Evidential grid mapping from asynchronous LIDAR scans and images for autonomous driving

Edouard Capellier, Franck Davoine, Vincent Frémont, Javier Ibañez-Guzman, You Li

Evidential grid mapping from asynchronous LIDAR scans and images for autonomous driving

Edouard Capellier[1,2], Franck Davoine[1], Vincent Fremont[1], Javier Ibanez-Guzman[2], You Li[2]

***Abstract*— In this paper, an evidential fusion scheme between LIDAR scans and RGB images is proposed. To assess the drivability of an area met by an autonomous vehicle, LIDAR points are classified as either belonging to the ground, or not, and the RGB image is processed by a state-of-the-art convolutional neural network to obtain semantic labels. The results are then fused into an evidential grid, to handle incoherences over time and between sensors. The dynamic behaviour of potentially moving objects can be estimated from the high-level semantic labels obtained from the semantic segmentation of the image. LIDAR scans and images are not assumed to be acquired at the same time, making the proposed grid mapping algorithm asynchronous. This approach is justified by the need for handling, at the same time, sensor uncertainties, incoherences of results over time and between sensors, and the need for handling sensor failure. In classical LIDAR/camera fusion, in which LIDAR scans and images are considered to be acquired at the same time (or synchronously), the failure of a single sensor leads to the failure of the whole fusion system. However, experiments on a challenging use case show how this approach can be used to fuse contradictory information over time, while allowing the vehicle to operate even in case of the failure of a single sensor.**

I. INTRODUCTION

Long range 3D perception, and semantic understanding of the environment, are key challenges for mobile robots and autonomous vehicles. Though it has been proposed to fully address, at the same time, both of these issues via a multi-task convolutional neural network [1], such approaches are not sufficient on their own. The fact that convolutional neural networks seem sensible to minimal distortions within an image, as shown in [2], justifies to only use them when paired with highly reliable sensors, such as LIDAR scanners [3]. A fusion framework generating 2D evidential grids from LIDAR scans, and segmentation results from a state-of-the-art convolutional neural network, is thus proposed in this paper. The evidential theory [4] was adopted since this formalism can efficiently handle incoherences between multiple pieces of information, while being able to model unobserved areas, and to manage sensor uncertainties. Additionally, 2D evidential grids are pertinent inputs for path planning systems, such as the one described in [5]. As explained in Fig. 1, Dempster's conjunctive rule is used to fuse, over time, evidential Cartesian grids from LIDAR scans, and image segmentation results projected on the ground plane. As this system presents a modular design, its components that process raw data, such as the neural network, can easily be replaced. One of the specificities of the proposed approach is the fact that it is asynchronous: the LIDAR scanner and RGB camera are not supposed to be synchronized and to acquire data at the same time. Instead, each sensor operates freely, and each sensor input is processed individually, and fused with the global state. The resulting system is thus flexible enough to undergo sensor failure. In the end, the main contributions of this paper are:

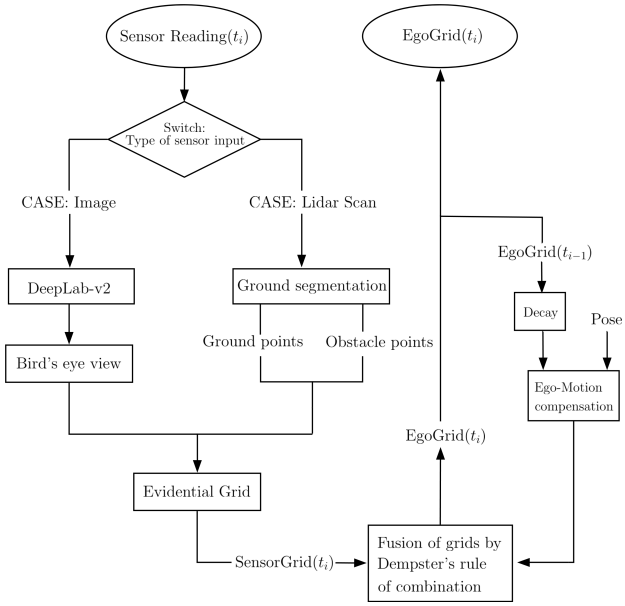


Fig. 1: The proposed asynchronous fusion algorithm. Images and LIDAR scans are individually processed when they are acquired. An evidential grid is then built from the sensor input, and fused based on the vehicle odometry with the current observed state of the world, denoted as EgoGrid. Thus, no LIDAR/camera synchronization is needed.

- A new Cartesian grid mapping framework for LIDAR sensors ;
- A new evidential, asynchronous, fusion scheme for semantic segmentation results generated by a convolutional neural network, and LIDAR scans ;
- An experimental analysis of the interest of the method on real-life data, that was collected in an urban environment ;

The paper is organized as follows: section II gives an overview of related work. Section III illustrates the proposed

*This work is supported by a CIFRE fellowship from Renault S.A.S

[1]Sorbonne universités, Université de technologie de Compiègne, CNRS, HeuDiaSyc, Centre de recherche Royallieu, CS 60319, 60 203 Compiegne cedex, France. Contact: name.surname@hds.utc.fr

[2]Renault S.A.S, 1 av. du Golf, 78288 Guyancourt, France. Contact: name.surname@renault.com

evidential framework and fusion scheme. Section IV explains how LIDAR scans and images are individually processed and fused over time, and section V presents the behaviour of the system in a challenging use-case: the failure of the camera while being overtaken by another vehicle.

II. RELATED WORK

A. Semantic segmentation via convolutional neural networks

State-of-the-art results on several semantic segmentation benchmarks have been achieved thanks to fully convolutional neural networks. Chen et al. [6] proposed Deeplab-v2: ResNet101 [7] is used as an encoder network with additional atrous convolutions, applied with different rates to detect details at multiple scales, and an additional CRF as a post-processing step to smooth the segmentation results. Deeplab-v2 reached 79.7% mIOU accuracy at the PASCAL VOC-2012 semantic image segmentation task. This result was overcame by PSPNet [8] in which feature maps initially encoded by ResNet-50 are reprocessed at different scales by parallel convolutional layers. PSPNet reached 85.4% mIoU accuracy on the PASCAL VOC-2012 semantic image segmentation task and 81.2% accuracy on the CityScapes dataset [9]. This result was recently overpassed by DeepLab-v3 [10] which reached 81.3% accuracy on CityScapes thanks to atrous convolutions in cascade and, similarly to PSPNet, the use of concatenated multi-scale features for the final segmentation. After exploring several CNN architectures, we decided to finetune DeepLab-v2 on additional driving scenes, as Deeplab-v3 was released too recently, and PSPNet's training framework is not publicly available.

B. Evidential occupancy grid mapping

Yu et al. [11] originally proposed an evidential sensor model to build polar occupancy grids from a Velodyne HDL-64E LIDAR sensor. Based on the angular resolution and beam divergence of the LIDAR sensor, a polar missed detection rate is estimated, and a false alarm rate is empirically defined. Hogger et al. [12] convert this mapping scheme to a fully Cartesian mapping scheme, by approximating the false alarm and missed detection rates by two ad hoc constants without considering the characteristics of their sensor. The authors of [11] also proposed an evidential mapping scheme from stereo-vision in [13].

C. LIDAR-camera fusion for autonomous driving

The first satisfying fusion scheme involving LIDAR scans and image segmentation results, relying on the use of a multi-layer perceptron and fuzzy reasoning, was proposed by Zhao et al. [3]. Recently, many new fusion schemes were proposed thanks to the public release of the KITTI dataset [14], which consists in calibrated and synchronized LIDAR scans and images, and localization information, captured in driving scenes. An evidential fusion framework between LIDAR classifiers and camera classifiers was proposed for instance in [15]. Li et al. [16] proposed to use a convolutional neural network within a monocular SLAM framework, in order to create a semantic map of a scene. Another recent proposition is also to directly use convolutional neural networks as fusion frameworks for LIDAR and camera data. Chen et al. [17] for instance proposed to train a convolutional neural network on multi-view data coming from the KITTI dataset, in order to perform object detection in the 3D space. Since most fusion schemes designed from the KITTI dataset benefit from the fact that the LIDAR scans and images were simultaneously acquired, they cannot handle sensor failures. Instead, asynchronous fusion can handle both sensor failures and timing errors in a flexible way, as sensor inputs are simply processed individually.

III. EVIDENTIAL FRAMEWORK FOR SENSOR FUSION

In order to generate the final evidential grid, individual grids for each sensor reading have first to be computed, and then fused. The final goal is to know, at every moment, whether a location in the perceived environment, i.e. a cell of a grid, can be passed through by an autonomous vehicle. A corresponding frame of discernment Ω was thus defined as $\{D,\ ND\}$, where the two singletons D and ND are propositions respectively indicating that a cell is either drivable or non-drivable. It is then possible to derive $2^{|\Omega|}$ subsets from Ω, the set of which form the power set $2^{\Omega} = \{\emptyset, D, ND, \Omega\}$. Each singleton of the power set is a proposition. The empty set $\emptyset$ indicates that the cell is not in a state that corresponds to the model, and Ω indicates that the state of the cell is unknown. This explicit quantification of ignorance is one of the specificities of evidential grids. Safer decisions can be taken from such grids compared to grids generated via a classical bayesian framework, as observed in [18].

The amount of proof corresponding to each proposition of the power set can be quantified via appropriate functions. It is assumed that, in each generated grid, a direct quantification of the belief is available. This quantification, or mass, is computed via a Basic Probability Assignment (BPA) for which a mass function m stands for.

Evidential theory offers powerful tools to fuse different sources of information in a flexible way. Let $m1$, $m2$ be two BPAs independently describing the state of a given cell. Typically, $m1$ and $m2$ can come from different sensor readings. Dempster's conjunctive rule [4], described in Eq. 1, can be used to compute a new joint mass $m_{1,2}$:

$$m_{1,2}(A) = \frac{1}{1-K} \sum_{B \cap C = A, A \neq \emptyset} m_1(B) m_2(C) \qquad (1)$$

$$K = \sum_{B \cap C = \emptyset} m_1(B) m_2(C) \qquad (2)$$

The normalization by $1 - K$ is a way to handle any conflict between the states during fusion, by attributing it to the empty set.

IV. FROM RAW LIDAR SCANS AND IMAGES TO EVIDENTIAL GRIDS

Before being able to use the fusion scheme described previously, evidential grids have to be generated. The fol-

lowing section describes how such grids can be produced from LIDAR scans and images, and how they are fused.

A. Generating evidential grids from LIDAR scans

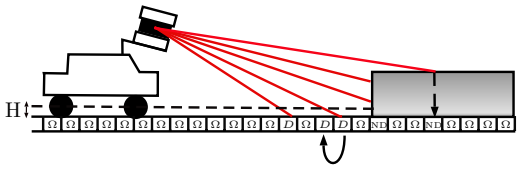


Fig. 2: Evidential mapping from a LIDAR scan. The state having the largest mass is reported for each cell. The curved arrow indicates the occurrence of backward propagation.

To convert LIDAR scans into evidential grids, an approach inpired by Yu et al.'s polar grid mapping [11] is proposed, to create evidential scan grids in a Cartesian coordinate system. The resulting grids can then be fused over time, based on the odometry of the vehicle. Contrary to the work in [12], the intrinsic characteristics of the LIDAR are still used to accurately characterize the information available for each Cartesian cell.

When converting a scan to an evidential grid, drivable areas are those where only points corresponding to the ground are detected, and non-drivable ones are those where an obstacle is detected. The ground is represented by a flat plane. Provided that the altitude and rotation of the LIDAR relatively to the ground is known, each point can be projected on this ground plane. This plane is then divided into regular cells to build an evidential grid. To quantify the amount of proof associated with each state in every cell, false alarm and missed-detection rates are defined. A *false alarm* happens if it is wrongly considered, due to sensor noise for instance, that an obstacle is present. On the opposite, a *missed detection* happens when an obstacle that is actually present is not detected by the LIDAR, often because of its size or reflectivity. Let α_{FA} be the false alarm rate in a given cell, n_o the number of points that hit this cell and are classified as obstacles, n_g the number of points that hit the cell and are classified as ground points, and $\alpha_{MD}(n_g)$ the corresponding missed detection rate. For each cell, the corresponding BPA, denoted as m_l, is computed as follows:

$$m_l(\emptyset) = 0 \tag{3}$$

If no point has hit the cell:

$$m_l(D) = 0, m_l(ND) = 0, m_l(\Omega) = 1 \tag{4}$$

If all the points that hit the cell are classified as ground:

$$m_l(D) = 1 - \alpha_{MD}(n_g), m_l(ND) = 0, m_l(\Omega) = \alpha_{MD}(n_g) \tag{5}$$

If at least one point that hit the cell is classified as obstacle:

$$m_l(D) = 0, m_l(ND) = 1 - \alpha_{FA}^{n_o}, m_l(\Omega) = \alpha_{FA}^{n_o} \tag{6}$$

On the one hand, the false alarm rate is considered to be the same for every cell. On the other hand, an unique missed detection rate is computed for every particular cell, as the number of laser impacts that can intercept a regular cell depends on its position. To compute the missed detection rate of a given cell, the maximum number of laser impacts that can occur in this area is computed, and compared with the actual number of points classified as ground points that hit the cell. Given the beam divergence and the horizontal angular resolution of a LIDAR, this maximum number of hits can be deduced from the maximum angle between two points belonging to the cell. For the sake of simplicity and computational workload, the maximum angle is considered to be, for every cell, the maximum angle formed by opposite corners of the cell, named A and B, and the origin of the grid, i.e. the origin of the LIDAR projected on the ground plane, named O. Let o be the size of the diagonal of a cell, b the distance between O and A, and a the distance between O and B. This maximum angle γ can be computed from the law of cosines, as follow:

$$\gamma = \cos^{-1}\left(\frac{o^2 - a^2 - b^2}{-2ab}\right) \tag{7}$$

Let l_{bd} be the beam divergence of a LIDAR. The LIDAR is assumed to be in a position such that any cell belonging to the ground can only be hit by a single rotating laser. Thus, for every cell, the missed detection rate can be estimated as follow:

$$\alpha_{MD}(n_g) = 1 - \frac{n_g \cdot l_{bd}}{\gamma} \tag{8}$$

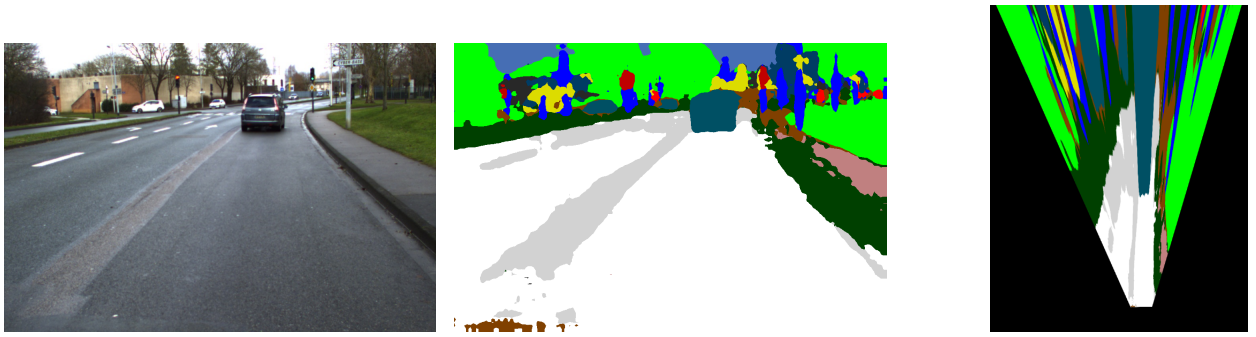
Finally, backward extrapolation, given a maximum threshold H, is performed: masses of cells classified as drivable are propagated to cells where nothing is detected, but obstacles taller than H cannot be present. Fig. 2 illustrates how evidential grids are built from LIDAR scans.

B. Generating evidential grids from segmented images

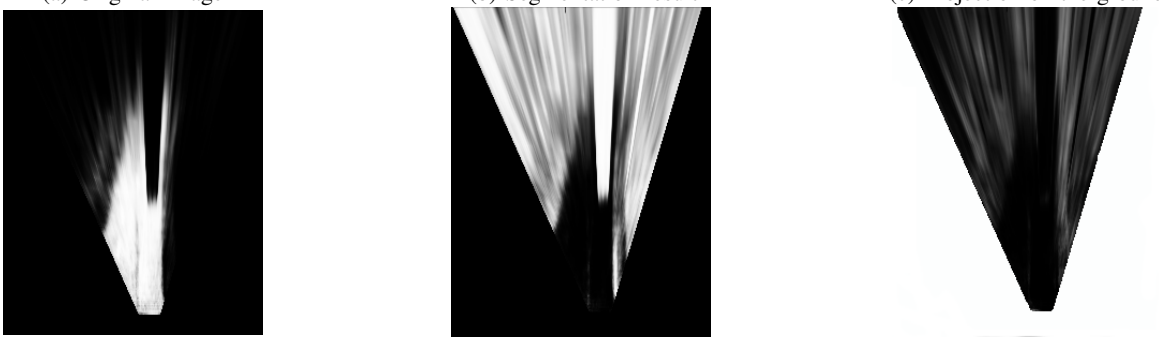
Fusing evidential grids generated from a LIDAR and pixel-wise segmentation results requires the later to be converted into an evidential grid. Contrary to the work in [13], stereo-vision is not used to estimate the depth of observed objects. Instead, the segmentation results generated from a mono-camera can be projected into a bird's eye view, corresponding to the ground plane in the LIDAR's coordinate system, to generate an evidential grid. The popular pinhole camera model to represent the behavior of an undistorted camera. Let $K \in \mathbb{R}^{3\times3}$ be the camera intrinsic matrix. The extrinsic matrix corresponding to the transformation between the camera's coordinate system and the LIDAR's coordinate system is supposed to be known. This matrix is composed of $T \in \mathbb{R}^{3\times1}$ and $R \in \mathbb{R}^{3\times3}$, respectively the translation vector and rotation matrix relating the two coordinate systems. Let $x = (X, Y, Z, 1)^T$ be a LIDAR point. It can be matched to a pixel $y = s \cdot (u, v, 1)^T$ as follow:

$$y = K \left[R \mid T \right] x \tag{9}$$

Then, the RANSAC algorithm, fed with matched LIDAR points and pixels belonging to the ground plane, can be used to compute the perspective projection matrix $H_{pg} \in R^{3\times3}$,



(a) Original image (b) Segmentation result (c) Projection on the ground



(d) Masses for the D proposition (e) Masses for the ND proposition (f) Masses for the Ω proposition

Fig. 3: Building process of an image-based evidential grid. The picture of a driving scene is segmented by DeepLab, before being projected on the ground plane. The masses for D (drivable), ND (non-drivable) and Ω (unknown) are then derived from the activations for each segmented class. In (d), (e), (f), the lighter a pixel is, the larger the mass is.

between the ground plane in the LIDAR's coordinate system, and the camera plane. It is then possible to match each pixel with a grid cell. To keep the computational workload low, only the center of each cell of the resulting grid is matched to a segmented pixel, instead of projecting every pixel in the evidential grid's coordinate system. Let x_{cell} be the known coordinates of the center of a cell, and y_{pixel}, the coordinates of the corresponding image pixel ; y_{pixel} can be computed as follow:

$$y_{pixel} = H_{pg} \cdot x_{cell} \tag{10}$$

Finally, the output of a fully convolutional neural network is used to build a new mass function, for every grid cell. The activation values are normalized, for every pixel mapped to a grid cell, via the softmax function, and then used to build a new mass function m_c. Let $\Omega_{cnn} = \{A_0, A_1, ..., A_n\}$ be the set of classes the convolutional neural network has been trained on. Let $z_p = (z_0^p, z_1^p, ..., z_n^p)$ be the corresponding activation values generated during inference by the neural network, for a pixel p. Let $\sigma(z_i^p)$ be the normalized activation for the class A_i and the pixel p. Then:

$$\sigma(z_i^p) = \frac{\exp z_i^p}{\sum_{k=0}^{n} \exp z_k^p} \tag{11}$$

$\sigma(z_i^p) \in [0,1]$ and $\sum_{i=0}^{n} \sigma(z_i^p) = 1$. Let $\cup_{B_i \in B}$ be the union of all the sets, depicted as B_i, that belong to the B set. It is assumed that there exists a partition of $\Omega_{cnn} = \{A_D, A_{ND}, A_\Omega\}$ such that:

$$\cup_{A \in A_D} = D \tag{12}$$

$$\cup_{A \in A_{ND}} = ND \tag{13}$$

$$\cup_{A \in A_\Omega} = \Omega \tag{14}$$

A mass m_p can then be computed for each pixel p mapped to a grid cell, as follow:

$$m_p(\emptyset) = 0 \tag{15}$$

$$m_p(D) = \sum_{A_i \in A_D} \sigma(z_i^p) \tag{16}$$

$$m_p(ND) = \sum_{A_i \in A_{ND}} \sigma(z_i^p) \tag{17}$$

$$m_p(\Omega) = \sum_{A_i \in A_\Omega} \sigma(z_i^p) \tag{18}$$

A new evidential grid is then obtained from the segmented image thanks to Eq. (10), which is used to map each cell of the grid with the corresponding segmentation results, and mass values. Fig. 3 illustrates how the mass value for each proposition, in each cell, is derived from the corresponding segmentation result. As showed in Fig. 3(c), when objects that do not belong to the ground are projected, they are stretched, which is normal since their presence occludes the ground in the distance. Since the information is meaningful, the grids are kept as they are.

C. Asynchronous fusion of LIDAR data and image segmentation results as an evidential grid

If the speed vector of the vehicle is available at any instant, and if all the sensor readings are accurately timestamped, grids corresponding to consecutive sensor readings can be processed independently from the type of sensor that issued the raw data. The grid obtained after fusion is called *EgoGrid* (cf. Fig 1), and has a BPA denoted as m_{eg} for each cell. At every new sensor reading, issued at a date t_i, a single evidential grid *SensorGrid(t_i)* is generated based on the type of sensor input (cf. Fig 1). After the ego-motion of the vehicle is compensated in *EgoGrid(t_{i-1})*, $m_{eg}(\Omega)$ is set to 1 for all the new cells that cover previously absent areas. Then, *SensorGrid(t_i)* can be fused with *EgoGrid(t_{i-1})* into a new *EgoGrid(t_i)* evidential grid via Dempster's conjunctive rule. Thanks to this framework, new sensors can easily be added to the fusion process, and a faulty sensor can be ignored without preventing the system from working in a degraded mode. Furthermore, contradictions between successive frames at a given locations are handled during fusion based on the mass associated to each state, in each frame. However, objects that have potentially moved between successive sensor readings have to be accounted for. No strong dynamic model, for any type of object, is presupposed. Instead, a decay factor is used to force every cell to eventually tend to the unknown state, similarly to [19]. Yet, as the types of the object present in the scene are available when using DeepLab-v2, an unique decay rate can be computed for each cell, based on the likelihood of the presence of a moving object. Let β be the decay rate associated with a cell of the current *EgoGrid*. Four main types of objects, that have similar dynamic behaviours, were identified: four-wheeled vehicles, two-wheeled vehicles, pedestrians and fixed objects. Each behaviour is associated with a typical decay rate: β_{4W}, β_{2W}, β_P and β_F. If no semantic information has been previously provided about a cell of the *EgoGrid*, a default value is used as the decay rate. Yet, if semantic information has been available for a given cell of the *EgoGrid* at previous dates, four values, indicating the likelihood that the object detected at this position correspond to one of those dynamic behaviours, are computed. Those indicators are denoted as L_{4W}, L_{2W}, L_P and L_F. Typically, when DeepLab-v2 is used, L_{4W} is equal to the sum of the activations corresponding to four-wheeled vehicles in the previous frames, for the given cell ; L_{2W} is equal to the sum of the activations corresponding to two-wheeled vehicles in the previous frames ; L_P is equal to the sum of the activations corresponding to pedestrians in the previous frames, and L_F is equal to the sum of the activations corresponding to fixed objects, such as the road, or buildings, in the previous frames. Then, the final decay rate for the cell is given by a weighted arithmetic mean:

$$\beta = \frac{L_{4W}\beta_{4W} + L_{2W}\beta_{2W} + L_P\beta_P + L_F\beta_F}{L_{4W} + L_{2W} + L_P + L_F} \tag{19}$$

Before fusing *EgoGrid(t_{i-1})* and *SensorGrid(t_i)*, each cell of *EgoGrid(t_{i-1})* is updated using β, as follow:

$$m_{eg}(A) = \beta \cdot m_{eg}(A), A \subset \Omega \tag{20}$$

$$m_{eg}(\Omega) = 1 - \beta + \beta \cdot m_{eg}(\Omega) \tag{21}$$

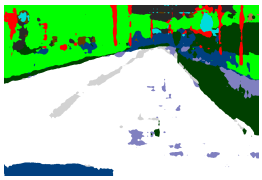
V. EXPERIMENTAL RESULTS

The evidential fusion scheme was implemented thanks to the software library proposed by Fankhauser et al. [20]. It was tested on real-life driving data collected around HeuDiaSyc Lab in Compiègne, France. The evidential grids are built from a VLP-16 LIDAR and a single HD camera, and the pose and speed of the vehicle were obtained from an IMU. Popular LIDAR/camera datasets, such as KITTI, were not considered for those tests, since one of the specificity of the proposed method is that it is intended to work in an asynchronous fashion. Evidential grids of $(90\times90)m^2$ are built from the collected data, with cells of size $(0.1\times0.1)m^2$. H was empirically set to 0.2m, α_{FA} to 0.05, the default value of β to 0.995, β_{4W} to 0.80, β_{2W} to 0.75, β_P to 0.95 and β_F also to 0.995. LIDAR scans were acquired at 10 Hz, and the camera was freely running at 30 Hz. The extrinsic calibration matrix between the LIDAR's coordinate system and the camera's coordinate system was estimated from the semi-automatic tool offered within the Autoware software stack [21]. The recent and fast algorithm described in [22] was used to classify each LIDAR point as either ground point or obstacle. DeepLab-v2 was finetuned on the publicly available Mapillary Vistas dataset [23], consisting in 25000 real-life driving scenes labelled into 66 object categories, to ake it usable in our experiments. To speed up and ease the finetuning of DeepLab-v2, the total number of classes was reduced, by factorizing some of them. A class for unlabelled objects in Mapillary Vistas was also reserved, and included in the loss calculation as an *unknown* class. Doing so, pixels are not forced to be classified into a meaningful class. Thus, in this set up, A_Ω ={*unknown*, *sky*}, as the pixels depicting the sky are not supposed to be part of the ground plane. A_D ={*road*, *road marking*, *crosswalk*}, and the remaining classes form A_{ND}. The classical stochastic gradient descent with momentum was used for finetuning DeepLab-v2, with the same parameters as in [6]. The loss function was modified to handle class imbalance within the dataset, by weighting the error for each pixel depending on the target class thanks to median class balancing [24]. The finetuning of DeepLab was performed during sixteen epochs, until the validation loss started increasing. Three cases, each highlighting specific advantages and drawbacks of the proposed approach, are presented. They were generated from the same driving sequence, but at different instants. The data collection vehicle was driven in a peri-urban environment and overtaken by another vehicle. During the overtaking, the camera was permanently switched off, to simulate a sensor failure. The full driving sequence is presented in the supplementary elements of this paper. Additional test sequences will be made publicly available.

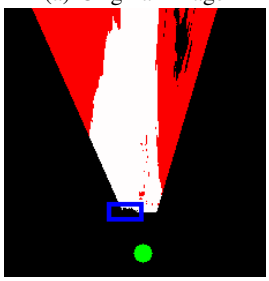
A. Handling sporadic semantic segmentation errors

First, the robustness of the fusion scheme against incoherences between successive sensor readings, and especially sporadic false alarms, is highlighted in Figure 4.

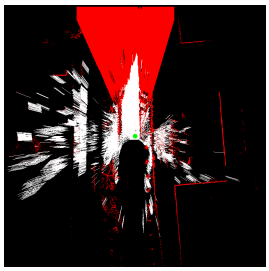
(a) Original image



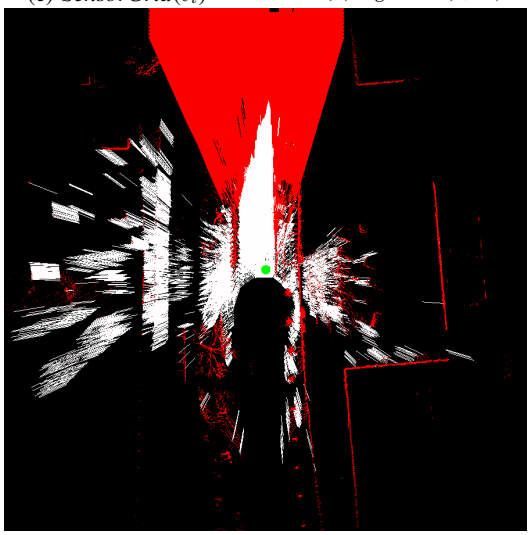
(b) Semantic segmentation



(c) *SensorGrid*(t_i)



(d) *EgoGrid*(t_{i-1})

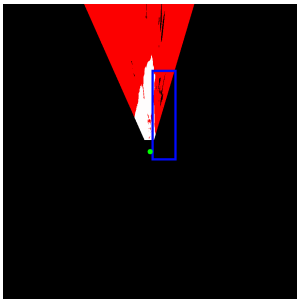


(e) Resulting *EgoGrid*(t_i) after fusion

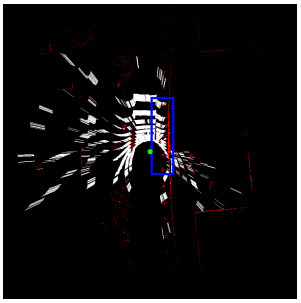
Fig. 4: Robustness of the fusion against sporadic errors. As displayed in (b), an object was wrongly detected by DeepLab. However, the conversion and fusion of this information in the evidential framework efficiently filtered the semantic segmentation result.

In the semantic segmentation result, white indicates that the class with the highest activation is "road" ; grey that it is "road marking" ; blue that it is "building" ; purple that it is "sidewalk" ; green that it is "border". In the grids, red cells are those where the largest mass is for ND ; white that the largest mass is for D ; black that it is for Ω ; the green point indicates the origin of the LIDAR, considered to be the vehicle's position. In Fig. 4b, many segmentation errors seem to come from the fact that the road is particularly damaged, and was repaired many times. As a result, objects are wrongly considered to be present, especially a building in the bottom-left corner. In the *SensorGrid*, a blue rectangle indicates the cells corresponding to this wrongly detected building. The mass of the cells in this area is larger for Ω, which indicates that even if the activation for the "building" class is the largest, the sum of the classes corresponding to A_Ω is larger. The segmentation result is thus very uncertain in this area. This is not the case for the pixels wrongly classified as "side-walk", but belonging in fact to the road, since small obstacles are detected in *SensorGrid* in front of the vehicle. The *EgoGrid*(t_{i-1}) to be fused with the *SensorGrid* was generated from 6 previous LIDAR scans and 7 previous images. The resulting *EgoGrid*(t_i) is marginally impacted, as no obstacle is considered to be present in front of the origin, even if a small area is considered to be unknown. This means that the mass for the areas falsely considered to be non-drivable in *SensorGrid* was not very high, compared to the corresponding mass in *EgoGrid*(t_{i-1}) for the D proposition. This shows the interest of fusing all the information over time, and to consider all the activations of the neural network.

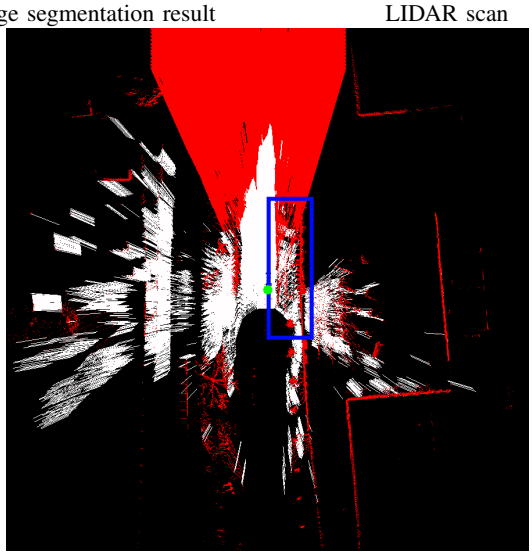
B. Handling systematically contradictory information



(a) *SensorGrid* from an image segmentation result



(b) *SensorGrid* from a LIDAR scan



(c) *EgoGrid* generated at this position

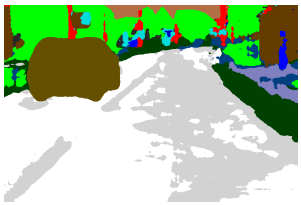
Fig. 5: Result of systematic inconsistencies between *SensorGrids*. The dark blue rectangle indicate the approximative position of a side-walk.

If temporal fusion among successive frames, and the use of mass values, can efficiently be used to handle sporadic errors while processing sensor inputs, the behaviour against systematic errors is not always as satisfactory. The ground segmentation algorithm used in this experimental set-up is

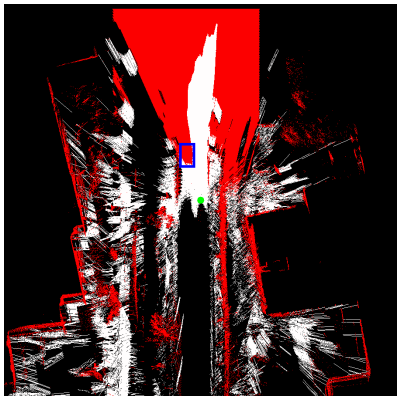
mainly designed to detect ground planes. As such, roads and side-walks are often both considered to be drivable in a *SensorGrid* generated from a LIDAR scan. Nevertheless, side-walk borders are efficiently detected, thanks to the gap between roads and side-walks, as shown in Fig. 5. Side-walks though remain uncertain, as the corresponding cells are not consistently classified as non-drivable.

C. Handling sensor failures

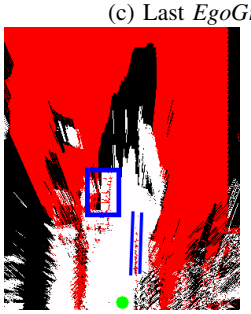
(a) Last image before camera failure



(b) Last segmentation result before camera failure



(c) Last *EgoGrid* before camera failure



(d) *EgoGrid* after 1s

(e) *EgoGrid* after 2s

Fig. 6: Handling camera failure. Blue rectangles indicate the actual position of the overtaking car, and blue lines highlight the border of the sidewalk.

The last case regards the moment when a camera failure was simulated. *SensorGrids* can then only be generated from LIDAR scans. As a result, the *EgoGrid* was updated less often, and the decay was less applied. As shown in Fig. 6d, this results in the conservation of outdated information, coming from previous detections. However, the car is still detected, and after a few more scans, the results become more consistent, as seen in Fig. 6e. Finally, as the side-walk borders are still detected, an autonomous navigation in such a fail-soft mode would have still be possible.

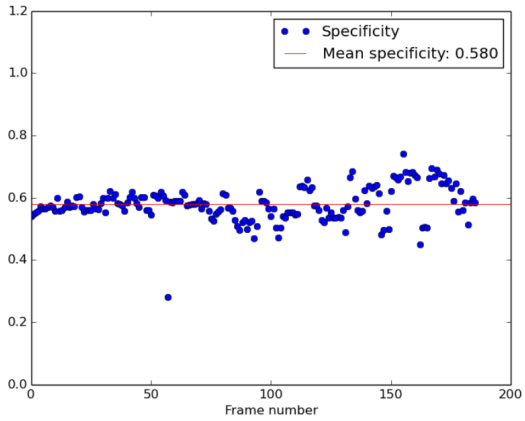
D. Evaluation of the importance of handling moving objects

Yager et al. proposed in [25] to evaluate mass functions by calculating entropy and specificity values. The effectiveness of the proposed grid mapping scheme can be evaluated from such indicators. Let E_m be the entropy of the mass function m, and S_m its specificity. Let the plausibility of a set A be $pl(A) = \sum_{B|B\in A} m(B)$.

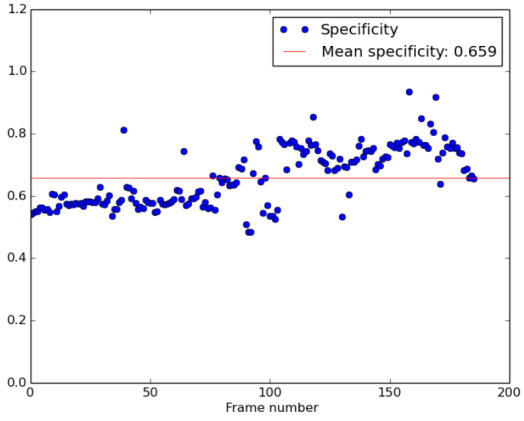
$$E_m = -\sum_{A\subseteq\Omega} m(A).ln(pl(A)) \tag{22}$$

$$S_m = \sum_{A\subseteq\Omega, A\neq\emptyset} \frac{m(A)}{card(A)} \tag{23}$$

A high degree of specificity and low-degree of entropy indicate tat the mass function is informative and non-ambiguous. The mean entropy and specificity of the mass assignment in the cells of an evidential grid are thus representative indicators of the quality of the whole representation.



(a) Mean specificity for a fixed decay rate



(b) Mean specificity when computing the decay rate from β_{4W}, β_{2W}, β_P and β_F

Fig. 7: Comparison of the average specificity for a fixed decay rate, and the proposed class-dependent decay rate

Those values were calculated for each frame of the sequence, in two cases: first, with a fixed decay rate of 0.98 for each cell as in [11], and then based on the values of β_{4W}, β_{2W}, β_P and β_F. The camera failure was simulated from the

frame 163. In both cases, the entropy was extremely low, and below 0.015. Yet, as shown in Fig.7a and 7b, the specificity is higher for this sequence when the class of the object in each cell is considered, making the resulting grids more informative. Indeed, the average specificity for the sequence is 0.580 when a fixed decay rate is used, and 0.659 when the decay rate is computed based on the activations of the neural network.

VI. CONCLUSIONS

In this paper, a grid-based asynchronous fusion scheme of LIDAR scans and images from a mono-camera was presented. A new Cartesian mapping scheme from LIDAR scans was proposed, and a way to handle possibly moving objects based on their semantic class was evaluated. The use of an adaptive decay rate, computed from semantic classification results, seem to be an efficient way to generate a meaningful representation, even when moving objects are present. Moreover, the interest of asynchronous fusion was highlighted. Processing each individual sensor reading, and temporally aligning the generated grids, is a flexible and efficient way to fuse information, while allowing the fusion system to continue working although one of the sensors has failed. Real-time performances have not been reached yet, especially because of the use of DeepLab-v2, and because each operation on grid cells are done iteratively. Those two problems will be solved by the use of a faster neural network, and the use of GPU operations to update the grid cells in parallel. Moreover, new modalities will be introduced within the fusion system, such as a priori knowledge from HD maps.

ACKNOWLEDGMENT

This work was realized within the SIVALab joint laboratory between Renault S.A.S and HeuDiaSyc UTC/CNRS Lab. We also thank NVidia for their donation of a Titan X GPU card that was used in this work.

REFERENCES

[1] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," *CoRR*, vol. abs/1705.07115, 2017. [Online]. Available: `http://arxiv.org/abs/1705.07115`.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013.

[3] G. Zhao, X. Xiao, J. Yuan, and G. W. Ng, "Fusion of 3D-lidar and camera data for scene parsing," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 165–183, 2014.

[4] G. Shafer, *A mathematical theory of evidence*. Princeton university press, 1976, vol. 42.

[5] H. Mouhagir, R. Talj, V. Cherfaoui, F. Guillemard, and F. Aioun, "A markov decision process-based approach for trajectory planning with clothoid tentacles," in *Intelligent Vehicles Symposium (IV)*, IEEE, 2016, pp. 1254–1259.

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *arXiv preprint arXiv:1606.00915*, 2016.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *arXiv preprint arXiv:1612.01105*, 2016.

[9] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The CityScapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[10] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[11] C. Yu, V. Cherfaoui, and P. Bonnifait, "An evidential sensor model for velodyne scan grids," in *Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on*, IEEE, 2014, pp. 583–588.

[12] A. Högger, "Dempster shafer sensor fusion for autonomously driving vehicles: association free tracking of dynamic objects," *Master's thesis, KTH Royal Institute of Technology, Stockholm*, 2016.

[13] C. Yu, V. Cherfaoui, and P. Bonnifait, "Evidential occupancy grid mapping with stereo-vision," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*, IEEE, 2015, pp. 712–717.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: the KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[15] P. Xu, F. Davoine, J.-B. Bordes, H. Zhao, and T. Denœux, "Multimodal information fusion for urban scene understanding," *Machine Vision and Applications*, vol. 27, no. 3, pp. 331–349, 2016.

[16] X. Li and R. Belaroussi, "Semi-dense 3d semantic mapping from monocular slam," *arXiv preprint arXiv:1611.04144*, 2016.

[17] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," 2017.

[18] C. Yu, V. Cherfaoui, and P. Bonnifait, "Evidential grids with semantic lane information for intelligent vehicles," in *RFIA-Journée Transports Intelligents*, 2016.

[19] J. Moras, V. Cherfaoui, and P. Bonnifait, "Moving objects detection by conflict analysis in evidential grids," in *Intelligent Vehicles Symposium (IV)*, IEEE, 2011, pp. 1122–1127.

[20] P. Fankhauser and M. Hutter, "A universal grid map library: implementation and use case for rough terrain navigation," in *Robot Operating System (ROS)*, Springer, 2016, pp. 99–120.

[21] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles," *IEEE Micro*, vol. 35, no. 6, pp. 60–68, 2015.

[22] P. Chu, S. Cho, S. Sim, K. Kwak, and K. Cho, "A fast ground segmentation method for 3D point cloud," *Journal of information processing systems*, vol. 13, no. 3, pp. 491–499, 2017.

[23] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kontschieder, "The Mapillary Vistas dataset for semantic understanding of street scenes," in *Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy*, 2017, pp. 22–29.

[24] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2650–2658.

[25] R. R. Yager, "Entropy and specificity in a mathematical theory of evidence," *International Journal of General System*, vol. 9, no. 4, pp. 249–260, 1983.