Multi-Agent Cooperative Camera-Based Evidential Occupancy Grid Generation

Antoine Caillot, Safa Ouerghi, Pascal Vasseur, Yohan Dupuis, Rémi Boutteau

Multi-Agent Cooperative Camera-Based Evidential Occupancy Grid Generation

Antoine Caillot[1], Safa Ouerghi[2], Pascal Vasseur [3], Yohan Dupuis [4], Rémi Boutteau [5]

***Abstract*— About a decade ago the idea of cooperation has been introduced to self-driving with the aim to enhance safety in dangerous places such as intersections. Infrastructure-based cooperative systems emerged very recently bringing a new point of view of the scene and more computation power. In this paper, we want to go beyond the framework presented in the vehicle-to-infrastructure (V2I) cooperation by including the vehicle's point of view in the perception of the environment. To keep the cost low, we decided to use only two-dimensional bounding boxes, thus depriving ourselves of depth information that contrasts with state-of-the-art methods. With this in-the-scene point-of-view, we propose a new framework to generate a cooperative evidential occupancy grid based on the Dempster-Shafer Theory and which employs a Monte Carlo framework to incorporate position noise in our algorithm. We also provide a new cooperative dataset generator based on the CARLA simulator. Finally, we provide an extended review of our new cooperative occupancy grid map generation method which improves the state-of-the-art techniques.**

I. Introduction

Nowadays, navigation and traffic management in intersections and roundabouts is still a complex task [1], [2]. The perception of the environment aims at providing an accurate and robust estimation of the state of other road users in order to make safe navigation decisions both for humans and autonomous vehicles.

To perceive the environment, we decided to use data from the infrastructure but also data from the in-the-scene users' points of view. The inclusion of vehicle data in the scene enhances the intersection's safety while allowing reducing the number of sensors on the edge of the road and, consequently, the cost of the infrastructure. Currently, to tackle this challenge, many systems propose the implementation of infrastructure allowing the observation of a scene [3], [4], [5], [6] but none of these solutions take into account the vehicles' point of view.

In this paper, we use only two-dimensional bounding box information from cameras to keep the cost of our cooperative system low and to reduce the required data rate and thus the communication burden of the cooperative systems. We noted that the state of the art systems uses range sensors such as LiDAR [3], [4], [5] or RADAR[6], which are more expensive than cameras. Bounding box information extraction is out of the scope of this paper and is therefore considered given either by an algorithm such as YOLO [7] or by an ADAS such as the Mobileye's solution used in [8].

In this work, we take advantage of the in-scene point of view from the vehicles in order to enrich a common map of the scene in the form of an evidential occupancy grid [9] and to build confidence while decreasing the costs. Usually, classical occupancy grids as defined in [10] are used [11], [12]. However, the latter use joint probabilities based method to merge the grids which do not take into account the unobserved cells.

We also present a new dataset generator with available source code[1] based on the CARLA simulator [13] allowing to have several instrumented agents in the scene.

In the remainder of this section, we review publications related to our work. Then, we start in section II by presenting the global architecture of the framework made available. The computation methods used are presented in the section III. Finally, we present our results in section IV, in which we present the cooperative vehicle-infrastructure dataset that has been generated and used to test our system, before bringing a conclusion in section V.

A. Related work

Cooperative systems are more and more present in the state of the art of perception in the autonomous driving context [14]. Today, two main categories of cooperative systems exist; vehicle-to-vehicle (V2V) systems [15] and vehicle-to-infrastructure (V2I) systems [3], [4], [5], [6]. Other systems also exist and are generally grouped under the name of vehicles-to-everything (V2X). During the last years, the V2I paradigm has been significantly developed with the aim of obtaining an omniscient point of view. However, with the development of the V2I paradigm, more interest has been devoted to the data at the infrastructure's level while the data from the vehicles to the infrastructure has been left aside which could have been used to refine the detections made by the infrastructure. This motivated us to take into consideration these data to propose a framework that is able to exploit multiple points of view.

In order to make a cooperative system, the question of the data sharing level arises. Several strategies have been investigated in the state of the art [16]. Either raw data,

[1] Antoine Caillot and [2] Safa Ouerghi are with Normandie Univ, UNIROUEN, ESIGELEC, IRSEEM, 76000 Rouen, France. caillot@esigelec.fr ouerghi@esigelec.fr

[3]Pascal Vasseur is with Laboratoire MIS, Université de Picardie Jules Verne, UFR des Sciences, Département Informatique, 80000 Amiens, France. pascal.vasseur@u-picardie.fr

[4]Yohan Dupuis is with LINEACT CESI, Paris La Défense, Paris, France. ydupuis@cesi.fr

[5]Rémi Boutteau is with Normandie Univ, UNIROUEN, UNILEHAVRE, INSA Rouen, LITIS, 76000 Rouen, France. remi.boutteau@univ-rouen.fr

[1]https://github.com/caillotantoine/carla-V2X-dataset-generator

directly provided by sensors can be shared [15], [4] or data in the form of labels where the agents do most of the processing locally before sharing them as in [5], [6], [3]. Sharing raw data has the advantage of densifying the measurements and thus having an impact on the accuracy of the detection whereas, for label sharing, the majority of the detections are done locally on each of the agents which reduce the impact of the cooperation on the system performances. However, raw data sharing comes at the cost of higher network pressure and bandwidth requirements. In our paper, we have decided to work with pre-processed data where only bounding boxes of other detected vehicles in the scene are shared without the use of the whole image. This relies on performing an early detection at the sensor's level. This can be generated by a detection algorithm in order to satisfy the requirements of a cooperative environment in terms of bandwidth as presented in [3], [17].

On the other hand, occupancy grids have been extensively used for mapping but generally using either multiple range sensors [10], [18] or with cameras but from a single point of view [19]. This motivated us to use vision-based occupancy grid mapping in a cooperative context using bounding boxes detection. We project bounding boxes corresponding to the detected vehicles onto the ground using back-projection as presented in [20]. In the latter, the back-projected images on the ground are merged by using the union method. However, our merging method is based on Dempster Shaffer Theory [21], [22] as presented in [9]. We have also used the average occupancy fusion method based on the intersection between detections to assess the efficiency of each fusion method in our vision-based cooperative context.

II. System architecture

In this section, we present our cooperative V2I bi-directional framework for creating occupancy grids from camera data based on ROS [23]. Our framework is made of two types of elements: agents and a Road Side Unit (RSU).

A. Agents

The agents can be intelligent roadside sensors or connected vehicles and can be of an arbitrary amount in the scene. They are equipped with an image sensor and a system to identify vehicles in their field of view that extracts bounding boxes. Every agent publishes their messages on a global topic containing every bounding box of every agent and will be read by the RSU as illustrated in Fig. 1.

In our work, we consider that the extraction of bounding boxes is derived from off-the-shelf solutions and is therefore not a topic covered here. We consider that timestamps are generated at the time of shooting from a GPS clock and thus the sensors are roughly synchronized. Therefore, we used the synthetic data from the ground truth to which we added random Gaussian noise.

B. Road Side Unit

The Road Side Unit (RSU) is the central element of our framework. It aggregates every agents' messages and compiles them to form an occupancy grid of the mobile objects in the scene. It is composed of different blocks as illustrated in Fig. 2. We explain the role of each of the blocks in the following subsections, the mathematical details will be given later.

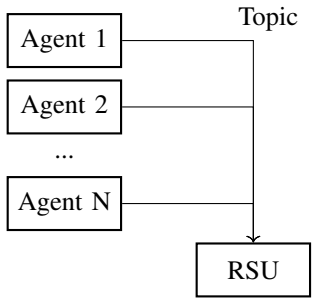


Fig. 1: Macro organisation of the agents and the RSU.

Monte Carlo Uncertainties Sampler:
This block takes the bounding boxes and the sensor pose from which they are extracted and models the uncertainties by applying noise to the parameters on N samples created from each original measurement, with N the larger possible.

Back Projector:
This block uses the bounding box parameters for each of the N samples, finds the 4 corners of the bounding box, and projects them on the ground by ray tracing.

Rasterizer:
This block takes the 4 projected points on the ground of each bounding box and N samples and rasterizes them on the N occupation grid.

Sample Merger:
This block merges the N occupancy grids forming a local occupancy grid (LOG) for a sensor.

Stack:
This block keeps the LOGs until the next block empties it.

Basic Belief Assignment (BBA):
This block assigns, from the observations, the masses to the different classes used with the DST method to each cell of the occupation grids. This block appears only for the DST fusion. In other cases, it is bypassed.

Combiner:
This block merges the occupancy grids of the stack either based on the DST and the BBA values or directly with the probabilities contained in the occupancy grids.

III. Methods

In this section, we provide more details about the functioning of the blocks composing the RSU. We start with the basic principle of our system: the Back Projection, which allows us to obtain the footprints of the bounding boxes. We also present the methods allowing us to generate the local occupancy grids (LOG). Finally, we present the details of the methods for merging the LOG. As we go along, we also give details on the use of Monte Carlo methods.

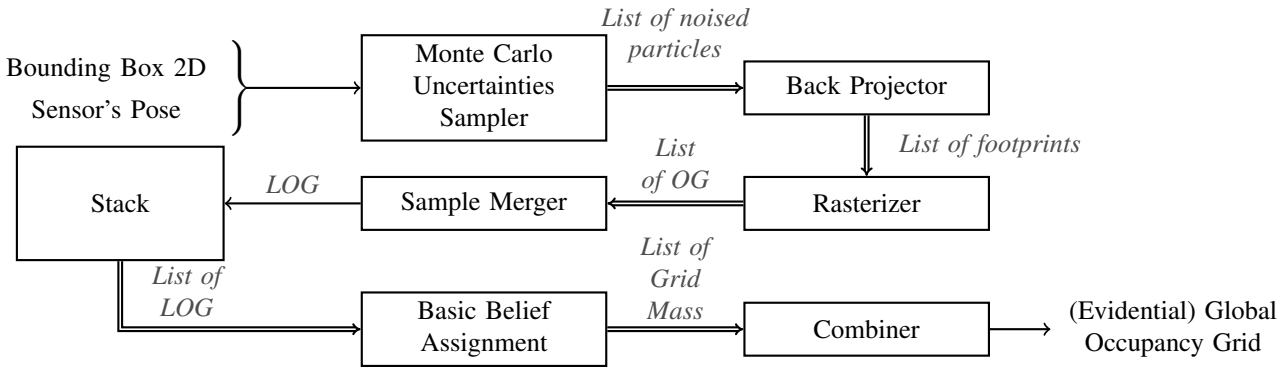


Fig. 2: Illustration of bounding box processing and fusion framework in the RSU.

A. Back Projection

The 2D bounding boxes are given in the camera's image frame in pixel coordinates. Therefore, we need to back-project them into the 3D world space. The projection of a point $\boldsymbol{P_W} = (X_W, Y_W, Z_W)^\intercal$ expressed in the world's reference frame into image point $\boldsymbol{p} = (u, v)^\intercal$, where u and v are pixel coordinates, involves two steps: The first one considers transforming the point into the camera's reference frame, given by the position of the optical center of the camera $\boldsymbol{t_c}$ in the world frame and the rotation $\boldsymbol{R_c}$ from the world back to the camera frame.

The second step consists in transforming the 3D point into the 2D image plane which requires the intrinsic parameters matrix $\boldsymbol{K}$ of the camera. The pinhole camera model can be expressed using homogeneous coordinates as:

$$s\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & u_c \\ 0 & f & v_c \\ 0 & 0 & 1 \end{pmatrix} [\boldsymbol{R_c}|\boldsymbol{t_c}] \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix} = \boldsymbol{K}\boldsymbol{T}_{\boldsymbol{C}}^{\boldsymbol{W}} \begin{pmatrix} \boldsymbol{P_W} \\ 1 \end{pmatrix} \tag{1}$$

where s is an unknown scale factor, f the focal length, the parameters (u_c, v_c) represent the camera's principal point coordinates in pixels and $\boldsymbol{T}_{\boldsymbol{C}}^{\boldsymbol{W}}$ the extrinsic transformation from the world's reference frame to the camera's reference frame.

It is therefore possible to compute the inverse transformation to estimate the 3D position of a point in 3D space up to a scale factor s.

In fact, for each bounding box, we have a set of five points including the camera's pose and the four corners of the bounding box and we need to compute their projection onto the ground. We start, first, by finding the corresponding 3D points up to a scale that belong to rays passing through both the optical center and the 2D points in front of the camera using the inverse transformation of the pinhole model. Second, we compute the projection of these points onto the ground and we use the Plücker coordinates which is a convenient representation for directed lines in affine 3D space, in our case the rays from the center to the corners.

The ground is assumed to be a plane $\boldsymbol{\pi}$ that can be represented in 3-space as in (2),

$$\pi_1 X + \pi_2 Y + \pi_3 Z + \pi_4 = 0 \tag{2}$$

where the vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3, \pi_4)^\intercal$ refers to its homogeneous representation [24].

Using Plücker coordinates, a ray is represented using a 4x4 Plücker matrix defined according to Equation (3),

$$\boldsymbol{L} = \boldsymbol{A}\boldsymbol{B}^\intercal - \boldsymbol{B}\boldsymbol{A}^\intercal \tag{3}$$

where $\boldsymbol{A} = [x_1, y_1, z_1, w_1]^\intercal$ and $\boldsymbol{B} = [x_2, y_2, z_2, w_2]^\intercal$ are two points in the 3D space expressed in homogeneous coordinates.

Given the ground plane and the set of four rays corresponding to the four corners of each bounding box, the intersection between each ray L and the ground plane π can be found according to equation (4),

$$\boldsymbol{X_{gnd}} = \boldsymbol{L}\boldsymbol{\pi} \tag{4}$$

where $\boldsymbol{X_{gnd}}$ is a four-dimensional vector in homogeneous coordinates. Thus, we obtain for each bounding box a set of four projected points on the ground plane forming a polygon that represents a footprint on the ground of the detected object at a given point of view.

B. Local Occupancy Grids generation

We consider a local occupancy grid (LOG) as an occupancy grid containing the information provided by a single sensor. Let $\mathcal{M}$ be the occupancy grid map over a region of interest divided into square cells $\mathcal{M}_{x,y}$ where $\langle x, y\rangle$ correspond to the position of the cell $\mathcal{M}_{x,y}$ within $\mathcal{M}$ as defined in [25]. Thus, the problem addressed is the determination of the probability of occupancy of each grid cell given the measurements. The assigned values to $\mathcal{M}_{x,y}$ are $\{m_{x,y} \in \mathbb{Z} | -1 \leq m_{x,y} \leq 100\}$ where -1 denotes a cell of unknown occupation, 0 denotes a free cell and 100 an occupied cell as given in ROS documentation.

1) From Ground coordinates to Occupancy Grid (OG) coordinates : Let δ be the length of the square cell in meters, (O_x, O_y) correspond to the position of the origin in the occupancy grid. The position of a footprint is obtained in cell coordinates from metric coordinates according to (5),

$$\mathbf{x_{grid}} = \left\lceil \begin{pmatrix} 1/\delta & 0 & 0 & O_x \\ 0 & 1/\delta & 0 & O_y \end{pmatrix} \boldsymbol{X_{gnd}} \right\rceil \tag{5}$$

where $\mathbf{x_{grid}}$ is a 2-vector and $\boldsymbol{X_{gnd}}$ a 4-vector.

2) Rasterisation: As explained earlier, detected objects are projected on the ground plane as polygons forming footprints which now need to be rasterized in order to fill the corresponding cells of the occupancy grid. To do so, we considered the occupancy grid as an image and used the fillpoly function from the OpenCV library [26]. We used the 8-connected line parameter, also known as the Moore neighborhood, for the polygon edges. At this point, the occupancy of a cell is not proportional to the area occupied by the polygon but is assigned in a binary manner. Therefore, the cells in the camera frustum take the value 0, those in the bounding box frustum takes the value 100 and all the others of unknown state take the value -1.

3) Modeling uncertainties: The position estimation of the camera in the scene is subject to noise as well as the bounding box position and dimension determination on the image. To model these uncertainties, we created N samples from each original measurement, with N the larger possible. Then, we applied noise to the pose estimation and bounding box estimation parameters for each of the sample. The noise follows a Gaussian distribution with parameters μ the original measurement and σ the standard deviation presented in [27] and in [28]. Each of the N samples is projected on N sample grids and then merged by averaging the cells.

C. Local Occupancy Grids Merging

Since each sensor provides a local occupancy grid, these latter have to be merged in order to create a global one. The LOG is already created with respect to a global frame reference and can therefore be directly merged without frame transformations. In fact, two main paradigms have been investigated in the state of the art to perform the merging namely the probabilistic approach and the Evidential approach.

Let $\mathcal{M}$ be a global occupation grid. Let's consider a local occupation grid $\mathcal{M}_l$ and $\mathcal{M}^i_{x,y}$ a given cell of $\mathcal{M}_l$ where $\langle x, y\rangle$ refer to the location of the cell and i to the index of the agent $1 \leq i \leq N_A$ with N_A the number of the available agents.

1) Probabilistic merging method: The probabilistic method is based on making the intersection between the probabilities of occupation of $\mathcal{M}^i_{x,y}, i \in \{1, 2, ..., N\}$ to erode detections to get the final shape at the intersection of the point of views. We consider the probability of each cell as independent, thus $P(\mathcal{M}^i_{x,y} \cap \mathcal{M}^j_{x,y}) = P(\mathcal{M}^i_{x,y}) \times P(\mathcal{M}^j_{x,y}),\ i \neq j \in \{1, 2, ..., N\}$. Therefore, we propose two methods that perform a product between cells to determine its occupancy probability as given in (6) [29]. The former, named **inter1**, considers the cells having an unknown state (-1) as having a probability of 0.5 before performing the product of the cells. The latter, named **inter2**, ignores the cells having an unknown value (-1) in the product. For both of them, values between 0 and 100 are divided by 100.

$$\mathcal{M}_{x,y} = \prod_{i=1}^{N} \mathcal{M}^i_{x,y} \tag{6}$$

Algorithm 1 Basic Belief Assignment

Require: $\mathcal{C} \in \mathcal{M}$
  $m(\varnothing), m(O), m(F), m(\Omega) \leftarrow 0$
  if C = -1 **then**
    $m(\Omega) \leftarrow 1$
  else
    if C is from an infrastructure sensor **then**
      $m(F) \leftarrow 1.0 - \frac{\mathcal{C}}{100}$
      $m(\Omega) \leftarrow \frac{\mathcal{C}}{100}$
    else
      $m(O) \leftarrow \frac{\mathcal{C}}{100}$
      $m(\Omega) \leftarrow 1.0 - \frac{\mathcal{C}}{100}$
    end if
  end if

2) Evidential merging method: We can also use the Dempster-Shafer Theory (DST) [21] to merge the LOGs as attempted in [9]. We give two possible statuses forming the universe given by the equation (7). O describes the status of an occupied cell and F that of a free cell. The elements of the power set 2^Ω represent the status of the cell and are formed by (8).

$$\Omega = \{O, F\} \tag{7}$$

$$2^\Omega = \{\varnothing, \{O\}, \{F\}, \Omega\} \tag{8}$$

$$\sum_{X \in 2^\Omega} m(X) = 1 \tag{9}$$

Thus, Ω represents an unknown state. For example, if a cell has the value -1, we know that it has not been observed and, consequently, it can be either free or occupied without being able to choose one state more than the other. Each state has a mass m corresponding to the probability of the cell being in that state respecting the distribution of the equation (9). Since a cell is either occupied or free, $m(\varnothing) = 0$.

The association of a mass with a 2^Ω status is performed by a function named basic belief assignment (BBA). Our basic belief assignment function is given by the Algorithm 1. We distinguish two distinct points of view, that of a vehicle with a grazing view on the scene giving an estimation of the occupation and that of the infrastructure with a quasi bird eye view giving an estimation of the absence of occupation of a cell. Once each cell of each LOG has had its masses assigned, it is possible to merge them one by one with Dempster's rule of combination given in the equation (10) to merge two cells where $X \in 2^\Omega$ is defined by equation (11) with $K = \sum_{Y \cap Z = \varnothing} m_1(Y) m_2(Z)$.

$$m_f(X) = m_1(X) \oplus m_2(X) \tag{10}$$

$$m_f(X) = \frac{1}{1-K} \sum_{Y \cap Z = X \neq \varnothing} m_1(Y) m_2(Z) \tag{11}$$

$$m_{out}(X) = \bigoplus_{i=0}^{N} m_i(X) \tag{12}$$

Dempster's rule of combination being commutative and associative, it is, therefore, possible to combine N masses

as expressed in equation (12). Thus, we combine the masses associated with cells of the same coordinate in each layer.

We propose two methods to associate the values to the cells from the masses of the final grid. The former, named **dst1**, directly assigns to the cell the mass of the set O while the latter one, named **dst2**, assigns to the cell the value of O if the mass of the set F is bigger than the mass of the set O and Ω, the value of -1 if Ω is bigger than O and F and the value of 100 in every other cases.

IV. Results

A. Carla dataset

To the best of our knowledge, we have not identified any dataset that delivers a vehicle-infrastructure cooperative experimental framework. Therefore, we created a dataset generator based on the CARLA simulator [13] allowing the generation of datasets with one or more viewpoints from infrastructure and vehicles. For the works of this paper, we generated a dataset with 4 agents: 3 connected vehicles and an infrastructure. The vehicles pass through the roundabout and are in the field of view of the infrastructure. Also, some vehicles will enter the field of view of one or more other vehicles and have their fields of view overlapping as shown in Fig. 3. Each agent can have different sensors:

- $1\times$ RGB camera ($90°$ fov, 1384×1032 pixels)
- $1\times$ Depth camera ($90°$, 1384×1032)
- $1\times$ Semantic segmentation camera ($90°$, 1384×1032)
- $1\times$ LiDAR (32 layers, $40°$ vertical fov)

For the different agents, the rigid transformation between each onboard sensor and the attached reference frame is the same. For the infrastructure, the sensors are positioned at $13m$ altitude at the center of the roundabout (located at the scene's center) and with a pitch of $-20°$. For the vehicles, the sensors are located at $1.9m$ above the chassis. In addition to the raw data, the vehicle's state is stored in a JSON file for each frame. This latter contains the sensor's transformation matrix with respect to the world's reference frame as well as the vehicle's transformation matrix, linear velocity, angular velocity, acceleration, forward vector, and 3D bounding box.

In order to generate the ground truth, we used the JSON files. We retrieve the 4 points forming the bottom plan of the bounding box and place them in the scene with the given transformation matrix to express their coordinates in the world reference frame. We get a perfect polygon forming the footprint of the vehicle which is then rasterized on the grid. Fig. 4c illustrates at the frame 155 of the dataset an example of the map saved where black color corresponds to a value of 100 and white color corresponds to a value of 0. Alongside, in Fig. 4, the outputs of each above-mentioned algorithm are featured.

B. Qualitative Evaluation

We evaluate our algorithm within 280 frames in which we can distinguish 6 sequences as described below:

Seq 0: This sequence corresponds to the best coverage from the cars. Each car sees at least one other vehicle.

Fig. 3: Synchronous video frames from each camera of our multi-agent dataset made with CARLA.

Seq 1: The infrastructure coverage is maximal: each vehicle is visible from the infrastructure's point of view.

Seq 2: The coverage is maximal from every agent. Each car is seen by at least one vehicle and by the infrastructure.

Seq 3: This sequence gives an example of partial coverage where both vehicle and infrastructure operate but not every car is seen by the infrastructure.

Seq 4: This sequence features monomodal detection. This means that cars are detected either by the infrastructure or by other vehicles but not both.

Seq 5: This sequence features single detection. The detected cars are detected by only one agent and thus is not a cooperative situation.

C. Quantitative Evaluation

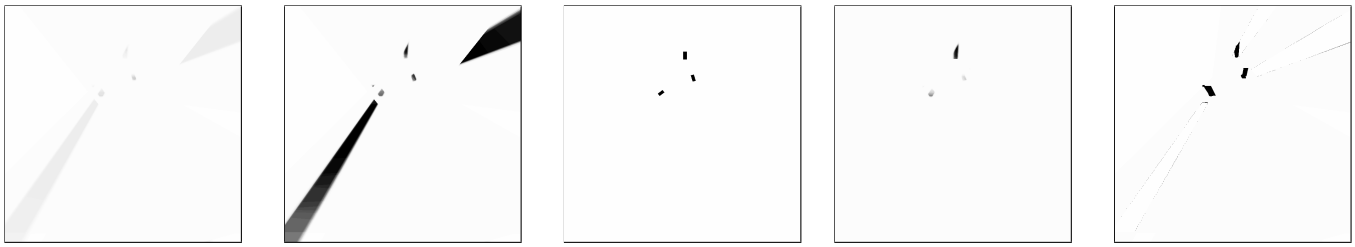
We based our quantitative evaluation on Intersection over Union (IoU) and F1-score which are two common metrics for occupancy grid evaluation. Both of them are based on the number of True Positive (TP), False Positive (FP), and False Negative (FN). To define if a cell is positive, we compare its value to a threshold. IoU is defined as in equation (13) and F1-score is defined in (14).

$$IoU = \frac{TP}{TP + FP + FN} \tag{13}$$

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{14}$$

Table I gives an overview of the IoU and the F1-score for each sequence and each algorithm. These results were given with a noise applied following a normal distribution with a standard deviation of $\sigma = 0.00243m$ on the lateral and longitudinal position and of $\sigma = 0.0518m$ on the altitude as we can find in [28]. For the rotations, the noise follows a normal distribution with a standard deviation of $\sigma = 0.1°$ on all axes as we can find in [27]. The bounding boxes have a normal distribution noise with a standard deviation of 5 pixels applied on each edge of the bounding box. The threshold was set at 0.5 but requires in-depth research to determine its impact on the results.

We note that **inter1** obtains a global result 0%, either on IoU and F1-score, showing that the basic probability-based



(a) Using inter1. (b) Using inter2. (c) Ground truth. (d) Using dst1. (e) Using dst2.

Fig. 4: Occupancy grid map for different methods (frame 155).

TABLE I: Example of the evolution of the IoU and F1 scores with a threshold of detection of 0.50 (normalized) with 3 vehicles transiting in a roundabout.

Algorithm	**Metric**	**Seq 0**	**Seq 1**	**Seq 2**	**Seq 3**	**Seq 4**	**Seq 5**	**Total**
inter1	IoU	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
inter1	F1 Score	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
inter2	IoU	0.052524	0.005595	0.004103	0.022008	0.214580	0.019923	0.013490
inter2	F1 Score	0.099806	0.011128	0.008172	0.043068	0.353341	0.039067	0.026620
dst1	IoU	0.055993	0.305245	0.287551	0.233977	0.174009	0.162757	0.223513
dst1	F1 Score	0.106048	0.467721	0.446663	0.379224	0.296435	0.279951	0.365362
dst2	IoU	0.175572	0.217005	0.213857	0.175982	0.172789	0.142080	0.188038
dst2	F1 Score	0.298700	0.356621	0.352359	0.299293	0.294663	0.248810	0.316551

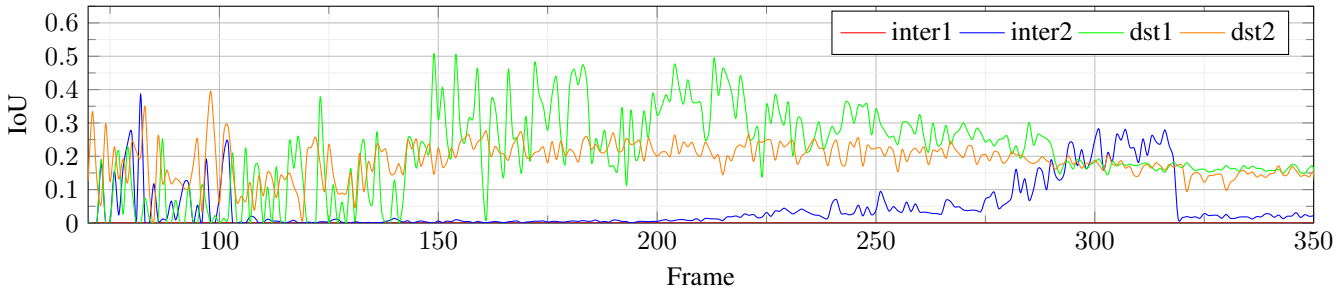


Fig. 5: Example of the evolution of the IoU with a threshold of detection of 0.50 (normalized) with 3 vehicles transiting in a roundabout.

occupancy grid fusion method as presented in [29], [30] is not usable in this situation. However, **inter2** offers slightly better results with a maximum IoU of 21.45% in sequence 4 as shown in Fig. 5 but a global IoU of 1.35%.

This behavior is explainable by the fact that **inter2** is a modified version of **inter1**. Indeed, although these two algorithms follow the same merging rule, **inter2** excludes the unobserved cells during the merging. This avoids that individual detections be removed from the final map because of the successive multiplications by 0.5 that **inter1** would perform in individual detections. However, this approach shows its limits since beams appear when projecting the frustums as shown in Fig. 4b, thus increasing the number of false negatives. In sequence 4, the position of the vehicles offers viewpoints to the agents allowing them to reduce the beams due to frustum and thus to reduce the number of false positives. Nevertheless, this situation disappears when moving to sequence 5, and the false-positive rate increases dramatically.

Regarding the results obtained with our method, the algorithms based on the Dempster-Shafer theory (DST) offers much better results than the standard method cited in the previous paragraph. The fusion algorithm **dst1** offers a maximum IoU of 30.52% in the sequence 1 and a global IoU of 22.35% while **dst2** offers a global IoU of 18.8%.

The fusion algorithms **dst1** and **dst2**, based on the DST, show much better results since the DST allows the management of cells with an unknown state. We can consider that the distribution of masses can give a hint on the confidence of a measurement. Thus, when a cell is not observed, the confidence associated with this measurement is null. The consequence of this behavior is the elimination of the beams as observed in the methods **inter1** and **inter2** and thus the reduction of the false positives. We notice a more erratic behavior on Fig. 5 until frame 140. This is due to the fact that a vehicle is too far away to be detected which corresponds to a false negative. Moreover, the vehicles are distant from each other, which has the consequence of amplifying the observation errors. As for the last sequences, the vehicles move away from each other and leave the field of view of the infrastructure, thus increasing the measurement errors. Therefore, we can conclude that the results are given at the

beginning and the end of the traffic circle transit as given in Fig. 5 are due to measurement errors. To conclude, we note that **dst1** and **dst2** do not seem to be affected by the arrangement of the vehicles as is **inter2** and therefore **dst1** and **dst2** are more robust than the state of the art methods while providing better results.

V. Conclusion and Future work

In this paper, we presented a new approach for cooperative perception in order to create an evidential occupancy grid map. We used the vehicle point of view in addition to the infrastructure point of view in order to build confidence at low cost.

In addition, we propose a method for cooperative generation of evidential occupancy grid using only the two-dimensional bounding boxes given by an image sensor as well as the position of that sensor with the aim of keeping the system's cost low as well as reducing the load on the communication system.

Finally, we propose a study on different data fusion methods based either on a Bayesian approach or on a Dempster-Shafer based approach on which we observe much better results. We have validated our results on a cooperative dataset that we have created from the CARLA simulator that we provide and to which we have added measurement noise.

In future work, we will explore decision taking approaches for the generation of the occupancy grid map more advanced than thresholding as well as the impact of the number of agents in the scene.

References

[1] L. C. Bento, R. Parafita, and U. Nunes, "Intelligent traffic management at intersections supported by V2V and V2I communications," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, Sept. 2012, pp. 1495–1502, iSSN: 2153-0017.

[2] M. Khayatian, M. Mehrabian, E. Andert, R. Dedinsky, S. Choudhary, Y. Lou, and A. Shirvastava, "A Survey on Intersection Management of Connected Autonomous Vehicles," *ACM Transactions on Cyber-Physical Systems*, vol. 4, no. 4, pp. 48:1–48:27, Aug. 2020. [Online]. Available: https://doi.org/10.1145/3407903

[3] B. Lv, H. Xu, J. Wu, Y. Tian, Y. Zhang, Y. Zheng, C. Yuan, and S. Tian, "Lidar-enhanced connected infrastructures sensing and broadcasting high-resolution traffic information serving smart cities," *IEEE Access*, vol. 7, pp. 79 895–79 907, 2019.

[4] Z. Li, T. Yu, R. Fukatsu, G. K. Tran, and K. Sakaguchi, "Proof-of-concept of a sdn based mmwave v2x network for safe automated driving," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[5] Bosch, "Conduite automatisée : comment les voitures et les infrastructures communiquent en milieu urbain," Website, July 2020, accessed 2020-07-29. [Online]. Available: https://www.bosch.fr/actualites/2020/conduite%2Dautomatisee%2Dcomment%2Dles%2Dvoitures%2Det%2Dles%2Dinfrastructures%2Dcommuniquent%2Den%2Dmilieu%2Durbain/

[6] D. G. Annkathrin Krämmer*, Christoph Schöller* and A. Knoll, "Providentia - a large scale sensing system for the assistance of autonomous vehicles," in *Robotics: Science and Systems (RSS), Workshop on Scene and Situation Understanding for Autonomous Driving*, 2019. [Online]. Available: https://sites.google.com/view/uad2019/accepted-posters

[7] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[8] M. Baek, D. Jeong, D. Choi, and S. Lee, "Vehicle Trajectory Prediction and Collision Warning via Fusion of Multisensors and Wireless Vehicular Communications," *Sensors*, vol. 20, no. 1, p. 288, Jan. 2020, number: 1 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/20/1/288

[9] F. Camarda, F. Davoine, and V. Cherfaoui, "Fusion of evidential occupancy grids for cooperative perception," in *2018 13th Annual Conference on System of Systems Engineering (SoSE)*, June 2018, pp. 284–290.

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

[11] 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.

[12] S. Steyer, G. Tanzmeister, and D. Wollherr, "Grid-based environment estimation using evidential mapping and particle tracking," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 384–396, 2018.

[13] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[14] A. Caillot, S. Ouerghi, P. Vasseur, R. Boutteau, and Y. Dupuis, "Survey on cooperative perception in an automotive context," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–20, 2022.

[15] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 514–524.

[16] Z. Zhang, S. Wang, Y. Hong, L. Zhou, and Q. Hao, "Distributed dynamic map fusion via federated learning for intelligent networked vehicles," *arXiv preprint arXiv:2103.03786*, 2021.

[17] S.-W. Kim, Z. J. Chong, B. Qin, X. Shen, Z. Cheng, W. Liu, and M. H. Ang, "Cooperative perception for autonomous vehicle control on the road: Motivation and experimental results," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 5059–5066.

[18] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings. 1985 IEEE international conference on robotics and automation*, vol. 2. IEEE, 1985, pp. 116–121.

[19] S. Richter, Y. Wang, J. Beck, S. Wirges, and C. Stiller, "Semantic evidential grid mapping using monocular and stereo cameras," *Sensors*, vol. 21, no. 10, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/10/3380

[20] S.-W. Kim, B. Qin, Z. J. Chong, X. Shen, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "Multivehicle cooperative driving using cooperative perception: Design and experimental validation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 663–680, 2014.

[21] K. Sentz, S. Ferson, *et al.*, *Combination of evidence in Dempster-Shafer theory*. Sandia National Laboratories Albuquerque, 2002, vol. 4015. [Online]. Available: https://www.osti.gov/servlets/purl/800792

[22] G. Shafer, "Dempster-shafer theory," *Encyclopedia of artificial intelligence*, vol. 1, pp. 330–331, 1992.

[23] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, *et al.*, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[24] R. Hartley and A. Zisserman, "Multiple view geometry in computer vision 2nd ed., 4th print," 2006.

[25] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous robots*, vol. 15, no. 2, pp. 111–127, 2003.

[26] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[27] L. Lecrosnier, R. Boutteau, P. Vasseur, X. Savatier, and F. Fraundorfer, "Camera pose estimation based on pnl with a known vertical direction," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3852–3859, 2019.

[28] R. Abdulmajed and R. ABBAK, "Accuracy comparison between gps only and gps plus glonass in rtk and static methods," Ph.D. dissertation, Doctoral dissertation, Selçuk Üniversitesi Fen Bilimleri Enstitüsü, 2017.

[29] J.-S. Franco and E. Boyer, "Fusion of multiview silhouette cues using a space occupancy grid," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, Oct. 2005, pp. 1747–1753 Vol. 2, iSSN: 2380-7504.

[30] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.