

# A Simulation-based End-to-End Learning Framework for Evidential Occupancy Grid Mapping\*

Raphael van Kempen<sup>ID</sup>, Bastian Lampe<sup>ID</sup>, Timo Wopen<sup>ID</sup>, and Lutz Eckstein

**Abstract**—Evidential occupancy grid maps (OGMs) are a popular representation of the environment of automated vehicles. Inverse sensor models (ISMs) are used to compute OGMs from sensor data such as lidar point clouds. Geometric ISMs show a limited performance when estimating states in unobserved but inferable areas and have difficulties dealing with ambiguous input. Deep learning-based ISMs face the challenge of limited training data and they often cannot handle uncertainty quantification yet. We propose a deep learning-based framework for learning an OGM algorithm which is both capable of quantifying first- and second-order uncertainty and which does not rely on manually labeled data. Results on synthetic and on real-world data show superiority over other approaches. Source code and datasets are available at <https://github.com/ika-rwth-aachen/EviLOG>.

## I. INTRODUCTION

Automated vehicles rely on an accurate model of their environment for planning safe and efficient behavior. Depending on the chosen representation of the environment in this model, different perception algorithms and different sensor modalities are best suited, each coming with corresponding advantages and disadvantages. Often, several different approaches are combined to compensate for the disadvantages of one perception algorithm with the advantages of another.

One common representation of the dynamic environment are object lists. They contain the state of all objects detected and tracked by the vehicle. Several methods for detection and tracking of objects in camera, radar and lidar data have been published during the past years [1], [2], [3]. A drawback of object lists is that the perception algorithms generating them can often only detect a fixed set of predefined object classes. Objects of these classes need to be explicitly contained in the perception algorithms' training data. Since there exist extremely many classes of objects in the world, it is difficult to ensure that all relevant objects are accounted for.

Occupancy grid mapping algorithms, which are agnostic to the specific class of an object, can compensate for this disadvantage by reducing their task to simply assigning an occupancy state to each cell in a grid, which describes a defined area around a vehicle [4], [5]. They usually take distance measurements, e.g. from a lidar sensor, as input. The resulting occupancy grid maps (OGMs) are for example

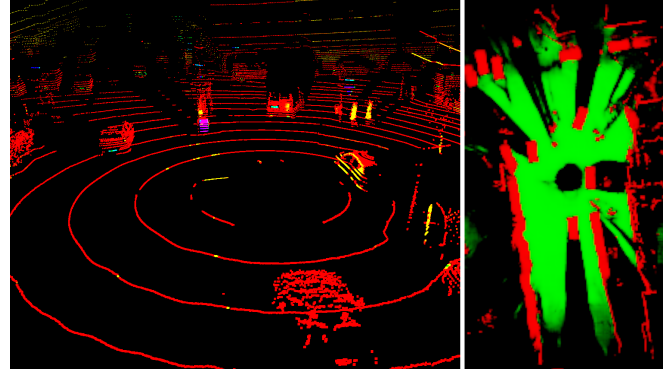


Fig. 1. A deep learning-based inverse sensor model predicts an evidential occupancy grid map (right) from a real-world lidar point cloud (left). Grid cells with a large belief mass for the state *Free* are colored green, those with a large belief mass for the state *Occupied* are colored red. The bigger the belief mass, the larger the respective value in the color channel. Black represents the maximum of epistemic uncertainty.

used in the automated vehicles developed in the UNICARagil project [6], [7], from which this paper also originates.

To determine cell occupancy states from sensor data, an inverse sensor model (ISM) is required. In the past, geometric models have mostly been used [5]. These approaches are often suitable for static and flat environments, but fail in dynamic and non-flat environments. Recent works also propose deep neural networks for this task as so-called deep ISMs [8], [9]. As usual with supervised learning, the gathering of training data poses a challenge here. Both [8] and [9] use cross-modal training. They make use of a lidar-based geometric ISM to generate training data for a radar-based deep ISM. This approach enables the learned model to infer occupancy information which cannot be deduced with the geometrical approach alone. Since the model is trained with data that does not constitute ground-truth data but only an estimation from the lidar-based geometric ISM, the trained models suffer from the restrictions of the lidar-based ISM.

If cross-modal training is discarded, one can make use of manually labeled data for training deep ISMs. This approach is infeasible for many though, because of the associated amount of manual labeling work. Fortunately, simulation software for automated driving is rapidly evolving. Better sensor models and the possibility to automatically generate ground-truth data make this approach viable. With more realistic synthetic data, the reality gap is closing and generalization of neural networks from synthetic to real-world data becomes possible.

In any event, there is a need for the quantification of

\*This research is accomplished within the project "UNICARagil" (FKZ 16EMO0289). We acknowledge the financial support for the project by the Federal Ministry of Education and Research of Germany (BMBF).

The authors are with the Institute for Automotive Engineering (ika), RWTH Aachen University, 52074 Aachen, Germany (`{firstname.lastname}@ika.rwth-aachen.de`)

uncertainty in the estimates of perception algorithms. The capability of neural networks to output a measure of their confidence in a prediction is not yet reflected in many deep learning-based approaches.

Our work takes into account all of the aforementioned challenges and contributes a framework in which they are dealt with.

## II. CONTRIBUTION

We present an end-to-end learning framework for training deep learning-based inverse lidar sensor models using synthetic data. First, a method for generating training samples consisting of lidar point clouds as input data and evidential ground-truth OGMs as labels is presented. Second, we propose a suitable neural network architecture that extends the popular PointPillars architecture [3] with an evidential prediction head (EPH). The EPH is capable of estimating an evidential OGM. Finally, we evaluate the performance of our approach both on synthetic data and on real-world data. We show that the trained model is able to generalize to different synthetic environments and also produces promising results on real-world data.

## III. BACKGROUND

In the following, the concept of OGMs and an overview of current approaches for computing OGMs through inverse sensor models (ISMs) is presented. A geometric ISM will be the baseline for the evaluation of the deep ISM proposed in this work. Additionally, a brief introduction into evidence theory and subjective logic is given as this is the basis for the loss function that is used with the proposed deep neural network.

### A. Occupancy Grid Maps

OGMs as introduced in [4] divide the vehicle’s environment into discrete cells containing occupancy information. The occupancy state of each cell at time  $k$  can be represented as a binomial random variable  $o_k \in \{O, F\}$  ( $O$ : *Occupied*;  $F$ : *Free*). The probability of each cell being occupied at time  $k$  can be derived from the current measurement  $z_k$  using an inverse sensor model (ISM)  $p_{z_k}(o_k|z_k)$ . Often, a binary Bayes filter is used to create an OGM from a set of distance measurements [5]. This approach has some deficiencies. First, the binary Bayes filter relies on the Markov assumption that there is no temporal correlation of the occupancy state, and that the state does not change in time, which is not adequate for a dynamically changing environment. Additionally, by representing the occupancy as a probability value  $p_k(o_k)$ , it cannot be distinguished between cells which are uncertain because they cannot be observed, e.g. due to occlusions, and because of conflicting evidence as both cases are described with  $p_k(o_k) \approx 0.5$ . This challenge can be tackled with evidence theory as described in Section III-C.

### B. Inverse Sensor Models

While a sensor model describes how the environment is represented in sensor data, an inverse sensor model (ISM) is required to reconstruct information about the environment from sensor data, as done in e.g. grid mapping.

**Geometric ISMs** process distance measurements and calculate a probability distribution for the occupancy of cells at the location of the measurement as well as for cells in between the sensor and the measurement. These models consider the sensor’s inaccuracy, which is usually hand-crafted, but can also be learned [5]. Hand-designed methods assume a ground model to separate ground from obstacles in the measurement. By filtering techniques, such as Bayesian filtering, the information gathered from each data point in the measurement can be combined into one measurement OGM. By combining the latter with a previous estimate of the OGM, a new estimate with reduced uncertainty can be found [5].

**Deep ISMs** are deep learning-based inverse sensor models, which take distance measurements as input for a deep neural network, which is used to predict an OGM. For this purpose, tensor-like representations for the sensor data as input and the OGM as output must be found such that they can be processed.

In [9], a deep radar ISM is introduced. Multiple radar measurements are combined into one bird’s-eye-view image serving as input for a semantic segmentation task with classes *Occupied*, *Free* and *Unobserved*. Training data is created from OGMs calculated from lidar measurements using a geometric ISM.

The model presented in [10] predicts future OGMs from lidar measurements. It is trained using unsupervised learning by creating training samples of OGMs with previous measurements. They use a naive model to create the ground-truth OGM by solely treating all reflection points in a height of 0.6 to 1.5 meters as obstacles. This only works in flat environments and with small pitch and roll angles of the ego-vehicle. They encode the lidar measurements into two matrices covering the vehicle’s environment with binary values describing the visibility and the occupancy of each cell. The OGM is encoded as a binary matrix which does not sufficiently allow describing uncertainty.

In [8], radar data is transformed into a two-channel bird’s-eye-view image with one channel containing static and the other containing dynamic detections. The OGM is represented as a three-channel image containing belief masses (cf. Section III-C) for the states *Occupied*, *Free* and *Unknown*. The task is treated as image segmentation problem. In [11], they combine a deep radar ISM with geometrical lidar and radar ISMs to increase the perception field and to reduce the time needed to populate the occupancy grid.

The deep neural network presented in [12] predicts a bird’s-eye-view image containing occupancy, object class and motion information in one shot. A sequence of lidar point clouds encoded as binary matrices which are stacked along a third dimension are used as input data. The model is trained using labeled data from the nuScenes data set [13].

A methodology capable of transforming segmented images from vehicle cameras to a semantic OGM is presented in [14]. They also use synthetic data and try to bridge the reality gap by using segmented images as an intermediate representation between real-world and synthetic sensor data.

### C. Evidence Theory

**Evidence Theory** as introduced by Dempster and Shafer (DST) [15] can be understood as a generalization of Bayesian probability theory [16]. It allows the explicit consideration of epistemic uncertainty and has also been used with OGMs [17]. Using DST, belief masses are assigned to all subsets of the frame of discernment  $\Theta$ . For cells in an evidential OGM, this can consist of all possible and mutually exclusive cell states *Free* ( $F$ ) and *Occupied* ( $O$ ):  $\Theta = \{F, O\}$ . The power set  $2^\Theta = \{\emptyset, \{F\}, \{O\}, \Theta\}$  contains all possible subsets of  $\Theta$  to which belief masses  $m$  can be assigned.

$$m : 2^\Theta \rightarrow [0, 1] \quad (1)$$

$$m(\emptyset) = 0 \quad (2)$$

$$\sum_{A \in 2^\Theta} m(A) = 1 \quad (3)$$

In this example,  $m(O)$  constitutes evidence for a cell being occupied and  $m(F)$  for a cell being free. Additionally, a state for which no evidence is available can be addressed with  $m(\Theta)$  while the empty set is no possible outcome.

**Subjective Logic** (SL) is a mathematical framework for reasoning under uncertainty. It explicitly distinguishes between epistemic opinions and aleatoric opinions. A direct bijective mapping between the belief mass distribution  $m(A)$  in DST and a subjective opinion, i.e. a belief mass distribution  $b_A$  and an uncertainty mass  $u$  in SL is given by [18].

$$b_A = m(A), \quad A \in \Theta \quad (4)$$

$$u = m(\Theta) \quad (5)$$

$$\sum_{A \in \Theta} b_A + u = 1 \quad (6)$$

Subjective opinions are equivalent to a Dirichlet probability density function (PDF)

$$\text{Dir}(\mathbf{p}, \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{A \in \Theta} p_A^{\alpha_A - 1} \quad (7)$$

with prior probabilities  $\mathbf{p}$  and parameters  $\boldsymbol{\alpha}$

$$\mathbf{p} = \left\{ p_A \mid A \in \Theta, \sum_{A \in \Theta} p_A = 1, 0 \leq p_A \leq 1 \right\} \\ \boldsymbol{\alpha} = \{ \alpha_A \mid A \in \Theta \} \quad (8)$$

and the multivariate beta function  $B$  in terms of the gamma function  $\Gamma$

$$B(\boldsymbol{\alpha}) = \frac{\prod_{A \in \Theta} \Gamma(\alpha_A)}{\Gamma(\sum_{A \in \Theta} \alpha_A)}. \quad (9)$$

Evidence for the singletons in the FOD  $e_A \geq 1, A \in \Theta$  can be converted to parameters of a Dirichlet PDF and to a

subjective opinion  $(\mathbf{b}, u)$  with the number of classes  $K = |\Theta|$  and the Dirichlet strength  $S = \sum_{A \in \Theta} \alpha_A$ :

$$\alpha_A = e_A + 1, \quad A \in \Theta \quad (10)$$

$$b_A = \frac{e_A}{S} \quad (11)$$

$$u = \frac{K}{S} \quad (12)$$

The authors of [19] show that a deep neural network can be trained to predict the parameters  $\boldsymbol{\alpha}$  of a Dirichlet PDF to express uncertainty in a classification task on the MNIST data set [20]. They propose three loss functions while the approach using the sum of squares led to the best results and is used in this work. The loss for one grid cell  $i$  with network parameters  $\mathbf{w}$ , expected class probabilities  $\hat{\mathbf{p}}_i$  and the true state  $\mathbf{y}_i = \{y_{i,A} \in \{0, 1\} \mid A \in \Theta\}$  with  $y_{i,A}$  being one if state  $A$  is true and zero if false or unknown is given:

$$\mathcal{L}_i(\mathbf{w}) = \|\mathbf{y}_i - \hat{\mathbf{p}}_i\|_2^2 \text{Dir}(\hat{\mathbf{p}}_i, \hat{\boldsymbol{\alpha}}_i) \quad (13)$$

This motivates us to create an evidential deep neural network for the task of occupancy grid mapping. We train a model to predict the parameters of a Dirichlet PDF describing the states of cells in an OGM.

## IV. LEARNING FRAMEWORK

Our learning framework processes lidar point clouds as input data and predicts evidential OGMs. In the following, the network architecture and a simulation-based method for generating and augmenting training data is presented.

### A. Network Architecture

The architecture of our deep neural network is based on the popular PointPillars architecture [3], which is capable of accurately detecting objects in lidar point clouds, while providing a relatively low execution time. The network is divided into three parts. First, there is the Pillar Feature Net, which encodes the reflection points from the lidar measurement into denser features. Second, there is a 2D CNN backbone, which transforms the features into a high-level representation. Last, there are detection heads estimating bounding boxes and motion states for the measured objects.

In this work, the detection heads used in [3] are replaced with a 2D convolutional layer with two output channels and ReLU activation. Each pixel represents a cell  $i$  in the predicted OGM where the channels contain evidence  $e_{i,A}$  for both singletons in the frame of discernment  $\Theta = \{F, O\}$ , i.e. the cell being occupied or free. The predicted evidence can be converted into estimated parameters of a Dirichlet PDF  $\hat{\alpha}_{i,A}$  with Equation (10). The expected probability values of a cell being occupied or free can be derived from the Dirichlet PDF as  $\hat{p}_{i,A} = \hat{\alpha}_{i,A} / S_i$ . Thus, following the simplification presented in [19], the loss function from Equation (13) in

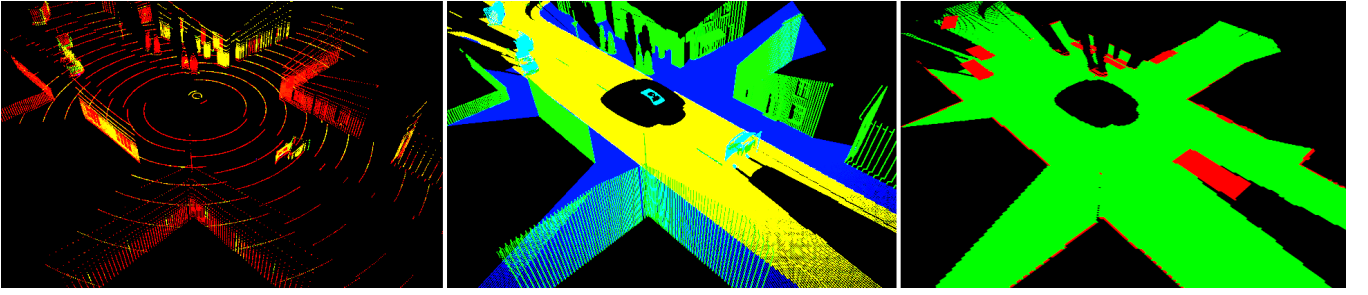


Fig. 2. The left image shows the point cloud of a simulated VLP32C lidar sensor where the point color indicates the intensity of reflection. In the middle, the corresponding high-definition (HD) point cloud with 3000 layers and points colored by the material causing the reflection is shown. This is used to create the ground-truth occupancy grid map that is shown in the right picture. Free cells are colored green, occupied cells are colored red and unknown cells are black. In addition to the information gained from the HD point cloud, all cells covered by other traffic participants are marked as occupied. A training sample consists of one point cloud as shown in the left image and a corresponding ground truth OGM as shown in the right image.

our application reduces to

$$\begin{aligned}\mathcal{L}_i(w) &= \mathbb{E} [y_{i,F}^2 - 2y_{i,F}p_{i,F} + p_{i,F}^2] \\ &\quad + \mathbb{E} [y_{i,O}^2 - 2y_{i,O}p_{i,O} + p_{i,O}^2] \\ &= (y_{i,F} - \hat{p}_{i,F})^2 + \frac{\hat{p}_{i,F}(1 - \hat{p}_{i,F})}{S_i + 1} \\ &\quad + (y_{i,O} - \hat{p}_{i,O})^2 + \frac{\hat{p}_{i,O}(1 - \hat{p}_{i,O})}{S_i + 1}.\end{aligned}\quad (14)$$

We extend the loss function with a Kullback-Leibler divergence term with an impact factor  $\lambda_t = \min(1.0, t/10)$  that increases from zero to one with the epoch number  $t$  as proposed in [19]. This regularization term penalizes a divergence of the Dirichlet parameters resulting from conflicting evidence  $\tilde{\alpha}_i = \mathbf{y}_i + (1 - \mathbf{y}) \odot \alpha_i$  from a distribution resulting from no evidence, i.e.  $\alpha_i = \mathbf{1}$ . Hence, it encourages the network to reduce conflicting evidence in its output. This leads to the total loss function for all  $N$  cells of the OGM in one sample.

$$\mathcal{L}(w) = \sum_{i=1}^N \mathcal{L}_i(w) + \lambda_t \text{KL}[\text{Dir}(\mathbf{p}_i | \tilde{\alpha}_i) || \text{Dir}(\mathbf{p}_i | \mathbf{1})] \quad (15)$$

with the Kullback-Leibler divergence in terms of the gamma function  $\Gamma$  and the digamma function  $\psi$ .

$$\begin{aligned}\text{KL}[\text{Dir}(\mathbf{p}_i | \tilde{\alpha}_i) || \text{Dir}(\mathbf{p}_i | \mathbf{1})] &= \\ \log \left( \frac{\Gamma(\tilde{\alpha}_{iF} + \tilde{\alpha}_{iO})}{\Gamma(2)\Gamma(\tilde{\alpha}_{iF})\Gamma(\tilde{\alpha}_{iO})} \right) & \\ + (\tilde{\alpha}_{iF} - 1) [\psi(\tilde{\alpha}_{iF}) - \psi(\tilde{\alpha}_{iF} + \tilde{\alpha}_{iO})] & \\ + (\tilde{\alpha}_{iO} - 1) [\psi(\tilde{\alpha}_{iO}) - \psi(\tilde{\alpha}_{iF} + \tilde{\alpha}_{iO})] &\end{aligned}\quad (16)$$

This loss function will be used to train the model in the experiments such that it approximates the evidence masses in the grid cells while loss for occupied cells is weighted 100 times compared to the other cells to compensate for their under-representation.

## B. Training Data

To create training data from the simulation, we use a method that we have already presented in [21]. The simulation environment [22] provides a lidar plugin that supports advanced ray tracing and physically-based rendering, i.e. it

considers the physical properties of materials to generate more realistic point clouds. We have modeled the sensor setup of one of our research vehicles, which has a Velodyne VLP32C lidar sensor mounted in the middle of its roof. In addition to this 32-layer lidar sensor, another lidar sensor with 3000 layers, which will be called high-definition (HD) lidar from now on, was added to the simulated vehicle. As the lidar plugin provides information about the material type that caused a reflection, it is possible to derive a dense ground-truth OGM from the simulated HD lidar data. The HD lidar covers the same field of view and is exposed to the same occlusions as the real sensor to ensure that the OGMs in the training data will only contain information which can theoretically be derived from the input point cloud. In addition, we augment the ground-truth OGMs based on a ground-truth object list that is also provided by the simulation and contains information about the position and shape of other traffic participants. As we expect the deep ISM to learn to recognize the shape of e.g. cars and trucks, we mark all cells covered by them as occupied in the ground-truth OGM if a minimum of 50 reflections on the vehicle is present in the input point cloud. Figure 2 shows that cells with reflections on obstacles such as cars and trees are marked as occupied in the ground-truth OGM, whereas the ground is marked as free. We decided to mark also sidewalks as free space as precise information on the lane geometry can be obtained from a static map. In the event of a serious obstruction of road traffic, the sidewalk could also be an option to pass by.

## C. Augmentation

During the training, we augment the training data by applying a random rotation to the input point cloud and the label OGM. Both are rotated around a vertical axis at the origin of the lidar sensor. This turned out to be important, as without augmentation, cells on the far left and right side of the OGM are almost always occupied, which makes it difficult for the deep ISM to classify those cells correctly.

## V. EXPERIMENTAL SETUP AND RESEARCH QUESTION

As explained in Section IV-B, training data for the deep learning-based inverse lidar model is created using a simula-

tion environment. We have selected a model of an urban area and created ten scenarios, each containing random variations of the dynamic environment. In half of the scenarios, the ego-vehicle takes a clockwise route, in the other half, it takes a counter-clockwise route. The randomly generated pulk traffic contains cars, trucks and motorcycles. The type of each of these objects is also chosen randomly from a larger catalogue of possibilities. Additionally, pedestrians are placed at various locations on the sidewalks. Parked vehicles are randomly put on parking lanes. With each scenario, 1.000 training samples were created at a sampling rate of one per second. In total, 10.000 samples were generated for training. For the validation data set, another 1000 samples were created in the same static, but a different dynamic environment. Another test dataset consists of 100 samples from a different scenario in another part of the simulated urban area and will be used for evaluation in Section VI-A.

The neural network is trained using the Adam optimizer. The Pillar Feature Net [3] creates 10.000 pillars with a maximum of 100 reflection points per pillar. The intensity of the reflection points in the simulated lidar point cloud is normalized such that a distribution similar to one observed in real-world lidar point clouds is achieved. The output grid maps have a length of 81.92 and a width of 56.32 meters. A cell's side length is 16 centimeters, resulting in a 512 by 352 cell grid map. The sensor origin is at the center of the grid map. The map dimensions and cell size correspond to the detection area and step size of the Pillar Feature Net. After training for 100 epochs with a batch size of 5, a minimum loss of 0.104 and a Kullback-Leibler divergence of 0.357 on the validation data was achieved.

In the following, we want to answer the following research questions: How well does a deep convolutional neural network that is based on our presented methodology perform when predicting dense OGMs from lidar measurements? In particular, we want to analyze whether it is capable of capturing the epistemic uncertainty for cells where no reflection point is located. Then, how well does the network perform when presented with real-world sensor data?

## VI. RESULTS AND DISCUSSION

First, we evaluate the performance of the trained model on a synthetic test dataset to analyze how well the trained ISM predicts OGMs in a scenario that was not contained in the training data.

Afterwards, we test our model on real-world sensor data that was recorded with one of our research vehicles.

### A. Evaluation on Synthetic Data

We compare the OGMs that are created using our proposed deep ISM to OGMs created using a geometric ISM where all cells containing reflection points in a height of 0.5 to 2.0 meters above ground are marked as occupied while all cells between the sensor origin and these cells are marked as free. Figure 4 shows the mean belief masses in both OGMs during the test scenario. It is apparent and confirmed by Figure 3 that the geometric ISM creates a considerably higher proportion

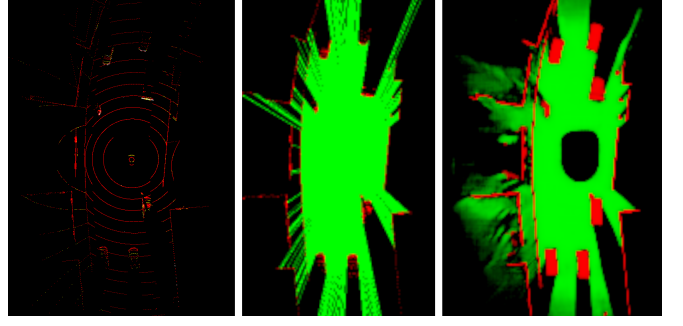


Fig. 3. The geometric ISM (middle) is only able to determine occupied cells containing reflection points of the measurement (left). The deep learning-based ISM (right) has learned to derive more information, e.g. the whole space occupied by vehicles.

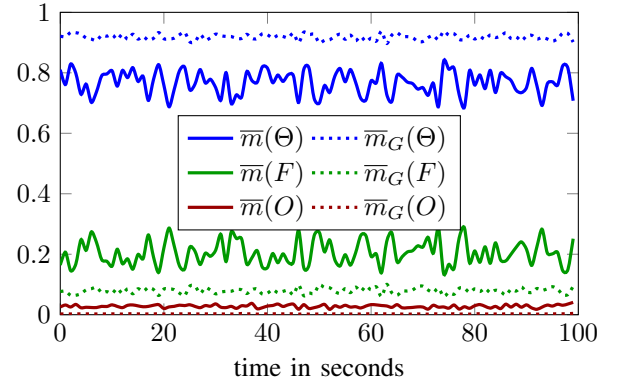


Fig. 4. Mean belief masses per OGM during the test scenario.  $\bar{m}(F)$  is the mean mass for a free,  $\bar{m}(O)$  for an occupied and  $\bar{m}(\Theta)$  for an unknown cell state. The dashed lines represent results based on a geometric ISM, whereas the continuous lines show results from our deep ISM.

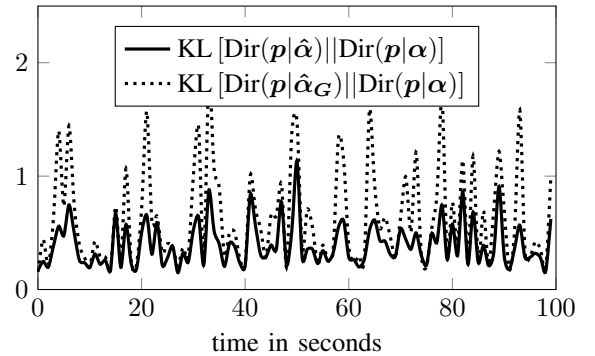


Fig. 5. Comparison of the mean Kullback-Leibler divergence of both estimated Dirichlet PDFs from the true Dirichlet PDF per OGM during the test scenario.  $\hat{\alpha}$  are the Dirichlet parameters estimated by our deep ISM and  $\hat{\alpha}_G$  are the parameters estimated using a geometric ISM.

of cells with an unknown state, hence, less cells are classified as free or occupied. Figure 5 shows the Kullback-Leibler divergence of both Dirichlet PDFs, the one predicted from the deep ISM  $\text{Dir}(p|\hat{\alpha})$  and the one from the geometric ISM  $\text{Dir}(p|\hat{\alpha}_G)$ , from the true PDF  $\text{Dir}(p|\alpha)$ . It is evident that the deep ISM estimates the cell states better than the geometric ISM at any time.



## B. Evaluation on Real-World Data

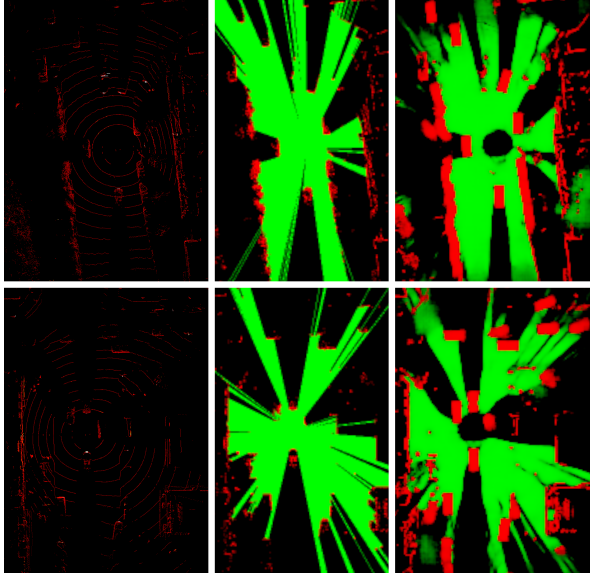


Fig. 6. Two representative samples showing the predicted OGMs (right) using the deep ISM trained with synthetic data on real-world measurements (left) compared to a geometric ISM (middle).

Finally, we want to analyze whether the deep ISM trained with synthetic data can also be successfully applied to real-world data. Thus, we test the model on real lidar measurements from one of our research vehicles that has a similar shape and sensor setup as the simulated vehicle. Figure 6 shows two representative samples of the model's performance on a dataset recorded in an urban environment. It is apparent that the performance cannot keep up with the evaluation on synthetic data. Nevertheless, it creates promising results in an environment that is considerably different from the synthetic environment. In particular, the occupancy states of cells that do not contain a reflection point are mostly rated correctly. A higher real-world performance is expected when the simulation scenarios would take place in a model of the actual static environment of the real-world scenarios. More samples and a higher diversity of the training data are relatively easy to achieve in the simulation and are also expected to further increase performance.

## VII. CONCLUSION

We presented a new methodology to train neural networks for the task of occupancy grid mapping using lidar point clouds. The trained network performs considerably better than a classical approach when presented with synthetic data. It also shows advantages when presented with real-world data, even though many options to increase the network's generalization capabilities remain. In contrast to many other deep learning-based approaches, our network is trained such that it can give an estimate of its first- and second-order uncertainty. Our methodology is especially promising because it does not rely on manually labeled data. Future research will investigate methods that further increase the network's performance on real-world data.

## REFERENCES

- [1] M. Aeberhard, "Object-level fusion for surround environment perception in automated driving applications," Ph.D. dissertation, TU Dortmund, Dortmund, 2017.
- [2] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490–4499.
- [3] A. H. Lang, S. Vora *et al.*, "Pointpillars: Fast encoders for object detection from point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 689–12 697.
- [4] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.
- [5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*, ser. Intelligent robotics and autonomous agents. Cambridge, Mass.: MIT Press, 2005.
- [6] T. Wopen, B. Lampe *et al.*, "Unicaragil - disruptive modular architectures for agile, automated vehicle concepts," in *27th Aachen Colloquium*, 2018, pp. 663–694.
- [7] M. Buchholz, F. Gies *et al.*, "Automation of the unicaragil vehicles," in *29th Aachen Colloquium*, 2020, pp. 1531–1560.
- [8] D. Bauer, L. Kuhnert, and L. Eckstein, "Deep, spatially coherent inverse sensor models with uncertainty incorporation using the evidential framework," in *2019 IEEE Intelligent Vehicles Symposium (IV 2019)*, 2019, pp. 2490–2495.
- [9] L. Sless, B. E. Shlomo *et al.*, "Road scene understanding by occupancy grid learning from sparse radar clusters using semantic segmentation," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 867–875.
- [10] J. Dequaire, P. Ondruška *et al.*, "Deep tracking in the wild: End-to-end tracking using recurrent neural networks," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 492–512, Apr. 2018.
- [11] D. Bauer, L. Kuhnert, and L. Eckstein, "Deep inverse sensor models as priors for evidential occupancy mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 10242020, pp. 6032–3067.
- [12] P. Wu, S. Chen, and D. N. Metaxas, "Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 382–11 392.
- [13] H. Caesar, V. Bankiti *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 618–11 628.
- [14] L. Reiher, B. Lampe, and L. Eckstein, "A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020.
- [15] G. Shafer, *A mathematical theory of evidence*, ser. Limited paperback editions. Princeton, NJ: Princeton Univ. Press, 1976, vol. 42.
- [16] A. P. Dempster, "A generalization of bayesian inference," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 30, no. 2, pp. 205–232, Jul. 1968.
- [17] D. Nuss, S. Reuter *et al.*, "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, Jul. 2018.
- [18] A. Jøsang, *Subjective Logic*. Cham: Springer International Publishing, 2016.
- [19] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," in *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, 2018, pp. 3179–3189.
- [20] Y. LeCun, C. Cortes, and C. J. Burges, "Mnist handwritten digit database," *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, vol. 2, 2010.
- [21] B. Lampe, R. van Kempen *et al.*, "Reducing uncertainty by fusing dynamic occupancy grid maps in a cloud-based collective environment model," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 837–843.
- [22] K. von Neumann-Cosel, M. Dupius, and C. Weiss, "Virtual test drive - provision of a consistent tool-set for [d,h,s,v]-in-the-loop," in *Proceedings of the driving simulation conference Monaco*, 2009.