Transformation-adversarial network for road detection in LIDAR rings, and model-free evidential road grid mapping

Edouard CAPELLIER[1,2], Franck DAVOINE[2], Veronique CHERFAOUI[2], You LI[1]

***Abstract*— We propose a deep learning approch to perform road-detection in LIDAR scans, at the point level. Instead of processing a full LIDAR point-cloud, LIDAR rings can be processed individually. To account for the geometrical diversity among LIDAR rings, an homothety rescaling factor can be predicted during the classification, to realign all the LIDAR rings and facilitate the training. This scale factor is learnt in a semi-supervised fashion. A performant classification can then be achieved with a relatively simple system. Furthermore, evidential mass values can be generated for each point from an observation of the conflict at the output of the network, which enables the classification results to be fused in evidential grids. Experiments are done on real-life LIDAR scans that were labelled from a lane-level centimetric map, to evaluate the classification peformances.**

I. INTRODUCTION

LIDAR sensors are traditionally used within occupancy grid mapping frameworks, to detect obstacles and infer the traversability of the environment. Evidential occupancy grid mapping frameworks usually assume that the ground is fully traversable, and evaluate the occupancy of cells from strong geometrical assumptions [1]–[3].

Yet, the applicability of such systems, in the context of autonomous driving, can be limited. First of all, they might fail to generate appropriate results, when the geometrical model they are based on is not satisfied anymore, which is likely to occur in complex urban areas. For example, the flat world assumption is not satisfied anymore at a speed bump. Then, areas that are traversable by an urban autonomous vehicle usually belong to the road: modelling the ground is thus not sufficient in most driving situations. Road detection in LIDAR scans is thus crucial, when aiming to implement evidential occupancy grid mapping algorithms in autonomous systems, that are intended to drive in urban areas. The use of machine learning could leverage the need for strong geometrical assumptions, as the system could be able to learn how to behave on edge-cases (speed-bumps, for instance), instead of relying on strong geometrical assumptions.

Inspired by the recent PointNet architecture [4] and novel advances in evidential classification [5], we propose to rely on a neural network that processes LIDAR rings individually, and can be used to output evidential mass values for each LIDAR point. Being able to represent the output of the neural network as evidential mass values is particularly valuable when trying to understand what was learnt, since the total amount of knowledge available at each position can be quantified. Moreoever, the evidential outputs of the network can directly be used in a model-free evidential grid mapping framework.

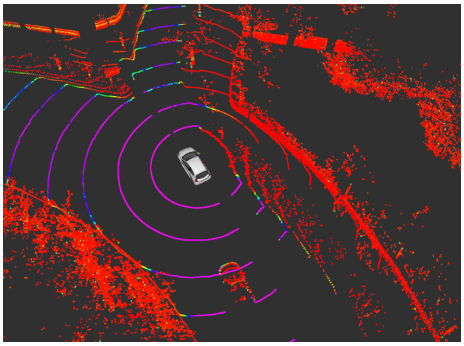


Fig. 1: Example of classification result. The grey ego-vehicle drives towards the road exit. The purpler a point is, the more likely it is to be on the road.

The paper is organized as follows: in Section II, we propose a short litterature review ; Section III presents how evidential mass values can be obtained from a neural network that was trained on coarse labels ; Section IV presents the ring-level neural network that we propose to perform road detection ; Section V presents the data collection and evaluation procedures used to train and evaluate the classifier and finally, Section VI presents a simple model-free evidential grid mapping system relying on the proposed classifier.

II. LITTERATURE REVIEW

A. Evidential grid mapping from LIDAR scans

Yu et al. [1] originally proposed an evidential sensor model to build polar occupancy grids from LIDAR scans. Based on the angular resolution and the beam divergence of the sensor, a polar missed detection rate was estimated, and a false alarm rate was empirically defined. From a ground-detection step relying on a flat-world assumption, the belief in the occupancy of each grid cell was then evaluated over time, according to an evidential framework. Such evidential polar grids however have to be interpolated, and mapped into a Cartesian coordinate system to perform fusion over time, at the cost of a loss in the correctness of the model. We ourselves proposed in [2] to evaluate a cartesian missed detection rate, to tackle this limitation while relying again on a ground detection algorithm and a flat world assumption. We

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

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

[2]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

however observed that such strong geometrical assumptions lack of flexibility, and are not always satisfied in practice. Simple ground detection also often fails to properly capture the actual drivable area. A road detection step, alongside a more flexible model, are thus needed to generate evidential grids from LIDAR scans in a more robust fashion.

B. Road detection from LIDAR scans

State-of-the-art approaches for road detection in LIDAR scans rely on image processing techniques. Fernandes et al. [6] proposed to project LIDAR points into a 2D image plane, to upsample them, and to detect the road in this image plane via an histogram similarity measure. Caltagirone et al. [7] proposed to project LIDAR points into a 2D sparse feature grid corresponding to a bird's eye view, and to train a convolutional neural network to predict a dense road region from this sparse representation. Lyu et al. [8] proposed to train a neural network on range images generated from the spherical projection of LIDAR scans, and to fit a polygon representing a dense drivable area on the predicted road points. Although those approaches are currently the best performing LIDAR-only road-detection approaches on the KITTI dataset, they all aim at predicting a dense road area from a sparse LIDAR scan, and thus rely on upsampling. All those approaches then predict the presence of road on locations where no actual LIDAR measurements were actually available, which is an undesirable behavior for a LIDAR-only road detection algorithm. Indeed, gaps or small obstacles could be present but remain unobserved due to the sparsity of a LIDAR sensor, in areas where those agorithms would predict the presence of road. Moreover, due to the limitations of the KITTI dataset, in which the road is only labelled in a front camera view, those systems do not detect the road on complete LIDAR scans. Point-level road detection should be performed in complete LIDAR scans, so as to only represent information in areas that are actually observed.

C. PointNet: Machine Learning on raw point clouds

The recent PointNet architecture, introduced by Qi et al. [4], processes vectors of raw point-clouds, in which the point corrdinates are directly stored. PointNet applies a multi-layer perceptron to each individual point, and produces a feature vector describing the whole point-cloud by applying a global max operator on the features extracted from each point. Althought simple, this solution has proven to approach, or overpass, state-of-the-art performances on several perception tasks relying on point-clouds. It was extended in [9], by extracting local features in a point-cloud at several contextual scales, based on the metric distances between points. The resulting system outperforms the original PointNet architecture, at the cost of an increased complexity and inference time. However, PointNet architectures suffer from several drawbacks. First of all, they require a fixed number of input points. Secondly, PointNets usually expect normalized, relatively dense and constrained inputs. This makes the architecture improper when aiming to process large-scale LIDAR scans [10], and often requires to split large point-clouds into individually processed voxels. Processing LIDAR points at the ring level could however leverage these limitations, as LIDAR rings are dense. Yet, a proper grid mapping framework relying on such a point-level classification is still to be defined. Especially, a proper way to represent the outputs of such a classifier into an evidential framework is still to be defined.

III. EVIDENTIAL REINTERPRETATION OF BINARY GLR CLASSIFIERS

T. Denoeux, in [5], proposed to reinterpret generalized logistic regression (GLR) classifiers as performing a fusion of evidential mass functions. With such a view, it is possible to construct evidential mass values, from the weights at the output of a neural network. Thanks to this technique, it becomes trivial to generate and accumulate evidential road detection results into an evidential 2D grid from a classifier, without relying on any explicit geometrical model. This is what we call *model-free evidential road grid mapping*.

Let a binary classification problem with $X = (x_1, ..., x_d)$, a d-dimensional input vector, and $Y \in \Theta = \{\theta, \neg\theta\}$ a class variable. Let $p_1(x)$ be the probability that $Y = \theta$ according to the fact that $X = x$. Let w be the output of a binary logistic regression classifier, trained to solve the aforementioned classification problem ; $p_1(x)$ is such that:

$$p_1(x) = S(w) = S(\sum_{j=1}^{d} \beta_j \phi_j(x_c) + \beta_0) \tag{1}$$

with S being the sigmoid function, and the β values being usually learnt alongside those of the potentially non-linear ϕ_i mappings. In Eq. 1, w exactly corresponds to the output of a deep neural network trained as a binary GLR classifier, with x_c being its input. There exist α_j values such that:

$$\sum_{j=1}^{d} \alpha_j = \beta_0 \tag{2}$$

$$w = \sum_{j=1}^{d} w_j = \sum_{j=1}^{d} (\beta_j \phi_j(x_c) + \alpha_j) \tag{3}$$

Each w_j can then be seen as a piece of evidence towards θ or $\neg\theta$, depending on its sign. Let w_j^+ be the positive part of w_j, and let w_j^- be its negative part. Let $w^+ = \sum w_j^+$, $w^- = \sum w_j^-$. An evidential mass function m_{LR} can be generated as follows:

$$m_{LR} = \{\theta\}^{w^+} \oplus \{\neg\theta\}^{w^-} \tag{4}$$

This means that any binary GLR classifier can be seen as a fusion of simple mass functions, that can be derived from the parameters of the final linear layer of the classifier. However, the α_j values in Eq. 2 have to be estimated. Let $\alpha = (\alpha_i, ...\alpha_d)$. T. Denoeux proposed to select the α vector that maximize the sum of the $m_{LR}(\Theta)$ mass values over the training set, so as to get the most uncertain and cautious solution. This leads to the following minimization problem:

$$minf(\alpha) = \sum_{i=1}^{n} \sum_{j=1}^{d} (\beta_j \phi_j(x_i) + \alpha_j)^2 \tag{5}$$

with $\{(x_i, y_i)\}_{i=1}^{n}$ being the training dataset.

An exact solution to this minimization problem exists [5], but it requires to perform an additional post-processing step after the training, and relies on the assumption that the parameters obtained after the training are reliable. When working with unperfect or coarse labels, an approximate solution is thus needed. We observed in [11] that an approximate solution to the minimization problem in Eq. 5 could be obtained directly during the training, by considering the α vector as the bias values of an Instance-Normalization layer present at the output of the network. Let $\upsilon(x_c) = (\upsilon_1(x_c), ..., \upsilon_d(x_c))$ be the mapping modelled by all the consecutive layers of the classifier but the last one ; let $\overline{\upsilon_j}$ be the mean value of the υ_j function on the training set, and $\sigma(\upsilon_j)^2$ its corresponding variance. Then, if it is assumed that Instance-Normalization is used as the final layer of the network, Eq. 5 becomes:

$$minf(\alpha) = \sum_{i=1}^{n}\sum_{j=1}^{d}((\beta_j \frac{\upsilon_j(x_c) - \overline{\upsilon_j}}{\sqrt{\sigma(\upsilon_j)^2 + \epsilon}}) + \sum_{j=1}^{d}\alpha_j)^2 \quad (6)$$

After development, the following expression is obtained:

$$minf(\alpha) = n\sum_{j=1}^{d}\beta_j^2 + n\sum_{j=1}^{d}\alpha_j^2 \quad (7)$$

By simply applying L2-regularization on the linear parameters of the final layer, this expression will be minimized during the training. The network can then be trained to generate relevant evidential mass values, even when the network is optimized on coarse labels.

IV. TRANSFORMATION-ADVERSARIAL NETWORK FOR POINT-LEVEL ROAD DETECTION IN LIDAR RINGS

A. Ring-level PointNet

Typically, dense LIDAR sensors rely on stacked lasers that individually sweep the scene. A LIDAR ring represents a set of points that is obtained after the sweep of the environment by a single laser of a LIDAR. To detect the road in LIDAR scans, without having to transform the raw points into another representation, a classifier inspired by PointNet can be used. To leverage the limitations of PointNet that were exposed in Sec. II, the processing is done at the ring level. Indeed, the maximum number of points that a LIDAR ring can include can be computed from the angular resolution of the LIDAR. Then, contrary to what was done in [4] and [10], no sampling of the point-cloud is needed. Moreover, LIDAR rings are often dense, especially at short range, since each laser sweeps the whole scene, which would facilitate the reasoning of a PointNet-like network. And in the event of missing points, the input vector can typically be padded with an already present point, since the point-cloud wise max-pooling operation used in PointNet can filter duplicate point features. Finally, the maximum number of points in each sweep is relatively small, which means that the LIDAR rings will be easily processed in parallel.

However, LIDAR rings vary significatively among each others: a ring acquired by a top laser and a bottom laser will include points that were acquired at very different distances. A training scheme, inspired by the recent successes of generative-adversarial networks (GAN) in the image domain [12], was proposed to cope with this issue.

B. Transformation-adversarial network for LIDAR rings

GANs rely on the conjunction of two alternatively trained systems. The first one, called the generator, is optimized to generate artificial samples that are as realistic as possible. The second one, called the discriminator, is trained to discriminate real and artificial samples. The two systems are competing against each other: the generator aims at fooling the discriminator, and the discriminator aims at detecting samples generated by the generator. Similarly, we propose a Transformation-adversarial network, or TAdNet, composed of a Transformation network, and a Classification/Discrimination network. In the original PointNet, T-Nets predict affine transformation matrices applied to the whole input cloud, and to intermediate features extracted by point-level MLPs. Those T-Nets are optimized during the training, alongside the other parameters of the network.

The Transformation network that we propose, which also applies a transformation predicted by a T-Net to the input, is optimized separately from the rest of the system. To cope with the variability among LIDAR rings, the Transformation network also includes an H-Net. This H-Net, or homothety network, processes the transformed point-cloud obtained from the transformation predicted by the T-Net, and predicts an explicit rescaling factor, that is applied to the coordinates of all the points. The input points are represented by their Cartesian coordinates (x,y,z), spherical coordinates (ρ,ϕ,θ), and their intensity. To account for the risk of redundancy among the point features, the ϕ and θ angles are the uncorrected azimuth and zenith at which the point was acquired, while the Cartesian coordinates are obtained after correction. Let h be the scale predicted by the H-Net. Then, the coordinates of the input points are rescaled as follows: $x_* = hx$, $y_* = hy$, $z_* = hz$, $\rho_* = h\rho$. All the other features are left unchanged.

The Transformation network can then learn to remap all the LIDAR rings into a constrained range, that is suitable for the road classification task. We assumed that it should be difficult to predict the ring ID of properly remapped and constrained LIDAR rings. The Transformation network is thus trained alongside a Classification/Discrimination network, and aims at generating similar LIDAR rings. This Classification/Discrimination network is in fact a multi-task PointNet, whithout any initial T-Net. It has to both perform road detection among the LIDAR points, and predict the ID of the LIDAR ring that it processes. This ring ID is predicted from the output of a small Pointnet-like subnetwork that is fed with the vector of concatenated point-level features and cloud-level features, that can be obtained after the max-pooling operation that every PointNet-like network uses. Following the results in Eq. 7, Instance-Normalization is

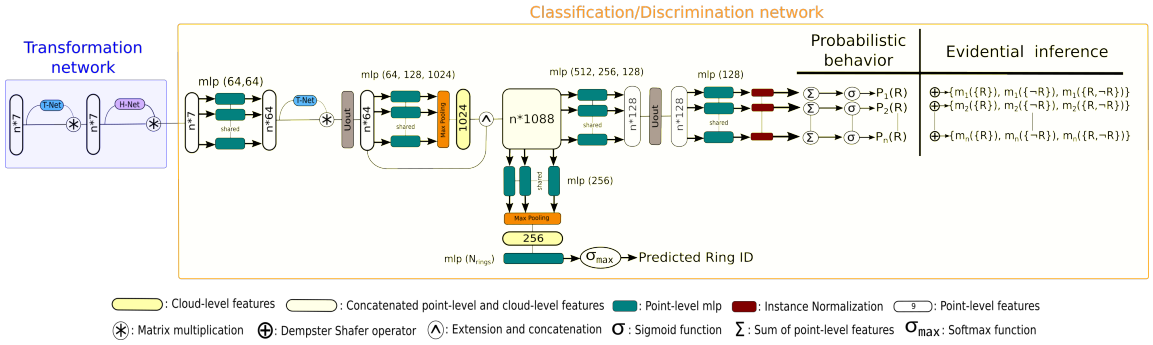


Fig. 2: Transformation-adversarial network for road-detection in LIDAR scans

used on the outputs used for road detection. The whole system is depicted in Fig. 2.

C. Training procedure

A PointNet-like system is typically trained with a multi-task loss. In the context of this study, the problem is point-level road detection in LIDAR rings. The loss chosen for this task, noted L_{ce} was the classical cross-entropy loss. The second component of the loss used for the training was a geometrical regularization loss. Let A be the transformation matrix predicted by the T-Net inside the Classification/Transformation network. This 64 by 64 matrix is more difficult to optimize than the simple transformation matrix predicted by the first T-Net, but should be as orthogonal as possible. Then, the loss on A to minimize, noted L_{geo}, is:

$$L_{geo}(A) = ||I - AA^T||^2 \tag{8}$$

Finally, the ring ID prediction error is again evaluated from the cross-entropy loss, calculated from the actual ring ID. We note this loss L_{ring}. Let L_{Tr}, the loss used to optimize the Transformation network, and L_{CD}, the loss used to optimize the Classification/Discrimination network. For each ring, let P_{road}, Y_{road}, P_{Ring} and Y_{Ring} be, respectively, the point-wise predicted probability that each point belongs to the road, the corresponding road labels, the predicted ring ID and the corresponding ring label. Then:

$$\begin{aligned} L_{CD} &= \lambda_{road} L_{ce}(Y_{road}, P_{road}) \\ &\quad + \lambda_{ring} L_{ce}(Y_{Ring}, P_{Ring}) \\ &\quad + \lambda_{geo} L_{geo}(A) \\ L_{Tr} &= \lambda_{road} L_{ce}(Y_{road}, P_{road}) \\ &\quad - \lambda_{ring} L_{ce}(Y_{Ring}, P_{Ring}) \\ &\quad + \lambda_{geo} L_{geo}(A) \end{aligned}$$

The whole system is trained thanks to the algorithm 1. To facilitate the training, UOut [13] was used. Originally, UOut was proposed because it was observed that Dropout shifts the mean and standard deviations of the features, which is not desirable when using Batch-Normalization, or Instance-Normalization. Uout, on the other hand, marginally affects those statistics. As Instance-Normalization is used on the output features of the network, due to the results of Eq. 7, UOut is a reasonnable choice to regularize the model.

Algorithm 1 Training of the proposed system

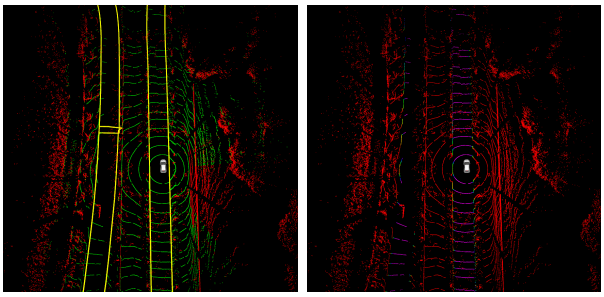
```
Transformation network: T ;
Classification/Discrimination network: CD ;
N training rings are available ;
for e epochs do
    for N/n iterations do
        Sample n batches (b_0, .., b_t) from the training set
        for i in range(n) do
            b_i^* = T(b_i)
            RoadClassif, RingID = CD(b_i^*)
            Update CD from L_CD
        end for
        for i in range(n) do
            b_i^* = T(b_i)
            RoadClassif, RingID = CD(b_i^*)
            Update T from L_Tr
        end for
    end for
end for
```

V. EXPERIMENTS AND EVALUATION OF THE CLASSIFICATION PERFORMANCES

A. Automatic labelling of a LIDAR dataset from a lane-level map

To properly evaluate the system, a dedicated LIDAR dataset was needed. No open-source LIDAR dataset including 360° point-level road labels was available when conducting this study. An autonomous perception platform equipped with a Velodyne VLP-32C running at 10 Hz was thus used to collect raw LIDAR scans in Guyancourt, France, in order to create a dataset with point-level road labels in LIDAR scans. Each LIDAR ring was composed of a maximum of 1800 points. The labelling was done automatically thanks to a pre-existing lane-level centimetric map, as shown in Fig 3. The data collection vehicle also included a Trimble BX940 inertial positioning system coupled with an RTK Satinfo modem, for localization.

A ground detection algorithm [14] was used to label obvious obstacles with a probability of being road-points equal to 0. The detected ground-points were projected into the map plane, for labelling. Following [15], the localization error was assumed to follow a zero-mean Gaussian model. Covariance matrices corresponding to the estimated position were provided by the localization system. The variance of



(a) Raw point-cloud, and the corresponding map available at the recording position. Green points belong to the pre-detected ground.

(b) Resulting point-cloud. Red points are labelled as obstacles ; the purpler a point is, the most likely of being a road point it is

Fig. 3: Automatic labelling procedure of a LIDAR point-cloud from a lane-level centimetric map.

the localization error was assumed to be the maximum variance on the easting/northing coordinates, noted σ_{xy}^2. This pessimistic assumption facilitates the computations, and accounts for possibly undetected timing or calibration errors. Let a detected ground-point x_i, with d_i the distance between its projection on the map plane and the closest mapped road-edge. The labelled probability of x_i being a road point y_i can be computed from the cumulative distribution function of the normal distribution. If x_i was projected into a mapped road:

$$y_i = \int_{-\infty}^{d_i} \frac{1}{\sigma_{xy}\sqrt{2\pi}} \exp^{-\frac{1}{2}(\frac{x}{\sigma_{xy}})^2} dx \tag{10}$$

Otherwise:

$$y_i = 1 - \int_{-\infty}^{d_i} \frac{1}{\sigma_{xy}\sqrt{2\pi}} \exp^{-\frac{1}{2}(\frac{x}{\sigma_{xy}})^2} dx \tag{11}$$

To prevent the presence of redundant data, the labelling procedure was only launched every ten scans. It was also disabled when the vehicle was stopped. The final dataset was finally generated from 2334 labelled LIDAR scans acquired in Guyancourt, France. In practice, when d_i was larger than $10 * \sigma_{xy}$, y_i was set to either 0 (the point is not projected into a road) or 1 (the point is projected into a road). 0-1 labels represent more than 96,5% of the labels. A 70/30 split was used to create a training and a validation set from this data. To ensure that the train and validation dataset are significantly different, the scans were first ordered according to their recording date. Then, the validation set was created from the earliest and latest fifteen percents of the dataset. With such a dataset of automatically and softly labelled LIDAR scans, being able to generate evidential mass values while training on coarse labels, as allowed by the use of Instance Normalization and L2-regularization, is valuable.

B. Evaluation procedure and results

We report the clssification results in Table I. Three systems were evaluated: the proposed Transformation-Adversarial Network (TAdNet), a ring-level PointNet, and a scan-level PointNet, to quantify the interest of the refinements introduced with TAdNet. The point-level MLPs were following

	All labels		0-1 labels	
Model	F1-score	Accuracy	F1-score	Accuracy
PointNet [4] - *ring*	0.868	0.973	0.907	0.983
PointNet [4] - *scan*	0.899	0.980	0.933	0.988
TAdNet - *ours*	0.933	0.987	0.959	0.993

TABLE I: Classification results for PointNet on LIDAR scans and tings, and for the proposed TAdNet, on the validation set

the original architecture proposed in [4], with a ReLU activation function and systematic use of Batch Normalisation. The three systems were implemented in PyTorch. The two PointNets consisted in exactly the same layers as TAdNet, except for the H-Net and the ring-ID prediction subnetwork that were removed. Instance Normalization and UOut were still used, as the resulting systems were all intended to be used for model-free evidential road-grid mapping. The Adam optimizer was used for the three networks, with a learning rate of 0.0001. Following the recommendations from the original authors of Uout, the random numbers generated by the Uout layers were sampled from a $[-0.1, 0.1]$ range. Empirical observations showed that, instead of only applying L2-regularization to the final layer of the networks, applying it to all the parameters led to better numerical stability. Then, a weight-decay of 0.0001 was applied to all the parameters of the three networks, except for the parameters of the Transformation-network in TAdNet, on which a weight-decay of 0.00001 was applied. All the T-Nets and the H-Net were initialized with identity transformations. Following [4], all the parameters of the multi-task losses were set to 1 for the regular PointNets, and for TAdNet, λ_{ring} was set to 0.8, λ_{road} was set to 1.2 and λ_{geo} was set to 1. TAdNet and the ring-level PointNet were trained on mini-batches including 64 rings, and the scan-level PointNet was trained on mini-batches of 2 scans, as each scan was composed of 32 rings. We report F1-scores and accuracies on the full validation dataset, and on only the 96.5% of 0-1 labels. In the case of non-binary labels, a point was considered to be labelled as a road-point if its labelled probability was higher than 0,5. And a point was considered to be classified as road if the predicted probability was higher than 0,5. Table I reports the respective results of those approaches in the validation set. All approaches have satisfactory results, even if TAdNet outperforms all the approaches in all the indicators. The interest of the rescaling performed by TAdNet is obvious, as the ring-level PointNet is by far the worst performing approach, while TAdNet outperforms the scan-level PointNet, even though it only processes rings.

VI. MODEL-FREE EVIDENTIAL ROAD GRID MAPPING FROM THE CLASSIFICATION RESULTS

Evidential road grids can easily be generated from TAdNet, and the expression in Equation 4. For each point, three evidential mass values can be extracted: m($\{$R$\}$), for the road class ; m($\{\neg$R$\}$), for the obstacle class ; and m($\{$R, $\neg$R$\}$), for the unknown class. Then, a grid can be obtained by projecting all the LIDAR points into the xy-plane. The

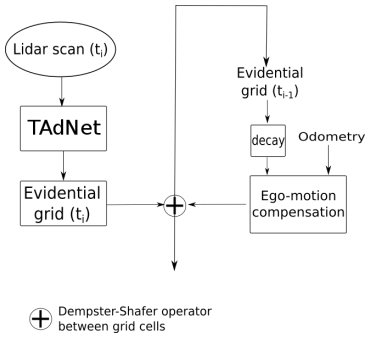
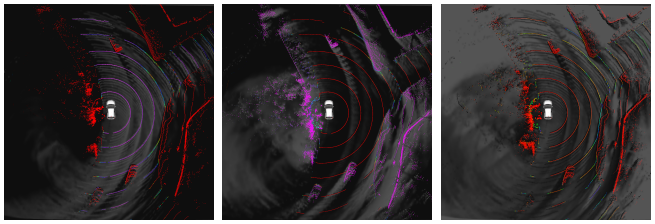


Fig. 4: Simple model-free evidential road grid mapping algorithm



(a) m({R}) **(b)** m({¬R}) **(c)** m({R, ¬R})

Fig. 5: Model-free evidential road grid mapping. The accumulated evidential grid is overlayed with the point-level evidential mass values generated from the LIDAR sensor

Dempster-Shafer operator can then be used to fuse the mass values of all the points that are projected into a given cell. Finally, the evidential grids can be fused over time thanks to the algorithm in Fig 4, which follows the approach in [1] but applies it to evidential mass values generated from TAdNET, instead of using a geometrical model. Figure 5 presents an example of model-free evidential road grid map generated from this algorithm, and TAdNet. A (45m×45m) area around the vehicle was covered by a road grid having a cell size of (0.1m×0.1m). A decay rate of 0.98 was used, and the odometry was coming from the IMU present in the localization system previously used for the collection of the labelled LIDAR scans. Only the 20 lowest LIDAR rings were used. A video of a grid accumulation in a roundabout is available[1].

VII. CONCLUSION

We presented TAdNet, a Transformation-adversarial network inspired by PointNet that performs road detection in LIDAR rings. The classification results can be used to generate evidential road grid maps without needing an explicit geometrical model, as showed by some experiments done on real-life data, and a TAdNet trained on coarse LIDAR labels obtained from a map. The next step will consist in evaluating other approaches for model-free evidential road grid mapping, in a more reliable fashion. To do so, a dataset of 368 LIDAR scans was already finely labelled by hand, and will be used for validation purposes in the future.

ACKNOWLEDGMENT

This work was realized within the SIVALab joint laboratory between Renault S.A.S, the CNRS and HeuDiaSyc.

REFERENCES

[1] 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.

[2] E. Capellier, F. Davoine, V. Frémont, J. Ibañez-Guzman, and Y. Li, "Evidential grid mapping, from asynchronous LIDAR scans and RGB images, for autonomous driving," in *21st International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2018, pp. 2595–2602.

[3] S. Wirges, C. Stiller, and F. Hartenbach, "Evidential occupancy grid map augmentation using deep learning," in *IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2018, pp. 668–673.

[4] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3D classification and segmentation," *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, vol. 1, no. 2, p. 4, 2017.

[5] T. Denoeux, "Logistic regression, neural networks and dempster-shafer theory: A new perspective," *ArXiv preprint arXiv:1807.01846*, 2018.

[6] R. Fernandes, C. Premebida, P. Peixoto, D. Wolf, and U. Nunes, "Road detection using high resolution lidar," in *2014 IEEE Vehicle Power and Propulsion Conference (VPPC)*, IEEE, 2014, pp. 1–6.

[7] L. Caltagirone, S. Scheidegger, L. Svensson, and M. Wahde, "Fast lidar-based road detection using fully convolutional neural networks," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2017, pp. 1019–1024.

[8] Y. Lyu, L. Bai, and X. Huang, "Chipnet: Real-time lidar processing for drivable region segmentation on an fpga," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2018.

[9] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems*, 2017, pp. 5099–5108.

[10] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe, "Exploring spatial context for 3D semantic segmentation of point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 716–724.

[11] E. Capellier, F. Davoine, V. Cherfaoui, and Y. Li, "Evidential deep-learning for arbitrary lidar-object classification in the context of autonomous driving," *Intelligent Vehicles Symposium (IV)*, 2019.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[13] X. Li, S. Chen, X. Hu, and J. Yang, "Understanding the disharmony between dropout and batch normalization by variance shift," *ArXiv preprint arXiv:1801.05134*, 2018.

[14] 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.

[15] K. A. B. Ahmad, M. Sahmoudi, and C. Macabiau, "Characterization of GNSS receiver position errors for user integrity monitoring in urban environments," in *ENC-GNSS 2014, European Navigation Conference*, 2014.

[1] https://drive.google.com/file/d/1R7WuZaIvUqPHVRbplDLglea5b46zugE5/view?usp=sharing