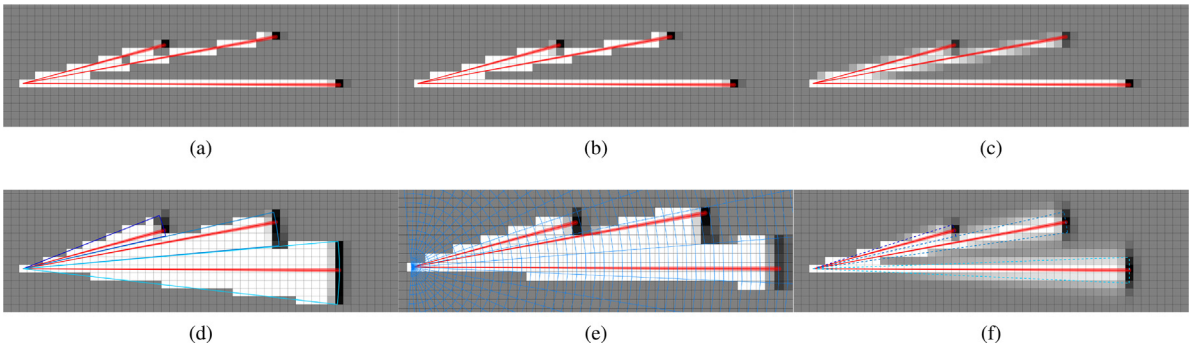


Fig. 2. Illustrative example of the tested methods: (a) Line-drawing based method, (b) Traversal based method, (c) Weighted-line based method, (d) Beam-by-beam based method, (e) Polar grid based method and (f) Weighted-angular based method. Occupancy probability is drawn in gray-scale. Laser beams are drawn as thin red cones. For the angular-based methods, the area covered by each beam is colored with bluish lines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

by bisectors formed with its contiguous beams. This algorithm was originally designed for LiDAR sensors with constant angular steps, but this condition may not be met by other sensors or if ego-motion correction is applied to the point cloud. In this case, the area covered by each beam is variable and a security upper threshold should also be included. An example of the resultant algorithm is shown in Fig. 2(d), where bluish lines denote the width of the beams.

4.2.2. Polar grid based method

Polar grid based methods are considered a balance between accuracy and computational cost. First, an OG centered on the sensor's origin and dividing the environment in terms of radial distance and angle is computed. Then, this OG is transformed into the final Cartesian OG. An example of polar occupancy grid can be found in [11]. The initial polar representation is accurate with respect to laser range-finders usual functioning, but the transformation from polar to Cartesian coordinates is a complex task — it has been extensively studied in [10], the sampling method is employed in this work.

4.2.3. Weighted-angular based method

Modeling the beams as wide angular sectors presents the drawback that not truly sensed cells can be also updated. In order to address this problem, an algorithm inspired by [14,25], named *Weighted-angular sector*, is considered in this evaluation.

Each beam is modeled as an angular sector with constant angular width. Cells gathered inside this area are weighted using a Gaussian probability density function, relative to the angular proximity between the cells' center and the beam's centerline. Since angular weighting may dismiss radially-closer cells, cells directly traversed by the centerline of the beam receive full confidence. The computation of this weight β^c is as follows:

$$\beta^c = \begin{cases} 1 & \theta^{c,min} \leq \theta^z \leq \theta^{c,max} \\ g(\theta^c, \theta^z) & otherwise \end{cases} \tag{13}$$

being:

$$g(\theta^c, \theta^z) = \exp\left(-0.5\frac{(\theta^c - \theta^z)^2}{\sigma_\theta^2}\right) \tag{14}$$

where θ^c and θ^z refer to the angle of the center of the cell and the angle of the beam respectively, and $\theta^{c,min}$ and $\theta^{c,max}$ denote the minimum and maximum angles of the cell. σ_θ defines the standard deviation computed accordingly to the beam's angular width.

Fig. 2(f) shows an example of the result obtained. The bluish angular sectors denote the reach of the angular standard deviation.

5. Implementation details

This section introduces the OG framework with which the six scan rendering methods are tested. The necessary details for their integration are provided.

5.1. Occupancy grid framework overview

Different OG frameworks can be found in the literature [40–43]. The selected framework is based on the DOG proposed

in [43]. This approach is popular in autonomous driving object detection and tracking tasks [1,3,4,36], since it properly fulfills the requirements of driving scenarios: (i) real-time estimation of ego-vehicle's surrounding environment, (ii) fast modeling of occupied and free spaces and (iii) robust modeling and estimation of objects' dynamics. It includes three main steps: (i) grid prediction, (ii) observed occupancy grid calculation and (iii) grid estimation update. In order to estimate the dynamic state of the cells, a particle filter is included by linking the particles' weight to the cells' occupancy value.

Two important considerations must be taken into account hereinafter: (i) the observed occupancy grid is computed taking into account all the available sensed information at the current frame; (ii) the occupancy of cells is represented in terms of the Dempster–Shafer Theory (DST) of Evidence [44,45] instead of in terms of occupancy probability. This is done in order to differentiate between unknown cells and cells with conflicting estimations.

5.2. Observed occupancy grid computation

The computation of an observed OG relies on two processes: (i) the sensed data rendering and (ii) the data fusion. Scan rendering methods compute the cells which are updated by each beam, but a single cell can be updated by multiple beams, for example when using multi-layered sensors. In this case, the occupancy estimations of the different beams have to be fused.

This data fusion is addressed following the strategy proposed in [46]. Each beam provides two values: the occupancy distribution and a confidence measurement for the modeled occupancy distribution. The estimations from multiple beams updating the same cell can be fused as a weighted average:

$$P(O^c|z_1,\dots,z_n) = \left(\sum_{i=1}^{n} w^c(z_i)\right)^{-1} \sum_{i=1}^{n} w^c(z_i) P(O^c|z_i) \tag{15}$$

where $P(O^c|z_i)$ and $w^c(z_i)$ are the occupancy probability and the confidence measure estimated for the cell c by the measurement z of the ith beam, respectively; and n defines the number of beams updating the cell. The computation of $P(O^c|z_i)$ and $w^c(z_i)$ is detailed in Section 5.3.

The resultant fused estimation is then expressed in terms of DST, being the frame of discernment the occupied and free events ($\Omega = \{O, F\}$). This way, each cell contains a mass for occupied $m(O)$ and a mass for free $m(F)$ computed as:

$$m(O^c) = W \cdot P(O^c|\mathbf{z}) \tag{16}$$

$$m(L^c) = W \cdot \left(1 - P(O^c|\mathbf{z})\right) \tag{17}$$

being W the overall confidence:

$$W = \min\left(1, \sum_{i=1}^{n} w^c(z_i)\right) \tag{18}$$

Fig. 3(a) shows an example of the occupancy distribution and confidence measurement modeled for a single beam and Fig. 3(b) shows its representation in terms of DST.

5.3. Occupancy distribution and confidence measurement

The occupancy distribution and the confidence measurement are calculated based on the sensor model. In this work, an inverse sensor model taking into account the noise in the measurement is employed. The two most common approaches are the Dirac model and the Gaussian model. Other more elaborated sensors models can be found in [40,47]. The Dirac model sets only the cell gathering the impact as occupied. On the contrary, the Gaussian model defines the occupancy as a Gaussian function depending on the distance between the center of the cell and the impact. [10] proposed to choose the Dirac model when the uncertainty in the measurement is far smaller than the cell's size and the Gaussian model in the opposite case. Opposite opinions can be found in terms of neglecting the uncertainty. [9] applies discrete values of occupied, free and unknown (Dirac model) seeking for a faster OG generation. On the other hand, [40] suggests to enlarge the value of uncertainty parameters since models tend to ignore some features and dependencies, for example, in the context of OGs, these may be the uncertainty on the localization or the discretization error. Therefore, both methods are tested in this work.

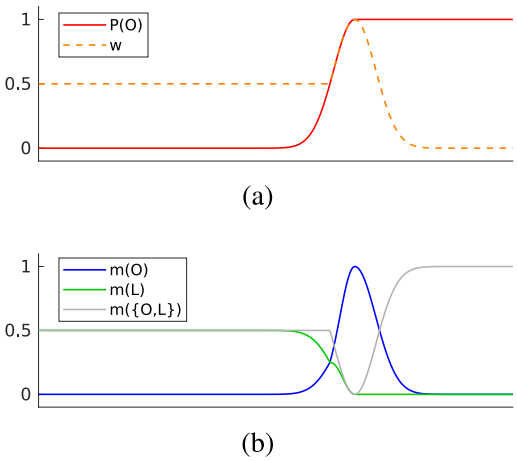


Fig. 3. Illustrative example: (a) occupancy probability and confidence measure computed for a single beam, and (b) representation in terms of DST.

5.3.1. Dirac model

The Dirac model assumes that the uncertainty on the measurement is negligible and the occupancy distribution $P(O^c|z)$ takes therefore discrete values separated by the cell gathering the impact C^z:

$$P(O^c|z) = \begin{cases} 0 & if (c \neq C^z \text{ and } d^c < d^z) \\ 1 & otherwise \end{cases} \tag{19}$$

$$w^c(z) = \begin{cases} \beta^c \cdot w_{occ} & if (c = C^z) \\ \beta^c \cdot w_{free} & if (c \neq C^z \text{ and } d^c < d^z) \\ 0 & if (c \neq C^z \text{ and } d^c > d^z) \end{cases} \tag{20}$$

where d^z denotes the distance measured by the beam and d^c the distance towards the center of the cell c. w_{free} and w_{occ} are design parameters that define the maximum confidence for a free and occupied measurement, respectively. β^c corresponds to the inherent weighting of the applied method.

In practice, this model can be directly applied for line-drawing, traversal and polar grid based approaches, but not for weighted-line, beam-by-beam and weighted-angular sector based approaches. For the weighted-line based approach, the cell pairing cell C^z is also considered as occupied and fully confident. For the methods beam-by-beam and weighted-angular, a first 1D discretization along the beam centerline has to be performed; then, each 2D cell obtains its corresponding occupancy and confidence values based on its radial distance. Fig. 4 illustrates the process.

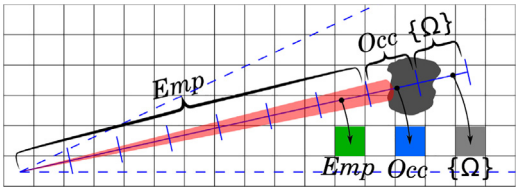


Fig. 4. Illustrative example of the Dirac model for the methods beam-by-beam and weighted angular sector.

5.3.2. Gaussian model

The Gaussian model assumes that the uncertainty on the measurement is not negligible and therefore models the occupancy distribution $P(O^c|z)$ and the confidence measurement $w^c(z)$ as a Gaussian function along the beam's centerline:

$$P(O^c|z) = \begin{cases} g(d^c, z) & if(d^c \leq d^z) \\ 1 & if(d^c > d^z) \end{cases} \tag{21}$$

$$w^c(z) = \begin{cases} \beta^c \cdot \max\left(w_{free},\ g(d^c, z)\right) & if(d^c \leq d^z) \\ \beta^c \cdot \min\left(w_{occ}, g(d^c, z)\right) & if(d^c > d^z) \end{cases} \tag{22}$$

being:

$$g(d^c, z) = \exp\left(-0.5\frac{(d^c - d^z)^2}{\sigma^2}\right) \tag{23}$$

where σ is the standard deviation of the measurement's uncertainty.

In contrast to the Dirac approach, this model can be directly applied in all the methods.

5.4. Height and ground points filtering

LiDAR sensors provide a large amount of data. Nevertheless, not all of it is valuable for OG representation.

- Ground filtering: A great percentage of the laser-beams impact on the ground and not on relevant obstacles. Therefore, a ground-obstacle segmentation step has to be performed before modeling the Observed OG. In this work the Channel-based with median filter approach proposed in [48] is used. For beams hitting the ground, Eqs. (20) and (22) are substituted by (24) and (25).

$$w^c(z) = \begin{cases} \beta^c \cdot w_{free} & if(c \neq C^z \text{ and } d^c < d^z) \\ 0 & otherwise \end{cases} \tag{24}$$

$$w^c(z) = \begin{cases} \beta^c \cdot w_{free} & if(d^c - s \leq d^z) \\ 0 & otherwise \end{cases} \tag{25}$$

  where s is the size of the cell.
- Height filtering: LiDARs with several layers emit laser-beams with different pitch angle (vertical inclination), capturing information at different heights. In this work, information captured above h_{max} is dismissed. For this purpose, when updating a cell, the height of the beam at the cell's center is computed.

6. Experiments

This section describes the set of experiments performed in order to compare the six evaluated scan rendering methods and showcase the usefulness of the evaluation proposed.

Experimentation is conducted with the public dataset nuScenes [34]. The subset *mini* is selected since it provides four hundred labeled samples from ten different representative urban scenes. Laser data was captured by a 32-layer LiDAR sensor mounted on top of the vehicle. The FOV covers 360° around the vehicle and with an angular resolution, between beams of the same layer, of approximately 0.33°. Fig. 5 shows a frame example.

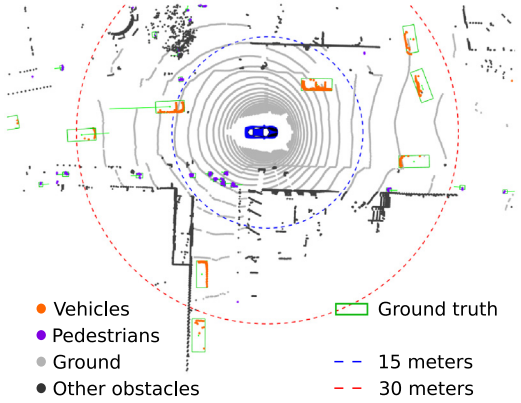


Fig. 5. nuScenes frame example. LiDAR data is drawn with colored dots depending on its class-label, and the ground truth labeled objects, with the corresponding bounding boxes and velocity vectors in green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6.1. Implementation details

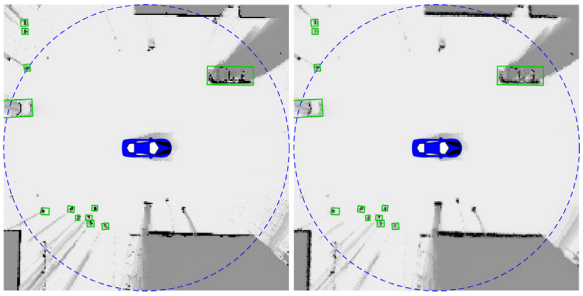
Hereunder, the implementation details related with the OG computation and benchmark evaluation are given.

The selected parametrization has been experimentally adjusted. The values chosen for the six OGs seek to obtain compact and well defined footprints. The evaluation parameters are defined intended to be strict with the clustering process and to bound significantly high errors that can be obtained from new objects entering in the field of view or poorly representative footprints.

- DOG general parameters: The grid is centered on the ego-vehicle, covering a FOV of 360° around. Because of parallel programming reasons the size of the grid is defined as 512×512, being the cells size is set to 0.15 m (the large and width length are therefore 76.8 m). 2^{21} particles are used for cell's dynamic state estimation.
- Observed OG: For the beam-by-beam method, the maximum bisector allowed is 0.5°. The polar grid resolution is also set as 0.5° with a radial resolution of 0.15 cm. In the weighted-angular strategy a standard deviation $\sigma_\theta = 0.25°$ is selected. The standard deviation for the Gaussian model is set as 0.075 m. The maximum confidence parameters for free and occupied space are set as $w_{occ} = 1$ and $w_{free} = 0.3$. Lastly, the height threshold is defined as $h_{max} = 1.5$ m.
- Ground truth data: The evaluation is performed with respect to the detection of vehicles and pedestrians within the range of the OG. Additionally, in order to avoid detection problems caused by wrong obstacle-ground point cloud segmentation, only objects with at least three laser impacts labeled as obstacle are taken into account.
- Evaluation: The occupied threshold is conservatively set as $\alpha_{occ} = 0.1$. The ideal clusters are expanded $i_t = 3$ times. The thresholds to consider a DCO as noise, merged and dynamic are set to $n_{cells} = 3$, $\beta_{merge} = 0.6$ and $v_{static} = 1$ m/s, respectively. The maximum acceptable errors are set to: $E_{TE}^{max} = 5$ m, $E_{SE}^{max} = 1$, $E_{VE}^{max} = 5$ m/s and $E_{VOE}^{max} = 180°$ and $E_{BOE}^{max} = 45°$.

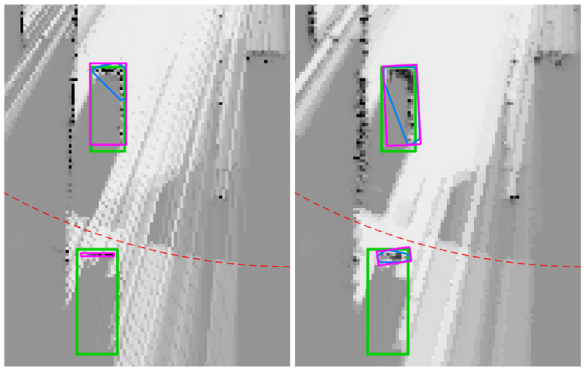
6.2. Qualitative evaluation

A qualitative evaluation seeking to illustrate (i) the differences between line-based and angular-based algorithms and (ii) the



(a) (b)

Fig. 6. OGs examples: (a) line-drawing Dirac based OG and (b) polar Gaussian based OG. Occupancy is represented in gray-scale, ground truth bounding boxes are drawn in green and the dashed blue circle denotes a distance range of 15 m. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) (b)

Fig. 7. Example of OG at farther distance: (a) line-drawing Dirac based OG and (b) polar Gaussian based OG. The red dashed line denotes the range of 30 meters from the LiDAR sensor. Ground truth bounding boxes are drawn in green, the estimated bounding box with ideal clustering in magenta and the estimated convex hull with proximity clustering in blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

feasibility of the proposed evaluation method, is shown in this section.

Since visual inspection of OGs is complex and subjective, and the most notable differences are caused by the line/angular shape assumption, this qualitative evaluation is performed comparing the considered two most popular but opposing methods: line-drawing method with Dirac model and polar grid method with Gaussian model.

- Line-drawing Dirac OG: this strategy models occupancy only along the centerline of the beam and assigns discrete values – occupied, free and unknown – to the traversed cells.
- Polar Gaussian OG: this strategy first discretizes beams' data in angular sectors which, all together, cover the 360° FOV, and then models occupancy along these angular sectors assigning continuous values.

As is mentioned in Section 1, the decision of fulfilling the uncovered space between beams with the data of these beams is a matter of discussion in the literature. Given the rotational behavior of common LiDAR sensors, the farther the radial distance the bigger the space between beams. As it can be seen in Fig. 5, with the nuScenes LiDAR's setup the captured information is dense at short distances, e.g. 15 m, and becomes relatively sparse at distances farther than 30 m. The obtained OGs at both ranges are shown in Figs. 6 and 7.

At a range of 15 m, the distance between beams of the same layer is approximately 0.08 m, less than the size of a cell. Therefore, closer cells are covered by one or more laser beams and both methods provide similar results despite of their beam's shape assumption. Indeed, the main differences are caused by the Dirac/Gaussian models. In the Line-drawing Dirac OG, objects' footprints are thinner and cells have high occupied values, while in the Polar Gaussian OG, objects are thicker but blurred.

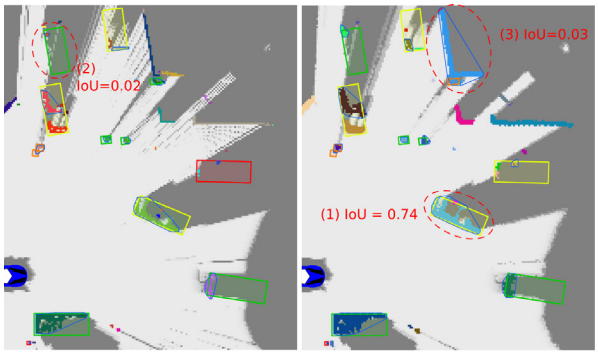
At farther distances, the space between beams increases, e.g. approximately 0.17 m at 30 m, and more notable differences appear between the two methods. When applying line-based scan rendering algorithms, the space between beams is modeled as unknown. Thus, an irregular free space estimation and non-compact objects are generated. On the contrary, when applying angular-based algorithms the space between beams is approximated by the angularly close estimated occupancy values. Therefore, a smoother occupancy distribution is obtained, but object's shapes are slightly enlarged and distorted.

This difference can be noticed in the vehicles of Fig. 7. The polar Gaussian OG models the upper car with a compact footprint, leading to a better object segmentation by distance-based clustering (blue line). On the contrary, when estimating the oriented bounding boxes, the line-drawing Dirac OG obtains better results, since it does not distort the L-shape and I-shape captured by the LiDAR.

Fig. 8 shows another scene where the detection benchmark is illustrated. Every road user in this scene is counted as detected since occupied space has been modeled inside its bounding boxes. However, the quality of each detection is not the same. For example, vehicle (1) is clearly detected by the LiDAR and, therefore, a good object representation is achieved in both OGs, which is correctly modeled by the IoU score. Nevertheless, additional clusters are also obtained inside the GTO's bounding box, which is taken into account in the split score. On the contrary, vehicle (2) is correctly segmented from the footprint perceived by the LiDAR. No merge or split effects appear, but this detection barely provides information and, therefore, obtains a low IoU value. Lastly, pedestrian (3) is correctly segmented in the line-drawing Dirac OG but is merged with the buildings in the polar Gaussian OG. This object is counted as detected for both because occupied space has been modeled in its location and by applying further segmentation steps – e.g. particle labeling – it may be correctly segmented. However, its clustering quality score is strongly penalized by the merge and IoU scores.

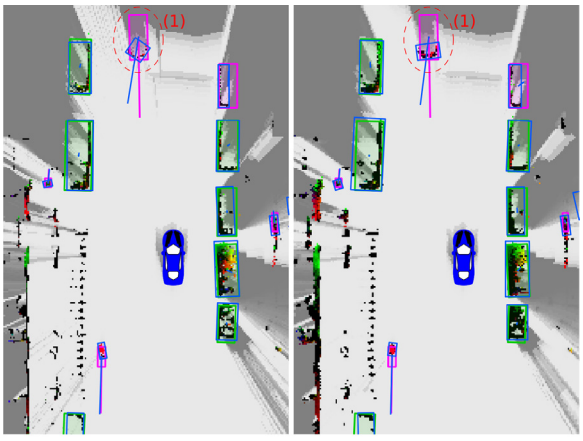
Fig. 9 shows the object features quality evaluation. As it can be seen, similar visual results are obtained from both methods, being the most notable difference the oriented bounding box estimated for vehicle (1). In the case of the line-drawing Dirac based OG, the number of occupied cells is not enough to accomplish a good orientation estimation. This example is intended to show the feasibility of the features estimation process, but also the usefulness of the reference metric IoU(PC). The bounding box estimation strongly relies on the footprint captured by the sensor, thus errors in location and shape are expected to increase inversely with IoU(PC).

As it has been shown, obtaining conclusions from qualitative evaluation is complex. The different methods may produce different results, but these discrepancies may imply an advantage or a disadvantage depending on the case. For example, employing the Gaussian model and the angular-shape assumption seeking to model more occupied space can be an advantage in situations such as Fig. 7(b), since objects are more easily identified. However, it also may promote noise, e.g. the vehicle (1) of Fig. 8. Similarly, a more conservative approach, where only the cells gathering the impacts are classified as occupied, may provide a



(a) (b)

Fig. 8. Object detection capability evaluation. (a) Line-drawing Dirac based OG and (b) Polar Gaussian based OG. The OG is colored with respect the results of the distance-based clustering method applied for object segmentation and the GTO's bounding boxes are colored following the clustering quality scores explained in Section 3.2: red for *noise*, orange for *merged*, yellow for *split* and green for correctly detected clusters. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



(a) (b)

Fig. 9. Object features evaluation. (a) line-drawing Dirac based OG and (b) polar Gaussian based OG. Cells are colored with respect the DOG obtained, free and unknown cells are drawn in gray-scale, occupied cells are drawn in black if classified as static or with colors corresponding to their orientation, if dynamic. The estimated oriented bounding boxes and velocity vectors are represented in blue, the ground truth bounding boxes are drawn in green if the IoU(PC) is higher than 0.35 and in purple if not. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

better representation of the objects real shapes (see Fig. 7(a)), but it may also lead to a reduced number of occupied cells with which an acceptable oriented bounding box adjustment cannot be obtained.

6.3. Quantitative evaluation

As exposed in Section 6.2, visual conclusions are hardly achievable and differences between methods can lead to both advantages and disadvantages. Therefore, in order to obtain concluding results, a quantitative evaluation has to be performed. In the following, the DOGs resulting from the six scan rendering methods tested are evaluated using the proposed evaluation method.

Table 2 shows the obtained results corresponding to the object detection capability evaluation. As it can be seen, all methods detect approximately the same number of objects.

Nevertheless, DOGs employing the Gaussian model with the weighted-line, polar grid and weighted-angular methods tend to perform the best with respect the scores of noise, split and IoU and the worst with respect the merge score. On the contrary, DOGs employing the Dirac model and line-drawing, traversal and beam-by-beam methods behaves reversely. These results can be explained by the amount of occupied space modeled by each strategy. Line-drawing and Traversal based methods with Dirac model only set as occupied the cells gathering the laser-impacts. As a result, they tend to model thin occupied spaces and unknown or free gaps between occupied cells (see Fig. 7). The beam-by-beam approach performs similarly because the assumption that each cell is only updated by one beam per layer makes that, at closer distances, some information is dismissed and objects can also be modeled with non-occupied gaps in between. While these facts benefit the objects differentiation (n_{merged}), they also result in less representative footprints (n_{noise}, n_{split}, $mIoU_{proximity}$).

Regarding the dynamic classification (F_1^{Dyn}), the variations on the scan rendering method does not severely affect the dynamic estimation performed by the DOG's underlying particle filter, but, again, the Gaussian model seems to perform slightly better. Table 3 summarizes the results obtained for the object features evaluation. Notice that the MAVOE score is only computed for dynamic objects ($v_{GTO} > v_{static}$) and the MABOE score does not include pedestrians, only vehicles. It can be also seen that, with an ideal clustering, all DOGs present similar object feature estimation capabilities. Note that the Gaussian model performs better than the Dirac model and that slightly better results are obtained by the weighted-line, polar grid and weighted-angular based methods. Nevertheless, there is no strategy which stands out the others significantly in all metrics.

Lastly, Table 4 shows the OES, summarizing the overall result. It can be easily noticed that methods employing the Gaussian model performs better. With this model, differences between the line and angular-based methods are minor, but there is a clear tendency to obtain better results when seeking for more accurate scan rendering methods.

Fig. 10 shows the obtained results derived from the footprint captured by the LiDAR sensor — IoU(PC) metric. It can be seen how the quality of the clustering (JQCS) decreases as the IoU(PC) increases. This can be explained by the fact that objects perceived with small footprints are unlikely to be split or merged, contrary to objects with higher footprints (see the vehicles in Fig. 8). Conversely, the object features estimation (JFMS) improves with the IoU(PC). This is an expected result since the JFMS is computed with ideal clustering and the best the object is described, the easier the object's features estimation is. Note in this connection that capturing the complete footprint facilitates the estimation of the center point and size. Regarding the quality of the estimated footprint (IoU), the results obtained with both clustering methods increase with the quality of the footprint captured by the sensor. The results obtained for the ideal clustering (dashed lines) show that in general the OG representation enhances the object's shape and location estimation with respect the LiDAR footprint. At the maximum IoU(PC) values, grid representation results decreases primarily because occupied space is modeled outside the GTOs bounding boxes. In contrast, the results for the proximity clustering (solid lines) shows that the obtained clusters in general performs worse than the IoU(PC). This is mainly explained because the obtained clusters do not gather all the occupied space modeled for the object, i.e. clustering split, while the IoU(PC) is computed with ideal clustering. Overall, the polar grid, weighted-angular and weighted-line based methods with Gaussian model tend to obtain the best results in general.

Table 2
Detection capability quantitative evaluation. The results for each score are colored from red (worse) to green (best).

	$n_{detected}$		n_{noise}		n_{merged}		n_{split}		$JQCS$		$mIoU_{proximity}$		F_1^{Dyn}	
	Dirac	Gaus	Dirac	Gaus	Dirac	Gaus	Dirac	Gaus	Dirac	Gaus	Dirac	Gaus	Dirac	Gaus
Line-drawing	4989	4992	95	19	420	491	2807	2476	0.778	0.801	0.299	0.336	0.851	0.853
Traversal	4987	4993	127	5	358	501	2930	2385	0.772	0.807	0.280	0.343	0.848	0.856
Weighted-line	4993	4990	25	4	484	533	2568	2256	0.795	0.813	0.319	0.349	0.854	0.860
Beam-by-beam	4991	4993	65	12	356	472	2869	2525	0.780	0.799	0.276	0.327	0.844	0.859
Polar grid	4993	4993	29	4	484	578	2507	2237	0.798	0.812	0.322	0.352	0.837	0.847
Weighted-angular	4993	4993	13	3	457	536	2572	2282	0.797	0.812	0.319	0.348	0.843	0.854

Table 3
Object features quality evaluation.

	$MATE$ m		$MASE$		$MAVE$ m/s		$MAVOE$ °		$MABOE$ °		$JFMS$		$JFMSS$		$mIoU_{ideal}$	
	Dirac	Gaus	Dirac	Gaus	Dirac	Gaus	Dirac	Gaus	Dirac	Gaus	Dirac	Gaus	Dirac	Gaus	Dirac	Gaus
Line-drawing	0.543	0.535	0.623	0.586	0.568	0.568	10.510	10.382	4.586	4.552	0.799	0.807	0.875	0.883	0.435	0.459
Traversal	0.540	0.524	0.637	0.586	0.578	0.571	10.752	9.951	4.584	4.367	0.795	0.809	0.870	0.885	0.421	0.466
Weighted-line	0.543	0.531	0.598	0.583	0.563	0.557	10.236	10.021	4.575	4.407	0.805	0.809	0.881	0.885	0.451	0.467
Beam-by-beam	0.553	0.532	0.617	0.592	0.571	0.555	10.699	10.397	5.045	4.868	0.797	0.805	0.876	0.883	0.432	0.459
Polar grid	0.536	0.533	0.589	0.560	0.603	0.574	11.374	10.533	4.664	4.570	0.803	0.812	0.882	0.891	0.455	0.474
Weighted-angular	0.531	0.522	0.593	0.574	0.579	0.566	10.352	10.068	4.722	4.402	0.804	0.811	0.882	0.888	0.456	0.473

Table 4
Overall result summarized with the Object Estimation Score.

	Line-drawing	Traversal	Weighted-line	Mono-beam	Polar grid	Weighted-angular
Dirac	0.698	0.691	0.710	0.694	0.707	0.708
Gaussian	0.714	0.719	0.722	0.714	0.721	0.722

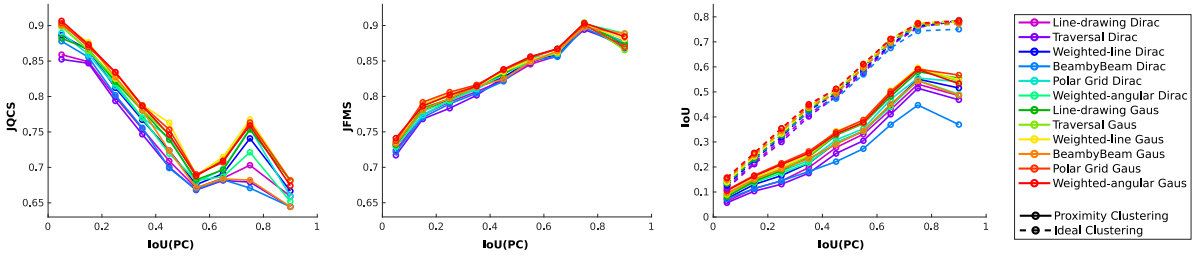


Fig. 10. Evaluation results represented against the IoU(PC) metric.

As explained in Section 2.1 and shown in Fig. 7, the main difference between line-based and angular-based scan rendering methods is the update of cells not truly traversed by the laser-beams, a condition which increases with distance. Additionally, estimation complexity also increases due to the increasing sparsity of the LiDAR data. Hence, Fig. 11 shows the results obtained with respect the distance to the ego-vehicle. It can be clearly noticed how as distance increases, the results for the JQCS begin to diverge, maintaining the best results the angular-based methods, followed by the weighted-line method. This results support the statement that fulfilling gaps between beams enhances the objects' shape estimation (see Section 2.1 and Fig. 7). This increasing complexity can be perfectly noticed in the object's features estimation and footprint estimation. Similar conclusions can be observed, methods based on Polar grid, weighted-angular and weighted-line, tend to perform the best in general. It is interesting to note that the beam-by-beam method with Gaussian model performs worse in terms of footprint quality (IoU) at closer distances. However, it is aligned with best angular-based methods at farther distances. This can be explained by the one beam per cell assumption, i.e. at closer distances important information may be dismissed, but, at farther distances, a cell is at most updated by one beam.

6.4. Results summary

All tested methods exhibit an acceptable behavior under the evaluation conditions, but some differences can be appreciated. The most significant dissimilarities are obtained from the choice of the selected model for the occupancy distribution and confidence measurement. In general, the Gaussian model provides significantly better results than the Dirac model, both in the qualitative and quantitative analysis.

Regarding the difference between using line-based or angular-based methods, qualitative differences are visually noticeable, but these are not translated to large quantitative dissimilarities. Angular-based methods perform slightly better at farther distances, but the line-based method focused on higher model accuracy – Weighted-line – obtains results alike.

With respect to the election between (i) fast computation, (ii) balance between computation speed and model accuracy and (iii) more accurate models; the results are consistent with the accuracy sought by each strategy. Methods focused on more accurate models perform better in general. Nevertheless, they are closely followed by the balanced methods. On the contrary, methods which make more simplifying assumptions, seeking for faster computation, perform worse.

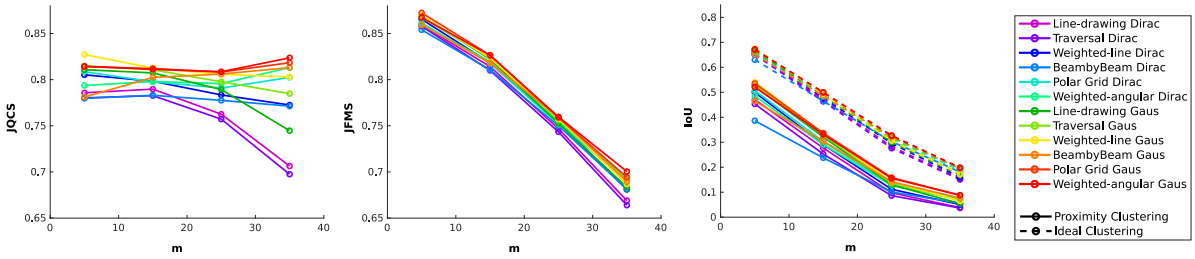


Fig. 11. Evaluation results represented against the distance to the ego-vehicle.

7. Conclusion

This work presents a comparison between six LiDAR scan rendering methods specifically selected to evaluate the main differences discussed in the literature. In addition, since no standardized OG benchmark has been found, an evaluation method focused on object detection is proposed.

Firstly, concerning the scan rendering methods, all tested methods perform acceptable under the evaluated conditions. Despite visual results, line-based and angular-based methods have been found to perform alike regarding the quantitative results. More noticeable differences have been found in relation to the occupancy distribution model, being clearly superior the Gaussian model over the Dirac model, and to the accuracy sought by the scan rendering method, obtaining better results those designed for accuracy and worse results those conceived for fast computation.

Secondly, an OG evaluation framework, intended to be repeatable with other OG approaches, has been proposed. It addresses the two main steps of object detection: (i) object segmentation and (ii) features estimation. Several metrics are provided and summarized into five easily understandable scores and one general score. In addition to that, the evaluation considers two fundamental factors, e.g. the distance to the sensor and the quality of the footprint captured by the laser sensor.

Future work will address the evaluation with other LiDAR sensors and with farther distance ranges in order to provide more generalized conclusions.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors are unable or have chosen not to specify which data has been used.

Acknowledgments

This work was supported in part by the Spanish Ministry of Science and Innovation with Project DISCERN, under Grant PID2021-125850OB-I00 and in part by the Community of Madrid through SEGVAUTO 4.0-CM Programme under Grant S2018-EMT-4362.

References

[1] J. Godoy, V. Jiménez, A. Artuñedo, J. Villagra, A grid-based framework for collective perception in autonomous vehicles, Sensors 21 (3) (2021) http://dx.doi.org/10.3390/s21030744, URL https://www.mdpi.com/1424-8220/21/3/744.

[2] S. Steyer, C. Lenk, D. Kellner, G. Tanzmeister, D. Wollherr, Grid-based object tracking with nonlinear dynamic state and shape estimation, IEEE Trans. Intell. Transp. Syst. 21 (7) (2020) 2874–2893, http://dx.doi.org/10.1109/TITS.2019.2921248.

[3] N. Engel, S. Hoermann, P. Henzler, K. Dietmayer, Deep object tracking on dynamic occupancy grid maps using RNNs, in: 2018 21st International Conference on Intelligent Transportation Systems, ITSC, 2018, pp. 3852–3858, http://dx.doi.org/10.1109/ITSC.2018.8569234.

[4] N. Rexin, M. Musch, K. Dietmayer, Fusion of object tracking and dynamic occupancy grid map, in: 2019 IEEE Intelligent Transportation Systems Conference, ITSC, 2019, pp. 4121–4127, http://dx.doi.org/10.1109/ITSC.2019.8917048.

[5] F. Gies, A. Danzer, K. Dietmayer, Environment perception framework fusing multi-object tracking, dynamic occupancy grid maps and digital maps, 2018, CoRR abs/1812.08449 arXiv:1812.08449.

[6] D. Rodriguez-Losada, P. de la Puente, A. Valero, P.S. Segundo, M. Hernando, Fast processing of grid maps using graphical multiprocessors, IFAC Proc. Vol. 43 (16) (2010) 342–347, http://dx.doi.org/10.3182/20100906-3-IT-2019.00060, 7th IFAC Symposium on Intelligent Autonomous Vehicles.

[7] y. Li, Stereo vision and LIDAR based Dynamic Occupancy Grid mapping: Application to scenes analysis for Intelligent Vehicles (Ph.D. thesis), (2013BELF0225) Université de Technologie de Belfort-Montbeliard, 2013.

[8] A. Souza, L.M.G. Gonçalves, Occupancy-elevation grid: an alternative approach for robotic mapping and navigation, Robotica 34 (11) (2016) 2592–2609, http://dx.doi.org/10.1017/S0263574715000235.

[9] F.M. Moreno, O. El-Sobky, F. Garcia, J.M. Armingol, Hypergrid: A hyper-fast ROS-based framework for local map generation, in: 2019 IEEE International Conference on Vehicular Electronics and Safety, ICVES, 2019, pp. 1–6, http://dx.doi.org/10.1109/ICVES.2019.8906334.

[10] M. Yguel, O. Aycard, C. Laugier, Efficient GPU-based construction of occupancy girds using several laser range-finders, in: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, pp. 105–110, http://dx.doi.org/10.1109/IROS.2006.281817.

[11] F. Homm, N. Kaempchen, J. Ota, D. Burschka, Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection, in: 2010 IEEE Intelligent Vehicles Symposium, 2010, pp. 1006–1013, http://dx.doi.org/10.1109/IVS.2010.5548091.

[12] J. Moras, V. Cherfaoui, P. Bonnifait, Credibilist occupancy grids for vehicle perception in dynamic environments, in: 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 84–89, http://dx.doi.org/10.1109/ICRA.2011.5980298.

[13] C. Yu, V. Cherfaoui, P. Bonnifait, An evidential sensor model for velodyne scan grids, in: 2014 13th International Conference on Control Automation Robotics & Vision, ICARCV, 2014, pp. 583–588, http://dx.doi.org/10.1109/ICARCV.2014.7064369.

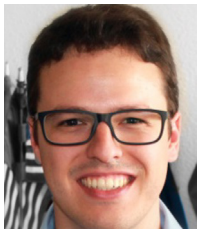
[14] J. Porębski, Customizable inverse sensor model for Bayesian and Dempster-Shafer occupancy grid frameworks, in: A. Bartoszewicz, J. Kabziński, J. Kacprzyk (Eds.), Advanced, Contemporary Control, Springer International Publishing, Cham, 2020, pp. 1225–1236.

[15] J.E. Bresenham, Algorithm for computer control of a digital plotter, IBM Syst. J. 4 (1) (1965) 25–30, http://dx.doi.org/10.1147/sj.41.0025.

[16] T. Rakotovao Andriamahefa, Integer Occupancy Grids : a probabilistic multi-sensor fusion framework for embedded perception (Ph.D. thesis), (2017GREAM010) Université Grenoble Alpes, 2017.
[17] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, OctoMap: An efficient probabilistic 3D mapping framework based on octrees, Auton. Robots (2013) http://dx.doi.org/10.1007/s10514-012-9321-0.
[18] C. Schulz, A. Zell, Sub-pixel resolution techniques for ray casting in low-resolution occupancy grid maps, in: 2019 European Conference on Mobile Robots, ECMR, 2019, pp. 1–6, http://dx.doi.org/10.1109/ECMR.2019.8870966.
[19] X. Wu, An efficient antialiasing technique, SIGGRAPH Comput. Graph. 25 (4) (1991) 143–152, http://dx.doi.org/10.1145/127719.122734.
[20] L. Roldão, R. De Charette, A. Verroust-Blondet, A statistical update of grid representations from range sensors, 2018, http://dx.doi.org/10.48550/ARXIV.1807.08483.
[21] H. Winner, S. Hakuli, F. Lotz, C. Singer, Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort, Springer International, 2015, pp. 1–1602, http://dx.doi.org/10.1007/978-3-319-12352-3.
[22] Y. Qian, X. Wang, Z. Chen, C. Wang, M. Yang, Hy-seg: A hybrid method for ground segmentation using point clouds, IEEE Trans. Intell. Vehicles (2022) 1–10, http://dx.doi.org/10.1109/TIV.2022.3187008.
[23] J. Huang, M. Demir, T. Lian, K. Fujimura, An online multi-lidar dynamic occupancy mapping method, in: 2019 IEEE Intelligent Vehicles Symposium, IV, 2019, pp. 517–522, http://dx.doi.org/10.1109/IVS.2019.8814006.
[24] T. Rakotovao, J. Mottin, D. Puschini, C. Laugier, Real-time power-efficient integration of multi-sensor occupancy grid on many-core, in: 2015 IEEE International Workshop on Advanced Robotics and Its Social Impacts, ARSO, 2015, pp. 1–6, http://dx.doi.org/10.1109/ARSO.2015.7428211.
[25] W. Xiao, B. Vallet, Y. Xiao, J. Mills, N. Paparoditis, OCCUPANCY modelling FOR MOVING object DETECTION FROM LIDAR point CLOUDS: A comparative STUDY, ISPRS Anna. Photogrammetry Remote Sens. Spatial Inform. Sci. IV-2/W4 (2017) 171–178, http://dx.doi.org/10.5194/isprs-annals-IV-2-W4-171-2017.
[26] T. Collins, J. Collins, D. Ryan, Occupancy grid mapping: An empirical evaluation, in: 2007 Mediterranean Conference on Control and Automation, 2007, pp. 1–6, http://dx.doi.org/10.1109/MED.2007.4433772.
[27] R. Grewe, M. Komar, A. Hohm, S. Lueke, H. Winner, Evaluation method and results for the accuracy of an automotive occupancy grid, in: 2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012), 2012, pp. 19–24, http://dx.doi.org/10.1109/ICVES.2012.6294297.
[28] J. Markiewicz, Developing occupancy grid with automotive simulation environment, Appl. Sci. 10 (21) (2020) http://dx.doi.org/10.3390/app10217629.
[29] S. Steyer, G. Tanzmeister, D. Wollherr, Grid-based environment estimation using evidential mapping and particle tracking, IEEE Trans. Intell. Vehicles 3 (3) (2018) 384–396, http://dx.doi.org/10.1109/TIV.2018.2843130.
[30] C.P. Diehl, E. Feicho, A. Schwambach, T. Dammeier, E. Mares, T. Bertram, Radar-based dynamic occupancy grid mapping and object detection, in: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), 2020, pp. 1–6.
[31] V. Jiménez, J. Godoy, A. Artuñedo, J. Villagra, Object-based velocity feedback for dynamic occupancy grids, in: 2022 IEEE Intelligent Vehicles Symposium, IV, 2022, pp. 1309–1314, http://dx.doi.org/10.1109/IV51971.2022.9827201.
[32] M. Schreiber, V. Belagiannis, C. Glaser, K.C.J. Dietmayer, Dynamic occupancy grid mapping with recurrent neural networks, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 6717–6724.
[33] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, J. Hays, Argoverse: 3D tracking and forecasting with rich maps, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 8740–8749, http://dx.doi.org/10.1109/CVPR.2019.00895.
[34] H. Caesar, V. Bankiti, A.H. Lang, S. Vora, V.E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, Nuscenes: A multimodal dataset for autonomous driving, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 11618–11628, http://dx.doi.org/10.1109/CVPR42600.2020.01164.
[35] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? The KITTI vision benchmark suite, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354–3361, http://dx.doi.org/10.1109/CVPR.2012.6248074.
[36] S. Hoermann, P. Henzler, M. Bach, K. Dietmayer, Object detection on dynamic occupancy grid maps using deep learning and automatic label generation, in: 2018 IEEE Intelligent Vehicles Symposium, IV, 2018, pp. 826–833, http://dx.doi.org/10.1109/IVS.2018.8500677.
[37] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, AAAI Press, 1996, pp. 226–231.
[38] A. Rosenfeld, J.L. Pfaltz, Sequential operations in digital picture processing, J. ACM 13 (4) (1966) 471–494, http://dx.doi.org/10.1145/321356.321357.
[39] X. Zhang, W. Xu, C. Dong, J.M. Dolan, Efficient L-shape fitting for vehicle detection using laser scanners, in: 2017 IEEE Intelligent Vehicles Symposium, IV, 2017, pp. 54–59, http://dx.doi.org/10.1109/IVS.2017.7995698.
[40] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics, MIT Press, Cambridge, Mass, 2005.
[41] Y. Yuan, H. Kuang, S. Schwertfeger, Fast Gaussian process occupancy maps, in: 2018 15th International Conference on Control, Automation, Robotics and Vision, ICARCV, 2018, pp. 1502–1507, http://dx.doi.org/10.1109/ICARCV.2018.8581356.
[42] M. Saval-Calvo, L. Medina-Valdés, J.M. Castillo-Secilla, S. Cuenca-Asensi, A. Martínez-Álvarez, J. Villagrá, A review of the Bayesian occupancy filter, Sensors 17 (2) (2017) http://dx.doi.org/10.3390/s17020344, URL https://www.mdpi.com/1424-8220/17/2/344.
[43] D. Nuss, S. Reuter, M. Thom, T. Yuan, G. Krehl, M. Maile, A. Gern, K. Dietmayer, A random finite set approach for dynamic occupancy grid maps with real-time application, Int. J. Robot. Res. 37 (8) (2018) 841–866, http://dx.doi.org/10.1177/0278364918775523.
[44] A.P. Dempster, Upper and lower probabilities induced by a multivalued mapping, in: Classic Works of the Dempster-Shafer Theory of Belief Functions, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 57–72.
[45] G. Shafer, A Mathematical Theory of Evidence, Princeton University Press, 1976.
[46] J.D. Adarve, M. Perrollaz, A. Makris, C. Laugier, Computing occupancy grids from multiple sensors using linear opinion pools, in: 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 4074–4079, http://dx.doi.org/10.1109/ICRA.2012.6224976.
[47] K. Pathak, A. Birk, J. Poppinga, S. Schwertfeger, 3D forward sensor modeling and application to occupancy grid based sensor fusion, in: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2007, pp. 2059–2064, http://dx.doi.org/10.1109/IROS.2007.4399406.
[48] V. Jiménez, J. Godoy, A. Artuñedo, J. Villagra, Ground segmentation algorithm for sloped terrain and sparse LiDAR point cloud, IEEE Access 9 (2021) 132914–132927, http://dx.doi.org/10.1109/ACCESS.2021.3115664.

Víctor Jiménez was born in Madrid, Spain, in 1993. He received the degree in industrial electronics and automation engineering from the Universidad Carlos III de Madrid in 2017, and the master degree in automation and robotics from the Universidad Politécnica de Madrid in 2019. During 2018 and 2019 he obtained a research grant focused on mobile robots and fuzzy logic and competed in the SEAT Autonomous Driving Challenge 2018 as a member of AUTOPIA Program's team. In 2020 he joined this program as a Pre-Doctoral researcher for the Centre for Automation and Robotics (CSIC-UPM). His research interests include autonomous driving, perception of the environment, localization and information integrity.

Jorge Godoy was born in Maracay, Venezuela, in 1986. He received the degree in electronics engineering from Universidad Simón Bolívar in 2008, and the M.E. and Ph.D. degrees in automation and robotics from the Universidad Politécnica de Madrid in 2011 and 2013, respectively. From 2013 to 2017, he was the Technical Coordinator of the AUTOPIA Program funded by research contracts from National and European research projects. His research interests include intelligent transportation systems, autonomous driving, path planning, and embedded AI-based control for autonomous vehicles. In 2009, he was granted with a Pre-Doctoral JAE Fellowship from CSIC for researching on autonomous vehicles at the Centre of Automation and Robotics (UPM-CSIC). In November 2017, he was granted with a Juan de la Cierva Fellowship for Post-Doctoral Research at the Universidad Politécnica de Madrid.



Antonio Artuñedo received the B.Sc. degree in electrical engineering from the Universidad de Castilla–La Mancha, Spain, in 2011, the M.Sc. degree in industrial engineering from the Universidad Carlos III de Madrid in 2014, and the Ph.D. degree in automation and robotics from the Technical University of Madrid (UPM), Spain, in 2019, with the AUTOPIA Program. During his Pre-Doctoral period, he made a research stay at the Integrated Vehicle Safety Group at TNO, The Netherlands, in 2017. He is currently a Post-Doctoral Researcher with the Centre for Automation and Robotics (CSIC-UPM), AUTOPIA Group, Madrid, Spain. He has been working on both national and European research projects in the scope of autonomous vehicles. He has published and peer-reviewed multiple journals and conference papers focused in this field. His research interests include system modeling and simulation, intelligent control, motion planning, and decision-making systems. His thesis won the prize to the best Ph.D. on Intelligent Transportation Systems 2020 by the Spanish Chapter of the IEEE-ITS Society.

Jorge Villagra graduated in industrial engineering from the Universidad Politécnica of Madrid in 2002. He received the Ph.D. degree in real-time computer science, robotics and automatic control from the École des Mines de Paris, France, in 2006. From 2007 to 2009, he was a Visiting Professor with the University Carlos III of Madrid, Spain. From August 2013 to August 2016, he led the Department of ADAS and Highly Automated Driving Systems at Ixion Industry and Aerospace SL, where he also coordinated all the activities in the EU Research and Development funding programmes. He has been leading AUTOPIA Program at CSIC, since October 2016. He has developed his research activity in six different entities with a very intense activity in project setup and management, through over 40 international and national R&D projects, where he is or has been IP of 17 of these projects. He has published over 100 articles in international journals and in international conferences on autonomous driving, intelligent transportation systems, model free control, and probabilistic approaches for embedded components in autonomous vehicles.